Derivatives and Integrals

An Annotated Discourse

Robert Beezer
Department of Mathematics and Computer Science
University of Puget Sound
Tacoma, Washington, USA
beezer@pugetsound.edu
Statement about support received by the first author.

A. Second Author
Department of Mathematics
University of Somewhere
Anytown, USA
asauthor@example.edu
Statement about support received by the second author.

October 8, 2026

Abstract. This is a sample of many of the things you can do with PreTeXt. Sometimes the math makes sense, sometimes it seems to be written in the first person, sort of like this Abstract.

1 Introduction

We consider definite integrals of functions $f(x)$. For example,

$$\int_0^2 \sin^2(x)\,dx.$$

This is also a demonstration of the capabilities of PreTeXt.

Generated: October 8, 2026, 07:31:05 (-07:00)

2 The Fundamental Theorem

There is a remarkable theorem:[1]

Theorem 2.1 The Fundamental Theorem of Calculus. *If $f(x)$ is continuous, and the derivative of $F(x)$ is $f(x)$, then*

$$\int_a^b f(x)\,dx = F(b) - F(a)$$

[1] And fortunately we do not need to try to write it in the margin!

Proof. Left to the reader.[2] ■

You will find almost nothing about all this in the article [2], nor in the book [J.3.1], since they belong in some other article, but we can cite them out-of-order for practice anyway.

When we are writing we do not always know what we want to cite, or just where subsequent material will end up. For example, we might want a citation to `[provisional cross-reference: some textbook about the FTC]` or we might want to reference a later `[provisional cross-reference: chapter about DiffEq's, and an_underscore]`.

We can also embed "todo"s in the source by making an XML comment that begins with the four characters `todo`, and selectively display them, so you may not see the one here in the output you are looking at now. Or maybe you do see it?

Because a definite integral can be computed using an antiderivative, we have the following definition.

Definition 2.2. Suppose that $\frac{d}{dx}F(x) = f(x)$. Then the **indefinite integral** of $f(x)$ is $F(x)$ and is written as

$$\int f(x)\,dx = F(x).$$

◆

3 Computing Integrals with Sage ($\int$)

Sage can compute definite integrals. The output contains the approximate numerical value of the definite integral, followed by an upper bound of the error in the approximation.

```
numerical_integral(sin(x)^2, (0, 2))
```

```
(1.189200623826982, 1.320277913471315e-14)
```

Given the Fundamental Theorem, we would find the antiderivative useful.

```
integral(sin(x)^2, x)
```

```
1/2*x - 1/4*sin(2*x)
```

The same command can be used to employ the antiderivative in the application of the Fundamental Theorem. Notice that the answer is *exact* and any further manipulation is likely to be simply producing a numerical approximation.

```
integral(sin(x)^2, (x, 0, 2))
```

```
-1/4*sin(4) + 1
```

There are integrals you really do not want to evaluate, or you do not want your reader to evaluate. A Sage cell can be configured for display purposes only—you can look but you cannot touch.

```
integral(e^(x^2), x)
```

[2] This is a footnote in a `<proof>` that is inside a `<theorem>`.

```
-1/2*I*sqrt(pi)*erf(I*x)
```

You can give a Sage element a `doctest` attribute, whose value mirrors the optional hash tags used in Sage doctests. Possible values are `random`, `long time`, `not implemented`, `not tested`, `known bug`, `absolute`, `relative`, and `optional`. The values `absolute` and `relative` refer to floating-point tolerances for equality and require a second attribute `tolerance` to specify a floating point value. The value `optional` refers to the test requiring that an optional Sage package be present. The name of that package should be provided in the `package` attribute.

The next cell is marked in the source as `doctest="random"`, and so is specified as unpredictable and not tested. But there is some "sample" output which will appear in the LaTeX version (and always be the same).

```
random()
```

```
0.11736021338650582
```

While the next cell is random, the returned value will never be more than 0.01 away from 12, since the `random()` function stays between 0 and 1. So we provide 12.005 as the expected answer, but test with an absolute tolerance of $\epsilon = 0.006$.

```
12 + 0.01*random()
```

```
12.005
```

Sage has some functions which affect output, generally making mathematics look more like mathematics via LaTeX syntax. This is a simple test, and you should see the variable and superscript in italics, properly formatted as output when viewed within HTML output. We have provided expected output for doctesting, but it is sort of silly to have this as part of LaTeX output, even if it is instructive.

```
pretty_print(html("$a^2$"))
```

```
<script type="math/tex">a^2</script>
```

Sage, and by extension, the Sage Cell Server, can interpret several languages. The next example has code in the `R` language, a popular open source language for statistics. As an author, you add the attribute `language="r"` to your `sage` element. (The default language is Sage, so you do not need to indicate that repeatedly.) Note that a language like `R` likes to use a "less than" character, <, special character in XML. You need to escape it by writing `<` as we have done in the source for this example. (See the discussion in Subsection 9.1.)

As a reader you learn that the "Evaluate" button for a pre-loaded Sage cell will indicate the language in use.

```
ruth <- c(22, 25, 34, 35, 41, 41, 46, 46, 46, 47, 49, 54, 54, 59, 60)
bonds <- c(16, 25, 24, 19, 33, 25, 34, 46, 37, 33, 42, 40, 37, 34, 49, 73, 46, 45, 45, 5, 26, 28)
dimaggio <- c(12, 14, 20, 21, 25, 29, 30, 30, 31, 32, 32, 39, 46)
summary(ruth)
summary(bonds)
summary(dimaggio)
boxplot(ruth, bonds, dimaggio)
```

The Sage Cell Server supports the following languages: `sage`, `gap`, `gp`, `html`, `maxima`, `octave`, `python`, `r`, and `singular`.

Here is another `R` cell. Unfortunately, it seems Sage's `doctest` facility cannot be used easily with code from other languages. In the source for this

example, we have employed the XML escape sequence, `<` several times (see Subsection 9.1).

```
age <- c(25, 30, 56)
gender <- c("male", "female", "male")
weight <- c(160, 110, 220)
mydata <- data.frame(age,gender,weight)
summary(mydata)
cor(mydata$age,mydata$weight)
mean(mydata$age)
sd(mydata$age)
plot(mydata$age,mydata$weight)
```

The Sage Cell server imports a few important R packages. As of 2022-06-04 these are `deSolve`, `ggplot2`, `pracma`, `survey`, `swirl`, and `tidyverse`. This next example uses the `ggplot` library for both a data set and the plotting capabilities. Note the initial use of the `library()` function. This is a modified version of the "Bubble plot" example at Top 50 `ggplot2` Visualizations —The Master List.

```
# load package and data
library(ggplot2)
data(mpg, package="ggplot2")

mpg_select <- mpg[mpg$manufacturer %in% c("audi", "ford", "honda", "hyundai"), ]

# Scatterplot
g <- ggplot(mpg_select, aes(displ, cty)) +
  labs(subtitle="mpg: Displacement vs City Mileage",
       title="Bubble chart")

g + geom_jitter(aes(col=manufacturer, size=hwy)) +
  geom_smooth(aes(col=manufacturer), method="lm", se=F)
```

Here is a blank Sage cell that you may use for practice and experimentation with the commands above. Note that this cell allows a choice of languages, and is not linked with any of the previous cells, so a reader would need to start fresh, or cut/paste definitions from other cells.

On the other hand a `<sage>` element with no content will also create an empty Sage cell for the reader's use, but now it will be specific to a particular language and linked to others of the same language. Run the `R` cell above that defines the variable `ruth` and then try typing `summary(ruth)` in the cell below.

You can make Sage blocks which are of `type="invisible"`, which will never be shown to a reader, but which get doctested. Why do this? If some code produces an error, and you hope it is fixed someday, use an invisible block to raise the error. Once fixed, the doctest will fail, and you can adjust your commentary to suit. There is such a block right now, *but* you will need to go to the source to see it.

```
graphs(augment='vertices').next()
```

```
Traceback (most recent call last):
...
NotImplementedError
```

Our maximum width for text, designed for readability, suggests you should format your Sage code with a maximum of about 54 characters. On a mobile device, the number of displayed characters might be as low as 28 in portrait

orientation, and again around 50 in landscape. You can use a variety of techniques to shorten long lines, such as using intermediate variables. Since Sage is just a huge Python library, you can use any of Python's facilities for handling long lines. These include a continuation character (which I try to avoid using) or natural places where you can break long lines, such as between entries of a list. Also, if writing loops or functions, you may wish to have your indentation be only two characters wide (rather than, say, four).

Sage output can sometimes be quite long, though this has improved with some changes in Sage's output routines. Output code should be included primarily for doctesting purposes, and in this use, you may break at almost whitespace character and the doctesting framework will adjust accordingly. You may wish to show sample output in a static format, like a PDF, so you can consider formatting your output to fit the width constraints of that medium. Or you may even adjust exactly what is output, to keep it from being too verbose. Sage doctesting also allows for a wild-card style syntax which allows you to skip over huge chunks of meaningless or unpredictable output, such as tracebacks on error messages.

This paragraph is just a placeholder. It has handful of index entries, all starting with the letters "gas", taken from *Indexing for Editors and Authors: A Practical Guide to Understanding Indexes* by Leise, Mertes, and Badgett. The intent is to test letter-by-letter versus word-by-word sorting of index entries. We use a word-by-word order, resulting in:

1. gas
2. gas masks
3. gas production
4. gas works
5. gasoline
6. gastritis

Titled Sage Cells.

```
integral(sin(x)^2, x)
```

```
1/2*x - 1/4*sin(2*x)
```

You can place Sage cells inside of a `paragraphs` if you want to give them a title, but no numbers, etc..

4 An Interesting Corollary

Objectives 4 Fundamental Structures

This is an `<objectives>` element you are reading, and this is its introduction. This early section has really grown and tries to accomplish many things. Not all of them are listed here.

1. Display various "blocks", fundamental units of the flow.
2. More.
3. Evermore.

This concludes the (incomplete) objectives for this section, so now we can carry-on as before.

This is a cross-reference to one of the objectives above, forced to use the `phrase-global` form of the text. It should describe the objective as belonging to the *section* (rather than the *objectives*), since objectives are one-per-subdivision and are numbered based upon the containing division: Objective 1 of Section 4. For comparison this is the (forced) `type-global` cross-reference: Objective 4.1.

The Fundamental Theorem comes in two flavors, where usually one is a corollary of the other.

4.1 Second Version of FTC

Corollary 4.1 (Leibniz, Newton) *Suppose $f(x)$ is a continuous function. Then*

$$\frac{d}{dx}\int_a^x f(t)\,dt = f(x). \tag{4.1}$$

Proof. We simply take the indicated derivative, applying Theorem 2.1 at (4.2)

$$\begin{aligned}\frac{d}{dx}\int_a^x f(t)\,dt &= \frac{d}{dx}(F(x)-F(a)) &(4.2)\\ &= \frac{d}{dx}F(x) - \frac{d}{dx}F(a)\\ &= f(x) - 0 = f(x). &(4.3)\end{aligned}$$

■

Justification. A justification, which is one of the variants of a proof. ■

Proof. You can have multiple proofs, and they can have titles which replace the word "Proof" as a heading. Here we just exercise displayed math with no automatic numbering, and an elective number on the middle equation.

$$\begin{aligned}\frac{d}{dx}\int_a^x f(t)\,dt &= \frac{d}{dx}(F(x)-F(a))\\ &= \frac{d}{dx}F(x) - \frac{d}{dx}F(a) &(4.4)\\ &= f(x) - 0 = f(x)\end{aligned}$$

■

The alternative version of the Fundamental Theorem (FTC) in (4.1) is a compact way to express the result.

For testing purposes, there is a simple bare Sage Cell here.

```
c = 832
c
```

Example 4.2 A Mysterious Derivative! So if we define a function with its variable employed as a limit of integration, like so

$$K(z) = \int_{345}^{z} x^4 \sin(x^2)\,dx$$

then we get the derivative of that function so easily it seems like a mystery,

$$\frac{d}{dz}K(z) = z^4\sin(z^2).$$

That's it.

For testing purposes, there is a simple Sage Cell here, buried inside an example that should be a knowl (embedded in the page).

```
2+2
```

We test a Sage cell inside a knowl, which should set the value of a variable that will be available to subsequent cells within the knowl.

```
a = 6
a
```

```
b = a + 10
b
```

Even if you ran the cell at the top of this page, within this knowl the value of the variable `c` is not known, so the next cell will cause an error.

```
c + 400
```

▲

The Sage cells on a page will "remember" results computed elsewhere on the page. If you rely on this feature, remind your readers to evaluate all the necessary cells and that they perhaps need to be evaluated in a certain order.

```
c/2
```

There are some Sage cells in the previous (knowled) `<example>`. The results there are restricted to the knowl. In other words, the scope of those cells is the knowl. So if you opened the example and executed the Sage cells there, or if you skipped the example entirely, the next cell should not "know" the values of those variables and will raise an error.

```
a + b
```

We cross-reference the example just prior, Example 4.2, to test the simple Sage cells that will now be part of a cross-reference knowl (an external file).

Claim 4.3 An Equivalent Claim. *This claim is an equivalence: it is true if and only if it is correct.*

Proof. Our purpose here is to show how you can structure a proof with cases, such as an equivalence structured with the arrows typically used to demonstrate the two "directions" involved in the proof, by using the `@direction` attribute on a `<case>` element.

(⇒) Nulla non lectus suscipit, bibendum leo quis, dignissim justo. In urna turpis, tincidunt id elementum id, faucibus ac tellus.

(⇐) Quisque auctor ligula turpis, ut aliquam urna consectetur hendrerit. Aenean porta dolor et justo facilisis feugiat in sed sapien. Nullam porta ex et commodo semper.

Case 3b: The inductive step. A case may also have a `title`, whose formatting and structure is entirely up to the author. This then becomes the text of a cross-reference, as well.

Why Not Try This? A `<case>` (or any other element with a default title) did not always handle title-ending punctuation correctly. So we try an example title with a question mark.

(⇒) Necessity. If you like, you can have both indications.

Case. No direction, no title, then just a generic title.

■

Proof. We test here that punctuation at the end of the title of a proof is handled correctly. ■

Proof. This proof should fill exactly three lines (as of defaults in place 2018-12-31) and so the tombstone/Halmos should be on a fourth line, and then *flush right.* xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx. ■

Claim 4.4 A List of Equivalent Statements. *The following are equivalent.*

(i) This statement is equivalent to all those below.
(ii) This statement is equivalent to the statement above and to all those below.
(iii) This statement is equivalent to the two statements above and to the statement below.
(iv) This statement is equivalent to all those above.

Proof. Our purpose here is to show how you can structure a proof with cases to address the circular logic required to prove the equivalence of a list of statements, by using the `@direction` attribute on a `<case>` element. You should order the list of statements in the order that you would like to prove ``this statement implies the next.''

((i) ⇒ (ii)) Here we would prove that the first statement implies the second.

((ii) ⇒ (iii)) Here we would prove that the second statement implies the third.

((iii) ⇒ (iv)) The trickiest case. This time we include a title to describe the nature of this case. But we would still need to prove that the third statement implies the last.

((iv) ⇒ (i)) Wrap-around. Finally, we would complete the cycle of logic by proving that the last statement implies the first.

■

We can also use `@direction` set to `cycle` in a stand-alone proof of our TFAE claim. If we include a `@ref` on the `<proof>` that points to the original claim, then the formatting of the markers on the statement list will be honored in our cases.

Proof. Once again we will prove that the four statements in 4.4 are equivalent.

((i) ⇒ (ii)) Another argument that the first statement implies the second.

((ii) ⇒ (iii)) And another argument that the second statement implies the third.

((iii) ⇒ (iv)) Not so tricky this time. Why did we find it so difficult before to prove that the third statement implies the last?

((iv) ⇒ (i)) Wrap-around. And once more we complete the cycle of logic.

■

A couple more times to check that the default list markers get applied to the directional cases properly.

Claim 4.5 Another List of Equivalent Statements. *The* `<ol>` *that creates the list below does not have* `@marker`*.*

1. *This statement is equivalent to the two below.*
2. *This statement is equivalent to both the statement above and to the statement below.*
3. *This statement is equivalent to the two above.*

Proof. You know the drill by now.

(1 ⇒ 2) Does the first statement imply the second?

(2 ⇒ 3) The "middle" case. Does the second statement imply the third?

(3 ⇒ 1) And finally, does the third imply the first?

■

This proof includes a `@ref` to the preceding claim.

Proof.

(1 ⇒ 2) Does the first statement imply the second?

(2 ⇒ 3) The "middle" case. Does the second statement imply the third?

(3 ⇒ 1) And finally, does the third imply the first?

■

This proof does not include a `@ref`, and so the direction indicators get default markers.

Proof.

(1 ⇒ 2) Does the first statement imply the second?

(2 ⇒ 1) And finally, does the second imply the first?

■

4.2 A Pedagogical Note about Subsection 4.1

4.2.1 Symbolic and Numerical Integrals

The Fundamental Theorem explains why we use the same notation for a definite integral, which is a numerical calculation,[1] and an antiderivative, which is a symbolic expression.

Checkpoint 4.6 Essay Question: Compare and Contrast. Write a short paragraph which compares, and contrasts, the definite and indefinite integral.

[1] Which I think sometimes students lose sight of.

This is an exercise which sits in the midst of the narrative, so is formatted more like an example or a remark. It can have a hint and a solution, but this one does not. It can have a title, which this one does.

Hint. Start writing!

4.2.2 Advice

Using an "integral sign" for an antiderivative (aka indefinite integral) would seem to make the Fundamental Theorem a *fait accompli*. So I would suggest not conflating the notation for two very different things until the Fundamental Theorem exposes them as being highly related.

Example 4.7 An Example of Structure. This is an example of an example with a bit more structure. Specifically, the example has a `title`, as usual, but then has a `statement`, which is separate from the `solution`. Why did we implement an example in two ways?

Solution. Authors asked for it and it seemed a very natural thing to do, even if we only had an unstructured version for a long time.

▲

Question 4.8 An Example of a Question. Any kind of question can be marked as such with `<question>`. Or similarly, as a `<problem>`. They behave identically to `example`s, such as the one preceding and are numbered along with theorems, examples. etc.

Solution 1. You can have a solution. Or several, even if you don't ask a question.

Solution 2. See?

▲

Checkpoint 4.9 An Inline Exercise. There are lots of exercises in this sample article, but mostly they are in special exercise sections. Sometimes you just want to sprinkle some exercises through the narrative. We call these **inline exercises**, in contrast to **divisional exercises**. The inline exercises look a bit more like a theorem or definition, with titles and fully-qualified numbers.

These may also have hints, answers and solutions.

Hint. A good hint.

Answer. 42.

Solution. If your exercise feels like proving a theorem, then you might want to make some comments, but also clearly delineate which part of the solution is a the complete proof.

Proof. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin lorem diam, convallis in nulla sed, accumsan fermentum urna. Pellentesque aliquet leo elit, ut consequat nunc dapibus ac. Sed lobortis leo tincidunt, vulputate nunc at, ultricies leo. Vivamus purus diam, tristique laoreet purus eget, mollis gravida sapien. Nunc vulputate nisl ac mauris hendrerit cursus. Sed vel molestie velit. Suspendisse sem sem, elementum at vehicula id, volutpat ac mi. Nullam ullamcorper fringilla purus in accumsan. Mauris at nunc accumsan orci dictum

vulputate id id augue. Suspendisse at dignissim elit, non euismod nunc. Aliquam faucibus magna ac molestie semper. Aliquam hendrerit sem sit amet metus congue tempor. Donec laoreet laoreet metus, id interdum purus mattis vulputate. Proin condimentum vitae erat varius mollis. Donec venenatis libero sed turpis pretium tempor.

Praesent rutrum scelerisque felis sit amet adipiscing. Phasellus in mollis velit. Nunc malesuada felis sit amet massa cursus, eget elementum neque viverra. Integer sagittis dictum turpis vel aliquet. Fusce ut suscipit dolor, nec tristique nisl. Aenean luctus, leo et ornare fermentum, nibh dui vulputate leo, nec tincidunt augue ipsum sed odio. Nunc non erat sollicitudin, iaculis eros consequat, dapibus eros. ■

Example 4.10 An Example of with $\frac{1}{2}$ math formula $\int e^x\,dx$ in the title. Just for testing math in knowls, and also extra whitespace in a `<p>`. ▲

There are many different blocks you can employ, and they mostly behave the same way. A `<project>` is very similar to a `<question>` or `<problem>`

Project 4.1 Start Exploring PreTeXt. You could grab the `minimal.xml` file from the `examples/minimal` directory and experiment with that.

Projects get their own independent numbering scheme, since they may be central to your textbook, workbook, or lab manual. If you process this sample article with level for project numbering set to `0` then you will get consecutive numbers from the beginning of your book, starting with 1.

Exploration 4.2 Exploring Explorations. This is an `<exploration>`. Other similar possibilities are `<project>`, `<activity>`, `<task>`, and `<investigation>`.

Note that projects, activities, explorations, tasks and investigations *share* the independent numbering scheme, so it is really only intended you use one of these. If you want a variant of the name (e.g. "Directed Activity") you can use the `<rename>` facility (Subsection 33.1).

Solution. This is a "solution" to the exploration. In practice, you might choose to not make this visible for students, but instead include it as part of some guidance you might provide to instructors (e.g. an *Instructor's Manual*).

Activity 4.3 Hints, Answers, Solutions. Another variant of these project-like items is to possibly include a `<hint>` and an `<answer>` before the `<solution>`.

Hint. Just a little help.

Answer. The result, but no help in getting there.

Solution. Everything to get it all done, in detail.

Note 4.11 A Note on Remarks. `<remark>`, `<convention>`, `<note>`, `<observation>` and `<warning>` are designed to hold very simple contents, with no additional structure (no proofs, no solutions, etc.).

But they do carry a title and a number, can be the target of a cross-reference, and may be optionally knowlized in HTML with the `html.knowl.remark` processing switch.

And distinctly different from a <note> in a <biblio>[2].

Aside An Aside with a *Formatted* Title. An <aside> is similar to a remark, but is not as critical to the narrative. It is not numbered, and so requires a title. It can be the target of a cross-reference. They are meant to be short, and so are not knowlized at their first appearance. If the content is appropriate, these can be marked as <historical> or <biographical>, though longer items should use subdivisions (e.g. sections, subsections) instead.

An <exercise> can be structured with <task>.

Checkpoint 4.12 A very structured exercise. This is an over-arching introduction to the whole exercise. We follow with some tasks.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(a) A super-simple task. This first task is very simple, just a paragraph. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(b) Now three paragraphs. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(c) A title of a task that has a subtask with an <answer> for the Solutions. This second task is further divided by more tasks. This is its introduction. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(i) A task with a title and an <answer> for the Solutions. A really simple subtask. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

A short paragraph, before an answer.

Answer. With a proof.

[2] A gratuitous footnote to test prior bug confusing this with a <note> in a <biblio>.

Proof. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis. ■

And a bit more to say.

(ii) A subtask with an answer. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Answer. Right! In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(iii) Three simple sub-sub-tasks. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(A) First subsubtask. Short paragraph.

(B) A second three-deep subsubtask! Second subsubtask. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(C) Third subsubtask. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

The conclusion of the structured subtask. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(iv) A simple task as the last subtask. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

This concludes our structured second task. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla

blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(d) This third top-level task is intermediate in complexity, you are reading the `statement`, which is followed by more items. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Hint. One hint. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Answer 1. First answer. In interdum suscipit ullamcorper.

Answer 2. Second answer. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Solution. At last, the solution. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

This is a conclusion where you could summarize the exercise. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

The following `<project>` is nearly identical to the preceding `<exercise>`.

Project 4.4 A very structured project. This is an over-arching introduction to the whole project. We follow with some tasks. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(a) This first task is very simple, just a paragraph. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(b) Now three paragraphs. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus,

malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(c) This second task is further divided by more tasks. This is its introduction. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

> **(i)** A really simple subtask. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.
>
> A short paragraph, before an answer.
>
> **Answer**. With a proof.
>
> *Proof.* In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis. ■
>
> And a bit more to say.
>
> **(ii)** A subtask with an answer. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.
>
> **Answer**. Right! In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.
>
> **(iii)** Two simple sub-sub-tasks. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.
>
> > **(A)** First subsubtask. Short paragraph.
> >
> > **(B)** Second subsubtask. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.
> >
> > In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada

mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(C) Third subsubtask. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

The conclusion of the structured subtask. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(iv) A simple task as the last subtask. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

This concludes our structured second task. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(d) This third top-level task is intermediate in complexity, you are reading the `statement`, which is followed by more items. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Hint. One hint. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Answer 1. First answer. In interdum suscipit ullamcorper.

Answer 2. Second answer. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Solution. At last, the solution. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

This is a conclusion where you could summarize the project. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

The following `<example>`, from Elise Desgreniers, is structured with `<task>`.

Example 4.13 Notation mathématique 2.
Écrivez chacun des exemples suivants avec les conditions pertinentes.

(a) Soit l'ensemble $A = \{1, 2, 3, 4, 5\}$.
On constate que cet ensemble contient uniquement des entiers positifs allant de 1 à 5.
Donc, on peut écrire $A = \{x \mid x \in \mathbb{N} \text{ et } 1 \leq x \leq 5\}$.

(b) $\{3, 6, 9, 12, 15, \ldots, 27, 30\}$

Hint. Ce sont des multiples de 3.

Answer. $\{x \mid x = 3y \text{ et } 1 \leq y \leq 10 \text{ et } y \in \mathbb{N}\}$

(c) $\{1, 3, 5, 7, 9, 11, ...\}$

Hint. Ce sont des nombres impairs.

Answer. $\{x \mid x \mod 2 = 1 \text{ et } x \in \mathbb{N}\}$

(d) $\{2, 3, 5, 7, 11, 13, 17, 19, 23, ...\}$

Hint. Ce sont des nombres premiers.

Answer. $\{x \mid x \text{ est un nombre premier positif}\}$

(e) $\{1, 4, 9, 16, 25, 36, \ldots, 961\}$

Hint. Ce sont des carrés parfaits.

Answer. $\{x \mid x = y^2 \text{ et } 1 \leq y \leq 31 \text{ et } y \in \mathbb{N}\}$

(f) $\{1, 8, 27, 64, 125, ...\}$

Hint. Ce sont des cubes parfaits.

Answer. $\{x \mid x = y^3 \text{ et } y \in \mathbb{N}^*\}$

▲

Notes or examples related to computation or technology can go in blocks of the same name.

Technology 4.14 Sample Use of Sage. This would be a good place to talk about Sage, including a cell or two.

```
diff(x^4, x)
```

```
4*x^3
```

But you might want to describe how to use some other calculator, or maybe some numerical method.

A <paragraphs> with a <project> with an <answer>. The solutions to a project (and similar) once did not migrate to the automatically-generated solutions.

Project 4.5. A simple project, no tasks, just an answer.

Answer. Here's the answer we are looking for.

4.2.3 Exercises

1. This is an exercise in an "Exercises" subdivision at the level of a subsubsection. There is no question other than if the numbering is appropriate. Here is a self-referential link: Exercise 4.2.3.1.

The subsubsection has no title in the source, so one is provided automatically, and will adjust according to the language of the document.

Solution. This solution will migrate to a list of solutions in the backmatter. We include a `sidebyside` as a test.

This is a skinny paragraph which should be just 30% of the width.

And another skinny paragraph which should also be just 30% of the width.

2. An `<exercise>` can be structured with parts, called `<task>`. This is the `<introduction>`.

(a) Do this.

(b) And the other thing.

4.2.4 Reading Questions

A set of reading questions may have an `<introduction>`, perhaps for preparatory explanation.

If a student has logged in to the HTML version, then they can answer the reading questions directly in the book. Inline math LaTeX can be entered using `$...$` or `\(...\)` delimiters, and inline AsciiMath using backticks `...` as delimiters. Here are some `gratuitous backticks` to check that AsciiMath is only active in the answers to reading questions.

1. This is a **reading question** that you might have a student answer prior to a class session, based on reading part of the book. A quick glance before class can help you tailor class time to the specific needs of your students. The perfect reading question will reveal whether the student has read and understood the material, and will be difficult to answer if they have not. What do you think of that?

2. And a second one, with a cross-reference to the first, as a check on numbering: Reading Question 4.2.4.1. Reading questions are allowed to have answers, but providing answers misses the point of a reading question, and the

answer knowl interacts poorly with the mechanism used to allow students to answer directly in the book. Do you think the schema should ban answers to reading questions?

And for symmetry, a `<conclusion>`.

4.2.5 Glossary

A glossary may have a `<headnote>`, perhaps with some explanation. This glossary is a specialized division within a section. Placement in the back matter is another option, see the Glossary.

bar. A part of **foobar**. See foobar.

foobar. A synonym for the acronym FUBAR.

4.2.6 Solutions for This Subsection

This is an introduction, where you might explain that this division of this subsection contains various hints, answers, solutions of inline exercises, divisional exercises, and/or project-like blocks. See the source to see just how this solutions division was built.

4.2.1 · Symbolic and Numerical Integrals

Checkpoint 4.6 Essay Question: Compare and Contrast.

Hint. Start writing!

4.2.2 · Advice

Checkpoint 4.9 An Inline Exercise.

Hint. A good hint.

Answer. 42.

Solution. If your exercise feels like proving a theorem, then you might want to make some comments, but also clearly delineate which part of the solution is a the complete proof.

Proof. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin lorem diam, convallis in nulla sed, accumsan fermentum urna. Pellentesque aliquet leo elit, ut consequat nunc dapibus ac. Sed lobortis leo tincidunt, vulputate nunc at, ultricies leo. Vivamus purus diam, tristique laoreet purus eget, mollis gravida sapien. Nunc vulputate nisl ac mauris hendrerit cursus. Sed vel molestie velit. Suspendisse sem sem, elementum at vehicula id, volutpat ac mi. Nullam ullamcorper fringilla purus in accumsan. Mauris at nunc accumsan orci dictum vulputate id id augue. Suspendisse at dignissim elit, non euismod nunc. Aliquam faucibus magna ac molestie semper. Aliquam hendrerit sem sit amet metus congue tempor. Donec laoreet laoreet metus, id interdum purus mattis vulputate. Proin condimentum vitae erat varius mollis. Donec venenatis libero sed turpis pretium tempor.

Praesent rutrum scelerisque felis sit amet adipiscing. Phasellus in mollis velit. Nunc malesuada felis sit amet massa cursus, eget elementum neque viverra. Integer sagittis dictum turpis vel aliquet. Fusce ut suscipit dolor, nec

tristique nisl. Aenean luctus, leo et ornare fermentum, nibh dui vulputate leo, nec tincidunt augue ipsum sed odio. Nunc non erat sollicitudin, iaculis eros consequat, dapibus eros. ■

Exploration 4.2 Exploring Explorations.

Solution. This is a "solution" to the exploration. In practice, you might choose to not make this visible for students, but instead include it as part of some guidance you might provide to instructors (e.g. an *Instructor's Manual*).

Activity 4.3 Hints, Answers, Solutions.

Hint. Just a little help.

Answer. The result, but no help in getting there.

Solution. Everything to get it all done, in detail.

Checkpoint 4.12 A very structured exercise.

(c) A title of a task that has a subtask with an `<answer>` for the Solutions.

(i) A task with a title and an `<answer>` for the Solutions.

Answer. With a proof.

Proof. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis. ■

And a bit more to say.

(ii)

Answer. Right! In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(d)

Hint. One hint. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Answer 1. First answer. In interdum suscipit ullamcorper.

Answer 2. Second answer. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Solution. At last, the solution. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Project 4.4 A very structured project.

(c)

(i)

Answer. With a proof.

Proof. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis. ■

And a bit more to say.

(ii)

Answer. Right! In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(d)

Hint. One hint. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Answer 1. First answer. In interdum suscipit ullamcorper.

Answer 2. Second answer. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Solution. At last, the solution. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Project 4.5.

Answer. Here's the answer we are looking for.

4.2.3 · Exercises

1.

Solution. This solution will migrate to a list of solutions in the backmatter. We include a `sidebyside` as a test.

This is a skinny paragraph which should be just 30% of the width.

And another skinny paragraph which should also be just 30% of the width.

And a conclusion to this solutions division, which may not be readily apparent as distinct from the final division's worth of solutions, but since it is not prefixed with a number, it may be different enough.

4.3 Theorem-Like Environments

There are a variety of pre-defined environments in PreTeXt. All take a title, and must have a statement. Some have proofs (theorems, corollaries, etc.), while some do not have proofs (conjectures, axioms, principles).

Principle 4.15 The Title Principle. It is a fundamental principle that many elements can have a title. Try it and see. If you get better formatting, then it is being recognized. If it looks very plain, check the documentation and perhaps make a feature request.

More precisely, `<theorem>`, `<corollary>`, `<lemma>`, `<algorithm>`, `<proposition>`, `<claim>`, `<fact>`, and `<identity>`, all behave exactly the same, requiring a statement (as a sequence of paragraphs) followed by an optional proof, and may have an optional title. The elements `<axiom>`, `<conjecture>`, `<principle>`, `<heuristic>`, `<hypothesis>`, and `<assumption>` are functionally the same, barring a proof (since they would never have one!). Definitions are an exception, as it is natural to place `<notation>` within—see the source for Definition 2.2 for an example.

4.4 Linking Sage Cells

Sage cells share their results on a per-webpage basis, or a per-knowl basis, so if you move to a new chapter, section, or subsection that happens to be on another webpage, your Sage computations are gone and you start fresh. But maybe you need some results from elsewhere. As an author, you can make an exact copy of a cell in another location by placing the code in an external file, which is pure text, freed from any need to format for XML processing. So, in particular, there is no need to escape ampersands and angle brackets, nor is

there employment of the CDATA mechanism. But the real value is that there is just one version to edit, and any changes will be reflected in both copies. We demonstrate this in the sample book, since it has the xinclude mechanism in place. In the chapter on groups, find the section on Sage and then find the discussion of subgroups, and you will find an example of two identical Sage cells produced from one source file.

You can also specify certain cells to be auto-evaluated, by setting the @auto-evaluate attribute to yes. The resulting cell will not have a button for evaluation (so editing it would be pointless). See the source of this sample article for the two examples below.

2023-08-17: support just now is for the use case of a small portion of code, not a huge library of helper routines.

Two cells with the default language sage.

```
k = 404
```

```
2*k
```

Two cells with language python.

```
k = 4112
print("Assigned an initial value to the variable k")
```

```
k = 2*k
print(k)
```

4.5 Hierarchy

Structure. This section of this article has subsections and subsubsections. In a book you can have chapters enclosing multiple sections. There is one finer subdivision, it is achieved with the paragraphs element.

It is basically a sequence of paragraphs, where the first one gets an inline title. You are reading the second, and final, paragraph of one right now. It is useful for organizing very short documents, where numbered subdivisions might be overkill.

A Second Paragraphs. This is a second consecutive paragraphs element, so should seem related to its title, but distinct from the two paragraphs in the grouping with the title “Structure” immediately prior.

Assemblages: Collections and Summaries.

An `<assemblage>` is a collection, or summary, that does not have much structure to it. So you are limited to paragraphs and friends (`p`, `blockquote`, `pre`) and side-by-sides that do not contain captioned items (`sidebyside`, `sbsgroup`). The intent is that contents are not numbered, so cannot be cross-referenced individually, and so also do not become knowls. You may place `<image>`, `<tabular>`, and `<program>` inside a `<sidebyside>`, in addition to other objects that do not have captions. Note that `p` may by extension contain lists (`ol`, `ul`, `dl`). Despite limited structure, the presentation should draw attention to it, because the contents should be seen as more important in some way. It should be "highlighted" in some manner. If you need to connect the entire assemblage with material elsewhere, you can do that with the usual `xref/xml:id` mechanism.

What have we seen so far in this (disorganized) sample?

- Theorems, definitions and corollaries. (Section 2)
- Sage cells, including with R. (Section 3)
- Lots of document structure, like introductions and conclusions (next). (Section 4)

A sample table, as a `tabular` inside a `sidebyside` with no caption, follows.

A	B	C
Uno	Dos	Tres

This is a small `assemblage` with no title, simply to make sure the surrounding box behaves properly, especially for LaTeX output.

Assemblages containing $\mu\forall\tau\mathbb{H} = \emptyset\kappa$.

It is acceptable for an assemblage to contain mathematical content, even in its title.

4.6 Introductions and Conclusions

Any subdivision may have a sequence of paragraphs within an `<introduction>` that precedes subsequent further subdivisions. You are reading one now. They are always leaves of the document structure, so are rendered on some pages that reference the following subdivisions.

An introduction or conclusion is an extremely restrictive container with simple presentation, and it may not have a title. Content is meant to be short and unstructured, in particular, nothing that can be numbered is allowed. If this feels *too* restrictive, then place your content in an initial numbered subdivision and perhaps title it "Introduction". Or make your entire subdivion unstructured and place whatever you want into it.

This ends this introduction to introductions.

4.6.1 Test One

An intervening subsubsection just after an introduction.

4.6.2 Test Two

An intervening subsection section which contains an `<exercises>` division which must be at the level of a Subsubsubsection.

Checkpoint 4.16. An inline exercise to examine any clash with divisional exercises below.

Answer. An answer so there is something to appear in a `<solutions>`.

4.6.2 What Did You Learn?

1. A mock exercise to appease validation.

Answer. An answer so there is something to appear in a `<solutions>`.

2. And a second to help with formatting the division heading.

4.6.2 Exercises

1. A mock exercise to appease validation.

Answer. An answer so there is something to appear in a `<solutions>`.

2. And a second to help with formatting the division heading.

4.6.3 Test Three

An intervening subsubsection just before a conclusion.
Entirely analogous to introductions are conclusions. Any subdivision may have a sequence of paragraphs within a `<conclusion>` that follows previous further subdivisions. You are reading one now. They are always leaves of the document structure, so are rendered on some pages that reference the preceding subdivisions.

This concludes this conclusion (and this subsection and this section).

4.7 Some Paragraph-Level Markup

Text within a paragraph may be *emphasized* with `<em>` or if you want to take it to the next level you can identify the text as an ***alert*** with `<alert>`.

Similarly, within a paragraph, you can identify edits between versions as inserted text that has been added with `<insert>` or as ~~deleted text that has been removed~~ with `<delete>`. Note that these identified edits are slightly different than ~~stale text that you want to retain, but which is no longer relevant~~, which is accomplished with `<stale>`. The original request for stale text came from an instructor with an online list of student topics for presentations, and as students claimed topics they were marked as no longer available for other students.

If you need a "fill-in blank", like this _____, it can be obtained with an empty `<fillin>` element that defaults to roughly a 10-character width. You can use the `@characters` attribute to make the rule longer or shorter, such as a 40-character blank: _____. The character count is approximate, based on typical character widths within a proportional font carrying English language text. Adjust to suit, or request a language-specific adjustment if it is critical.

This paragraph is intended to make a `<fillin>` appear right at the start of _____ the second line in print and then the next paragraph has nothing but a `<fillin>`. Both are for testing purposes.

_____

The following are `<fillin>` with `@rows` and/or `@cols` attributes (at least one of which is greater than 1): _____, _____, _____.

Long after we started this mess, we added PreTeXt tags to mark up tags and attributes. The elements are: `<tag>`, `<tage>`, `<attr>`. Examples of how these render are (respectively): `<section>`, `<hash/>`, `@width`. Perhaps this document will make greater use of these tags.

We supply two provisional cross-references for testing purposes only: `[provisional cross-reference: a first incomplete cross-reference]`, `[provisional cross-reference: a second incomplete cross-reference]`. A conclusion here, which we fill with some numbering tests.

This is a cross-reference to one of the outcomes, forced to use the `type-global` form of the text. It should describe the outcome as belonging to the *section* (rather than the *outcomes*), since outcomes are one-per-subdivision and are numbered based upon the containing division: Outcome 2 of Section 4. For comparison this is the (forced) `type-global` cross-reference: Outcome 4.2.

Outcomes 4 Fundamental Structures, Revisited

This is a `<outcomes>` element you are reading, and this is its introduction. This early section has really grown and we have tried to accomplish many things. Not all of them are listed here.

1. Display various "blocks", fundamental units of the flow.
2. More, and this is what the cross-references above are pointing to.
3. Evermore.

This concludes the (incomplete) outcomes for this section, so now we can carry-on to the next section.

5 Some Facts and Figures

Because of the Fundamental Theorem[1], for every derivative we know, there is an antiderivative we might find useful. Because of the Fundamental Theorem of Calculus[2], we recycle the "$\int$" symbol as notation for an antiderivative.

- Derivatives
  (a) $\dfrac{d}{dx}x^n = nx^{n-1}$
  (b) $\dfrac{d}{dx}e^x = e^x$
- Antiderivatives
  i) $\displaystyle\int x^n\,dx = \frac{x^{n-1}}{n+1}$ if $n \neq -1$
  ii) $\displaystyle\int e^x\,dx = e^x$

[1] First test footnote
[2] Second test footnote

(c) $\dfrac{d}{dx}\cos(x) = -\sin(x)$ iii) $\displaystyle\int \sin(x)\,dx = -\cos(x)$

Remark 5.1. You can[3] gain a greater understanding of derivatives by studying the graphs of functions with their derivatives. Can[4] you discern the derivative–antiderivative[5] relationship in Figure 5.2?

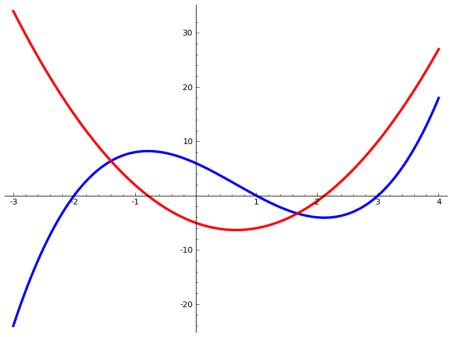


Figure 5.2. A function and its derivative

Lists[6] can have multiple columns. With HTML items displayed in row-major order (horizontally first) and[7] with LaTeX items are displayed in column-major order (vertically first). When one order, or the other, becomes workable in both variants, maybe we will be consistent in presentation. (Note that with just one row, it makes no difference.) We used it above for the two items—derivatives and integrals—where each item was a list of its own. Here are two more examples, one with short snippets and lots of columns, the other with lots of text in paragraphs.

1. Red
2. Blue
3. Green
4. Purple
5. Yellow
6. Black
7. Orange
8. Pink
9. Salmon
10. Aqua
11. Cyan
12. Puce

- Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin lorem diam, convallis in nulla sed, accumsan fermentum urna. Pellentesque aliquet leo elit, ut consequat nunc dapibus ac. Sed lobortis leo tincidunt, vulputate nunc at, ultricies leo. Vivamus purus diam, tristique laoreet purus eget, mollis gravida sapien. Nunc vulputate nisl ac mauris hendrerit cursus. Sed vel molestie velit. Suspendisse sem sem, elementum at vehicula id, volutpat ac mi. Nullam ullamcorper fringilla purus in accumsan. Mauris at nunc

- Donec vestibulum auctor nisl. Nullam placerat interdum dui. Quisque lobortis scelerisque augue imperdiet placerat. Maecenas ultricies massa tempor, laoreet urna a, eleifend enim. Integer sed suscipit odio. Pellentesque non dapibus diam, eget tempus dui. Maecenas sollicitudin magna viverra, egestas velit nec, tristique sem. Cras iaculis mattis dui ac cursus. Integer volutpat, urna vel tempus convallis, erat nisi consectetur turpis, id varius dolor lorem vitae mauris. Phasellus erat orci, laoreet commodo gravida

[3] Third test footnote
[4] Fourth test footnote
[5] Fifth test footnote
[6] Sixth test footnote
[7] Seventh test footnote

accumsan orci dictum vulputate id id augue. Suspendisse at dignissim elit, non euismod nunc. Aliquam faucibus magna ac molestie semper. Aliquam hendrerit sem sit amet metus congue tempor. Donec laoreet laoreet metus, id interdum purus mattis vulputate. Proin condimentum vitae erat varius mollis. Donec venenatis libero sed turpis pretium tempor.

Praesent rutrum scelerisque felis sit amet adipiscing. Phasellus in mollis velit. Nunc malesuada felis sit amet massa cursus, eget elementum neque viverra. Integer sagittis dictum turpis vel aliquet. Fusce ut suscipit dolor, nec tristique nisl. Aenean luctus, leo et ornare fermentum, nibh dui vulputate leo, nec tincidunt augue ipsum sed odio. Nunc non erat sollicitudin, iaculis eros consequat, dapibus eros.

- Ut nec sem vitae ipsum interdum vestibulum sit amet sed velit. Aliquam tempor nibh vitae augue pulvinar, at ultricies urna commodo. Donec in porta lectus, ac sagittis felis. Vestibulum tincidunt quis metus facilisis luctus. In lobortis lacus vel ornare vehicula. Duis aliquet, ligula semper sodales adipiscing, augue nibh ornare ante, quis pulvinar justo mi eget mi. Mauris varius imperdiet vehicula. Duis dignissim magna quis velit mattis, in cursus lectus vehicula. Morbi quis tempus felis, ut gravida nisi.

quis, congue in lacus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Praesent at bibendum turpis. Pellentesque est nisl, dapibus at sagittis non, ultricies in nunc. Etiam ipsum arcu, porta sed feugiat eget, facilisis nec libero. Mauris tempor convallis felis.

Cras iaculis sapien elit, at convallis ligula convallis nec. Duis ante tortor, euismod a libero vitae, ornare viverra purus. Pellentesque facilisis urna a velit volutpat, in malesuada tortor porttitor. Sed vehicula mauris id lectus dignissim, eget consectetur dui pellentesque. Sed vel quam molestie, euismod ligula ac, venenatis arcu. Fusce sit amet sapien non urna dignissim tempus in vitae metus. Aliquam arcu turpis, mattis non libero eu, lacinia feugiat turpis. Phasellus rhoncus lacinia lacus facilisis ullamcorper. Praesent hendrerit accumsan neque, eu dignissim est consequat sed. Nulla facilisi. Proin at mi scelerisque, scelerisque felis ut, tristique diam. Proin in leo in lorem porttitor varius. Praesent condimentum in dui sit amet blandit. In imperdiet blandit congue.

- Sed varius leo urna. Phasellus tempus mollis ultricies. Curabitur non neque aliquet, facilisis tortor in, sodales dui. Donec hendrerit ultricies nulla mollis rhoncus. In vel lobortis est. Vestibulum consectetur lacus vel sem dignissim vestibulum. Etiam sed elementum ligula, vel congue turpis. Morbi nec diam mattis, venenatis eros et, elementum tellus. Integer sed orci ornare, elementum elit id, lacinia augue. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; In et libero id turpis pharetra faucibus. Integer consequat dignissim

Vivamus eu commodo est, pretium fringilla dolor. Curabitur vel sollicitudin libero. Integer sit amet auctor felis. Maecenas sagittis erat at ante feugiat, in tincidunt ligula pretium. Integer eget auctor ipsum, quis volutpat felis. Morbi id dignissim eros. Suspendisse aliquet pulvinar lorem gravida egestas. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Praesent nec massa dui. Suspendisse convallis lacus sit amet adipiscing varius. Suspendisse tempus diam vitae justo ornare, in condimentum metus pharetra. Curabitur sem dolor, auctor vitae sagittis vestibulum, posuere imperdiet metus. Etiam pretium lacus urna, vel auctor diam tincidunt non. Etiam viverra sodales iaculis.

semper. Donec pretium magna at ullamcorper ultricies. Nam quis suscipit elit. Donec cursus tellus et venenatis feugiat. Mauris dictum molestie leo, vitae aliquet metus luctus vitae.

Ut id iaculis leo. Sed nec vestibulum mi. Mauris est mauris, porta in nulla eget, bibendum luctus nisl. Praesent et posuere felis, molestie vehicula velit. Nulla a nunc venenatis, aliquam orci nec, congue felis. Vestibulum a dolor nisi. Morbi sed nisi nulla. Nam iaculis felis a enim blandit, at venenatis diam congue. Nulla augue diam, egestas eget fermentum nec, posuere eget risus. Praesent egestas nulla eros, eget accumsan augue euismod vel. Pellentesque pellentesque non erat vitae posuere. Curabitur lacus arcu, varius sed risus ut, ullamcorper tincidunt lorem. Sed et lacus dignissim, tincidunt nisl ac, porttitor sapien.

Example 5.3 Footnotes in Nested Boxes. An `<example>` is implemented as one box in print, and this one[8] holds two more kinds of boxes, each holding a footnote of its own.

Aside A Box within a Box. An `<aside>` is a second box, so this footnote[9] lives two boxes deep.

A description list makes a third kind of box.

Depth Two boxes down once more,[10] now in a list item.

▲

6 Some Advanced Ideas

The multi-row displayed mathematics in the proof of the Fundamental Theorem had equations aligned on the equals signs via the & character. Sometimes you don't want that. Here is an example with some differential equations, with each equation centered and unnumbered,

$$\mathcal{L}(y')(s) = s\mathcal{L}(y)(s) - y(0) = sY(s) - y(0)$$
$$\mathcal{L}(y'')(s) = s^2\mathcal{L}(y)(s) - sy(0) - y'(0) = s^2Y(s) - sy(0) - y'(0).$$

[8] Eighth test footnote, one box deep

[9] Ninth test footnote, in an aside two boxes deep

[10] Tenth test footnote, in a description list item two boxes deep

Just prior to this sentence, in the middle of this paragraph, is an <idx> and a <notation>, adjacent, but separated by some whitespace in the authored source. That insignificant whitespace will be removed akways, which will be a (slightly) noticeable improvement in the LaTeX output. We test referencing notation here, placed *before* the sentence-ending period and right after some inline mathematics—for $\mathbb{Z}_n$.

LaTeX has a device where you can interrupt a sequence of equations with a small amout of text and preserve the equation alignment on either side. Here are two tests of that device, with aligned equations and non-aligned equations. Study the source to see use and differences. (The math does not make sense.)

Aligned and numbered first.

$$\mathcal{L}(y')(s) = s\mathcal{L}(y)(s) - y(0) = sY(s) - y(0) \tag{6.1}$$
$$\mathcal{L}(y'')(s) = s^2\mathcal{L}(y)(s) - sy(0) - y'(0) = s^2Y(s) - sy(0) - y'(0). \tag{6.2}$$

And so it follows that,

$$\mathcal{L}(y')(s)^{++} = s\mathcal{L}(y)(s) - y(0) = sY(s) - y(0) \tag{6.3}$$
$$\mathcal{L}(y'')(s)^{5} = s^2\mathcal{L}(y)(s) - sy(0) - y'(0) = s^2Y(s) - sy(0) - y'(0). \tag{6.4}$$

Now with no numbers and no alignment. We include two cross-references in the intertext portion for testing.

$$\mathcal{L}(y')(s) = s\mathcal{L}(y)(s) - y(0) = sY(s) - y(0)$$
$$\mathcal{L}(y'')(s) = s^2\mathcal{L}(y)(s) - sy(0) - y'(0) = s^2Y(s) - sy(0) - y'(0).$$

First an external reference to example.com and internal cross-reference to Corollary 4.1. And so it follows that,

$$\mathcal{L}(y')(s)^{++} = s\mathcal{L}(y)(s) - y(0) = sY(s) - y(0)$$
$$\mathcal{L}(y'')(s)^{5} = s^2\mathcal{L}(y)(s) - sy(0) - y'(0) = s^2Y(s) - sy(0) - y'(0).$$

Tables can get quite complex. Simple ones are simpler, such as this example of numerical computations for Euler's method in just a bit.

But first we make a figure with two very simple tables next to each other. This causes the very first instance of <table> to actually be a "subtable", which exposes a bug provoked by Emiliano Vega and fixed around 2020-08-06. (So we have to place this early to create the same behavior that exposed the bug.)

Table 6.1(a). First

One

Table 6.1(b). Second

Two

Figure 6.1. Buggy sub-tables

Table 6.2. Euler's approximation for Duffing's Equation with $h = 0.2$

i	t_i	x_i	y_i
0	0.00	0.0000	0.5000
1	0.20	0.1000	0.4800
2	0.40	0.1960	0.4560
3	0.60	0.2872	0.4295
4	0.80	0.3731	0.4027
5	1.00	0.4536	0.3783
6	1.20	0.5293	0.3591
7	1.40	0.6011	0.3480
8	1.60	0.6707	0.3474
9	1.80	0.7402	0.3603
10	2.00	0.8123	0.3900

7 Mathematics

To be able to create both LaTeX and HTML output (plus variations), we rely on MathJax, which in turn supports an extensive subset of the mathematical symbols normally available. The AMSMath symbol set is a good approximation. The PreTeXt Guide has a link to the complete list of macros supported by MathJax. We load the `AMSsymbols` library.

7.1 Basic Mathematics

The following is from the MathJax demonstration page, an identity due to Ramanujan:

$$\frac{1}{\left(\sqrt{\phi\sqrt{5}}-\phi\right)e^{\frac{2}{5}\pi}} = 1+\frac{e^{-2\pi}}{1+\frac{e^{-4\pi}}{1+\frac{e^{-6\pi}}{1+\frac{e^{-8\pi}}{1+\cdots}}}}$$

And again, from the MathJax demonstration page, Maxwell's equations:

$$\begin{aligned}\nabla\times\vec{\mathbf{B}}-\frac{1}{c}\frac{\partial\vec{\mathbf{E}}}{\partial t} &= \frac{4\pi}{c}\vec{\mathbf{j}}\\ \nabla\cdot\vec{\mathbf{E}} &= 4\pi\rho\\ \nabla\times\vec{\mathbf{E}}+\frac{1}{c}\frac{\partial\vec{\mathbf{B}}}{\partial t} &= \vec{\mathbf{0}}\\ \nabla\cdot\vec{\mathbf{B}} &= 0\end{aligned}$$

Look back at the top of the source file of this document to see how to include your TeX macros just once. For best results keep your macros simple and semantic.

PreTeXt once provided modest built-in support for "slanted", or "beveled", or "nice" fractions. To wit, we mean fractions such as: 3/8. Use the pre-defined `\sfrac{}{}` macro in your mathematics to achieve this presentation. The presentation in HTML is subpar, but could improve as MathJax provides support. It is now an author's responsibility to add support for superior

typesetting for PDF output by loading the `xfrac` LaTeX package with the following in `<docinfo>`:

```
<math-package latex-name="xfrac" mathjax-name=""/>
```

which is what we have done here as a test. See the Guide for more details.

We consider a system of equations. We number the first and last equation (there are just two) and include an `@xml:id` attribute on each. We reference the whole system later as the range of equations from the first to the last.

$$\frac{dx}{dt} = x^2 - 4x - y + 4 \tag{7.1}$$
$$\frac{dy}{dt} = x^3 - y. \tag{7.2}$$

7.2 Displayed Mathematics

Display mathematics is achieved with the `<md>` tag ("math display"), always used within a paragraph (`<p>`). As a good example of how XML syntax is superior, you author n lines of equations by enclosing each line inside of a `<mrow>` tag, rather than using $n-1$ separators (such as \\).

If you use no ampersands to express alignment (read ahead), then each equation is centered independently on the width of the text. This is implemented according to the AMSmath LaTeX package's `gather` environment. Example:

$$\frac{dx}{dt} = x^2 - 4x - y + 4$$
$$\frac{dy}{dt} = x^3 - y.$$

An ampersand is used, in two ways, to describe positioning several equations per line, organized in columns. We have created the pre-defined LaTeX macro `\amp` as one way specify these, but the escape sequence `&` may be used also. The second, fourth, sixth, … ampersands separate columns, and the spacing between columns will be provided automatically. The first, third, fifth, … ampersands are alignment points for the equations in each column. Typically this is placed just prior to a binary operator, such as an equal sign (`\amp =`), or for a column of explanations or commentary, just prior to the `\text{}` macro. Note that this scenario suggests always having an odd number of ampersands in each `<mrow>`. In the example below, alignment is on the equals sign in the first two columns, and provides left-justification to the explanations in the third column. N.B.: the use below of the `\text{}` macro does not include mathematics within its argument. Doing so may yield unpredictable results depending on your choice of delimiters for the mathematics (and using an `<m>` tag will be ineffective).

$$\frac{dx}{dt} = x^2 - 4x - y + 4 \qquad \frac{dy}{dt} = x^3 - y \qquad x, y \text{ version}$$
$$\frac{dw}{dt} = z^3 - w \qquad \frac{dz}{dt} = z^2 - 4z - w + 4 \qquad z, w \text{ version}$$

PreTeXt will automatically detect the presence or absence of ampersands, but by defining macros for entire aligned equations, you can effectively hide the ampersands. So the `@alignment` attribute can override automatic detection. We use a simple LaTeX macro to demonstrate setting `alignment='align'` to

override the use of a `gather` environment and use a `align` environment instead. Example:

$$\frac{dx}{dt} = x^2 - 4x - y + 4$$
$$\frac{dy}{dt} = x^3 - y.$$

The AMSmath LaTeX package's `alignat` environment is a third variant of alignment. It never happens automatically, you need to ask for it with `alignment="alignat"`. It is very similar to `align` but adds no space between the equation columns. So you can leave it that way, or you can add your own "extra" space to suit. Here is a previous example with no inter-column space:

$$\frac{dx}{dt} = x^2 - 4x - y + 4 \quad \frac{dy}{dt} = x^3 - y \qquad x, y \text{ version}$$
$$\frac{dw}{dt} = z^3 - w \qquad \frac{dz}{dt} = z^2 - 4z - w + 4z, w \text{ version.}$$

This modified example has a middle row with three columns, while the other rows have just one column, as a test of our routines for determining the `<mrow>` with the greatest number of ampersands (and how many there are),

$$\frac{dw}{dt} = z^3 - w$$
$$\frac{dx}{dt} = x^2 - 4x - y + 4 \quad \frac{dy}{dt} = x^3 - yx, y \text{ third column}$$
$$\frac{dw}{dt} = z^3 - w.$$

Final example demonstrates that ampersands in other objects (matrices here) can wreak havoc with computing the number of columns. So we provide yet another attribute to override automatic detection, `alignat-columns`. This is the number of *columns* not the number of *ampersands*. Generally, for c columns, there will be $2c - 1$ ampersands.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

One caveat: if your number of ampersands is even (see advice above about using an odd number) behavior should still be correct, as in next example.

If you want super-precise control over alignment of the terms of a system of equations (linear or not) you can use the `alignat` option to advantage by not including any extra space. This example is modified slightly from a post by Alex Jordan:

$$\begin{aligned} 2x + \ y + 3z &= \ 10 \\ x \quad\quad + \ z &= \ 6 \\ x + 3y + 2z &= 13. \end{aligned}$$

Beautiful.

Aside Overlap Test. This is simply to test what happens when an aside bumps into something like the content below that may extend past the normal right content boundary. It likely only is of interest with the theme `default-modern`.

A long equation, to check layout on various screen sizes. This is Weil's "explicit formula" for the Riemann ζ-function:

$$\sum_{\gamma} S_{-}(\gamma) = \frac{\log Q}{\pi}\hat{S}_{-}(0) + \frac{1}{2\pi}\sum_{j=1}^{d} \Re\left\{\int_{-\infty}^{\infty} \frac{\Gamma'}{\Gamma}\left(\frac{1}{4} + \frac{it}{2} + \mu_j\right) S_{-}(t)dt\right\} - \frac{d}{2\pi}\hat{S}_{-}(0)\log\pi. \quad (7.3)$$

Example 7.1 Excessive Display Mathematics. In print versions, a long run of displayed equations often needs to be broken across pages. If you are reading some other version of this, then there is nothing to see here. But for LaTeX output it could be interesting. First, with no extra effort, this page-long display should break naturally, no matter how the preceding material changes.

$$x^2 + y^2 = z^2$$
$$a^2 + b^2 = c^2$$
$$\alpha^2 + \beta^2 = \gamma^2$$
$$m^2 + n^2 = p^2$$
$$x^2 + y^2 = z^2$$
$$a^2 + b^2 = c^2$$
$$\alpha^2 + \beta^2 = \gamma^2$$
$$m^2 + n^2 = p^2$$
$$x^2 + y^2 = z^2$$
$$a^2 + b^2 = c^2$$
$$\alpha^2 + \beta^2 = \gamma^2$$
$$m^2 + n^2 = p^2$$
$$x^2 + y^2 = z^2$$
$$a^2 + b^2 = c^2$$
$$\alpha^2 + \beta^2 = \gamma^2$$
$$m^2 + n^2 = p^2$$
$$x^2 + y^2 = z^2$$
$$a^2 + b^2 = c^2$$
$$\alpha^2 + \beta^2 = \gamma^2$$
$$m^2 + n^2 = p^2$$
$$x^2 + y^2 = z^2$$
$$a^2 + b^2 = c^2$$
$$\alpha^2 + \beta^2 = \gamma^2$$
$$m^2 + n^2 = p^2$$
$$x^2 + y^2 = z^2$$
$$a^2 + b^2 = c^2$$
$$\alpha^2 + \beta^2 = \gamma^2$$
$$m^2 + n^2 = p^2$$
$$x^2 + y^2 = z^2$$
$$a^2 + b^2 = c^2$$
$$\alpha^2 + \beta^2 = \gamma^2$$
$$m^2 + n^2 = p^2$$
$$x^2 + y^2 = z^2$$
$$a^2 + b^2 = c^2$$
$$\alpha^2 + \beta^2 = \gamma^2$$
$$m^2 + n^2 = p^2.$$

In this version we have turned off page breaking for the entire display, but then allowed a break at every fourth equation, so you should see a reasonably attractive page break right after one of the $m^2 + n^2 = p^2$ equations.

$$x^2 + y^2 = z^2 \tag{7.4}$$
$$a^2 + b^2 = c^2 \tag{7.5}$$
$$\alpha^2 + \beta^2 = \gamma^2 \tag{7.6}$$
$$m^2 + n^2 = p^2 \tag{7.7}$$
$$x^2 + y^2 = z^2 \tag{7.8}$$
$$a^2 + b^2 = c^2 \tag{7.9}$$
$$\alpha^2 + \beta^2 = \gamma^2 \tag{7.10}$$
$$m^2 + n^2 = p^2 \tag{7.11}$$
$$x^2 + y^2 = z^2 \tag{7.12}$$
$$a^2 + b^2 = c^2 \tag{7.13}$$
$$\alpha^2 + \beta^2 = \gamma^2 \tag{7.14}$$
$$m^2 + n^2 = p^2 \tag{7.15}$$
$$x^2 + y^2 = z^2 \tag{7.16}$$
$$a^2 + b^2 = c^2 \tag{7.17}$$
$$\alpha^2 + \beta^2 = \gamma^2 \tag{7.18}$$
$$m^2 + n^2 = p^2 \tag{7.19}$$
$$x^2 + y^2 = z^2 \tag{7.20}$$
$$a^2 + b^2 = c^2 \tag{7.21}$$
$$\alpha^2 + \beta^2 = \gamma^2 \tag{7.22}$$
$$m^2 + n^2 = p^2 \tag{7.23}$$
$$x^2 + y^2 = z^2 \tag{7.24}$$
$$a^2 + b^2 = c^2 \tag{7.25}$$
$$\alpha^2 + \beta^2 = \gamma^2 \tag{7.26}$$
$$m^2 + n^2 = p^2 \tag{7.27}$$
$$x^2 + y^2 = z^2 \tag{7.28}$$
$$a^2 + b^2 = c^2 \tag{7.29}$$
$$\alpha^2 + \beta^2 = \gamma^2 \tag{7.30}$$
$$m^2 + n^2 = p^2 \tag{7.31}$$
$$x^2 + y^2 = z^2 \tag{7.32}$$
$$a^2 + b^2 = c^2 \tag{7.33}$$
$$\alpha^2 + \beta^2 = \gamma^2 \tag{7.34}$$
$$m^2 + n^2 = p^2 \tag{7.35}$$
$$x^2 + y^2 = z^2 \tag{7.36}$$
$$a^2 + b^2 = c^2 \tag{7.37}$$
$$\alpha^2 + \beta^2 = \gamma^2 \tag{7.38}$$
$$m^2 + n^2 = p^2 \tag{7.39}$$
$$x^2 + y^2 = z^2 \tag{7.40}$$
$$a^2 + b^2 = c^2 \tag{7.41}$$
$$\alpha^2 + \beta^2 = \gamma^2 \tag{7.42}$$
$$m^2 + n^2 = p^2 \tag{7.43}$$
$$x^2 + y^2 = z^2 \tag{7.44}$$
$$a^2 + b^2 = c^2 \tag{7.45}$$
$$\alpha^2 + \beta^2 = \gamma^2 \tag{7.46}$$
$$m^2 + n^2 = p^2 \tag{7.47}$$
$$x^2 + y^2 = z^2 \tag{7.48}$$
$$a^2 + b^2 = c^2 \tag{7.49}$$
$$\alpha^2 + \beta^2 = \gamma^2 \tag{7.50}$$
$$m^2 + n^2 = p^2. \tag{7.51}$$

So. Do not take any extra steps and let LaTeX figure out the breaks. If you do not like a break, modify the `<md>` to go back to the AMSmath default behavior and not break at all. Ever. Or rather, go further and modify an individual `<mrow>` to suggest that it is a good place for a break. ▲

Displayed mathematics in LaTeX has a device called **intertext**, allowing for short snippets of text in-between displayed equations, but alignment continues across these short breaks. For any other output format besides PDF/LaTeX there really is not the technical means to make this continued alignment happen. So we mostly do the best we can. The following is a test example. Since almost all the equations are very similar in shape, it appears that there is alignment throughout. However, for almost all output formats, the very last equation changes the horizontal positioning of the block with the final three equations, and they are not aligned with the previous equations. The exception is a PDF where we are able to take advantage of LaTeX's support and the equations should *all* be aligned on the equals signs.

$$A = Z + 1 \tag{7.52}$$
$$B = Y + 2 \tag{7.53}$$

One intertext.

$$C = X + 3 \tag{7.54}$$
$$D = W + 4 \tag{7.55}$$
$$E = V + 5 \tag{7.56}$$

Two intertext.

$$F = U + 6 \tag{7.57}$$

Three intertext.

$$G = T + 7 \tag{7.58}$$
$$H = S + 8 \tag{7.59}$$
$$I + J + K = P + Q + R + 9 + 10 + 11 \tag{7.60}$$

Now for some more testing of the `<intertext>` feature. Numbering some equations, and not others, requires some care in the conversion to LaTeX.

First, a numbered equation in the first group of equations.

$$A = B + C \tag{7.61}$$

Intertext: between first and second equations.

$$P = Q + R$$

Intertext: between second and third equations.

$$X = Y + Z$$

Second, a numbered equation in the intermediate group of equations.

$$A = B + C$$

Intertext: between first and second equations.

$$P = Q + R \tag{7.62}$$

Intertext: between second and third equations.

$$X = Y + Z$$

Third, a numbered equation in the final group of equations.

$$A = B + C$$

Intertext: between first and second equations.

$$P = Q + R$$

Intertext: between second and third equations.

$$X = Y + Z \tag{7.63}$$

Fourth, no numbered equations at all.

$$A = B + C$$

Intertext: between first and second equations.

$$P = Q + R$$

Intertext: between second and third equations.

$$X = Y + Z$$

This is a poorly-authored paragaph to test the conversion to HTML. There are two displayed equations, separated by a period ending the first one's sentence, which should migrate into the display, and not leave behind an empty paragraph:

$$z + y = z.$$

$$a + b = c.$$

This final sentence should remain, inside another HTML paragraph, without the second equation's period.

7.3 LaTeX Packages and MathJax Extensions

If you would like to enhance your mathematics by using a macro from a LaTeX package *and* there is a MathJax extension *which implements the same macro*, then you may use this with your mathematics as we demonstrate here.

This example is from Jason Underdown. The package is named `cancel` and is included in the TeXLive distribution, so is fairly standard. The particular macro being demonstrated is `\cancelto{}{}`.

$$\lim_{b\to\infty}\left[\cancelto{0}{-\frac{1}{s}e^{-sb}}+\frac{1}{s}\right].$$

Look at the source of this article to see the package name being supplied in a `<math-package>` element within the `<docinfo>` section. That is the only setup required to make the macro usable in LaTeX and HTML output.

See the *PreTeXt Guide* subsection about "MathJax Extensions" for more detail.

7.4 Advanced Mathematics

MathJax is extremely capable in rendering a subset of LaTeX in web browsers, and improving all the time. You can get fairly fancy with some of its supported commands. In particular, if you need to mix in a few words with your mathematics, the `\text{}` macro is supported. For example, you might use an "if" or an "otherwise" in the definition of a piecewise function.

Consider that the first line below is text sandwiched in-between two Greek letters, wrapped in a `\text{}` macro. In HTML output we have taken care that the font for text material within display mathematics should match the font of the surrounding paragraph, as also happens with LaTeX output. The second

line is nearly identical in the source, but is just naked text being rendered like a slew of variables.

$$\alpha \text{ is not equal to } \beta$$
$$\alpha isnotequalto \beta$$
$$\alpha \neq \beta.$$

We are not suggesting here that using words in place of symbols, as in the first line, is a good practice. (It is not.)

The following example is a good stress-test of using the `\text{}` macro to achieve certain effects. Note the Unicode left and right smart quotes. This a contribution from Alex Jordan as part of his work on *APEX Calculus*.

$$y \to \frac{\sin(0)}{0} \to \text{“ } \frac{0}{0} \text{ ”}.$$

And another one from Alex. Note the use of the `\mathord{}` and `\mathrel{}` macros to control spacing around the mathematical symbols. Examine the source to see how the quotation marks have been authored with XML syntax for Unicode characters, since we do not allow most markup inside mathematics.

$$\zeta(1) = \sum_{n=1}^{\infty} \frac{1}{n} \quad \text{“=”} \quad \prod_p \left(\frac{1}{1-1/p} \right) = \prod_p \left(\frac{1}{1-p^{-1}} \right)$$

Generally, you cannot use any XML elements inside of the mathematics elements. An exception is the `<xref>` element which you might want to use to provide justifications for the steps of a derivation. Here is a visual example that is mathematically meaningless,

$$\begin{aligned} A &= B + C && \text{Corollary 4.1} \\ &= D + E && \text{The Fundamental Theorem of Calculus} \\ &= F + G && \text{A nice result.} \end{aligned}$$

Scott Beaver likes to write short chains of equalities all in one line, with the cross-references sitting on each equals sign. Here we test the LaTeX `\overset`, `\underset`, and `\overunderset` macros wrapping a PreTeXt `<xref>`, with and without content, inside an `<md>` element. Note that `\stackrel` is obsolete, and that MathJax has supported `\overunderset` since its version 3.2 (see GitHub #2704), so the final equals sign carries a reason above *and* below. The mathematics is Scott's, the reasons are totally unrelated to the math.

$$AC - AD \overset{2.1}{=} A(C-D) \underset{2.2}{=} A0_{n\times p} \overset{\text{Thm. } 2.1}{\underset{2.2}{=}} 0_{m\times p}$$

We suggest using cross-references that only display numbers (`<xref>` with `@text` set to `global`) since if you stick to elements like `<theorem>`, `<lemma>`, `<definition>`, or `<axiom>`, then the numbers will be unambiguous and the target of the cross-reference will contain full information. But note that if you mix in divisions, or perhaps figures, as reasons then there is a possibility that numbers will need to be qualified by their type. We have provided an abbreviation for one cross-reference to Theorem 2.1 (which will not benefit from automatic translation to other languages).

7.5 Numbered Equations

There is flexible control for numbering displayed equations, largely provided by the `@number` attribute on the `<md>` and `<mrow>` elements, taking values of `yes` or `no`. First, a single displayed equation, with a number:

$$x^2 + y^2 = z^2. \tag{7.64}$$

Now several displayed equations, all numbered.

$$x^3 + y^3 = z^3 \tag{7.65}$$
$$x^4 + y^4 = z^4 \tag{7.66}$$
$$x^5 + y^5 = z^5. \tag{7.67}$$

Similar, but now the first equation's numbering is controlled by the global default value, the second is explicitly not numbered, and the third is explicitly numbered.

$$x^6 + y^6 = z^6$$
$$x^7 + y^7 = z^7$$
$$x^8 + y^8 = z^8. \tag{7.68}$$

The global default numbering choice is set in `<docinfo>`.

7.6 Local Tags on Equations

If you are not writing a research monograph, maybe (a) you will not use many numbered equations, or do not like the looks of them, or feel they scare your readers, and (b) maybe your cross-references are always local-ish, like strictly within an `<example>` or a `<proof>`. For this situation you can create, and employ, a "local" tag on a displayed equation. Nothing enforces the idea of what constitutes local, and there is nothing to stop you from using the same symbols more than once. With freedom comes responsibility.

Use the `@tag` attribute on an `<mrow>`, or directly on a single-line `<md>` (one with no `<mrow>` children). The value of the `@tag` attribute is a symbol name. The prefix `d` means "double", and the prefix `t` means "triple". So allowed values are

```
star, dstar, tstar
dagger, ddagger, tdagger
daggerdbl, ddaggerdbl, tdaggerdbl
hash, dhash, thash
maltese, dmaltese, tmaltese
```

Cross-references to these tagged equations happens in the usual way and should behave as expected. We test the double versions to make sure the symbols render properly in various output formats.

$$c^2 = a^2 + b^2 \tag{✶✶}$$
$$c^2 = a^2 + b^2 \tag{††}$$
$$c^2 = a^2 + b^2 \tag{‡‡}$$
$$c^2 = a^2 + b^2 \tag{\#\#}$$
$$c^2 = a^2 + b^2 \tag{✠✠}$$
$$z^2 = x^2 + y^2 \tag{7.69}$$

Here are the local cross-references: (✶✶), (††), (‡‡), (\#\#), (✠✠). We test another farther away in Section 21, contrary to our advice above.

When there is nothing to align, the `@tag` attribute may live directly on a single-line `<md>`:

$$c^2 = a^2 + b^2. \tag{✶}$$

This is equivalent to placing the content in a single `<mrow>` child carrying the `@tag`. A cross-reference to this equation looks like (✶). Note that `@tag` and `@number` are mutually exclusive on a single-line `<md>`.

7.7 Commutative Diagrams

This diagram is authored by Tom Judson using the syntax of the AMS LaTeX "CD" package. Inside of a `<md>` element start with `\begin{CD}`. Remember to escape the less-than character.

$$\begin{array}{ccc} E[x]/\langle p(x)\rangle & \xrightarrow{\psi} & F[x]/\langle q(x)\rangle \\ \downarrow{\scriptstyle\sigma} & & \downarrow{\scriptstyle\tau} \\ E(\alpha) & \xrightarrow{\overline{\phi}} & F(\beta) \\ \downarrow & & \downarrow \\ E & \xrightarrow{\phi} & F \end{array}$$

While this package is not as flexible as some generic drawing packages, it has the advantage of full support by MathJax, and thus the HTML version will be more accessible.

7.8 Line-Breaking after Mathematics

As of 2021-05-14, in HTML output the next sentence should just fill a full line across the page. We take active measures to bind the concluding period to the final bit of mathematics, the variable x. The prevents a bad line break which could see the period *begin* a new line, all by itself. In the event that the line-breaking siutation improves, we could relax these measures. This testing is only relevant to HTML output, not LaTeX output.

xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx x.

7.9 Fonts and Mathematics

This section is about testing types and sizes of fonts, not so much about using different typefaces. First, one long displayed equation, which is designed to be full-width for LaTeX output when using defaults as of 2020-01-29 (commit `defd4bffd462e7ea`).

Start paragraph.

$$a^2+b^2+a^2+b^2+a^2+b^2+a^2+b^2+a^2+b^2+a^2+b^2+a^2+b^2+a^2+b^2+a^2+b^2+a^2+b^2$$

End paragraph.

The next paragraph has five ways to write the sine of x, bracketed by plain text versions. This tests font size and the fonts employed. The raw source of this paragraph is (spread over two lines)

```
sin x | <m>\sin x</m> | <m>\text{sin}\ x</m> | <m>\mathrm{sin}\ x</m> |
<m>\text{sin}x</m> | <m>sin x</m> | sin x
```

The five ways, from good to bad,

1. The best way, using `\sin`. Note the distance to the x.
2. With a `\text{}` macro.
3. With a `\mathrm{}` macro. Not recommended for PreTeXt.
4. Without a space. Note that the previous two require explicit spacing.
5. No extra effort, so LaTeX renders as a product of four variables.

sin x | $\sin x$ | sin x | sin x | $\sin x$ | $sinx$ | sin x

Finally a simple paragraph that places a text “x” next to a variable “x”. wordxx$x$ + $x$xxword

7.10 Special Characters in Text Mode

LaTeX designates ten characters as **special** because they carry syntactic meaning. When these characters appear inside text-mode macros (such as `\text{}`) within mathematics, they must be escaped to produce their literal glyph. Seven of the ten have simple single-character escapes and are demonstrated in the following table. The remaining three (`~`, `^`, `\`) require macros from LaTeX's `textcomp` package (`\textasciitilde`, `\textasciicircum`, `\textbackslash`) which are not available in MathJax's text mode as of 2026-04-03.

Character	Escaped form	Rendered
Hash `#`	`\#`	5#
Ampersand `&`	`\&`	P & Q
Percent `%`	`\%`	5%
Dollar `$`	`\$`	$3
Underscore `_`	`\_`	x_1
Left brace `{`	`\{`	{a, b}
Right brace `}`	`\}`	{a, b}

The hash character also requires escaping within the `\tag{}` macro. See (##) for an example where the `tag="dhash"` attribute produces a double hash as an equation tag.

7.11 Miscellaneous

In HTML output, a MathJax workaround for a Safari rendering bug was inserting extra spaces after textual subscripts and superscripts (MathJax thread). PreTeXt CSS now applies a correction. The following tests if the CSS fix is sufficient, and could be used to test the necessity of our fix in the future. Following is the original report, though `NOT` has been moved to a superscript:

$$T_{\text{clk}} - t_{\text{su}} > t_{\text{clk-Q}} + \max\left(t_{\text{XOR}}, t^{\text{NOT}}\right).$$

There should not be anything to see in LaTeX/PDF output. (2021-10-21)

7.12 Wide and Narrow Displays

This subsection stresses how a displayed equation is placed when it does, or does not, fit the width of the text, and how a number or a custom tag interacts with that placement. In print (PDF) output a display that fits is centered, with any number at the right margin; a display too wide to fit is set flush with the left margin, so it overruns only the right margin and never the left.

A narrow display, numbered, centers with its number at the right margin:

$$a^2 + b^2 = c^2 \tag{7.70}$$

The same narrow display, unnumbered, simply centers:

$$a^2 + b^2 = c^2$$

A display wider than the text, numbered, is set flush left and overruns the right margin (and triggers an authoring warning):

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = \sum_{i=1}^{15} x_i \quad (7.71)$$

The same wide display, unnumbered, behaves the same way:

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = \sum_{i=1}^{15} x_i$$

A wide display with a custom tag flushes left as well, the tag taking the place of a number:

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = \sum_{i=1}^{15} x_i \quad (\ast)$$

A multi-line display, numbered, with one narrow line and one too wide, exercises per-line numbering:

$$a + b = c \quad (7.72)$$

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = \sum_{i=1}^{15} x_i \quad (7.73)$$

Inside a narrow container the available width is smaller, so a display centers within the container rather than the full text. Pushing a number all the way out to such a container's own right margin awaits a future refinement, so for now a number stays beside its centered equation. Here are two panels side by side, each with a numbered display.

A panel display:

$$a^2 + b^2 = c^2 \quad (7.74)$$

Another panel display:

$$u + v = w \quad (7.75)$$

A numbered display inside a block quotation, which is indented on both sides:

> An indented, numbered display:
>
> $$e^{i\pi} + 1 = 0 \quad (7.76)$$

A numbered display inside a doubly-nested list, where the item is indented twice:

1. Outer item.

   (a) Inner item, with a numbered display:

   $$\int_0^1 x^2 \, dx = \frac{1}{3} \quad (7.77)$$

8 Grouping Samples

While building or testing a rendering of PreTeXt, especially in HTML, it can be useful to see all the various elements that potentially create visual blocks in one place. So they are collected here.

We will omit content specific blocks like figures, images, tables, etc. as those elements have significant stress testing of their own elsewhere.

Please add any similar elements that are created or that you discover are missing from this page.

Aside A title. There are more <aside>es below, but here is a solor one.

Objectives 8 Our goals
A minimal <objective>.

- Stress test HTML themes
- Locally, test objectives

8.1 Remark-like Blocks

Remark 8.1 A title. A minimal <remark>.

Convention 8.2 A title. A minimal <convention>.

Note 8.3 A title. A minimal <note>.

Observation 8.4 A title. A minimal <observation>.

Warning 8.5 A title. A minimal <warning>.

Insight 8.6 A title. A minimal <insight>.

8.2 Example-like Blocks

Example 8.7 A title. A minimal <example>. ▲

Example 8.8 A title.
A structured <example>.

(a) A structured <example>.

Hint. A <hint>

Answer. An <answer>

Solution. A <solution>

The <conclusion>.

▲

Question 8.9 A title. A minimal <question>. ▲

Problem 8.10 A title. A minimal <problem>. ▲

Observation 8.11 A title. A minimal <observation>.

Warning 8.12 A title. A minimal <warning>.

Insight 8.13 A title. A minimal <insight>.

8.3 Theorem-like Blocks

Theorem 8.14 A title. A minimal `<theorem>`.

Theorem 8.15 A title. *A theorem with a proof.*

Proof. The proof of the theorem. ■

Theorem 8.16 A title. *A theorem with an* `<argument>`*, a proof-like alternate.*

Argument. The argument for the theorem. ■

Theorem 8.17 A title. *A theorem with a* `<reasoning>`*, a proof-like alternate.*

Reasoning. The reasoning for the theorem. ■

Theorem 8.18 A title. *A theorem with an* `<explanation>`*, a proof-like alternate.*

Explanation. The explanation of the theorem. ■

Corollary 8.19 A title. A minimal `<corollary>`.

Lemma 8.20 A title. A minimal `<lemma>`.

Algorithm 8.21 A title. A minimal `<algorithm>`.

Conundrum 8.22 A title. A minimal `<proposition>`.

Claim 8.23 A title. A minimal `<claim>`.

Fact 8.24 A title. A minimal `<fact>`.

Identity 8.25 A title. A minimal `<identity>`.

Proof. A stand-alone proof. ■

8.4 Axiom-like Blocks

Axiom 8.26 A title. A minimal `<axiom>`.

Conjecture 8.27 A title. A minimal `<conjecture>`.

Principle 8.28 A title. A minimal `<principle>`.

Heuristic 8.29 A title. A minimal `<heuristic>`.

Hypothesis 8.30 A title. A minimal `<hypothesis>`.

Assumption 8.31 A title. A minimal `<assumption>`.

8.5 Definition-like Blocks

Definition 8.32 A title. A minimal `<definition>`. ◆

8.6 Attributed Blocks

A theorem-like, axiom-like, definition-like, or open-problem block can credit a `<creator>`, and can cite its `<origins>` with one or more `<xref>`, typically to bibliography entries. Whatever combination of title, creator, and origins is present, the heading renders the title first, then the creator and the origins together in parentheses, with commas between them. The blocks that follow exercise the combinations, and a couple of the cross-references carry a `@detail`.

Theorem 8.33 Every Attribution. (Hadamard, [11.4.3, Chapter R], [2]) *A title, a creator, and two origins, the first with a locator.*

Lemma 8.34 ([1, Theorem 9.4]) *Origins alone, with a locator on a structured bibliography entry.*

Corollary 8.35 (Gauss, [1]) *A creator and one origin, no title.*

Conundrum 8.36 Titled, with Origins. ([2]) *A title and one origin, no creator.*

Axiom 8.37 ([11.4.3]) An axiom-like block with origins alone.

Conjecture 8.38 Titled and Credited. (Goldbach, [1], [2], [11.4.3]) An axiom-like block with a title, a creator, and three origins.

Definition 8.39 (Cantor) A definition-like block with a creator alone, no title. ◆

Definition 8.40 Titled, Credited, and Cited. (Dedekind, [2, Section 2]) A definition-like block with a title, a creator, and one origin with a locator.◆

Open Problem 8.41 An Open Problem, Attributed. (Hilbert, [1, Chapter 23]) An open problem with a title, a creator, and one origin with a locator.

8.7 Aside-like Blocks

Three `<aside>`s are below.

Aside A title. A minimal `<aside>`.

Biographical Aside A title. A minimal `<biographical>`.

Biographical Aside A title. A minimal `<biographical>`.

Historical Aside A title. A minimal `<historical>`.

8.8 Computation-like Blocks

Computation 8.42 A title. A minimal `<computation>`.

Technology 8.43 A title. A minimal `<technology>`.

Data 8.44 A title. A minimal `<data>`.

8.9 Project-like Blocks

Project 8.1 A title. A minimal `<project>`.

Activity 8.2 A title. A minimal `<activity>`.

Exploration 8.3 A title. A minimal `<exploration>`.

Investigation 8.4 A title. A minimal `<investigation>`.

8.10 Blocks Closing with a List or a Display

Some blocks end with a mark, such as the filled square that closes a proof. In print the mark rides the last line of the block's final paragraph, pushed to the right margin. A paragraph may also hold a list, a code display, or displayed mathematics, and then the mark cannot ride the paragraph's last line without disturbing the layout of the paragraph and of the list, so it takes a line of its own. Each block here ends with such a paragraph.

Example 8.45 Closing with a Bulleted List. This `<example>` ends with a paragraph whose final child is an unordered list, and the first item has two words so that a justified last line would be visible.

- first item
- second
- third

▲

Example 8.46 A List, Then More Text. Here the paragraph continues after the list, so its last line is running text, but the list is still inside the paragraph.

1. one item
2. another item

The paragraph resumes here and ends with words, not a list.

▲

Example 8.47 Closing with a Code Display. A paragraph can also end with a `<cd>`, a short display of code, as in

```
print("the last thing")
```

▲

Example 8.48 Closing with Displayed Mathematics. And a paragraph can end with displayed mathematics, the case that has always placed the mark on its own line, given here for comparison:

$$a^2 + b^2 = c^2$$

▲

Definition 8.49 A Statement Closing with a Description List. A **closing list** is a list that is the final child of a block's final paragraph, of any of three kinds; here the paragraph closes a `<statement>`.

ordered numbered or lettered

unordered	bulleted
description	this one, with a term for each item

◆

Theorem 8.50 A Proof Closing with a List. *Every proof that ends with a list is followed by its mark on a line of its own.*

Proof. The proof has two cases.

1. The proof ends with running text, and the mark rides its last line.
2. The proof ends with a list, as this one does, and the mark takes the next line.

■

8.11 Miscellaneous Blocks

A minimal `<assemblage>`.

Aside A title. A final `<aside>` to test behavior towards the end of a page. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Outcomes 8 Our results
A minimal `<outcome>`.

- Stress test HTML themes
- Locally, test outcomes

A minimal `<conclusion>`.

9 Entering Text in Paragraphs, Titles, Captions

XML, and therefore PreTeXt, is a markup language. But by and large, what you type into your source will be what you see in your output. So there is not much to say. Except that Subsection 9.1 eventually will be essential. However we do test various tricky situations here (which have technical explanations we avoid). See the Author's Guide for a superior treatment of the topics addressed here.

9.1 Special XML characters

One of the goals of PreTeXt is to relieve an author of managing the numerous conflicts when mixing languages that use different characters for special purposes. But, of course, XML has its own special characters.

If you type a "less-than" symbol in your source, the XML processor thinks you are starting an opening, or closing, tag. So how do you get a less-than sign into your source so that it survives into your output, like this: < ? You use an

escaped version. Type literally, the four characters `<` in your source. Then the XML processor will know you want the character and will not mistake it for a tag. But now we want to get an ampersand into our source like: &. How? Another escaped version of a character, literally the five characters `&`.

Otherwise, keys on your keyboard, even international versions, should be fine in your source and behave as expected. WYTIWYG = What You Type Is What You Get. So the principal concession to using XML markup is the following very simple rule.

> Rather than pressing the < and & keys on your keyboard, instead always enter the escape sequences `<` and `&` as replacements.

Simple. And it will work in "running text," verbatim text (like when authoring the content of `<c>` or `<pre>` elements), and mixed into LaTeX syntax to desribe mathematical expressions. XML has three other escape sequences `>`, `'`, and `"`, for the characters >, ', and " (respectively). But they seem largely unnecessary for authoring in PreTeXt, as we now demonstrate by typing them directly from our keyboard into our source: >, ', and ".

Checkpoint 9.1. How was `&` authored? Work it out, and then check the source here for the answer.

The next subsection should start on a new page in PDF output if the standard publisher file is used.

9.2 Quotations

The <q> tag will provide beginning and ending double quotations, while the <sq> tag will behave similarly but provide single quotes. Given the complexity of quotations, the different symbols used in different languages, and the over-simplified versions provided on keyboards, it is necessary to use markup.

"The roots of education are bitter, but the fruit is sweet." (Aristotle)

'It is always wise to look ahead, but difficult to look further than you can see.' (Winston Churchill)

A large quote can be accomodated with the <blockquote> tag, which can carry within itself an <attribution> element.

> The problem with writing a book in verse is, to be successful, it has to sound like you knocked it off on a rainy Friday afternoon. It has to sound easy. When you can do it, it helps tremendously because it's a thing that forces kids to read on. You have this unconsummated feeling if you stop.
>
> —Dr. Seuss

We say that again, to test a multiline attribution of a block quotation. Notice how the dash appears automatically, and that it is a **quotation dash** in HTML, distinct from other sorts of dashes.

> The problem with writing a book in verse is, to be successful, it has to sound like you knocked it off on a rainy Friday afternoon. It has to sound easy. When you can do it, it helps tremendously because it's a thing that forces kids to read on. You have this unconsummated feeling if you stop.
>
> —Dr. Seuss
> Children's Author

Sometimes a quote may extend across several paragraphs. Or a balanced pair of quotations marks crosses an XML boundary, so we need left, right, single and double versions. (For example, see Section 29 on poetry.) Here are all four in a haphazard order: ”, ‘, “, ’. These should be a last resort, and *not* a replacement for the q and sq tags. The left/right versions are used for the following quote from Abraham Lincoln, which we have edited into two paragraphs.

“I am not bound to win, but I am bound to be true. I am not bound to succeed, but I am bound to live by the light that I have.

I must stand with anybody that stands right, and stand with him while he is right, and part with him when he goes wrong.”

And as a tests, we try some crazy combinations of quotes, which would normally give LaTeX some trouble where the quotation marks are adjacent. View these examples in PDF output, with lots of magnification and note the spacing between the quotation marks.

- “we use ‘single quotes inside of double quotes’”
- ‘“double quotes inside of single quotes” with more’
- “‘single quotes tight inside of double quotes’”
- ‘“double quotes tight inside of single quotes”’
- An “‘‘“absurd test”’’” of two adjacent single quotes inside a pair of double quotes
- you would never do this, but a ‘‘pair of single quotes’’

Quotation marks are different in different languages. The following paragraphs contain short examples.

English (United States). Primary only: “Government of the people, by the people, for the people, shall not perish from the earth.” Primary and secondary: “I have a dream that one day this nation will rise up and live out the true meaning of its creed: ‘We hold these truths to be self-evident, that all men are created equal.’” (Abraham Lincoln, Gettysburg Address; Martin Luther King Jr., I Have a Dream)

French (France). Primary only: «La liberté commence où l’ignorance finit.» Primary and secondary: «Elle se répétait : “J’ai un amant ! un amant !” se délectant à cette idée comme à celle d’une autre puberté qui lui serait survenue». (Victor Hugo, Les Misérables; Gustave Flaubert, Madame Bovary)

French (Canada). Primary only: «La liberté commence où l’ignorance finit.» Primary and secondary: «Elle se répétait : ‘J’ai un amant ! un amant !’ se délectant à cette idée comme à celle d’une autre puberté qui lui serait survenue». (Victor Hugo, Les Misérables; Gustave Flaubert, Madame Bovary)

German (Germany). Primary only: „Grau, teurer Freund, ist alle Theorie, und grün des Lebens goldner Baum.” Primary and secondary: „‚Meine Luise’, sagte er mir, ‚hat mich zu Boden geworfen. Meine Luise wird mich auch aufrichten.’” (Johann Wolfgang von Goethe, Faust I; Friedrich Schiller, Kabale und Liebe)

9.3 Groupings

It is possible to make some other groupings like quotations, such as {some *emphasized text* grouped within braces}, or [a *Book Title* inside brackets], an “Article Title”, ⟨some *foreign words* inside angle brackets⟩, or ⟦just a bit of text within double brackets⟧. Some of these are used extensively by scholars who study texts to note various restorations or deletions. Note that the `<foreign>` element may have a `xml:lang` attribute.

Note that the angle brackets, ⟨ and ⟩, are *not* the keyboard characters, < and >. Your best bet is to use the provided `<angles>` element when constructing a balanced pair. Similarly, `<dblbrackets>` is provided to make the double-bracket characters easily available, since they are likely not on your keyboard.

9.4 Characters, Symbols, and Constructions

Verbatim, inline text, is accomplished with the `<c>` element. One special consideration: if your editor splits a line in the middle, we will fix it. This sentence has `two words` that we purposely split with a “newline” character in the middle. In output you should see a space between the words.

Some keyboard characters are ambiguous. Is the character ' an apostrophe or a right single quote? We presume the former, ’, and provide markup as an alternative for the latter (described above). Is / used to separate words, or to form a fraction? We presume the former, /, and provide `<solidus/>`, ⁄, for the latter. We test some other characters straight from our US keyboard (with two being escape sequences).

~ ` ! @ # $ % ^ & * ( ) _ - + = [ ] { } | \ ; : ’ " , < . > ? /

And again as verbatim text.

~ ` ! @ # $ % ^ & * ( ) _ - + = [ ] { } | \ ; : ' " , < . > ? /

Note that for a long time PreTeXt had empty elements for many of these characters, as a consequence of naïveté. So you might see `<dollar/>`, `<ampersand/>`, or others in old source. They will be deprecated and will raise warnings.

Now, when a character is nowhere to be found on your keyboard, we provide conveniences as markup. Or a keyboard character may have a different variant which we implement as an empty element. Here we test many of these. Read the Author's Guide for tags and more detail.

© Ⓟ ℗ ® ™ ℠ … · ~ ‰ ¶ § — × ⁄ ÷ ±

There are a few common abbreviations of Latin phrases that can be achieved in HTML one way, and in LaTeX with a slightly different mechanism. These are due to LaTeX's treatment of a period (full stop), depending on its surroundings. So not reserved characters, but just divergent treatment. Using these will lead to the best quality in all your outputs. See Will Robertson's informative and arcane blog post on the topic if you want the full story for the treatment of a full stop in LaTeX.

Tag	Realization	Meaning
`ad`	AD	*anno Domini*, in the year of the Lord
`am`	AM	*ante meridiem*, before midday
`bc`	BC	English, before Christ
`ca`	ca.	*circa*, about
`eg`	e.g.	*exempli gratia*, for example
`etal`	et al.	*et alia*, and others
`etc`	etc.	*et caetera*, and the rest
`ie`	i.e.	*id est*, in other words
`nb`	NB	*nota bene*, note well
`pm`	PM	*post meridiem*, after midday
`ps`	PS	*post scriptum*, after what has been written
`vs`	vs.	*versus*, against
`viz`	viz.	*videlicet*, namely

We also distinguish between abbreviations (vs.), acronyms (SCUBA) and initialisms (XML). This is a test of the text version of a multiplication symbol: 2×4.

Simple coordinates with degrees, minutes, seconds, or temperature, or distance in feet and inches. "We parked the car at 36°16′0.83″N, 122°35′47.27″W, and since it was 93°F, we walked 505′3.6″ so we could swim in the bay."

An **em dash** is the long dash used much like parentheses (not an **en dash** used to denote a range, such as a range of page numbers). It should not have spaces around it, but some style guides allow for a *thin* space, which—we test right now. A publication file entry can be set to `none` or `thin` to control this.

9.5 Currency

For best results, be certain the right Unicode characters are in your source. If you only need a certain symbol rarely, you can enter it in your source via its Unicode number. For example, to obtain a peso, type `₱`. This table has been tested with our default fonts, and should be fine for HTML output. Please report any difficulties with different LaTeX fonts, as there are extra measures we can take to make these more robust. (We've already done this for the Paraguayan guaraní.)

Table 9.2. Supported Currency

Sign	Unicode	Name
$	U+0024	dollar
¢	U+00A2	cent
£	U+00A3	sterling
¤	U+00A4	currency
¥	U+00A5	yen
f	U+0192	florin
฿	U+0E3F	baht
₡	U+20A1	colon
₤	U+20A4	lira
₦	U+20A6	naira
₩	U+20A9	won
₫	U+20AB	dong
€	U+20AC	euro
₱	U+20B1	peso
₲	U+20B2	guarani

9.6 Icons in Text

A limited supply of icons can be used when explaining how to use some computer application. The empty element is `<icon/>` and the attribute is `@name`.

We sprinkle a few into a few sentences to check baselines and font sizing. We sprinkle a few into a few sentences to check baselines and font sizing. We sprinkle a few into a few sentences to check baselines and font sizing. We sprinkle a few into a few sentences to check baselines and font sizing.

Table 9.3. User-Interface Icons

Name	Icon	Name	Icon	Name	Icon
arrow-down		arrow-left		arrow-right	
arrow-up		file-save		gear	
menu		wrench			

Table 9.4. Creative Commons License Icons

Name	Icon	Name	Icon	Name	Icon
`cc`		`cc-by`		`cc-sa`	
`cc-nc`		`cc-pd`		`cc-zero`	

Nominations of new icons must

- Have a Unicode character representation.
- Be in the HTML/CSS/JS Font Awesome 5 catalog.
- Be in the LaTeX `fontawesome5` package.
- Have a reasonably semantic PreTeXt name.

Please supply all this information, including the official Unicode name, with your request. Better yet, form a pull request.

Warning 9.5 Icons, `xelatex`, and Fonts. When processing a LaTeX file with `xelatex` the Font Awesome 5 icons are expected to be in a *system* font whose name is `Font Awesome 5 Free`. This is not a filename, and installing the LaTeX `fontawesome5` package into your LaTeX installation does not always guarantee that this font will automatically be available as a *system* font.

The Publisher's Guide contains some discussion about installing fonts into a system, as part of the documentation of creating a LaTeX style, and has particular warnings about only using the LaTeX `fontawesome5` package as a vehicle for installing and accessing these fonts.

9.7 Keyboard Keys

Your text can include specialized text meant to look like a key on the keyboard of a calculator or other device. So you can go b Enter < or F1. Or maybe a sequence as: Tab > Ctrl > T. Use the `<kbd>` element, with the label of the key as content.

There is a growing supply of keys which are labeled with graphics rather than text, such as a left arrow ←, right arrow →, up arrow ↑, down arrow ↓, and Enter ↵. See *The PreTeXt Guide* for the definitive list. In 9.6 the literal column means the symbol/character is the content of a `<kbd>` element, while the named column means the symbol/character has been chosen via the value of the `@name` attribute of an empty `<kbd/>` element.

Table 9.6. Named keys

	Literal	Named
Ampersand	&	&
Less than	<	<
Greater than	>	>
Dollar	$	$
Per cent	%	%
Open brace	{	{
Close brace	}	}
Hash	#	#
Backslash	\	\
Tilde	~	~
Circumflex	^	^
Underscore	_	_

Table 9.7. Upper Case

~	!	@	#	$	%	^	&	*	(	)	_	+		
Tab	Q	W	E	R	T	Y	U	I	O	P	{	}	|	
CapsLock	A	S	D	F	G	H	J	K	L	:	'	Enter		
Shift	Z	X	C	V	B	N	M	<	>	?	Shift			

Table 9.8. Lower Case

`	1	2	3	4	5	6	7	8	9	0	-	=		
Tab	q	w	e	r	t	y	u	i	o	p	[	]	\	
CapsLock	a	s	d	f	g	h	j	k	l	;	'	Enter		
Shift	z	x	c	v	b	n	m	,	.	/	Shift			

9.8 URLs, such as `http://example.com`

The `<url>` element can be used to create an external reference. The mandatory `@href` attribute is the actual URL complete with the protocol (e.g. `https://`). Content for the element is optional, and if provided will be the "clickable" text. In this case, a `@visual` attribute can be provided. In producing the print version of a PDF, this will become a parenthetical with a more friendly version of the URL. When no content is provided, the "clickable" will be the URL with a preference for an optional `@visual`. This subsection has some (extreme) tests and we leave complete documentation and full details for the PreTeXt Guide.

A long URL for testing: `www.pcc.edu/enroll/registration/dropping.html#withdraw`. Notice in the source that you *do not* put any tags inside the `@href` or `@visual` attributes, but you may need to provide XML escape sequences (see Subsection 9.1).

A `<url>` element in the print version of a PDF will get a trailing parenthetical containing the (simplified) URL in the highly-recommended `@visual` attribute. If you do not provide the `@visual` attribute in this case, then you will get the `@href` value repeated, possibly with some editing. If you insist, you can make the `@visual` attribute identical to the `@href` attribute. Some tests:

- With a useful `@visual` attribute
- With a duplicate `@visual` attribute
- With no `@visual` attribute, so an edited one is formulated (no protocol)

Here is a totally bogus URL, which contains every possible legal character, so if this fails to convert there is some problematic character. In order to test the use of a percent sign (`%`) in a URL, we follow it by two hex digits, specifically, `58`, which is a way to represent the character `X` in a URL. Normal text, monospace text, `<url>` with just `@href`, `<url>` with `@href` and `@visual`, `<url>` with `@href`, `@visual` and content. Notice how the various versions do, or do not, line-break in LaTeX/PDF output, including the (potentially confusing) use of a use of a hyphen in the normal text version. Note that `#` starts a fragment identifier, so it is the penultimate character. A `?` would normally start a query string, but as the fragment identifier, it is not special. Finally, square brackets (`[` and `]`) are only legal in limited situations (authority component for IPv6 host literals), so they are present as percent-encoded literals, which really does not demonstrate anything. All these efforts to make a comprehensive, but legal, URL are meant to appease the EPUB validator which will otherwise flag them as errors.

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789%-._~:/%5B%5D@!$&'()*+,;=#?

`ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789%-._~:/%5B%5D@!$&'()*+,;=#?`

`http://example.com/ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789%58-._~:/%5B%5D@!$&'()*+,;=#?`

`example.com/ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789%58-._~:/%5B%5D@!$&'()*+,;=#?`

Characters to a footnote

Line-breaking in LaTeX/PDF output is specialized, using path separators (slashes) as candidates for splitting across lines: `http://example.com/long/long/long/long/long/long/long/long/long/long/long/long/long/long/long/long/long/long/long/longer/longer`

We are not fans of footnotes, they are totally unstructured.[1] A URL in a footnote migrates around, and so care must be taken.[2] This paragraph has two footnotes, one with a real URL from Jesse Oldroyd, another with a fake

[1] Carleson's Theorem

[2] `http://example.com/ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789%58-._~:/%5B%5D@!$&'()*+,;=#?`

URL from the above suite. For good measure, we repeat the URL found in the first footnote[3]: Carleson's Theorem. And we include a no-content version of the same link, with a visual version provided and employed: `en.wikipedia.org/wiki/Carleson%27s_theorem`.

9.9 Biological Names

The `taxon` element can be used all by itself to get an italicized scientific name, as in *Escherichia coli*. It can also be structured with the elements `genus` and `species`, as in using both together in *Cyclopskolensis*. Or the subelements can be used individually. Rules for capitalization are presently your responsibility as an author. Possible improvements include new subelements, attributes for database identifiers, and checks on capitalization. Also, we might automatically abbreviate the genus after first use.

There is an attribute, `@ncbi` that you can use on the `taxon` element to precisely identify the organism you are discussing using an identification number from the National Center for Biotechnology Information. Their taxonomy is at `www.ncbi.nlm.nih.gov/taxonomy`. Right now, we do not do anything with this attribute, but things like links are certainly possible. See the source of this document to see it in use with *Drosophilamiranda* which could be used to construct a link to further information via id number or even further information via just the name.

9.10 Verbatim in titles, `\a&b#c%d~e{f}g$h_i^j`, OK

You can test the migration of the LaTeX special characters in this section title by requesting a 2-deep Table of Contents via the publication file.

9.11 Special Situations

Sage defines a nice syntax for generators of algebraic structures, but we must remember to use an escape sequence for the < symbol (see Subsection 9.1).

```
P.<t> = ZZ[]
P
```

```
Univariate Polynomial Ring in t over Integer Ring
```

There is an alternate Sage syntax, which avoids the less-than and greater symbols.

```
R = ZZ['u']
u = R.gen(0)
(u, R)
```

```
(u, Univariate Polynomial Ring in u over Integer Ring)
```

Ampersands, less-than, and greater symbols are likely to be necessary in source code, such as Sage code (think generators of field extensions) or TikZ code (think arrowheads), and in matrices (think separating entries). If you have a big matrix, or a huge chunk of TikZ code, you can protect it all at once from the XML processor by wrapping it in `<![CDATA[ ]]>`. It should be

[3] And if you do provide a link like Carleson's Theorem in a footnote you should see an easy-to-read parenthetical version in some output formats, such as `en.wikipedia.org/wiki/Carleson%27s_theorem`.

possible to write without ever using the "CDATA" mechanism, but it might get tedious in places to use the supplied macros or XML escape sequences. This construction is often mis-understood as a solution better remedied by reading Subsection 9.1 again.

We test the three pre-defined LaTeX macros for &, <, and > with a pair of aligned equations:

$$a^2 + b^2 < c^2$$
$$c^2 > a^2 + b^2$$

9.12 Jupyter Notebook, Markdown, MathJax, Delimiters

A Jupyter notebook allows a mix of HTML (our logistical preference for a conversion) and Markown (another set of special characters and their escaped versions). Certain pairs of delimiters, when appearing in consecutive HTML `<code>` elements require extraordinary care. But the one nut we cannot crack is pairs of dollar signs. So the next paragraph is known to render badly in a Jupyter notebook, but should otherwise be a bit boring.

`$` and `$`

9.13 LaTeX Characters, Ligatures, and More

This section is just for testing, and the more you know about LaTeX, the more we would encourage you to *not* to read this. Look to the Author's Guide for the *right* way to author your source.

The ten reserved characters, directly in the source: # $ % ^ & _ { } ~ \. And again: X#X$X%X^X&X_X{X}X~X\X, but smashed up tight to intermediate characters.

In a verbatim presentation: `# $ % ^ & _ { } ~ \`.

And `X#X$X%X^X&X_X{X}X~X\X`. (These verbatim versions are authored in different paragraphs to work around the Jupyter notebook bug described above.)

We also disrupt certain constructions from LaTeX. Attempting to sneak-in any traditional macro for the purposes of LaTeX-only output, such as, say a \newpage, will fail since the leading backslash will be caught and converted to `\textbackslash`. (See? It just happened twice.) For technical reasons we want to particularly test \textbackslash, \textbraceleft, and \textbraceright.

Four "TeX ligatures", `--`, `---`, ``, and `''`, authored in running text, --, ---, ``, ''. It may be hard to tell that the two consecutive apostrophes have not coalesced into a curly left smart quote, but see below, the spacing is subtly different.

We want the double quote mark from your keyboard, ", to not morph into some other character: " .

More testing: runs of hyphens. Such as: - (one), -- (two), --- (three), ---- (four), ----- (five), ------ (six), ------- (seven). Use the empty elements `<ndash/>` and `<mdash/>` for the longer dashes/hyphens.

Runs of apostrophes should not become smart right double quotes: ' (one), '' (two), ''' (three), '''' (four), ''''' (five), '''''' (six), ''''''' (seven). You might want to cut-and-paste these into a text file to convince yourself there are the right number of characters. Here are two smart right double quotes, separated by a non-breaking space, for visual comparison: " ". Or 30 apostrophes on a line of their own (longer) followed by 15 smart right double quotes (shorter).

''''''''''''''''''''''''''''''.

”””””””””””””””””””.

Runs of backticks (accent grave) should not become smart left double quotes when the output is processed by LaTeX: ` (one), `` (two), ``` (three), ```` (four), ````` (five), `````` (six), ``````` (seven). Furthermore, in a context where Markdown syntax is recognized as well (e.g. a Jupyter notebook), paired backticks should *not* produce `inline verbatim text`.

The next paragraph has a long run of words separated/joined by the keyboard forward-slash character. With this input, LaTeX will not line-break at the slash, nor will it hypenate anywhere. PreTeXt automatically provides an improved slash, which will line-break, as you should see below in LaTeX output. There is a bad right margin, but that is due to the absurdity of this test. This sort of problem should be no better or worse for the use of this character. Further refinements (zero-width space) and packages can be used to get hyphenation. HTML will line-break rationally with no extra help. Remember the `<solidus/>` character for super-simple text fractions like 7/32 (which will *not* line-break), and math elements or SI unit markup for technical work.

A/test/of/some/short/words/that/go/off/the/end/of/a/line/A/test/of/some/short/words/that/go/off/the/end/of/a/line/A/test/of/some/short/words/that/go/off/the/end/of/a/line/A/test/of/some/short/words/that/go/off/the/end/of/a/line/A/test/of/some/short/words/that/go/off/the/end/of/a/line.

9.14 HTML and accidental mathematics

We render mathematics in web pages with the fantastic MathJax Javascript library. Simplifying just a bit, it recognizes LaTeX syntax within a page, takes control of that text, and replaces it wth nice fonts and formatting. Now, if you write about LaTeX you might well have some mathematics in your examples. Best practice would be to use verbatim text for that, and we mark off such text as being off-limits to MathJax.

But if you are writing running text, then you can (accidentally) author some text which MathJax recognizes and converts to something (unintended). And if you are doing this intentionally, then you have ignored PreTeXt markup for mathematics, and are missing out on some features.

A few tests that we can prevent any accidents.

Inline mathematics: \(x^2\).

Display mathematics: \begin{align}x^2+y^2=z^2\end{align}

Double backticks is a common LaTeX construction, which in LaTeX/PDF output *should not* become an opening quote-mark. Also, a single backtick in HTML is a signal for MathJax to interpret ASCIIMath, and then a double backtick causes “random” pieces of mathematics on a page to not render at all. So we have a quotation authored LaTeX-style: ``We have nothing to fear, but fear itself.''

10 Graphics

In addition to including images created externally (e.g. photographs), PreTeXt supports several languages for describing diagrams and pictures with human-readable source code (i.e. plain text), rather than using a “paint” program. This section describes the various methods for incorporationg, or generating, graphis, images or diagrams.

10.1 Images from External Sources

If you have raster images (photographs, etc.) then they are specified with complete filenames, as above in Figure 5.2 or just below.



Figure 10.1. New Zealand Landscape, commons.wikimedia.org, CC-BY-SA-2.0

Raster formats are not ideal in that they do not scale well. But sometimes they are the only option. An ideal size (in pixels) for a raster image is at least 150 pixels per inch (PPI). If you have a raster image that is too small, then it will be scaled up and will look pixelated. If you have a raster image that is larger than the content width, then it will be scaled down and will generally look fine. And in some presentations there may be a way for the reader to see the high-resolution version. Providing an image that is significantly larger than needed can bloat your book and increase loading times, so avoid providing images that are many multiples of the required size.

Here is a small raster image based on a screenshot, as a test of scaling. The natural size of the image is 139 pixels by 127 pixels but many presentations will scale it beyond the natural size.

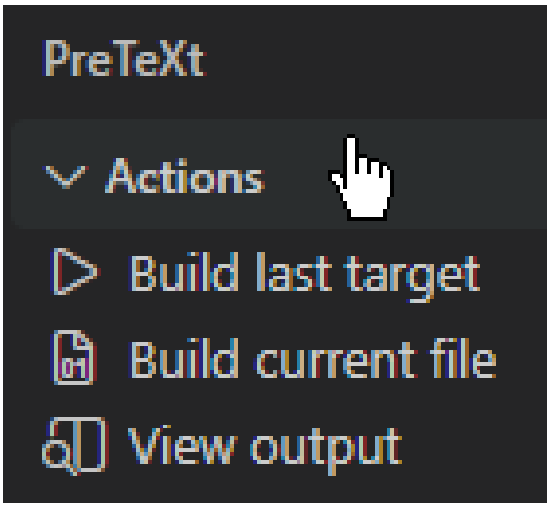


Figure 10.2. A portion of the UI for the PreTeXt tools plugin for VSCode.

If you have existing images that are vector graphics, then PDF format works best for LaTeX output and SVG format works best for HTML. The utility `pdf2svg` works well for converting PDF to SVG. In this case, specify your source as a filename, but leave off the file extension, and the appropriate version will be used for the current output format.

The image below is provided from a PDF file in LaTeX output, and was converted to an SVG for use with the HTML output. It has been explicitly scaled to a width of 65% of the text width. It has a `<description>`, but no `<shortdescription>`, so is testing that scenario.

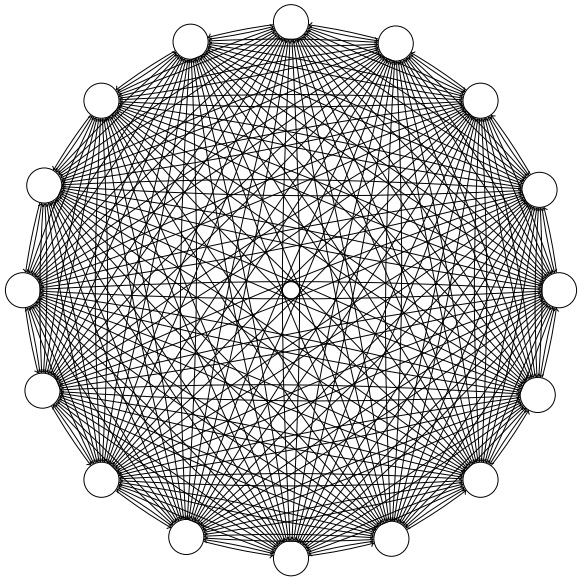


Figure 10.3. Complete graph on 16 vertices, from `www.texample.net`

A long description is not constrained by the width of its image. In HTML and EPUB output a revealed description is wide enough to read comfortably, however narrow the image. In HTML output it is centered on the image, but stays within the text. EPUB output ignores the margins of an image, so there the description is centered on the text. (Within a panel of a `<sidebyside>` the description fills the panel instead.) The two figures below are narrow versions of the previous image, first centered, then pushed to the left by its margins, so in HTML output the second description begins at the left edge of the text.

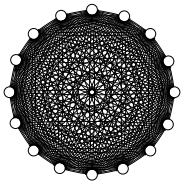


Figure 10.4. A narrow complete graph, with a long description

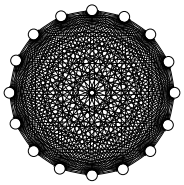
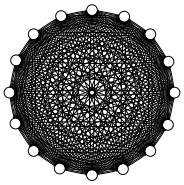


Figure 10.5. A narrow complete graph, off-center, with a long description

The next image is *not* inside a `<figure>`. It is narrow and centered, and its long description should be as wide as those of the figures above.



Remark 10.6 Footnote Buried. Nested `tcolorbox` (in LaTeX conversion) need special care when footnotes are interior.

A paragraph with a footnote[1] inside the paragraph.

A second paragraph, just to avoid a one-panel warning.

The final paragraph of this remark, randomly placed, to test footnotes in LaTeX conversions.

10.2 PreFigure

PreFigure is a standalone project for authoring mathematical diagrams (see prefigure.org). Its philosophy and approach are much like that of PreTeXt, and PreFigure is tightly integrated into PreTeXt. One key feature is excellent support for the creation of accessible output formats.

As of 2024-11-06 development continues for PreFigure itself, and fine-tuning of its integration within PreTeXt. But it is usable now for projects that want to use it.

- You can author PreFigure diagrams, and then generate SVG, PDF, and PNG output versions with the `pretext/pretext` script (see the PreTeXt Guide), and expect them to render in HTML, PDF, and EPUB output formats.
- Look for automatic generation to come to the PreTeXt-CLI sometime very soon.
- PreFigure has excellent support for annotations within an SVG diagram, supporting their use by screenreaders, for example. See an example below.
- Production of tactile versions is now possible, though they are not incorporated explicitly into any of the output formats. Perhaps they will become available via archive links or as a zip archive.

[1] Interior footnote.

This is a basic PreFigure diagram, in the sense that it does not use all the features of PreFigure or PreTeXt.

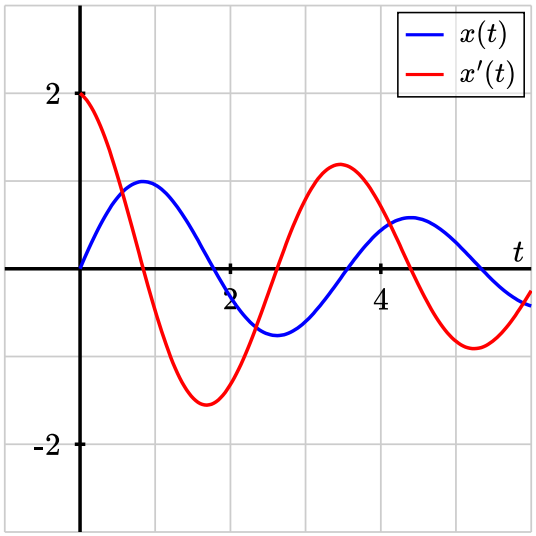
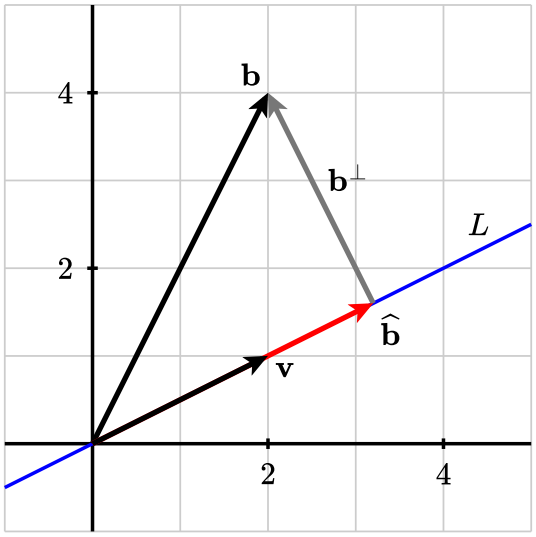
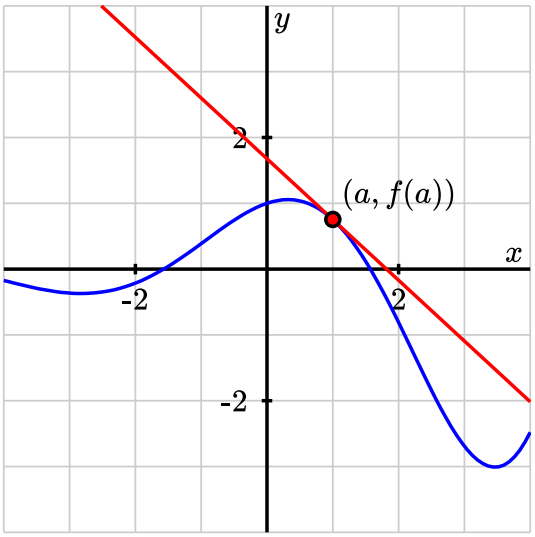


Figure 10.7. Solution to a differential equation

This next diagram employs some LaTeX macros that are defined in the usual way in `<docinfo>` and are employed to produce the names of some vectors in the labels. The blue line is colored blue by a global PreFigure declaration, also in `<docinfo>`.



The next PreFigure diagram is authored with annotations, arranged in a hierarchy of increasing refinement and detail. Each identified graphical component will read its annotation and show it on the screen below the diagram. When a reader clicks on the image, a high-level summary will be read using the author-provided annotation. The down and up arrow keys enable a reader to explore the diagram in more or less detail while the right and left arrow keys reveal features at the same level of detail. When the focus is on the graph, pressing "O" will produce a sonification of the graph.



Including annotations enables a new type of interactive diagram within a PreTeXt document offering potential benefits for all readers. In particular, annotations allow an author to call the reader's attention to specific details in a diagram and how they are related to one another so that the diagram and surrounding text are more tightly integrated. The annotated diagram below introduces Fibonacci tilings, which are one-dimensional analogs of Penrose tilings, and offers an explanation of their aperiodicity. Of course, surrounding text would usually provide a richer context for a diagram like this.

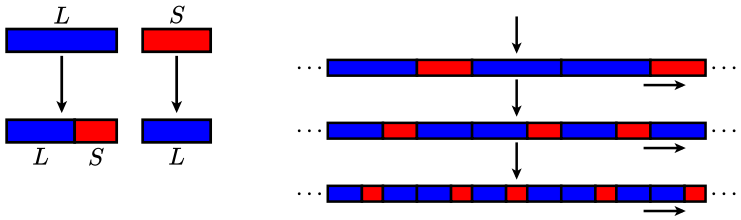
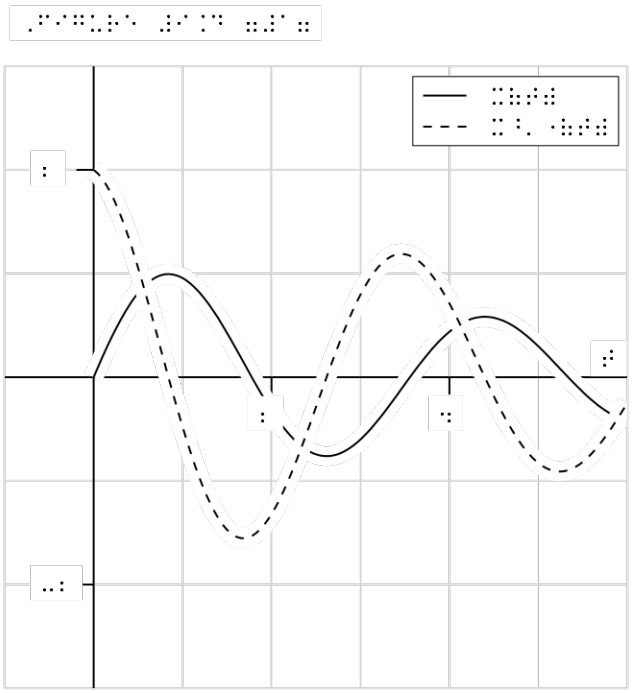


Figure 10.8. Fibonacci tilings are one-dimensional analogs of Penrose tilings. The diagram on the left introduces the process of deflation that is used to produce tilings while the diagram on the right explains why they are aperiodic.

For sighted readers, here is an example of a tactile version of one of the above diagrams. It is generated automatically from the same source as the other version. Imagine this being "printed" with an embosser so that the parts of the diagram, and the braille labels, are raised up from the paper and can be explored with one's fingertips. This image is a PNG produced specifically for this document. Typically, a tactile diagram produced by PreFigure will be a PDF ready to be sent to an appropriate embosser. Notice that PreTeXt adds a caption indicating the diagram's location in the document along the top of the diagram.



10.3 LaTeX images

There are several graphics engine packages that a LaTeX document can employ. Code from these packages renders diagrams automatically as part of normal processing of LaTeX files. For HTML output the `pretext` script produces SVG versions of the pictures. The script can also produce standalone TeX source files, PDFs, PNGs, and EPSs. The packages should be loaded in `docinfo/latex-image-preamble`, which is also where global package settings should be made. If any ampersands occur in your LaTeX code you should use the `\amp` macro pre-defined by PreTeXt. These first examples are from the TeXample.net site. Note that any LaTeX macros used in the rest of your document may be employed in the LaTeX-standalone or Asymptote diagrams (with this feature coming to Sage graphics next?).

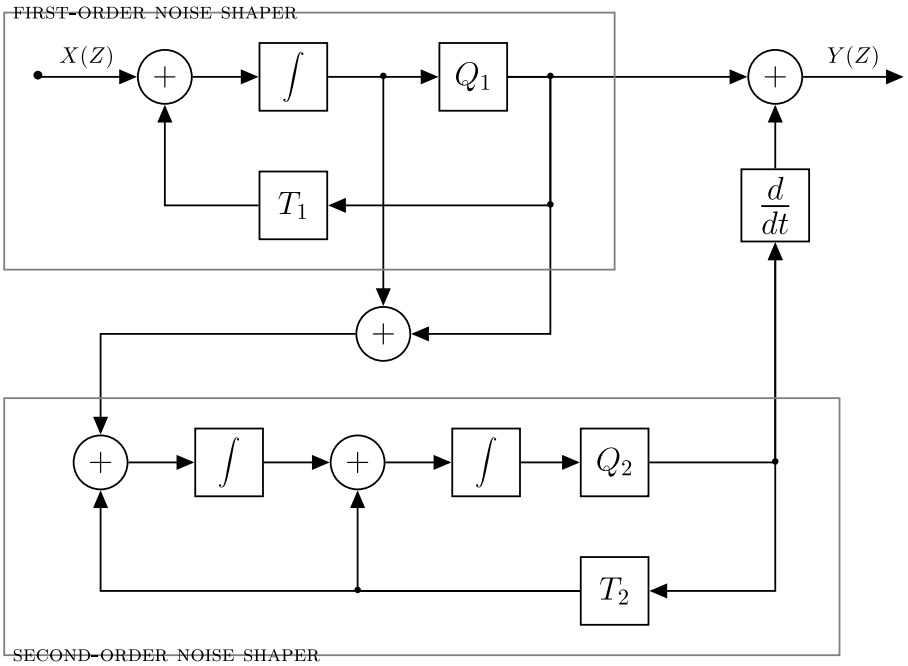


Figure 10.9. TikZ Electronics Diagram

The next example began life in Sketch, which will output TikZ code (though the code has been edited by hand for readability).

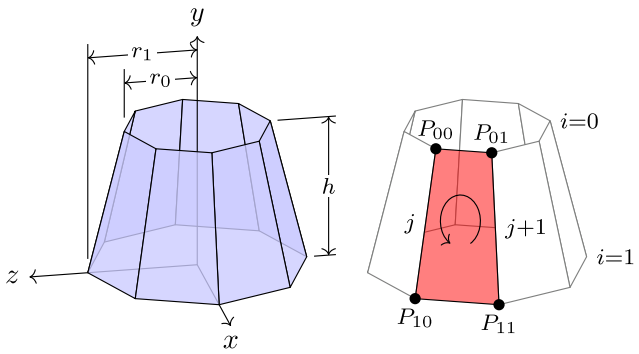


Figure 10.10. TikZ Cone Drawing

The pgfplots package was included in `docinfo/latex-image-preamble`. Here, it is used. Also, here we demonstrate using `\amp` where you would normally use an ampersand in LaTeX. There are known issues with `xelatex` processing any gradient shading in `tikz`. To (successfully) create the gradient shading in the 3D image here, you may need to use `pdflatex` until LaTeX developers resolve this issue.

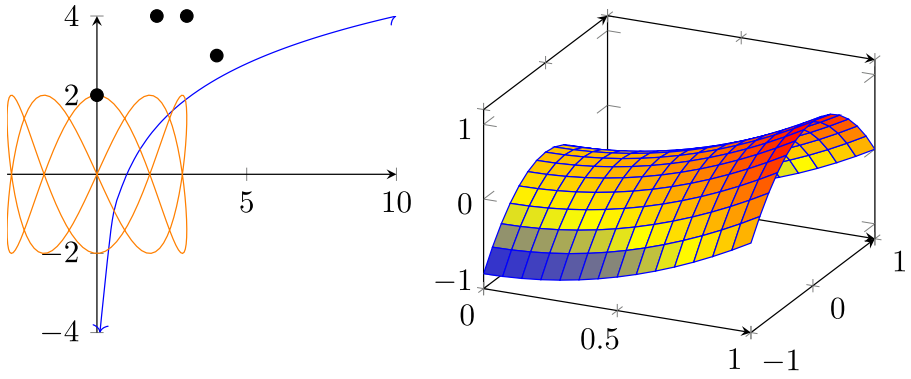


Figure 10.11. Sample pgfplots plot

A plot might use a graphics language to draw the axes and grid, but the data might be from an experiment and live in an external file that you do not wish to place in your source. Place such a file in a subdirectory directly below the directory where your master source file resides. Then indicate this directory in a `docinfo/directories/@data` attribute of your source. But you *must* prefix the path with `data/` as in the source below.

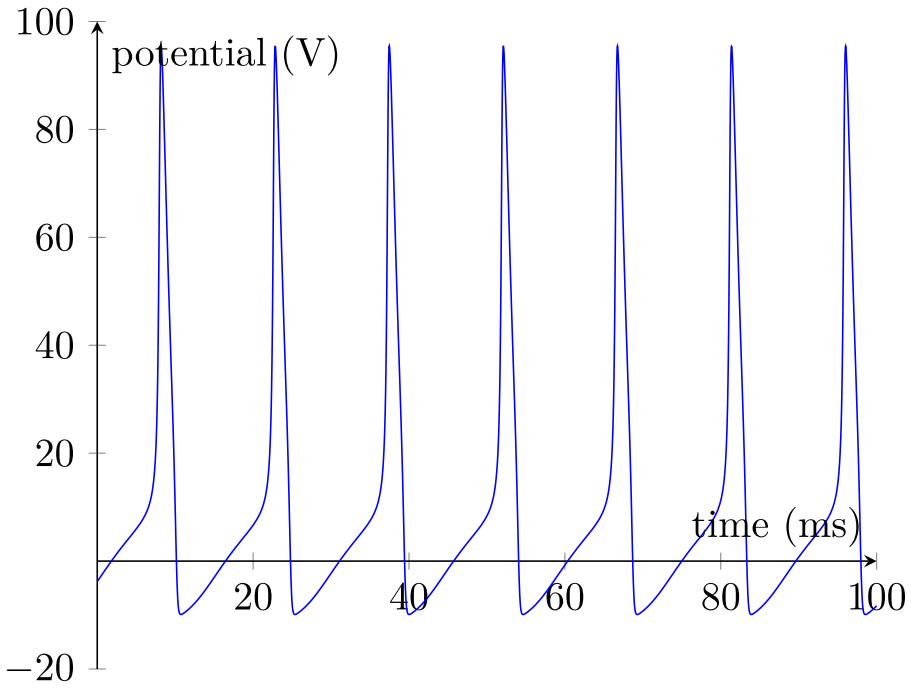


Figure 10.12. External data in a pgfplots plot

The TikZ image in the next figure is made up from two PNG images (the shark and the swimmer), in addition to various TikZ commands. The images reside in a source directory, `numerical/attack`, so the image files (`shark.png`, `swimmer.png`) are prefixed in the TikZ code with `data/attack` so the creation of the image will be successful. This example is courtesy of Stephen Brown.

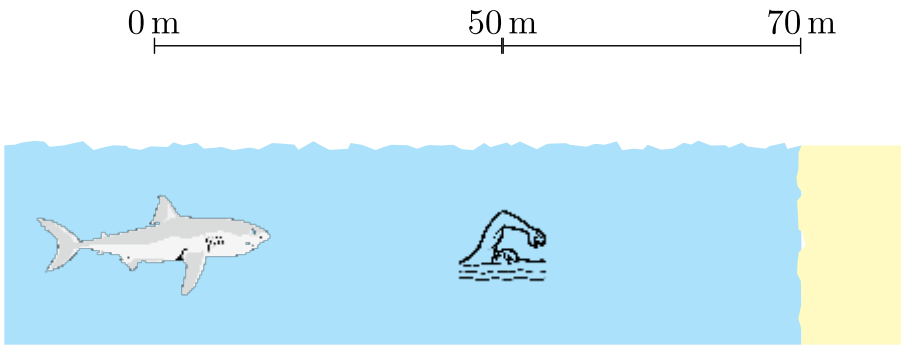


Figure 10.13. An illustration of the shark trying to catch you as you swim to shore.

A Cartesian plot might benefit from having a `<description>` with a `<tabular>` that lays bare the data used to to plot points.

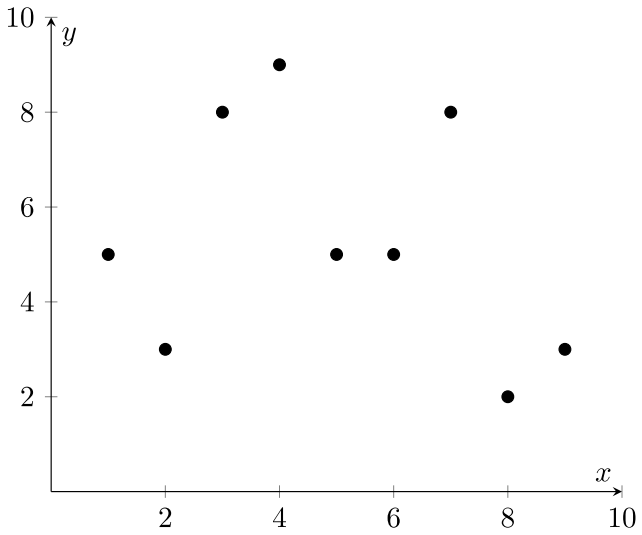


Figure 10.14. Full description with tabular

A QR code is a way to send a reader of a printed copy off to an online resource. The `qrcode` package, by Anders O. F. Hendrickson, will draw one from a URL with no other graphics package required (`ctan.org/pkg/qrcode`). It is loaded in `docinfo/latex-image-preamble`, and here the code sits on a white rectangle drawn by TikZ, so it stays readable whatever color the page behind it might be. The automatic cropping of a LaTeX image trims away the excess, so there is no need to size that rectangle carefully. The `nolinks` option keeps the code black—without it, the `hyperref` package would also make the code a live link in PDF output, colored like any other link. This example is courtesy of Lenore Horner.



Figure 10.15. A QR code for the `example.com` website, drawn by TikZ

PreTeXt also makes QR codes of its own, for the print version of a video or an interactive element, so a reader holding paper can still reach the live version. The `pretext` script does that with the `qrcode` Python package, by Lincoln Loop, distributed under a BSD license (`pypi.org/project/qrcode/`). You can drive that same package yourself, as in the short program below. Notice that its result is a PNG file, which we made ahead of time and stored with the other images of this document, since the `pretext` script does not run an author's programs. A PNG is the robust choice here, since every conversion will use it as-is.

Listing 10.16. A Python driver for the `qrcode` package

```
import qrcode

code = qrcode.QRCode(box_size=10, border=4)
code.add_data("https://example.com")
image = code.make_image()
image.save("qrcode-example-website.png")
```



Figure 10.17. A QR code for the same website, made by the Python `qrcode` package

The next image requires three passes with LaTeX to get everything in place. It is placed here to test that the code in the Pythion script correctly recognizes this requirement.

$$\left[\begin{array}{ccc|cc} 3 & 4 & 0 & 1 & 4 \\ -2 & 3 & 1 & -2 & 3 \\ 0 & 2 & 1 & 0 & 2 \end{array}\right]$$

Figure 10.18. A matrix with colored entries

PSTricks is a LaTeX package for drawing diagrams and pictures, dating back to the days before PDF, when PostScript (PS) was king. Given its history, it does not seem to work easily with the `pdflatex` engine. But it will work easily with the `xelatex` engine. We try to keep this present sample document workable with both engines, so we have presented an example of the use of PSTricks in the `xelatex`-exclusive sample document where we test obscure fonts and characters. So your best bet is to look there.

There are suggestions online that

```
\usepackage[pdf]{pstricks}
```

along with

```
pdflatex --shell-escape *.tex
```

is workable. We could not make it happen, and a "shell escape" can be a dangerous security hole. That said, updates to this approach are welcome.

10.4 Asymptote, 2D

The Asymptote graphics language may be placed in your source to draw graphs, diagrams or pictures. Rules for formatting code are identical to those for Sage code. For more on Asymptote see `asymptote.sourceforge.net`.

This is a simple physics diagram about levers, taken from the Asymptote documentation. In the HTML version of this article, the images are SVG's and so should scale nicely when you zoom in on the page.

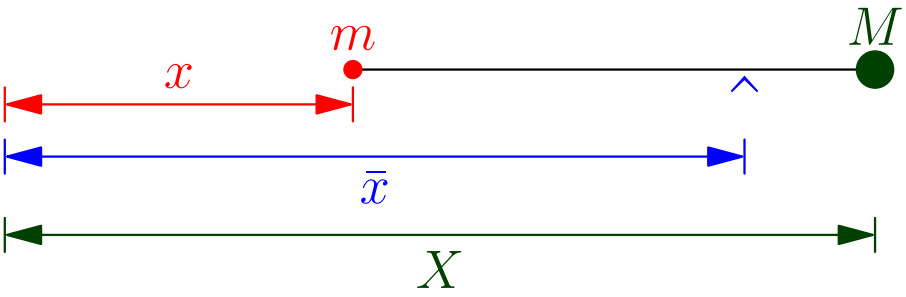


Figure 10.19. Asymptote Lever Demonstration

And a colorful contour plot with logarithmic scale. Again, from the Asymptote documentation. This SVG image employs two additional PNG images for the two parts where the color varies continuously.

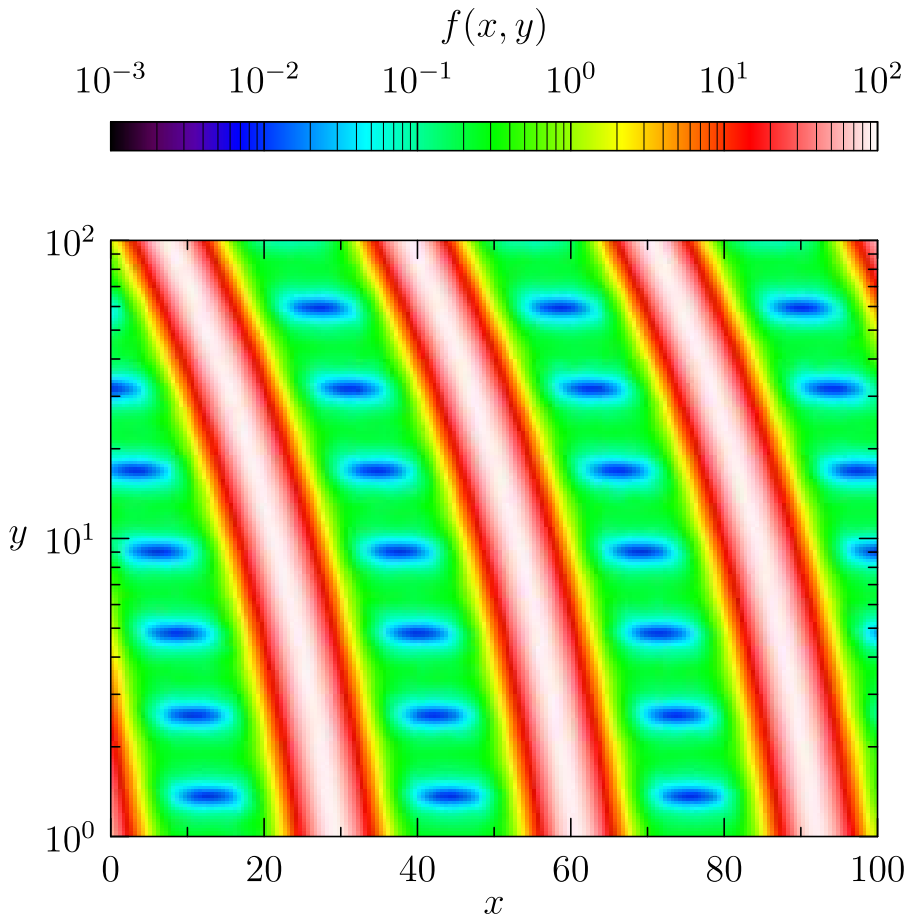


Figure 10.20. Asymptote Contour Plot

Here is the lever diagram again, but now we have added an integral to one of the legends, *using a LaTeX macro of our own,* which is idential to one we used in the early part of this article. The point is, we only needed to define the macro once for the entire document, and it is available as we make Asymptote diagrams. This device can be used to maintain flexibility and consistency in your choice of notation.

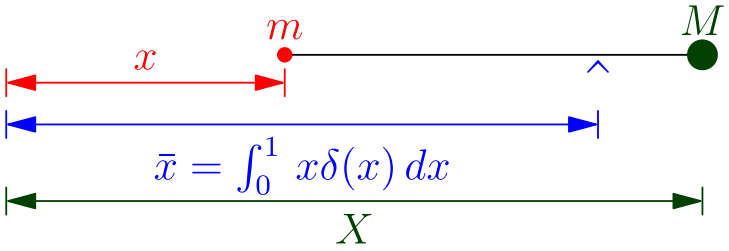


Figure 10.21. Aymptote Lever, plus Integral

10.5 Asymptote, 3D via WebGL

Asymptote can create an HTML file that is an interactive version of a 3D shape. At this writing (2020-05-18) support via the `pretext` script is evolving. Plus, you will need newer versions of Asymptote and the `dvisvgm` utility to

duplicate all of the results being displayed here in this testing document. The other distinction is that the author needs to provide the aspect ratio of the figure, and this should be placed on the `<asymptote>` element (not on the `<image>` element). Figure 10.22 is from the Asymptote Gallery.

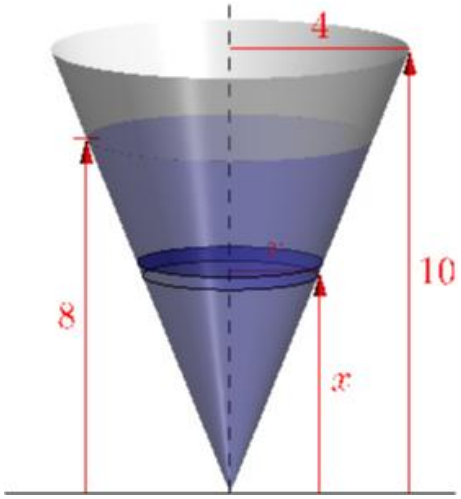


Figure 10.22. Work Cone (Asymptote Interactive 3D Image)

These 3D images in HTML output are rotable with a pointing device (mouse, trackpad) with a click-and-drag. A finger should suffice on touch-sensitive devices (phones, tablets). Zooming in and out can be accomplished with a mouse wheel, or by pinching. As a contribution to the accessibility of PreTeXt HTML output, keyboard controls will also allow for exploration of these images. (Make sure the image has focus when you attempt to use these.)

Table 10.23. 3D Image Keyboard Controls

Key	Action
x	Rotate around x-axis
y	Rotate around y-axis
z	Rotate around z-axis
+	Enlarge image
-	Shrink image
h	Return to home position

And finally, an example of a 3-D graph (from the Asymptote documentation again). This WebGL image is a beautiful example of a Riemann surface. As you rotate the image, notice how the reflection of the light source varies, along with the brightness of various regions of the surface. This example is accomplished with just 10 lines of Asymptote code.

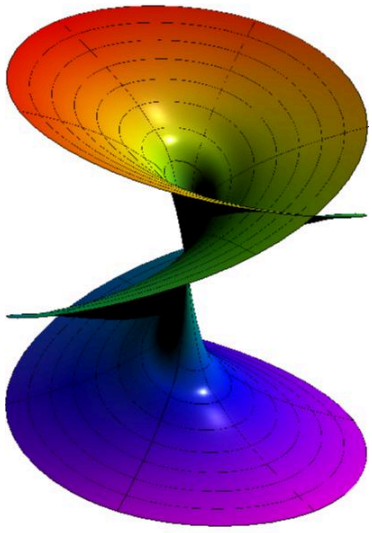


Figure 10.24. Asymptote 3-D Surface

10.6 Mermaid Diagrams

Mermaid is a Markdown-inspired tool for authoring various kinds of diagrams. Below, three of the available diagram types are demonstrated. For a full listing of diagram types, see the Mermaid Documentation. The Mermaid live editoris a great tool for testing the syntax of your mermaid diagrams.

In PreTeXt, you can specify a Mermaid theme via the `@theme` in the `common/mermaid/` publisher variable.

For HTML output, if you switch back-and-forth between light-mode and dark-mode, you will need to refresh the page to see the changes in the Mermaid diagrams.

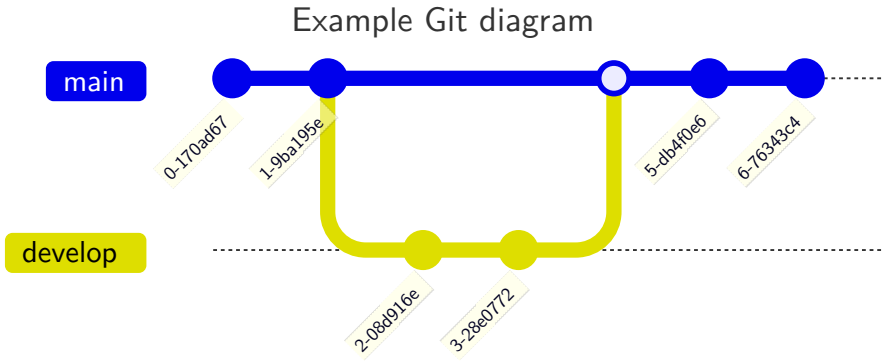


Figure 10.25. Mermaid Git Diagram

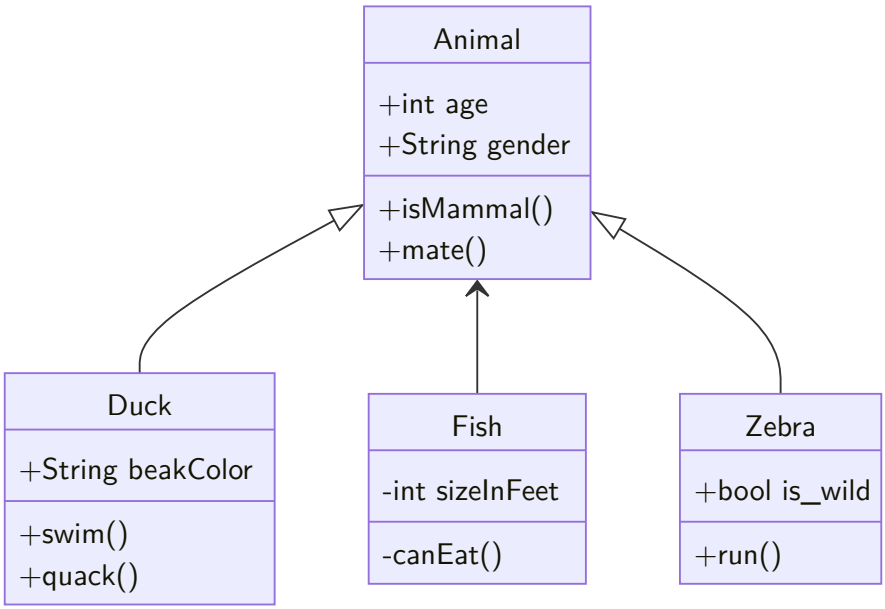


Figure 10.26. Mermaid Class Diagram

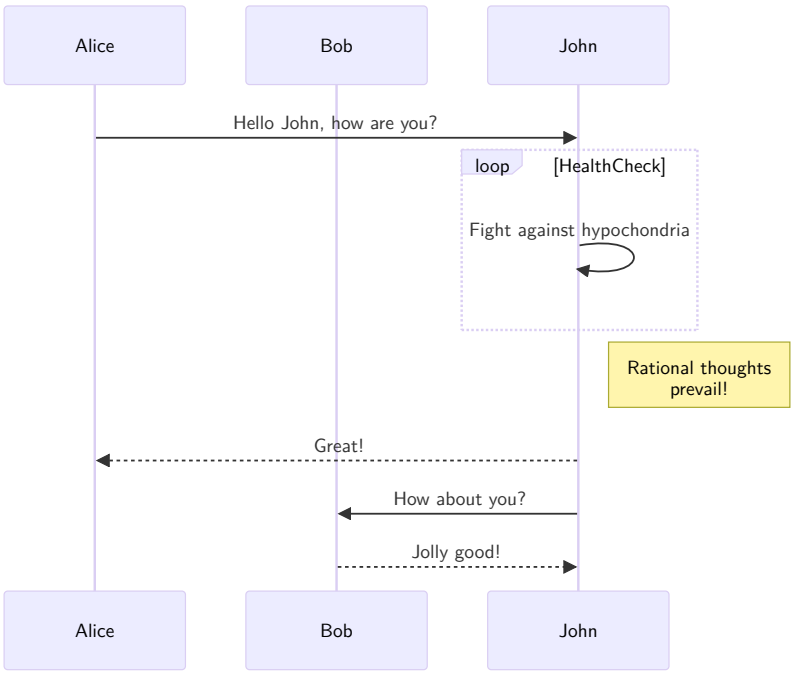


Figure 10.27. Mermaid Seuqnece Diagram

Mermaid has two layout engines—the default and `elk`. The ELK engine often does a better job with complex diagrams. You can specify it as a default by setting the publisher variable `<common/mermaid/@layout-engine>` to `"elk"`, or specify it in a single diagram by using standard Mermaid config frontmatter as shown in 10.28 below.

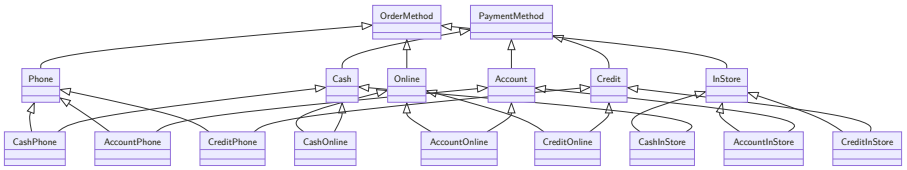


Figure 10.28. Mermaid Class Diagram using ELK layout engine

10.7 Sage Plots

Any of the numerous capabilities of Sage may be used to produce any graphics object, be it the simple graph of a single-variable function or some realization of a more complicated object. All of the usual rules about formatting Sage code (esp. indentation) apply, along with one more caveat. The last line of your Sage code ***must*** return a Sage `Graphics` object (or 3D plot). The `pretext` script will isolate this last line, use it as the RHS of an assignment statement, and the Sage `.save()` method will be called to generate the image, which is either a Portable Document Format (PDF) file amenable to LaTeX output, or a Scalable Vector Graphics (SVG) file amenable to HTML output. It is also possible to make a PNG image, which is necessary for an EPUB destined for a Kindle book. For visualizations of 3D plots, Sage will only produce Portable Network Graphics (PNG) files, which can be included in HTML pages or LaTeX output. For complete documentation, see the PreTeXt Guide as this subsection is not comprehensive.

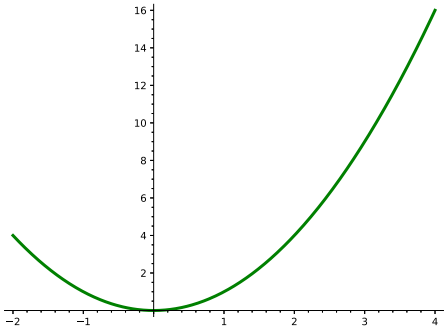


Figure 10.29. A Sage standard parabola, on $[-2, 4]$

Pay careful attention to the requirement that the last line of your code be a graphics object. In particular, while `show()` might appear to do the right thing, it evaluates to Python's `None` object and that is just what you will get. The code for Figure 10.30 illustrates creating two graphics objects and combining them into an expression on the last line that evaluates to a graphics object.

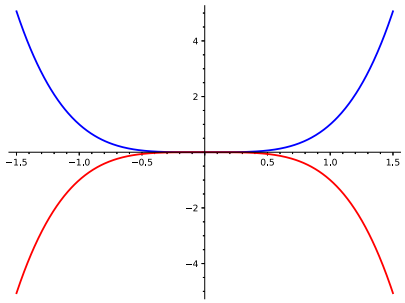


Figure 10.30. Two Sage plots on one set of axes

Sage code comprised of just a single line was once mishandled, leading to *no ouput*. From Jean-Sébastien Turcotte we have the example that revealed the problem.

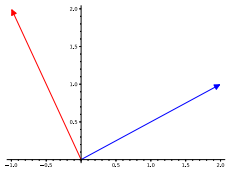


Figure 10.31. Les vecteurs $\vec{u}$ et $\vec{v}$

The following examples are from the Sage Tour. We package them into a `sidebyside` layout element, see Section 26.

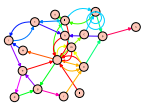


Figure 10.32. A Sage multigraph of a sentence

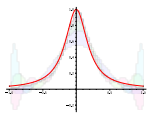


Figure 10.33. Sage polynomial approximations of $f(x) = 1/(1 + 25x^2)$

From the Sage documentation, with slight modifications, credited to Douglas Summers-Stay. A plot of the implicity defined surface

$$2 = \cos(x + ty) + \cos(x - ty) + \cos(y + tz) + \cos(y - tz) + \cos(z - tx) + \cos(z + tx)$$

in rectangular xyz coordinates, with t equal to the golden ratio. If you set `plot_points=100` in the Sage code, you will get a very smooth rendering, but also a quite large HTML file. We have used `plot_points=50` to reduce the file size by a factor of four. Note the need for a value of `3d` for the `@variant` attribute, and an explicit aspect ratio with `@aspect`. Arrow keys, a mouse scroll wheel, plus grabbing with a left or a right mouse button, can be used to manipulate the image.

Figure 10.34. A Sage implicitly defined 3D surface

10.8 Inkscape Images

Inkscape is a great tool for creating images. It ticks all the boxes: open source, mature, cross-platform, standards-compliant. Read much more about it in The PreTeXt Guide. In HTML output the two images below are both in SVG format. The first is "pure" SVG, while the second has embedded information that makes it easier to edit in Inkscape. You could view the source for this page in the HTML version, deduce the filename of the second image, download it, and manipulate it profitably with Inkscape. Both files are quite small, but the first is half the size of the second. In PDF the two images come from files that are identical, so nothing is being tested. The PDF version is smaller still.

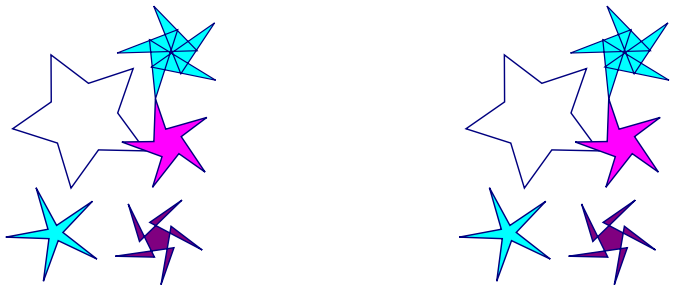


Figure 10.35. Inkscape Stars, Plain SVG (left), Inkscape SVG (right), from Bethany Llewellyn

10.9 Copies of Images

Sometimes you want to use the same image more than once. Here we just point to a PNG file that we repeat often throughout this sample.

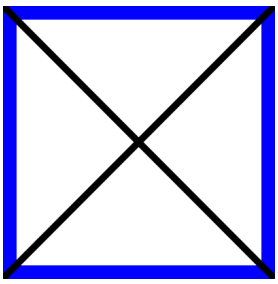


Figure 10.36. Copy of raster image, in a figure, so now numbered and captioned

For images described by code, such as TikZ code in a `<latex-image>` element, this is a bit subtler. See the PreTeXt Guide for a complete description. We also demonstrate this with the sample book, since it is all set up with the `xinclude` mechanism. See the two plots of the 8-th roots of unity in the complex numbers section of the chapter on cyclic groups.

10.10 Caption Testing

A caption could be as substantial as a paragraph, here we test out one such example.

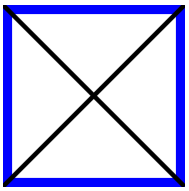


Figure 10.37. A caption can be a whole paragraph with lots of technical details, and maybe a hyperlink to something external, such as `pretextbook.org` or PreTeXt. There could be some inline mathematics, such as $x^2 + y^2 = c^2$. Would a knowl open here? Recursively? Let's see: 10.37. Display mathematics, side-by-sides, theorems, and lots of other things should be banned. Footnotes sound like a bad idea. Strange characters should be fine: §.

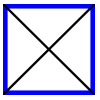
10.11 Captionless Images

We strongly suggest placing images within a `<figure>`, as we have done above, so that you can reference them, and use the (required) `<caption>` to explain what they are. However there are places, such as a `<preface>`, where numbered items are not permitted. So you might want a solo image there. Or maybe graphics are an illustration of sorts, and a numbered figure feels like overkill. Or it is part of an `<exercise>` or `<proof>` of a `<theorem>`. But notice that you cannot then use this image as the target of a cross-reference, so you may need to refer to some enclosing container.

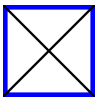
The image can be scaled by specifying the `@width` as a percentage, including the percent-sign (%). The height is scaled to preserve the aspect ratio. There is no facility to change the height, it is your responsibility to manage the aspect ratio independently. The `@margins` can be given as a pair of percentages, separated by a space. The `@width` defaults to 100%, while `@margins` defaults to the value `auto`, which will center the image. Missing values are computed

sensibly, and there is robust error-checking. The layout control here is a subset of what is available for the more elaborate `<sidebyside>` element, see Section 26.

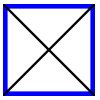
Two simple examples. The first has width `10%` and so defaults to being centered, and the second has width `10%` and left margin of `25%`.

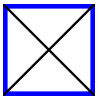


A paragraph, just to show where the first stops and the second ends.

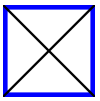


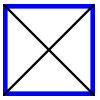
You might wish to place a single image flush-left, or flush-right. You can specify the `margins` attribute as a pair of percentages for different left and right margins. The following are laid out with two margins, with a 0% left margin and right margin (respectively).





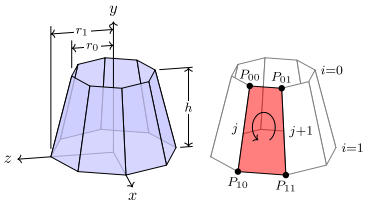
We place two images right above one another, to test spacing of consecutive images (provided they stay on the same page!).



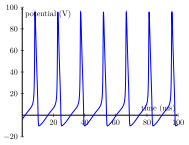


Testing (2019-06-02). All the images above are specified by filenames. We need to test how various options behave when incorporated into the (new) implementation for images, being introduced with solo images.

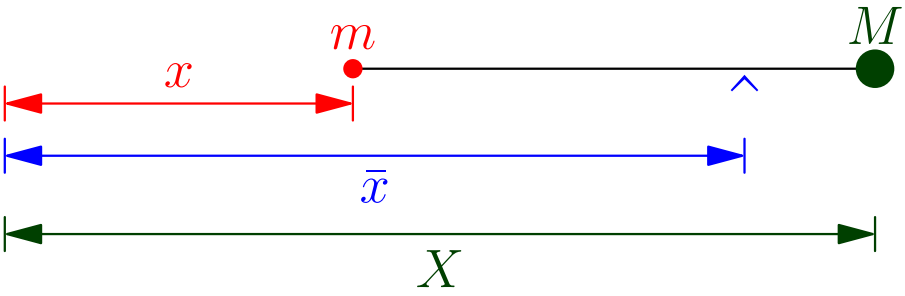
A `tikz` image recycled from above, now 40% width, with 40% left margin, 20% right margin.



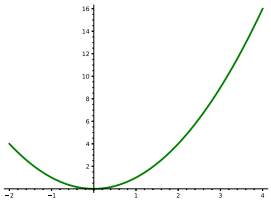
A `pgfplot` image recycled from above, now 20% width, with 40% left margin, 40% right margin, and no longer legible.



An Asymptote image, with zero layout control, so 100% width.



A Sage example, pushed to the right margin.



10.12 Technical Details

The table below is a summary of how graphics and images are specified, constructed and manipulated. Additional processing is indicated by reference to the Python script `pretext`. Images need to be placed relative to the LaTeX file that includes them during compilation, and placed relative to the HTML files which reference/include them. Author-provided image files may be placed in any subdirectory, and the `@source` attribute should include the complete relative path with the subdirectory. Files generated by the `pretext` script will be specified in the output using the relative directory `images`, which can be changed. There is no reason author-provided files cannot also be placed in this same directory (presuming no duplicate names). [This table is presently more readable in HTML, the PDF version will improve.]

Element	Specification	LaTeX/ Print	HTML	Notes
`image/ @source`	full relative path, w/ extension	directly included	directly included	author-provided PNG, JPEG
`image/ @source`	full relative path, w/ o extension	presumes PDF	presumes SVG	author-provided
`image/latex-image-code`	LaTeX-compatible source	directly included	SVG via `pretext`	e.g. tikz, pgfplots, xypic
`image/ sageplot`	Sage code	PDF via `pretext`	SVG via `pretext`	PNG for 3-D
`image/ asymptote`	Asymptote code	PDF via `pretext`	SVG via `pretext`	

In the early stages of a writing project, it may be best not to track provisional image files built with `pretext` under version control, and just regenerate them periodically (see the `-r` option for `pretext`). As a project

matures, then it makes sense to put stable files under version control for collaborators and others. In every case, managing graphics files (and other aspects of production), is much more pleasurable with a script (shell, Makefile, etc.)

10.13 Paragraphs as Figures

A `<figure>` usually holds an image or similar planar media, but it may instead hold a single paragraph. This serves content that is honestly a captioned, numbered display, just built from text. Displayed mathematics is the archetype: the addition table of a finite ring is no less a figure than a plot would be.

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Figure 10.38. Addition table for $\mathbb{Z}_5$

The humanities have the same need. The transcription of an inscription on a monument, a statue, or a coin is display material bound to its caption, not a paragraph of the surrounding argument.

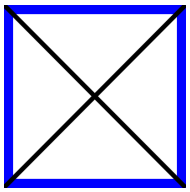
M · AGRIPPA · L · F · COS · TERTIVM · FECIT, "Marcus Agrippa, son of Lucius, three times consul, built this."

Figure 10.39. Inscription on the architrave of the Pantheon, Rome

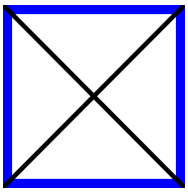
10.14 Accessibility

An `<image>` should either have a non-empty `<description>`, a non-empty `<shortdescription>`, or set `@decorative` to the value `yes`. Some of the following images comply, and some do not. There's not really anything to see here visually, this is testing notifications made elsewhere.

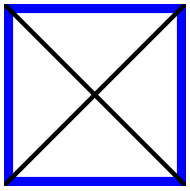
No description, no shortdescription, no `@decorative` set to `yes`



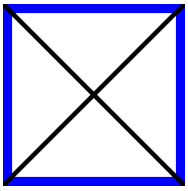
No description, no shortdescription, `@decorative` set to `yes`



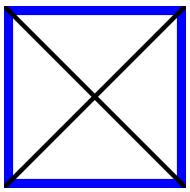
No description, empty shortdescription



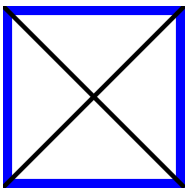
No description, too long shortdescription



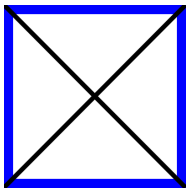
No description, good shortdescription



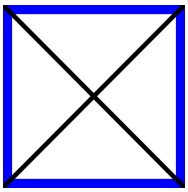
Description, no shortdescription



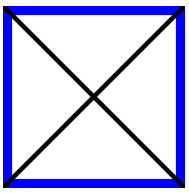
Description, empty shortdescription



Description, too long shortdescription



Description, good shortdescription



11 Further Reading

11.1 Specialized Subdivisions

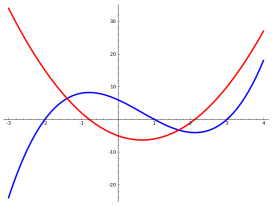
In a longer work you might wish to have some references on a per-chapter basis, or similar. You can make a "references" subdivision anywhere to hold bibliographic items, and you can reference the items like any other item. For example, we can cite the article below [11.4.3, Chapter R], included an indication that a specific chapter may be relevant.

11.2 Exercises

1. No problem here, but the next two are in an "exercise group" with an introduction and a conclusion, along with an optional title. The two problems of the exercise group should be indented some to indicate the grouping.

N.B. An `<exercisegroup>` is meant to hold a collection of (short) exercises with common, shared, instructions. Do not use this structure to subdivide an `<exercises>` division, as you will eventually be disappointed. Instead, use the available, but under development as of 2019-11-02, `<subexercises>`, which *requires* a `<title>`.

In the next two problems compute the indicated derivative.



You could "connect" the image above with the exercises following as part of this `introduction` for the `exercisegroup`.

2. $f(x) = x^3$, $\frac{df}{dx}$. This sentence is just a bunch of gibberish to check where the second line of the problem begins relative to the first line.

We cross-reference the next problem in this exercise group. For the `phrase-global` form, the common element of the cross-reference and the target should be the `exercises` division, and not the enclosing `exercisegroup`: Exercise 3 of Exercises 11.2.

3. $y = \cos(x)$, y'.

Note that the previous two problems used very different notation for the function and the resulting derivative.

4. This isn't really an exercise, but an explanation that the next `<exercisegroup>` has a title and no `<introduction>`, which once resulted in some aberrant formatting in LaTeX output.

Some common instructions would go here in the `<introduction>`

5. $f(x) = x^3$, $\frac{df}{dx}$. This sentence is just a bunch of gibberish to check where the second line of the problem begins relative to the first line.

We cross-reference the next problem in this exercise group. For the `phrase-global` form, the common element of the cross-reference and the target should be the `exercises` division, and not the enclosing `exercisegroup`: Exercise 3 of Exercises 11.2.

6. $y = \cos(x)$, y'.

7. Compute $\int 3x^2\,dx$.

8. One of the few things you can place inside of mathematics is a "fill-in" blank. We demonstrate a few scenarios here. See details on syntax in Subsection 4.7– the use is identical within mathematics.

- Inside inline math (short, space for x): sin( )
- Inside inline math (default, space for XXX): sin( )
- Inside exponents and subscripts (each is space for the string "12"). In this case, be sure to wrap your exponents and subscripts in braces, as would be good LaTeX practice anyway: x^{5+} y
- Inside inline math (too long for this line probably, 40 characters long): tan( )
- So use inside a displayed equation
$$16 \log$$
like this one.
- Inside the second line of a multi-line display:
$$y = x^7\, x^8$$
$$= x$$
- This fillin has the historical `@characters` attribute for a fillin inside math: $1 +$ $+ 4 = 10$, which may be more convenient, but may not side properly in places like subscripts, superscripts, fractions, limits of integrals, and so on.

11.3 More Exercises

This exercises section should start on a new page in PDF output if the standard publisher file is used.

1. This is not a real exercise, we just want to explain that this is another subsection of exercises, which has two consecutive exercise groups.

Introduction to first exercise group.

2. Only exercise of first group.

Conclusion to first exercise group.
Introduction to second exercise group.

3. First exercise of second group.

4. Second exercise of second group.

Conclusion to second exercise group.
An `<exercisegroup>` can have a `cols` attribute taking a value from 2–6. Exercises will progress by row, in so many columns. On a small screen, the HTML exercises may reorganize into fewer columns.

5. $1 + 2$

6. $3 + 4 + 5$

Hint.
Addition is associative.

Answer. 12

Solution.
First, add 3 and 4 to get 7, then add 5 to arrive at 12.

7. $5 + 6$

8. Add seven to eight.

Answer. 15

9. $9 + 10$

10. $1 + 2$

11. $3 + 4 + 5$

Hint.
Addition is associative.

Answer. 12

Solution.
First, add 3 and 4 to get 7, then add 5 to arrive at 12.

Proof. A simple argument. ■

12. $5 + 6$

And a bit more.

13. Add seven to eight.

Answer. 15

14. $9 + 10$

15. $1 + 2$

16. $3 + 4 + 5$

Hint. Addition is associative.

Answer. 12

Solution. First, add 3 and 4 to get 7, then add 5 to arrive at 12.

17. $5 + 6$

18. Add seven to eight.

Answer. 15

19. $9 + 10$

20. $1 + 2$

21. $3 + 4 + 5$

Hint. Addition is associative.

Answer. 12

Solution. First, add 3 and 4 to get 7, then add 5 to arrive at 12.

22. $5 + 6$

23. Add seven to eight.

Answer. 15

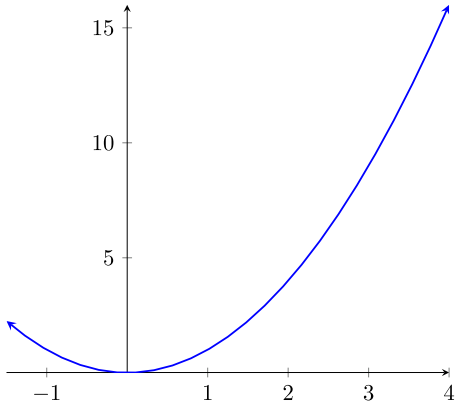
24. $9 + 10$

This feature was designed with short "drill" exercises in mind. With long exercises, or exercises with long hints, answers, or solutions, there is a risk that the LaTeX output will have bad page breaks in the vicinity (just before) such an exercise that occupies too much vertical space. Edit, rearrange, or use fewer columns to see if the situation improves.

25. Make a table and a graph for the function $f(x) = x^2$.

Answer.

x	$f(x)$
0	0
1	1
2	4
3	9



11.4 References

These items are here to test basic formatting of references. Did you see that the entry for [11.4.3] has a short annotation? So you can make annotated bibliographies easily.

[1] Gilbert Strang, *The Fundamental Theorem of Linear Algebra*, The American Mathematical Monthly November 1993, **100** no. 9, 848–855.

[2] J. B. Conrey and D. W. Farmer, *Mean values of L-functions and symmetry*, Internat. Math. Res. Notices no. 17 (2000) pp. 883-908.

[3] Robert A. Beezer, *A First Course in Linear Algebra*, 3rd Edition, Congruent Press, 2012.

An online, open-source[1] offering.

[4] H. Davenport, *Multiplicative Number Theory*, GTM, **74** Springer-Verlag New York, NY, (2000) xiv+177, A note may accompany a bibliographic item, such as saying the manuscript is under review. But it cannot contain any formatting..

[5] Alexander Rosswell, *Diffeomorphisms of Penciled Fiber Bundles*, Mathematicians of America (2020), **2** no. 6, 884–888.

[6] Ibid., *Diffeomorphisms of Penciled Fiber Bundles, Part 2*, Mathematicians of America (2021), **3** no. 4, 102–103.

12 List Calisthenics

12.1 Lists, Generally

Use `ol` to make an ordered list, and `ul` to make an unordered (bulleted) list. In both cases, use `li` for each entry. If an entry contains more than one paragraph, then each must be wrapped in `p`.

This section contains nested lists, to demonstrate how they get assigned labels (numbering, symbols). But we begin with two simple lists, demonstrating an ordered list and an unordered list. See the end of section for an example of

[1] A gratuitous footnote to test prior bug confusing this with a REMARK-LIKE `<note>`.

a description list. Note in the source the optional use of a paragraph (`p`) for the list items of the list of colors.

1. First.
2. Second[1].
3. Third.

- Red[2]
- Green
- Yellow
- Purple

Next, we have a list with no customization and multiple levels to test the defaults. LaTeX allows a maximum of four levels of ordered/numbered lists, and a total of six levels if some unordered lists are mixed in. Note that to have nested lists you *must* structure your list items as paragraphs, since a list may only appear within a `<p>` element.

1. *A title on a top-level item.*
   Level 1, first.
2. Level 1, second.
   (a) Level 2, first.
   (b) Level 2, second.
      i. Level 3, first.
      ii. Level 3, second.
         A. Level 4, first.
         B. Level 4, second.
         C. *Title on xref'ed list item.*
            Level 4, third.
      iii. Level 3, third.
   (c) *A title on a nested item.*
      Level 2, third.
3. Level 1, third.

Items in ordered lists (only) may be be give an `xml:id` and then may be the target of an `xref`. We test three here, referencing down into the hierarchy above. Level 1, second: 2. Level 3, second: 2.b.ii. Level 4, third: 2.b.ii.C. Note that if a list item of an ordered list is contained within a list item of an unordered list, then its number will not be defined.

And now a four-level deep unordered list with the default labels supplied by PreTeXt (disc, circle, square, disc). Again, the defalt order for Markdown/ Jupyter (disc, square, circle, circle) is different than for LaTeX and HTML (disc, circle, square, disc)

- *A title on a top-level item.*
  Level 1, first.
- Level 1, second.
  - Level 2, first.
  - Level 2, second.
    - Level 3, first.
    - Level 3, second.

[1] Footnote in an unstructured list item
[2] One of our favorite colors

- Level 4, first.
- Level 4, second.
- Level 4, third.

▪ *A title on a nested item.*
Level 2, third.

◦ Level 2, third.

• Level 1, third.

And a total of six levels with a mix of ordered and unordered lists, the most that out-of-the-box-LaTeX is able to handle.

1. Level 1, first.
2. Level 1, second.

(a) Level 2, first.
(b) Level 2, second.

- Level 3, first.
- Level 3, second.

i. Level 4, first.
ii. Level 4, second.

A. Level 5, first.
B. Level 5, second.

◦ Level 6, first.
◦ Level 6, second.
◦ Level 6, third.

C. Level 5, third.

iii. Level 4, third.

- Level 3, third.

(c) Level 2, third.

3. Level 1, third.

Now, nested lists with the defaults replaced by custom choices. First, an ordered list, three deep, upper Roman numerals, then upper-case Latin, then more traditional Arabic numerals on the three elements of the third level. The adornments of the labels render in any browser supporting the `@counter-style` CSS rule (Safari from version 17, and much earlier for Firefox and Chrome); an older browser falls back to the plain numeration, without adornments.

I Level 1, first.
II Level 1, second.

+ Level 2, first.

+A

+ Level 2, second.

+B

1) Level 3, first.
2) Level 3, second.
3) Level 3, third.

+ Level 2, third.

+C

III Level 1, third.

A nested unordered list, with labels given as squares on the outer list and nothing (blank) on the inner lists.

▪ Level 1, first.
▪ Level 1, second.

Level 2, first.
Level 2, second.

- Level 1, third.

A nested ordered list, to test intramural cross-references.

1. Level 1, first.
2. Level 1, second.

   Level 2, first.
   Level 2, second.
3. Level 1, third. With a cross-reference to second list item, 2.
4. Level 1, fourth. Whose number should not change when the knowl just prior is opened.

An ordered list may begin at zero by using a numeral zero in the `@label` attribute, instead of numeral one.

(0) First
(1) Second

   -0- Uno
   -1- Dos
   -2- Tres

(2) Third

The next definition is very poorly worded. It is meant to test leading off with a list (bad form), for which LaTeX normally begins right after the heading.

Definition 12.1 Group.

a) There is a binary operation, denoted "$\cdot$".
b) The operation is associative.
c) There is an identity element, e.
d) For every element b, there is an element c (the inverse), such that

$$b \cdot c = c \cdot b = e.$$

If these conditions are met for a set G, then we say G is a **group**. ◆

Exercises and References are specialized subdivisions you can put anywhere. They are implemented as top-level lists, so should share behavior. For example, an exercise may have many parts and when expressed as a list, should have the expected labels.

Similarly, References may have lists in their annotations. Unlikely? But possible.

The next two subdivisions are an Exercises subdivision and a References subdivision, which have lists within an exercise and a bibliographic item (respectively).

12.2 List Spacing, I

This is a short list that ends a subsection, so can be used to address the necessary spacing. We also test two XML elements separated by a space (which should not go missing).

(a) One item.

(b) *Two* ***ducks***.

(c) Three items. Plus a few more words to check that long entries in a two column list look good.
(d) Four items.
(e) Another long entry that simultaneously tests that long entries look good in a list, and also tests an odd number of entries in a two column list.

12.3 List Spacing, II

This is another short list that ends a subsection, so can be used to address the necessary spacing.

- Uno item.
- Dos items.

  Aside List column test. An aside in a multi-column list. We don't want this to float away from its context and/or break the layout. This also tests if list title formatting affects things inside the list.
- Tres item.
- Quattro items.

And a paragraph after that list so that spacing can be checked.

12.4 List items containing only inline math

Testing list items containing only math.

There are many places where it makes sense to have a list of mathematical terms, or possibly equations. For example, one might wish to provide a list of derivative formulas. With such lists, the author may wish to have display mathematics, but almost certainly they don't want it centered. One can work around this by using the LaTeX `\displaystyle` command. However, it would be nice if a list item containing only math used display mode by default.

1. A list item containing some text in a paragraph, as well as some inline math: $\int_a^b x^2\,dx = \frac{x^3}{3}$.
2. A list item with text and math $\int_a^b x^2\,dx$, not in a paragraph.
3. $\dfrac{d}{dx}\tan(x) = \sec^2(x) = \dfrac{1}{\cos^2(x)}$
4. $\displaystyle\sum_{i=0}^{n} r^i = \frac{1-r^{n+1}}{1-r}$

Now, a `p` that isn't in a list, followed by a list that's in a `p`.

- $\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$ A list item starting with some math, followed by text, all in a `p`
- $\dfrac{a}{b} \div \dfrac{c}{d} = \dfrac{ad}{bc}$
- $\tan\theta = \dfrac{\sin\theta}{\cos\theta}$

The above assemblage had some lists in it, just to see what will happen. While we're at it, we might try adding lists that are in a `list`.

List 12.2. A list of items, some of which contain math

1. A first list item, containing some text. The next list item will contain only math, with the `m` tag inline with the `li` tag.
2. A list item with text and math $\int_a^b x^2\,dx$, not in a paragraph.
3. $$\int_a^b f'(x)\,dx = f(b) - f(a)$$
4. The next two list items will contain, respectively, a list item containing only math, where the math is on a new line, then the same again, but with two new lines, and a list item containing math within a `p`, first inline, and then after a line break.
5. $$\frac{d}{dx}\sec^{-1}(x) = \frac{1}{x\sqrt{x^2-1}}$$
6. $$\frac{d}{dx}\sec^{-1}(x) = \frac{1}{x\sqrt{x^2-1}}$$
7. $$\frac{d}{dx}\sec^{-1}(x) = \frac{1}{x\sqrt{x^2-1}}$$
8. $$\frac{d}{dx}\sec^{-1}(x) = \frac{1}{x\sqrt{x^2-1}}$$

And now, a list in a paragraph.

- A paragraph that begins with text, then some math: $\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$ And now some more text. The next two list items contain:
    1. Math only, inline.
    2. Math only, with a newline.
    3. Math only, but in a paragraph. Also the next item on this list has math, just to see what happens in a nested list.
    4. $$\int_a^x \frac{d}{dt} f(t)\,dt = f(x) - f(a)$$
- $$\frac{a}{b} \div \frac{c}{d} = \frac{ad}{bc}$$
- $$\tan\theta = \frac{\sin\theta}{\cos\theta}$$
- $$\frac{a}{b} > \sum_{i=3}^{76} \frac{x^2}{y^2}$$

Inclusion of any text other than math will kill the automatic display style. For example, this would happen if one were to add punctuation after the math.

- $\int_a^b \frac{\sin(x)}{x}\,dx$,
- $$\int_a^b \frac{\sin(x)}{x}\,dx$$

List items can have titles. We try that here, along with testing list items structured with paragraphs.

- *With* `\displaystyle` *added automatically.*
  $$\sum_{n=1}^{\infty} ar^n$$
- *Two paragraphs.*
  $\sum_{n=1}^{\infty} ar^n$
  $\sum_{n=1}^{\infty} ar^n$
- *One paragraph, extra text.*
  So, $\sum_{n=1}^{\infty} ar^n$

- *Two elements (only).*
  LaTeX$\sum_{n=1}^{\infty} ar^n$

12.5 Difficult List Items

In Subsection 12.4 we were careful about lone bits of math inside list items. The `<cd>` element is used with indentation, which is likely superfluous inside a list item that is already being indented. Here we test lone `<cd>` elements inside of list items in various configurations.

Unordered list, one-deep.

- `Foo Bar Foo`
  `Bar Foo Bar`
- This list item is a long paragraph with a `<cd>` in the middle which should be indented some to indicate its participation in the paragraph.

  ```
  Foo Bar Foo
  Bar
  Bar Foo Bar
  Foo
  ```

  This list item is a long paragraph with a `<cd>` in the middle which should be indented some to indicate its participation in the paragraph.
- `Foo Bar Foo`
  `Bar`
  `Bar Foo Bar`
  `Foo`

Intervening paragraph, to illuminate spacing at both the top and bottom of a list. Intervening paragraph, to illuminate spacing at both the top and bottom of a list. Intervening paragraph, to illuminate spacing at both the top and bottom of a list. Intervening paragraph, to illuminate spacing at top and bottom of a list.

Ordered lists, two-deep, mixed.

1. First item, outer level.
   (a) `First inner item, cd only`
   `Foo Bar Foo`
   `Bar`
   `Bar Foo Bar`
   `Foo`
   (b) Second inner item, a paragraph in a list item.
   (c) `Third inner item, cd only`
   `Foo Bar Foo`
   `Foo Bar Foo`
   `Bar Foo Bar`
2. `cd inside second item, outer level`
   `Bar`
   `Bar Foo Bar`
   `Foo`

12.6 Description Lists

Use `dl` to make a description list. Inside of those tags, use `li` for each entry. Then, use `title` to specify the term being described and `p` to specify the description.

A "description" list has a short term or phrase that is prominent, followed by a short description. It is modeled on the lists of similar structure in both LaTeX and HTML. It makes for a nice medium-weight way to define terms, somewhere in-between the `term` tag which just makes a term prominent in a sentence, and a `definition`, which is set off, has a heading, a number, and a title. Do not try to manage the separation between the title and the description by employing punctuation (but you can include a question-mark or exclamation-point if necessary). For example, do not include a colon to the end of the title. This example is from Bob Plantz.

Central Processing Unit (CPU) Controls most of the activities of the computer, performs the arithmetic and logical operations, and contains a small amount of very fast memory.

This is a second paragraph that should appear indented in PDF output.

Memory Provides storage for the instructions for the CPU and the data they manipulate.

Input/Output (I/O) Communicates with the outside world and with mass storage devices (e.g., disks).

Bus! A communication pathway with a protocol specifying exactly how the pathway is used. (The punctuation is just for testing.)

$$\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}$$ A geometric series. The formula is valid if $|x| < 1$.

Some presentations can be assisted by a hint from the author about the lengths of the titles. You can choose to provide a `width` attribute on a `dl` element with possible values `narrow`, `medium`, and `wide`. The value refers (somewhat confusingly) to the distance between the left margin and the description. The default is `medium`, which is illustrated above. An example with `narrow`:

Red The color of the sun at sunset.

Blue The color of a clear sky. Also a synonym for "depressed or sad", the title of a 1971 Joni Mitchell album (and more than a dozen other musical albums), the period of Picasso's work between 1901 and 1904, and much more!

Aqua The color of shallow tropical waters.[3]

Math x^2 Sorry, not a color but testing titles with math in them.

"i" before "e" except after "c," unless it sounds like "a" as in "neighbor" Get feisty about that weird counterfeit rule: seize the day and don't have a heifer, man.

[3] On a sunny day! (Testing footnotes in description lists for LaTeX output.)

and "weigh"

Avocado Avocado is the the color with hex code `#568203`, and also the main ingredient in guacamole.

Magenta Magenta is a color, and a character in Rocky Horror.

Zymurgist A scientist who studies the chemical process of fermentation in brewing and distilling. Also the alphabetically last 9-letter word in the English language.

Byzantium Byzantium is the the color with hex code `#702963`, and also an ancient Greek city which later became known as Constantinople, and today is called Istanbul.

Convection Circulating motion in a fluid.

Elementary No literary detective ever said "Elementary my dear Watson." In particular, Sherlock Holmes never said that.

Understand Perceive the intended meaning of.

Washington A state, a district, the man on the US $1 bill and on the US quarter. Did you ever notice that on the US dime, the value is stated as "one dime"? But how is one to know that a dime is worth 10 cents?

Aquamarine Aquamarine is a color, and a mineral.

Those who cannot remember the past are condemned to repeat it. George Santayana wrote those words in 1905. A similar aphorism is misattributed to Winston Churchill. The idea is embodied in the 4th principle: PreTeXt respects the good design practices which have been developed over the past centuries.

$\zeta(s) = \sum_{n=1}^{\infty} n^{-s}$ The Riemann ζ-function is defined by a Dirichlet series, valid for $\Re(s) > 1$.

`main()` is a void function A `dl` with `width="narrow"` might be a useful way to give commentary on a program listing.

And the same example with `wide`:

Red The color of the sun at sunset.

Blue The color of a clear sky. Also a synonym for "depressed or sad", the title of a 1971 Joni Mitchell album (and more than a dozen other musical albums), the period of Picasso's work between 1901 and 1904, and much more!

Aqua The color of shallow tropical waters.[4]

[4] On a sunny day! (Testing footnotes in description lists for LaTeX output.)

Math x^2 Sorry, not a color but testing titles with math in them.

"i" before "e" except after "c," unless it sounds like "a" as in "neighbor" and "weigh" Get feisty about that weird counterfeit rule: seize the day and don't have a heifer, man.

Avocado Avocado is the the color with hex code `#568203`, and also the main ingredient in guacamole.

Magenta Magenta is a color, and a character in Rocky Horror.

Zymurgist A scientist who studies the chemical process of fermentation in brewing and distilling. Also the alphabetically last 9-letter word in the English language.

Byzantium Byzantium is the the color with hex code `#702963`, and also an ancient Greek city which later became known as Constantinople, and today is called Istanbul.

Convection Circulating motion in a fluid.

Elementary No literary detective ever said "Elementary my dear Watson." In particular, Sherlock Holmes never said that.

Understand Perceive the intended meaning of.

Washington A state, a district, the man on the US $1 bill and on the US quarter. Did you ever notice that on the US dime, the value is stated as "one dime"? But how is one to know that a dime is worth 10 cents?

Aquamarine Aquamarine is a color, and a mineral.

Those who cannot remember the past are condemned to repeat it. George Santayana wrote those words in 1905. A similar aphorism is misattributed to Winston Churchill. The idea is embodied in the 4th principle: PreTeXt respects the good design practices which have been developed over the past centuries.

$\zeta(s) = \sum_{n=1}^{\infty} n^{-s}$ The Riemann ζ-function is defined by a Dirichlet series, valid for $\Re(s) > 1$.

`main()` is a void function A `dl` with `width="narrow"` might be a useful way to give commentary on a program listing.

You can nest description lists, though items might get crowded horizontally. We expand our computer list a bit.

Expanded CPU Controls most of the activities of the computer, performs the arithmetic and logical operations, and contains a small amount of very fast memory.

Arithmetic The kind of computations that give a computer the name "computer."

Logic Operations that make decisions, "if this, then that", plus: and, or, and not.

Memory Space to store information and results, permanently or temporarily.

This is a second paragraph that should appear indented in PDF output.

Memory Provides storage for the instructions for the CPU and the data they manipulate.

Input/Output (I/O) Communicates with the outside world and with mass storage devices (e.g., disks).

Bus! A communication pathway with a protocol specifying exactly how the pathway is used. (The punctuation is just for testing.)

$$\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}$$ A geometric series. The formula is valid if $|x| < 1$.

12.7 Named Lists

A list can be wrapped with a `<list>` element, so that it earns a number, can be given a title and have an introduction and conclusion. Cross-references to individual list items get a bit involved as they are prefixed with the number of the list and then the number of the item, so conceivably you could get a number like `4.5.3:2.a.ii`. The colon is used to indicate the transition from the number of the list within divisions and the numbers coming from the list hierarchy, since it has two small dots.

List 12.3. Colors of the Rainbow

Because the colors are always in the same order, an ordered list is natural here. The colors change continuously, but are often divided up into large ranges that human perception can easily distinguish.

1. Red
2. Orange
3. Yellow
4. Green
5. Blue
6. Indigo
7. Violet

So some people use the acronym ROY-G-BIV to remember this sequence.

This next list is used for testing cross-references to it. See Section 21.

List 12.4. A named list of targets

This is the introduction to this named list, which references an item within, via the hybrid `@text` attribute: Item B.c. At one time this paragraph was inadvertently centered—that bug has been fixed.

A i A and i
ii A and ii
iii A and iii

B a B and a
b B and b
c B and c (target of some cross-references)

C
- The next three cross-references point to a list item, just above. It is interesting because the list is named, hence numbered. The global reference uses the full number, while the local reference uses the number from within the list. The hybrid reference recognizes that the target is within the same named list, so the number can be shorter. An identical hybrid cross-reference appears within the `<introduction>` to this list, an immediately following, but outside the `<list>`.
- Cross-reference within named list (`global`): Item 12.4:B.c
- Cross-reference within named list (`hybrid`): Item B.c
- Cross-reference within named list (`local`): Item B.c
- 1 C and bullet and 1
  2 C and bullet and 2
  3 C and bullet and 3
- C and bullet
- C and bullet

This is a paragraph just outside the preceding named list, which references an item within, via the hybrid `@text` attribute: Item 4:B.c. This paragraph is in the same section as the list, so the number of the list is shortened, just as it is for a hybrid cross-reference to the list itself: List 4.

This is a paragraph with three lists contained within it. For HTML output we have to "inside-out" the lists.

1. A one item ordered list.

In other words, the text before, after, and between, needs to each be encapsulated as an HTML `p` element of its own.

- A one item unordered list.

Including definition lists.

Define Me A one item definition list.

That's all!

A one item list, whose item is a paragraph with two contained ordered lists, separated by text.

- Introductory text.

  A First item, first list.

  Intermediate text.

  a First item, second list.

  Concluding text.

12.8 Testing List Decompositions

A list in a paragraph is a construction in HTML that browsers try to correct, which leads to unpredictable results, so we have to decompose an author's paragraph with lists into a sequence of HTML paragraphs, interrupted by lists. This subsection is only relevant to HTML output, and only for testing.

1. This paragraph opens with an ordered list.
2. Testing the id, and other info that should be at the top of the paragraph.

Now the paragraph continues, and we have an index item here, so we can test cross-references back here.

12.9 List Column Testing

This is a list arranged into two columns with some intentional layout challenges. The math is too long to fit in one column and can't be wrapped - it reduces the number of columns in its row. The long text items can be wrapped and stay within their column.

(a) One item.

(b) *Two **ducks***.

(c)
$$\begin{aligned} 3u + 1v + 3w + 2x + y &+ 3z = 10 \\ 2u + 2v - 2w + x &+ z = 6 \\ 1u - 3v + 1w + x &+ 3y + 2z = 13 \end{aligned}$$

(d) Short item.

(e) A long item that can be wrapped over multiple lines.

(f) Four items.

(g)

(h) Another long entry that simultaneously tests that long entries look good in a list, and also tests an odd number of entries in a two column list.

12.10 Exercises (with lists)

1. This exercise has several parts, each a `<task>`, so the parts are labeled with letters, and a part within a part with Roman numerals. An ordered list inside a part is an anonymous list, with labels following the defaults for a top-level list, and with no number from the exercise.

(a) Exercise 1, first part, with a list.

1. First item of the list.
2. Second item of the list.

(b) Exercise 1, second part.

(i) Exercise 1, second part, first refinement.

(c) Exercise 1, third part.

Table 12.5. Table Alignment Example

1111, 2222		3333
aaaa	bbbb,cccc	
AAAA	BBBB	CCCC

This exercise (a list item really) has a `table` first. Default LaTeX aligns it vertically above the exercise number. Placement here tests correcting that alignment.

A small test of cross-references to the tasks of an exercise. Exercise 1, third part: 12.10.1.c. Exercise 1, second part, first refinement: 12.10.1.b.i.

12.11 References (with lists in Annotations)

[1] Some book would be listed here.
Here is the annotation and an ordered list as part of that annotation.

1. Book 1, first part.
2. Book 1, second part.
3. Book 1, third part.

13 Table Calisthenics

```
c = 832
c
```

That was a Sage cell just now, which has nothing to do with tables. But we needed someplace to test placement right after a division heading. Carry on.

A very minimal table, hence with left-justified cells, no borders. We do wrap the tabular element in a table element to get centering, numbering and a caption. A single table note (the `<tn>` element) is tested here.

Table 13.1. Some Colors

Red	Green[a]	Yellow
Blue	White	Pink

[a] Green can be a very sick looking color.

Table notes get a thorough workout in the next table: one in a header cell, one in a cell spanning two columns, two in a single cell of two paragraphs (one note anchored at the end of each paragraph), and one anchored mid-text of an ordinary cell. Letters accumulate in reading order, and begin again with each table.

Table 13.2. Table Notes in Demanding Locations

Construction[a]	**Behavior**	**Year**
A single cell of text long enough that it plainly runs across the boundary between the first and second columns.[b]		1971
A first paragraph, with a note anchored at its end.[c] A second paragraph, with its own note.[d]	A note[e] amid running text	2026

[a] A header cell may carry a note.

[b] The mark trails the spanning content.

[c] Anchored after the first paragraph.

[d] Anchored after the second paragraph.

[e] Anchored mid-text, at the word “note”.

Note that tables may be constructed using the LaTeX Complex Table Editor tool online at `latex-tables.com`and then exported in PreTeXt syntax.

An example of aligning table cells’ contents horizontally. See the source for comments.

Table 13.3. Horizontal Alignment Example

1234567890	1234567890	1234567890	1234567890
[First	Second	Third	Fourth
A	B	C	D
1	2	3	4

Example from above, but now with horizontal rules, plus an extra row to test the bottom border. See the source for comments.

Table 13.4. Horizontal Rules Example

1234567890	1234567890	1234567890	1234567890
First	Second	Third	Fourth
A	B	C	D
1	2	3	4
1	2	3	4

For a table without a caption, create a `<tabular>` and place it directly within the current division. This will allow control over the horizontal placment, but without a caption, there is no number, and the tabular cannot be cross-referenced.

One

Same example as before, but now with vertical rules. See the source for comments.

Table 13.5. Vertical Rules Example

1234567890	1234567890	1234567890	1234567890
First	Second	Third	Fourth
A	B	C	D
1	2	3	4
1	2	3	4

Table 13.6. Progressively Thicker Rules Example

1111	2222	3333
aaaa	bbbb	cccc
AAAA	BBBB	CCCC

Generating empty header cells as `<th>` elements is an accessibility issue. They should render as `<td>` elements instead. This next table tests that behavior.

Table 13.7. Empty Header Stress Test

	??	??
??	??	??

Table 13.8. Column Span Example

1111, 2222		3333
aaaa	bbbb,cccc	
AAAA	BBBB	CCCC

A list whose first item is a table. In LaTeX output a `\leavevmode` is necessary to keep this organized (item number, *then* table as content).

1. **Table 13.9.** Table Alignment Example

1111, 2222		3333
aaaa	bbbb,cccc	
AAAA	BBBB	CCCC

Example 13.10 Example Environment with Leading Table.

Table 13.11. Column Spans, No `col` Elements, Nine Columns

1	2+3		4	5+6+7			8+9	
1	2	3	4	5	6	7+8		9
1	2	3	4	5	6	7	8	9

This example tests several things. In LaTeX output, figures, tables, listings and side-by-sides are "floats" whose placement can migrate, but we have tries to supress this behavior. However, a float that is the first item of an "environment" (like a theorem or an example) can still float to a position *before* its title. If that does not happen here, then our additional defenses are working.

This example also checks that the total number of columns is correctly computed from the first row, which features several `colspan` attributes. ▲

A bare minimum table (one row with one cell) to test edge cases:

Table 13.12. One entry table

One

Table cells with a fixed width where text wraps are known as "paragraph cells". A cell will be created as a paragraph cell if and only if it has `<p>` children. And such cells should *only* have `<p>` children. The width of a paragraph cell is determined by a `width` attribute on the corresponding `<col>` (as a percentage). If the column has a non-paragraph cell with contents that are wider than the paragraph cells, results will be undesirable. There is presently no implementation for a paragraph cell that has a `colspan` greater than 1, although cells with `colspan` greater than 1 that are above or below a paragraph cell will behave. Setting `width` on a `<col>` that has no paragraph cells may produce unexpected results. A `valign` for the parent `<row>` (or the ambient `<tabular>`) can control vertical alignment (top, middle, or bottom). A paragraph cell's `halign` attribute (left, center, right, or justify) controls how the text is justfied. Cells inherit `halign` from `<row>`, `<col>`, and `<tabular>` in that order of preference. In a non-paragraph cell where `halign='justify'`, the horizontal alignment will match the behavior of `halign='left'`.

Table 13.13. Time Units

Unit	Stands For	Definition	Roughly
s	second	the duration of 9192631770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the cesium-133 atom an extraneous paragraph just to demonstrate the inter-paragraph formatting.	the time it takes you to say the phrase "differential calculus"
min	minute	exactly 60 seconds	how long it takes to microwave a full dinner plate from the refrigerator
h	hour	exactly 3600 seconds; exaclty 60 minutes	the length of one episode of a premium cable television show

Table cells can have multiline content using `<line>` elements. This is not the same thing as a paragraph cell—line breaking will happen precisely where the author tells it to. A `<line>` will not break, even on a narrow screen. If a cell uses a `<line>`, it must only use a sequence of `<line>`s and no other content. As with paragraph cells, you can use a `valign` attribute for the row.

Table 13.14. Dr. Seuss lines

		Look at me! Look at me!
One Fish		Look at me NOW!
Two Fish	I am the Lorax.	It is fun to have fun.
Red Fish	I speak for the trees.	But you have
Blue Fish	Self-referential: Table 13.14	to know how.

This is a table torture test with many combinations of `halign`, `valign`, `colspan`, `<p>` children, and `<line>` children.

Table 13.15. Table Torture Test

							Cell too wide	
Lf md	Lef mid par cel	Rt md	Rig mid par cel	Cn md	Cen mid par cel	Js md	Jus mid par cel jus mid par cel	
Colspan=2 lef mid with lines		Colspan=3 rig mid			Lines Between Par	Lines Between No Par	Par in row with lines	
L t	Lef top par cel	R t	Rig top par cel	C t	Cen top par cel	J t	Jus top par cel jus top par cel	
L b	Lef bot par cel	R b	Rig bot par cel	C b	Cen bot par cel	J b	Jus bot par cel jus bot par cel	
Colspan=3 lef bot			Colspan=2 rig bot with lines		Lines Under Par	Lines Under No Par	Par in row with lines	

Long division of polynomials, described with vertical and horizontal borders on individual entries of a `<tabular>`. The division lines are slightly thicker than the subtraction lines. This is an ill-advised torture test of the `<tabular>` machinery—the typical abuse of tables for horizontal and vertical layout—so do not take it as a model, but do notice what the machinery can do. It is also a *bare* `<tabular>`, sitting between paragraphs with no wrapper at all, and so with no number and no caption. A `<tabular>` is either bare, like this one, or wrapped in a `<table>` when it deserves a number and a title.

$$
\begin{array}{rr|rrrrr}
 & & & & x & - & 5 \\
\cline{3-7}
x & + 2 & x^2 & - & 3x & - & 8 \\
 & & x^2 & + & 2x & & \\
\cline{3-5}
 & & & & -5x & - & 8 \\
 & & & & -5x & - & 10 \\
\cline{5-7}
 & & & & & & 2
\end{array}
$$

And now a `<sidebyside>` with a `<table>` and a `<tabular>` to check that width is scaled appropriately. See Section 26 to learn about `<sidebyside>`s.

A1.S1	All legislative Powers herein granted shall be vested in a Congress of the United States, which shall consist of a Senate and House of Representatives. Should be 50% of 45% except perhaps on small screens.	A1.S2.C1	The House of Representatives shall be composed of Members chosen every second Year by the People of the several States, and the Electors in each State shall have the Qualifications requisite for Electors of the most numerous Branch of the State Legislature. Should be 50% of 55% except perhaps on small screens.

Figure 13.16. Some text from the US Constitution

Tables are formed in LaTeX output with copious use of the `\multicolumn` macro to override more global alignment settings, and to spread the content of one cell across several columns. However, sometimes LaTeX's special characters have behaved badly in this situation. So the table below, two items per row, is just designed for LaTeX testing. But of course, it should still render fine in other formats. The three test cases are from 9.8, but without 50 alphabetic characters and 8 digits, which should not be problems in this context. That section says more about making a comprehensive, yet legal, test URL. In order to test the use of a percent sign (`%`) in a URL, we follow it by two hex digits, specifically, `58`, which is a way to represent the character `X` in a URL. The first column's entries are *forced* to be wrapped in a `\multicolumn` by specifying their horizontal alignment. The second column's entries *will not* be wrapped in a `\multicolumn`. So the two columns will look identical, other than the first having a left alignment, and the second has the default center alignment. (This table is known to render poorly in a Jupyter notebook. The cause is four dollar signs present in rows 1 and 3, and is explained in Subsection 9.12.)

Table 13.17. Problematic Table Cells for LaTeX

1	`09az%-._~:/%5B%5D@!$&'()*+,;=#?`	`09az%-._~:/%5B%5D@!$&'()*+,;=#?`
2	e.com/09az%58-._~:/%5B%5D@!$&'()*+,;=#?	e.com/09az%58-._~:/%5B%5D@!$&'()*+,;=#?
3	`example.com/09az%58-._~:/%5B%5D@!$&'()*+,;=#?`	`example.com/09az%58-._~:/%5B%5D@!$&'()*+,;=#?`

Now, the same table repeatedly, but with different headers. No care has been taken with alignment or rules, which could improve how these look.

Table 13.18. No Headers

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Table 13.19. One Row Header

State	**Population**	**Area (sq. mi.)**	**Statehood (Year)**
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Table 13.20. One Row Header, Multiline

State	**Population**	**Area (sq. mi.)**	**Statehood (Year)**
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Table 13.21. Two Row Headers

State	**Population**	**Area (sq. mi.)**	**Statehood (Year)**
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Table 13.22. One Vertical Row Header

State	**Population**	**Area (sq. mi.)**	**Statehood (Year)**
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Table 13.23. One Vertical Row Header, Multiline

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Table 13.24. Two Vertical Row Headers

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Table 13.25. Test of vertical header orientation and padding with borders

	0 ***(MTH111)***	**1** ***(MTH112)***	**2** ***(CS160)***	**3** ***(CS161)***	**4** ***(CS162)***	**5** ***(CS205)***
0 ***(MTH111)***	0	1	0	1	0	0

Table 13.26. One Row Header, with Rules

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Table 13.27. One Row Header, Multiline, with Rules

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

The next table has a progression of thicker rules in the header, plus a progression of thicker rules across the columns. For testing, not for aesthetics.

Table 13.28. Two Row Header, Many Rules

State	Population	Area	Statehood
		(sq. mi.)	(Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

We now finish this section with some *long* tables. Ones that will not fit on a single printed page. So this is only of interest when producing this sample article as a PDF. First a "naked" tabular, which should force a new page to start, and then still overrun the end of the page.

Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink

Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink

Now the same `<tabular>`, but within a `<table>`. Behavior should be similar, a new page and then it overruns the bottom of the page.

Table 13.29. A Lot of Colors

Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow

Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink

When you wish to allow a `<tabular>` to split itself at a page break, you can add the attribute `@break` with the value `yes`. Certainly this will control a `<table>` or `<tabular>` which is longer than a page, but will also allow a shorter one to break. This might useful in a draft stage before undertaking page-fitting.

Here is the long `<tabular>` again, but with the `@break` attribute set to `yes`.

Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink

Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink

Here is the long `<table>` again, but with the `@break` attribute set to `yes` on the `@tabular`.

Table 13.30. A Lot of Colors, Breaking

Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow

Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink
Red	Green	Yellow
Blue	White	Pink

This device is not ideal, as some features of tables are not behaving as expected. More precisely, we switch from the standard LaTeX `tabular` environment to the `longtable` environment from the package of the same name when table-breaking is requested. So there may be some undesirable interaction with other packages. For one, full-width horizontal rules seem to become as wide as the page (rather than as wide as the tabular). The following table is Table 13.4 repeated but as a breakable `<tabular>`. The `longtable` package documentation suggests it accomodates the `array` package, but it also seems to make a variety of redefinitions. Furthermore, a panel of a side-by-side cannot be a breakable tabular, or a LaTeX compilation error occurs.

Table 13.31. Horizontal Rules Example

1234567890	1234567890	1234567890	1234567890
First	Second	Third	Fourth
A	B	C	D
1	2	3	4
1	2	3	4

Here is a consecutive pair of "bare" `<tabular>` to test vertical space between them.

t	2/3	2/303	2/30003	2/3000003
$g(t)$	-1	-1	-1	-1

t	2/5	2/505	2/50005	2/5000005
$g(t)$	1	1	1	1

The `longtable` package allows for headers and footers indicating continued tables. A possible enhancement is to support this feature in the case of a long table.

14 Interactive Elements, Authored in Javascript

When outputting Web page versions, it is possible to embed a variety of dynamic interactive elements. In a LaTeX/PDF version, these will necessarily need to be replaced by some static substitute, such as a screenshot. See Section 3 for the specifics of embedding instances of the Sage Cell Server, which is more elaborate, and not entirely similar.

Interactives in this section are those for which you provide code you have authored. Generally, the libraries involved to support this have open licenses, though the player for GeoGebra may be an exception. Creating these assumes some familiarity with HTML and Javascript. See Section 15 for more interactives that are perhaps simpler to create or use.

(2018-06-22) Almost everything in this section is under active development and not stable yet. Feel free to experiment and make suggestions and requests. This page takes a while to completely load, so be patient.

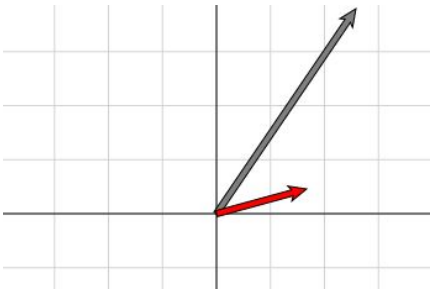
14.1 HTML5 Canvas

HTML5 introduced the `<canvas>` element, which can be thought of a blank slate, a place to draw or write on. So PreTeXt has the `<slate>` element for a similar purpose. Generally, but not exclusively, HTML5 writes on a `<canvas>` using the Javascript language. We demonstrate this approach to interactive diagrams in this subsection.

The following examples are from David Austin's excellent *Understanding Linear Algebra* textbook, which can be found at

`understandinglinearalgebra.org`

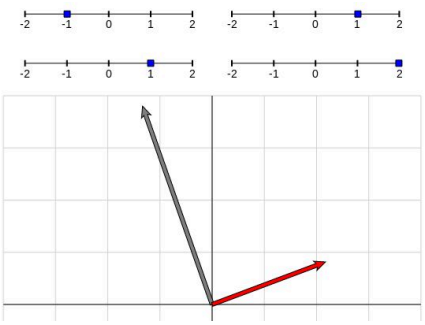
David's contribution of examples, and assistance designing the PreTeXt elements is greatly appreciated. Alright, let's learn some linear algebra. Yes, there are some learning opportunities in this subsection.



Standalone
In Context
Embed

Figure 14.1. A simple eigenvector demonstration

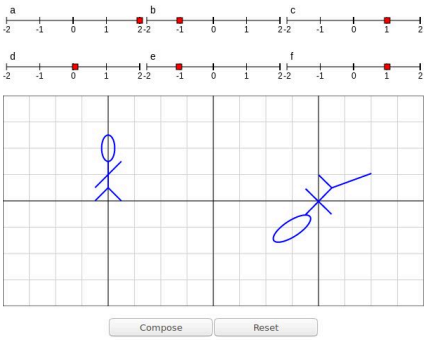
Checkpoint 14.2. The interactive in Figure 14.1 shows a vector $\vec{x}$ in red, and the matrix-vector product $A\vec{x}$ in grey, for a particular 2×2 matrix A. The four entries of the matrix A are coded into the interactive. Can you deduce A simply by using the interactive? Which theorem is the key?



Standalone
In Context
Embed

Figure 14.3. Eigenvector demonstration

The next example has ten `<slate>` elements communicating with each other, and arranged with the layout features of a `<sidebyside>` (see Section 26).



Standalone
In Context
Embed

Figure 14.4. Affine Transformations

Warning 14.5. If your `<interactive>` employs a `<slate>` with a `@surface` attribute whose value is `html`, then you are advised to augment each top-level (HTML) element within the `<slate>` with the attribute:

```
xmlns="http://www.w3.org/1999/xhtml"
```

This will identify *all* of the elements within the `<slate>` as HTML elements and not as PreTeXt elements. The danger is that elements with the same name in both languages, such as `<li>` and `<table>`, will be mis-identified. This could be harmless, but could also create chaos, such as disrupting numbering of PreTeXt elements.

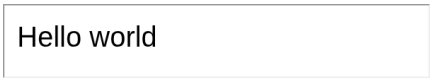
See the source code of this document for examples: Figure 14.4 Figure 14.12, Figure 14.18.

Note that the HTML that is output can vary slightly from your source in small, harmless ways, such as empty (self-closing) elements being output with both an opening and a closing tag. Please report any significant discrepancies. Soon this requirement will be enforced in the code.

It is also possible to add `<script>` elements within an interactive that contain properly escaped JS code. These elements will be placed at the end of the document created to hold the interactive content and can interactive with the other elements within the interactive but can not directly interact with the surrounding page.

Authors are strongly discouraged from trying to incorporate complex code in the form of a `<script>`, but it can be a useful tool to call more complex code that is linked via `@source` on the `<interactive>`.

This example uses a `<script>` to draw "Hello World" to a `<slate>`





Standalone
In Context
Embed

Figure 14.6. A simple embedded script example

This next interactive uses ECMAScript module based code. That will not function unless the `<script>`'s `@type` is set to `module` in the HTML output. So let's set it here and pass it through. It also generates its own HTML elements, so we hopefully don't get a default empty div.





Standalone
In Context
Embed

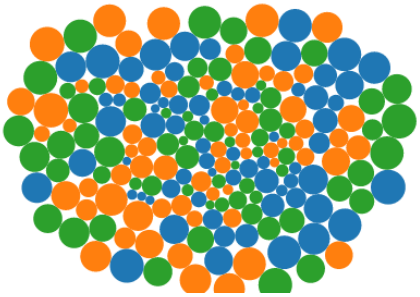
Figure 14.7. A simple JS module test

14.2 D3.js

D3 is a Javascript library for "Data-Driven Documents", which might greatly enhance some data you wish to display. In short, it uses the animation capabilities of SVG. Available examples seem sensitive to the version of the library, so we have examples using different versions. Use the `@version` attribute on `<interactive>` to specify the version number. The default is `5`.

The first example uses the force layout and collision detection from Version 3. (The necessary commands are very different in Version 4.) Pretend you are a working shepherding dog. Can you separate, and catch, one of the herd?

This is adapted from a block by Mike Bostock with a GPL license. A similar demonstration, only using an HTML5 canvas is at `bl.ocks.org/mbostock/3231307`.



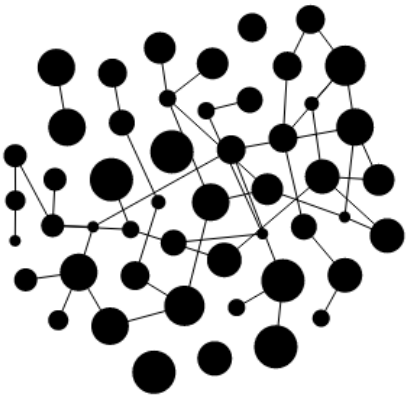
Standalone
In Context
Embed

Figure 14.8. Force layout and collision detection

Similar, but different, this demonstration of a graph layout uses Version 4 of the library. Technical notes:

- We have changed the size of the nodes, and their number, to fit in a smaller space.
- The Javascript script uses introspection to size itself, which would be a good general practice. (Though it does not resize itself on window resize, only initial load.) The `<interactive>` also has the `@resize-behavior` attribute set to `responsive`.

This is adapted from a block by shimizu with a GPL license.



Standalone
In Context
Embed

Figure 14.9. Graph Layout

Checkpoint 14.10 Graph Planarity. Can you move the vertices to new locations such that the resulting graph is planar? (In other words, no edges cross?)

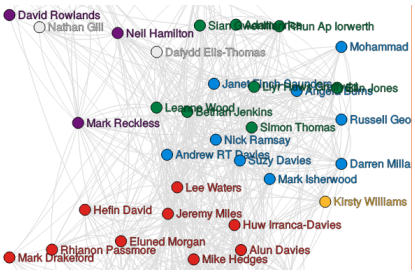
Finally an example that actually uses some data. Here is the description from the original block by Martin Chorley with an MIT License.

> This visualisation uses a D3 force simulation to show the Twitter relationships between the Assembly Members in the Welsh Assembly in terms of the number of times each assembly member has mentioned another assembly member in a tweet.
>
> Twitter relationships were mined on 22/03/2017, and are representative of the conversational relationships on that date. Links between AMs represent a conversational relationship: one AM has mentioned the other. Party colour indicates the direction of the mention.
>
> Hover over the nodes to fade out non-connected nodes.

Rather than using intermediate nodes to create curved links (as in Mike Bostock's block), this adds curves by adding a calculated control point for each edge.

Technical notes:

- Once the nodes organize themselves (automatically in the beginning), they cannot be moved.
- We have adjusted the margins in an attempt to keep names visible on the right side, but without giving up too much space.
- We have adjust the repelling force, and the collision buffer, to better fit the available space.
- This example required its own CSS, which we have included as part of the `<interactive>`.
- The `@aspect` ratio of the `<interactive>` is set to slightly taller than that of the `<slate>` it contains. This accounts for the fact that the contents generated by the js/css in the slate occupy slightly too much vertical real estate.
- The data collected from the Twitter analysis is contained in a JSON file, `mention_network.json`, and where the script loads that file, it has a hardcoded path. So this example is a bit brittle, should that file move.



Standalone
In Context
Embed

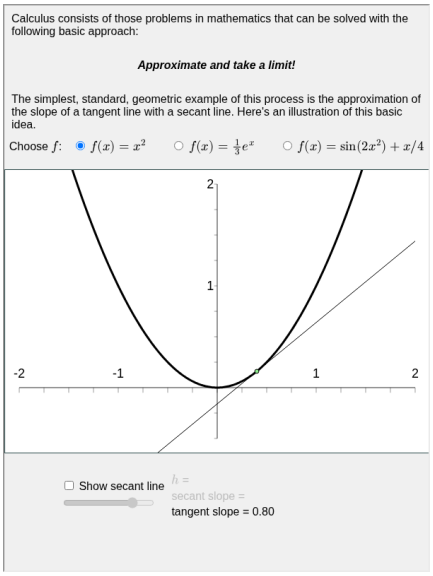
Figure 14.11. Tweet mentions within the Welsh Assembly

14.3 SVG

Entirely similar to using an HTML5 `canvas` element (14.1), it is possible to control an SVG element with Javascript. This example is from Mark McClure.

Look carefully at the source and rendering of this example as HTML. The functions to choose from via radio buttons, and the change in x, denoted later as h, are being rendered as mathematics by the Javascript MathJax library. However, this cannot be accomplished with simply `$...$` nor by simply `\(...\)`, but instead by using the `\(...\)` and then also wrapping that in a `<span>` element with a `@class` attribute set to `process-math`. (The latter is how we instruct MathJax as to which parts of a page to process, or not). So to have MathJax render a nice x^2 in this context (math inside HTML inside a `<slate>` inside an `<interactive>`) would be accomplished with

```
<span class="process-math">x^2</span>
```



Standalone
In Context
Embed

Figure 14.12. Tangent and secant slopes

Checkpoint 14.13 Changing Secant Lines. When discussing the derivative as a limit, we think of the point of tangency as being fixed (the green point in 14.12) and the "other" point defining the secant line as changing (the red point in 14.12). Switch it up! Fix a large value of h (positive or negative) and then change the point of tangency (the green point). Discuss what you observe.

14.4 JSXGraph

JSXGraph is a "cross-browser JavaScript library for interactive geometry, function plotting, charting, and data visualization in the web browser." Now a `<slate>` will be what JSXGraph calls a **board**. Again, you use Javascript to write onto a `<slate>`, but have some powerful shortcuts available from the JSXGraph library. For this reason, PreTeXt calls JSXGraph a "language", similar in may respects to how Sage is a language, but is really a Python library. So realize that the syntax for using JSXGraph is that of Javascript.

Place Javascript inside a file that is specified with the `@source` attribute of the `<interactive>` element. Then just be certain that `@xml:id` of the `<interactive>` element is passed as the HTML `id` in an (early) call to JSXGraph's `initBoard()` method.

The following example was contributed by Rick Roesler. The `<figure>` is comprised of four `<stack>` elements within a `<sidebyside>`. By varying the time in the top box, the reader can observe the displacement, velocity, and acceleration of a ball thrown upward with an initial velocity of 30 m/s.

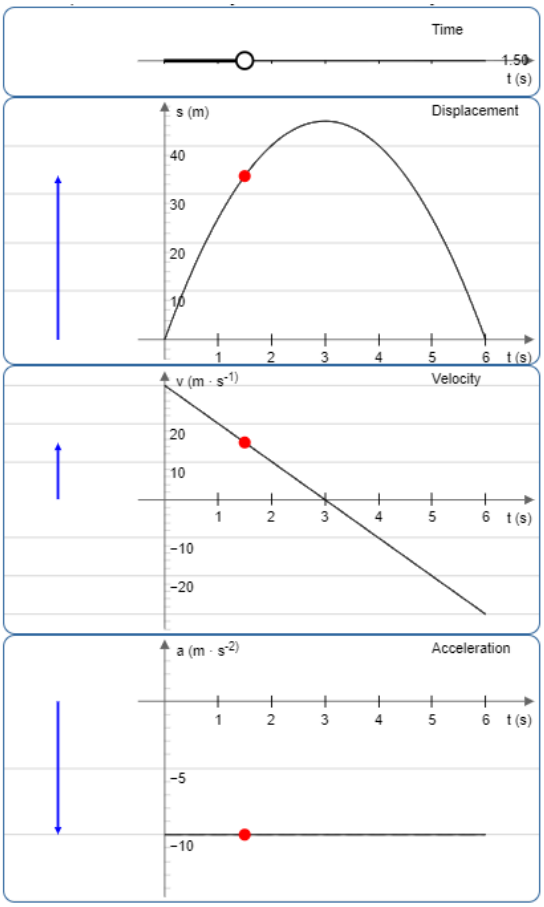
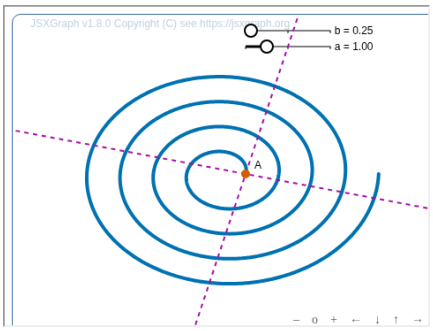


Figure 14.14. 1-Dimensional Kinematics

The plot below is the curve $r = a + b\theta$ in polar coordinates, for $0 \leq \theta \leq 8\pi$. It may be manipulated with the sliders to control the shape of the curve. Point A is contrained to the curve, but may be dragged to a new location. At A the tangent line and normal line are plotted as dashed red lines. Use the controls in the lower left to adjust the viewing window. This Archimedean Spiral is taken from the JSXGraph example wiki. The code could be written in 7 lines. Width is 80% and aspect ratio is 4:3.

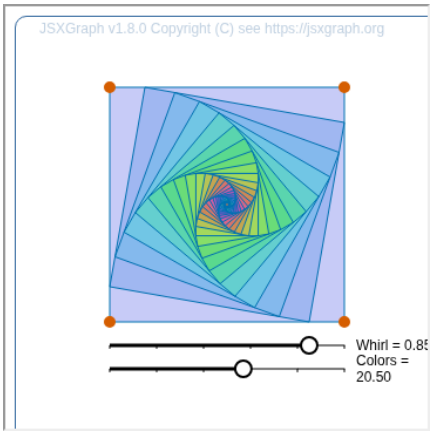


Standalone
In Context
Embed

Figure 14.15. The Archimedian Spiral $r = a + b\theta$, $0 \leq \theta \leq 8\pi$

Here is a more elaborate example, from the JSXGraph Showcase, titled Infinity.

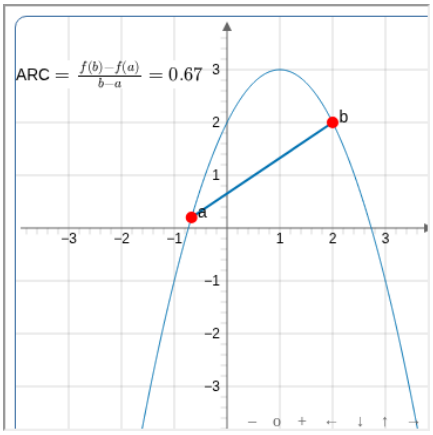
There are two active sliders to control the shape and shading of the graphic, and hovering the mouse near one of the edges will highlight the entirety of one of the 30 quadrangles. Finally, each of the four red corners may be dragged to a new location. Code is 47 lines. Width is 60% and aspect ratio is the default, 1:1, i.e. a square.



Standalone
In Context
Embed

Figure 14.16. Infinity, from the JSXGraph Showcase

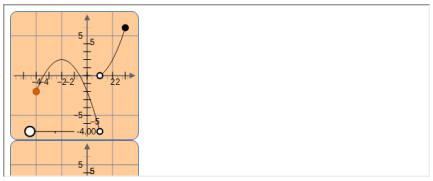
A test where the JSXGraph code indicates it needs MathJax to render.



Standalone
In Context
Embed

Figure 14.17. An interactive whose JSXGraph code uses MathJax to render mathematics.

Finally, a piecewise function you can control, with traces of the domain values and range values in two other JSXGraph boards. Boards and HTML buttons have been laid out using the `sidebyside` layout element.



Standalone
In Context
Embed

Figure 14.18. Piecewise Function

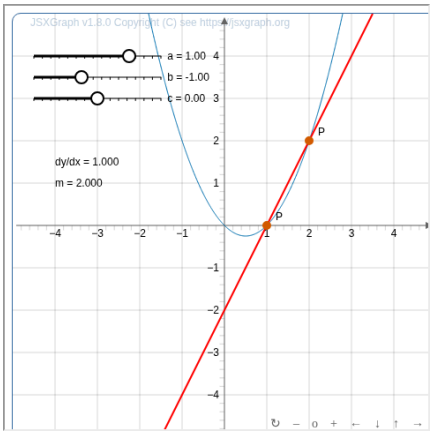
Generally, we load an interactive into an HTML `iframe` to sandbox (isolate) it from other interactives. We does this for your own protection. So, for example, one interactive cannot talk to another. If two `<slate>` need to communicate, then they are related, and should be placed into a single `<interactive>`, allowed to layout themselves, or grouped within a `<sidebyside>` allowing finer control. Even if we have this under control, you might still enjoy reading Your JS is a Mess at `mikecavaliere.com/your-js-is-a-mess-javascript-namespacing/`.

14.5 JessieCode

JessieCode is a scripting language for JSXGraph. It provides the core geometric and graphing features of JSXGraph without accessing the underlying JavaScript. In order to use JessieCode, you simply create the HTML `<div>` element as you would for any other JSXGraph interactive plot and then provide the JessieCode script, which focuses on the geometric elements.

Because JessieCode is provided by JSXGraph, the interactive platform is `jsxgraph`. The `<slate>`, however, uses `@surface` with value `jessiecode`. The script can be embedded directly in your code. As usual, you would need to remember to escape the special characters. JessieCode uses `<` and `>` for inequalities as well as for declaring objects used to style geometric elements, and `&&` is the boolean AND operator. Alternatively, you can provide the file as a separate resource, providing the URL with a `@source` attribute. Attributes defining the JSXBoard at the time it is created should be included as attributes of the `<slate>`, including `@boundingbox`, `@axis`, and `@grid`.

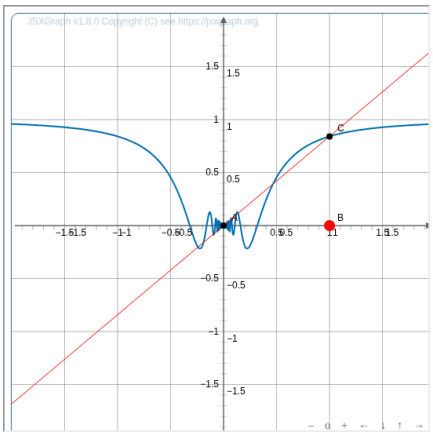
For this first example, the JessieCode was included directly in the XML source as the contents of the associated slate.



Standalone
In Context
Embed

Figure 14.19. Finding the Tangent Line of a Quadratic Function

For this second example, the JessieCode was included through an associated script file, loaded by the browser.



Standalone
In Context
Embed

Figure 14.20. Graph of a function that is continuous but not differentiable at $x = 0$ because the slope of the secant line has no limit.

14.6 Sage Interacts

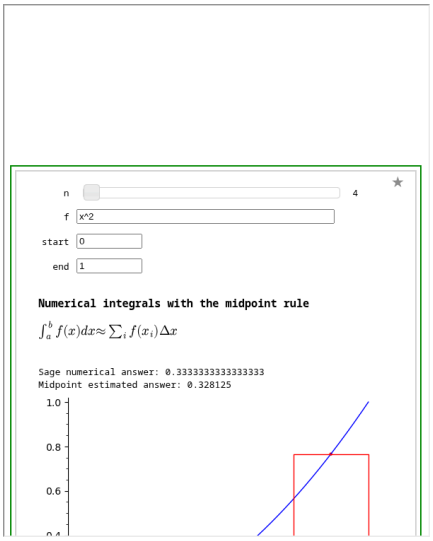
Sage, and the Sage Cell Server, support interactive demonstrations, called **interacts**.

- The interactive elements are nearly trivial to construct.
- An interact is simply a single Python function (acted on by a decorator).
- You have the full mathematical power of Sage at your disposal, so can do some very powerful computations with high precision (or exactly).
- The interface is not as polished as what you can achieve with Javascript libraries.
- Graphics refresh with a round-trip to the server, so are not nearly as fluid as with other tools.

Note that each interact is insulated from the others, unlike our other employment of the Sage Cell Server.

This example is by Marshall Hampton, taken from the Sage interact wiki, specifically Numerical integrals with the midpoint rule.

Also notice that when viewed in dark mode, this sample respects the rendering intent. That is due to `@dark-mode-enabled` applied to it.



Standalone
In Context
Embed

Figure 14.21. Numerical integrals using the midpoint rule

14.7 Geogebra

To embed a GeoGebra applet as-is from GeoGebra's Classroom Resources site (by material ID), see Subsection 15.1. To design your own applet (either from scratch, or modifying something that already exists in one of those three forms) you may use one of Geogebra's "Apps" to embed the material in your document. (But note, use of the App comes with licensing restrictions.) PreTeXt will handle most of the technical details for you.

Do one of the following:

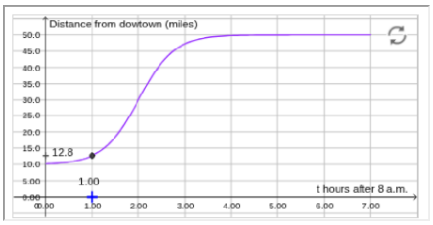
1. Identify a material ID from GeoGebra's Classroom Resources site. You might even make the material yourself on that site.
2. Obtain a `.ggb` file from GeoGebra. You might construct something on a desktop installation of GeoGebra and save it. If you have a base64-encoded string for a GeoGebra applet, but you don't have a `.ggb` file, you can decode the string and save the result. For example, at `https://www.opinionatedgeek.com/codecs/base64decoder`.
3. Obtain a base64 encoded string for a GeoGebra applet. You might first open a `.ggb` file in a desktop installation of GeoGebra, and push ctrl-shift-B (command-shift-B on a Mac) and then the string will be in your clipboard.
4. None of the above, with the intention to make an applet from scratch.

Then mimic the examples that follow, using GeoGebra API commands documented at Geogebra API Manual, but do not include the `ggbApplet.` or `applet.` used in examples to prefix the functions—that part of the code will be provided automatically by PreTeXt.

There are some optional control elements that Geogebra provides, such as the presence of the toolbar and the reset button. These can be controlled by adding the following additional attributes to the `slate` and are automatically included in the initialization for the Geogebra applet.

- `toolbar="yes"`: add the Geogebra toolbar above the material
- `algebra-input="yes"`: add an entry box below the material to add graphical objects using algebra formulas or Geogebra graphical commands
- `reset-icon="yes"`: enable the reset icon
- `shift-drag-zoom="yes"`: enable ability to drag and zoom the viewing context
- `zoom-buttons="yes"`: enable buttons that control zoom

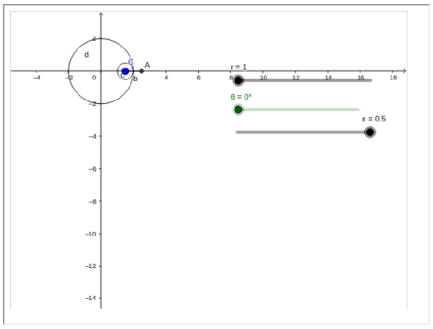
Jack Green created an applet on the Classroom Resources site with ID `D4s2v4ft`, which you may view at `www.geogebra.org/m/D4s2v4ft`. Suppose you would like to use this in your project, but change something about it. We will change something trivial, making the y-axis ticks be separated 5 apart instead of 10 apart. We also decide we want a different aspect ratio and overall width. One gotcha: the original applet is loaded and then PreTeXt uses `@width` and `@aspect` attributes to resize the viewing window using the top left corner as an anchor. This does not rescale axes and that may leave you with important elements missing from the viewing window. So here we reset the viewing window to return to values that are in the original. Lastly, we disable zooming, which is not helpful for this applet. To do each of these things, we rely on the GeoGebra API manual at Geogebra API Manual. It is important to use one command per line.



Standalone
In Context
Embed

Figure 14.22. GeoGebra: modified material ID

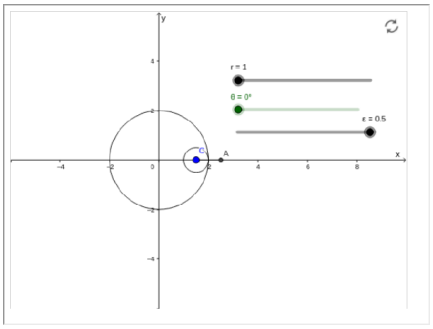
The same can be done with a `.ggb` file. Here we use two provided by David Rosoff, and one provided by Tevian Dray. The path to the file needs to be relative. First, David's original.



Standalone
In Context
Embed

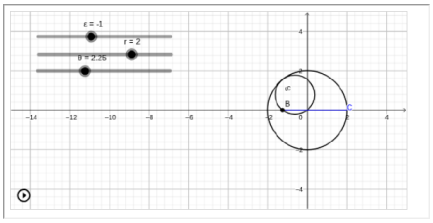
Figure 14.23. GeoGebra: `.ggb` file

Now we modify David's applet in a few ways.



Standalone
In Context
Embed

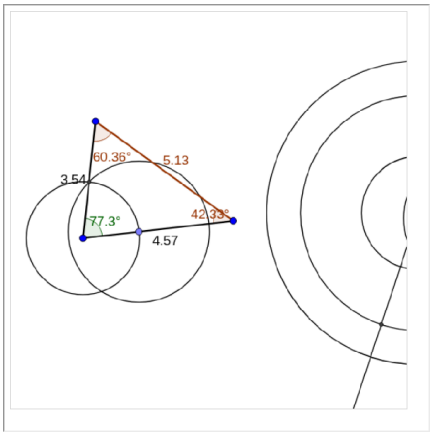
Figure 14.24. GeoGebra: modified from `.ggb` file



Standalone
In Context
Embed

Figure 14.25. GeoGebra: modified from `.ggb` file

In this one provided by Tevian Dray, we make no modifications (except for those imposed by the scaling). You will need to zoom out a bit, and then pan over some, to see all the pieces.



Standalone
In Context
Embed

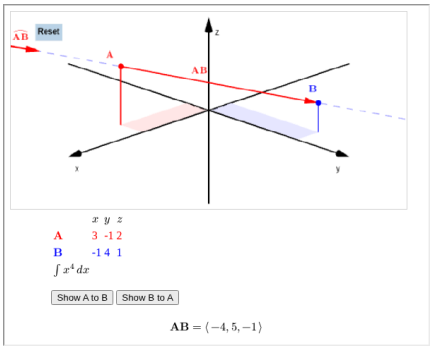
Figure 14.26. GeoGebra: a constructive "proof" that SAS congruence holds in Euclidean geometry (from Tevian Dray)

You could also use a base64-encoded string of the `.ggb` file. You might come across such a string somewhere, or you might generate one by opening a `.ggb` file in a desktop installation of GeoGebra, and pushing ctrl-shift-B (command-shift-B on a Mac) to get the string in your clipboard. If you do this, you could use a `@base64` attribute in place of the `@source` attribute in the previous example. We don't do that here because such a string is generally over 5000 characters long and we are keeping the sample article source a bit cleaner.

The next example shows how you can communicate between a GeoGebra applet and a `<slate>` contained in the interactive. The process is mediated by javascript code specified in the `@source` attribute of the interactive. Event listeners in the code update the HTML when the diagram changes or vice-versa. MathJax is also notified when it needs to update math.

Note that this example has a `<slate>` whose `@surface` is `pretext` *and* a `<slate>` whose `@surface` is `html`, the latter requiring a namespace declaration. The former produces HTML according to the PreTeXt templates, which is fairly predictable, but never guaranteed to always be identical over time. A PreTeXt slate uses familiar syntax, produces results styled consistently, but might break in the future. While an HTML slate is similar, the results will not be styled, but it does allow for a wider range of HTML elements (a `button` element here) and will not change over time.

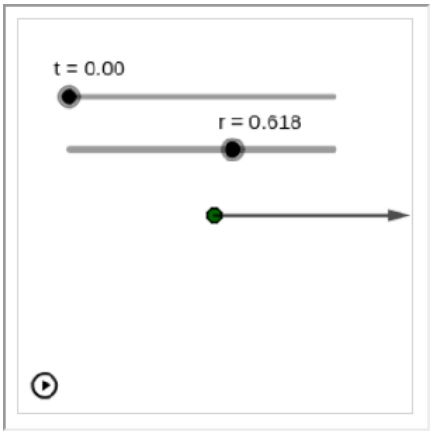
The indefinite integral in the last row of the table is a gratuitious test that aurhors' macros from `<docinfo>` are available to MathJax. Finally, a PreTeXt slate will only recognize `<p>`, `<tabular>`, `<sidebyside>`, and `<sbsgroup>` as children. Make a feature request if you have a good case for more.



Standalone
In Context
Embed

Figure 14.27. Geogebra/PreTeXt Communications

Lastly, you may just wish to build something from scratch using GeoGebra API. Note that for accessibilty reasons, some objects are rendered unselectable with the `setFixed` command. Perhaps this should have been done with the previous examples, but that is more difficult when you do not know all of their names. Note that the GeoGebra scripting command `setAttribute` also changes the webpage's focus, so it is better to set the perspective using an attribute of the slate.

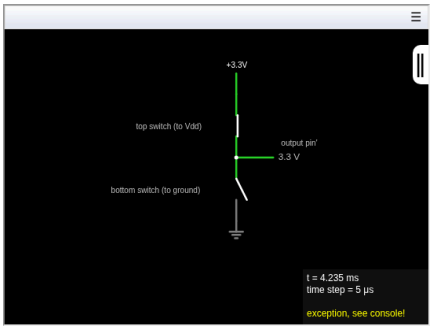


Standalone
In Context
Embed

Figure 14.28. GeoGebra: from scratch

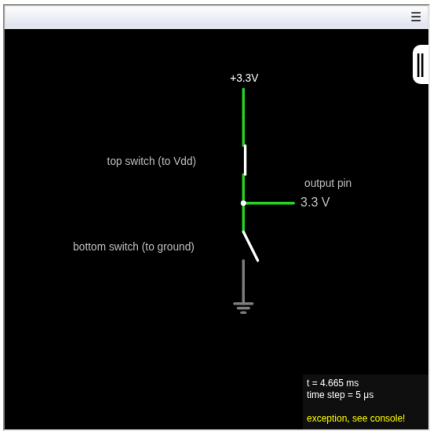
14.8 CircuitJS

CircuitJS is an electronic circuit simulator. A circuit can be described by a language, which PreTeXt will interpret and submit for rendering. The next two examples are identical, but provided in slightly different ways, see the PreTeXt source for more. Preview images for PDF will be added later.



Standalone
In Context

Figure 14.29. CircuitJS Example (source via an encoded attribute)



Standalone
In Context

Figure 14.30. CircuitJS Example (source authored directly)

14.9 IFrames from Files

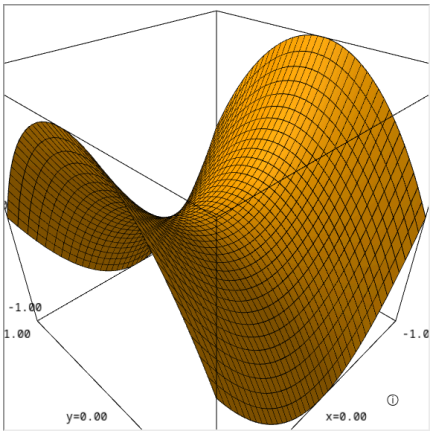
An `iframe` is an HTML element that allows embedding of a complete web page within another one. Here we use this device to provide interactive 3D diagrams built with other tools.

We begin with a Jupyter notebook hosted on CoCalc.com. News of success on other hosts for Jupyter notebook servers will allow us to expand this description. We use a Sage kernel and create a 3D surface suggested by Juan Carlos Bustamante:

```
var('x,y')
plot3d(x^2 - y^2, (x,-1,1), (y,-1,1), color="orange", mesh=true)
```

News of other computational engines that produce similar graphics will also allow us to further expand this description. Note that for the case of Sage 3D plots, support for the `<sageplot>` element makes this even easier. For example, see 10.34.

A button in the lower right allows for several options, one is `Save as HTML`, which will produce a complete self-contained web page we can recycle. We save this file with our other externally-created images, in a directory that we choose to name `iframe`—you can use another name. Then we make an `<interactive>` with an `@iframe` attribute that has the filename, starting from `iframe/` (in other words, do not include the name of your managed directory of external images).



Standalone
In Context

Figure 14.31. Sage+Jupyter `iframe`

Note that the downloaded file has links to specific versions of the `three.js` library, which are beyond our control, and beyond your control. So there is a future where these images may need updating. You could put your source code into a (large) comment with your project's source for safe-keeping in the future. See Subsection 15.4 for the server version.

14.10 threejs

Once upon a time there was an example here using the `threejs` 3D Javascript library. It was one of the project's examples, licensed with an MIT License, with minimal modifications.

But it would seem to have become a bit more complicated to embed and our example was not rendering. As of 2022-08-08, we have removed it. Of course, you can find it in the git repository, perhaps searching on the date string just

mentioned. It woulld be interesting to see if our <interactive> framework could still support the changes.

The following two examples are meant to be instructive (*only*). The end result is accomplished in a much more straightforward way be the method in Subsection 14.9. We illustrate a way to get a three.js image out of an HTML page as a Javascript file and render it on a PreTeXt <slate>. We follow the second method in a blog post from the *n*-Category Cafe.

1. Open a Jupyter Notebook that utilizes a Sage kernel. This can be done easily at CoCalc (and for free initially).
2. Sketch a surface using Sage code. We recycle the suggestion from Juan Carlos Bustamante in Subsection 14.9:

```
var('x,y')
plot3d(x^2 - y^2, (x,-1,1), (y,-1,1), color="orange", mesh=true)
```

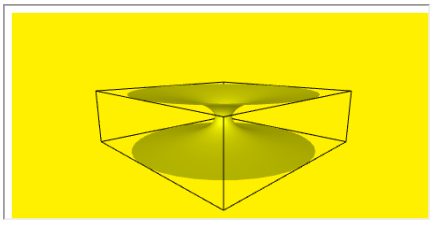
3. Look for a button (in the lower-right) which will provide a menu option Save as HTML. Save the resulting HTML file, and open a *copy* in a text editor.
4. You are now looking for two HTML script elements. One will tell you just which version of three.js is being used, vis a @src attribute. For the second example below we located

```
https://cdn.jsdelivr.net/gh/sagemath/threejs-sage@r122/build/three.min.js
```

   The @r122 will likely be a version number, which is a good thing for the longevity of your work. This will get used in the @source attribute of your <interactive>, which will have @platform set to javascript.

   The second script element is likely huge and has many generic functions defined it. There may be a huge variable full of data points computed by Sage. Copy the *contents* of this script element into a new Javascript file (so use a .js extension). Do not edit in any way until you read further. Once adjusted, this file too gets specified in the @source attribute of your <interactive>.
5. Your <interactive> needs a <slate> element for the graphic to draw on, and you will need to give it a proper @xml:id attribute, plus teh @surface will be set to div. Then you need to edit the Javascript file to connect the graphic with the slate, via IDs in your PreTeXt source and on teh HTML div created by the slate. Look at the provided examples to see how. Do not make any other edits to this file, even if tempted.
6. Study the two examples below, and mimic how they were constructed.

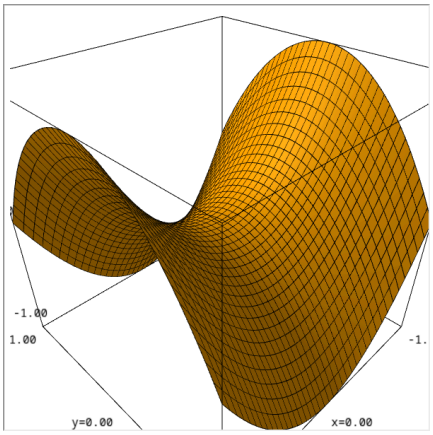
First, the example given in the blog post referenced above.



Standalone
In Context
Embed

Figure 14.32. threejs catenoid surface, from *n*-Category Cafe

Second, the example from JC Bustamante.



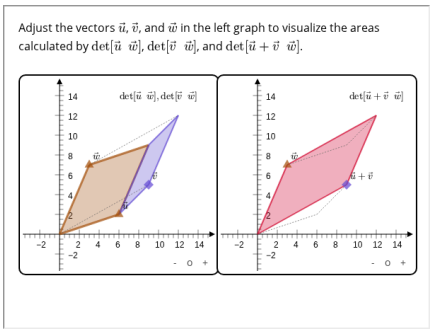
Standalone
In Context
Embed

Figure 14.33. `threejs` saddle by Sage

14.11 DoenetML

DoenetML is a markup language inspired by PreTeXt for semantically describing interactive mathematics applets for the web. Use `interactive[@platform="doenetml"]` with a `slate[@surface="doenetml"]` to include DoenetML content within your document.

Since DoenetML is similar to, but not completely compatible with, XML, it cannot be authored directly within a PreTeXt document without painstakingly escaping every `<` as `<` and every `&` as `&`. However, a convenient workflow to avoid this is to author your DoenetML in a separate file like `example.doenetml`, then include this file within your PreTeXt document using `<xi:include parse="text" href="path/to/example.doenetml"/>`. Altenatively, you can enclose your DoenetML in a `CDATA`, as we do below.



Standalone
In Context
Embed

Figure 14.34. DoenetML example

15 Interactive Elements, Server & Layouts

When outputting Web page versions, it is possible to embed a variety of dynamic interactive elements. In a LaTeX/PDF version, these will necessarily need to be replaced by some static substitute, such as a screenshot. See Section 3 for the specifics of embedding instances of the Sage Cell Server, which is more elaborate, and not entirely similar.

Interactives in this section have code that lives on a server somewhere (in the "cloud"). So maybe you uploaded an interactive demonstration, or maybe

somebody else did. In this sense, these are easier to create or use. But pay attention, the code may come with restrictive licenses, even if you are the author originally. See Section 14 for more interactives that can be free as in "freedom" or *liberté*. CalcPlot3D is the notable exception here.

(2018-06-22) Almost everything in this section is under active development and not stable yet. Feel free to experiment and make suggestions and requests. This page takes a while to completely load, so be patient.

15.1 GeoGebra

A Geogebra **material** is something you might use in a class. It could also be called an interactive demonstration. Go browsing at `www.geogebra.org/materials/` and find something appropriate for your project. Or make an account and create your own.

Once you find a material that looks instructive, it will be at a URL such as

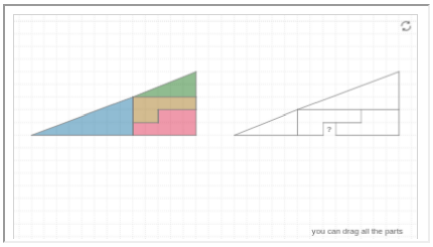
```
https://www.geogebra.org/m/KGn2d4Qd
```

and you want to pick off the identifier on the end, in this case `KGn2d4Qd`. Then author

```
<interactive geogebra="KGn2d4Qd" />
```

At this writing (2018-02-04) you will want to place this inside a `figure`, with a caption, as we do right now with material `KGn2d4Qd`.

The shape of the material will be fixed, so determine the dimensions (measure with an on-screen ruler or inspect the underlying definition). Use `@material-width` and `@material-height` with the actual material width and height (units are pixels). These are treated by Geogebra as internal coordinate dimensions and are separate from the physical layout dimensions that appear in your text. In the absence of these attributes, default values are 800 pixels for the width and a height based on the provied aspect ratio.



Standalone
In Context
Embed

Figure 15.1. Right Triangle Paradox

There are some optional control elements that Geogebra provides, such as the presence of the toolbar and the reset button. These can be controlled by adding the following additional attributes to the `interactive`.

- `toolbar="yes"`: add the Geogebra toolbar above the material
- `algebra-input="yes"`: add an entry box below the material to add graphical objects using algebra formulas or Geogebra graphical commands
- `reset-icon="yes"`: enable the reset icon
- `shift-drag-zoom="yes"`: enable ability to drag and zoom the viewing context
- `zoom-buttons="yes"`: enable buttons that control zoom

Note that materials hosted at `geogebra.org` have a non-standard, non-commercial license you must agree to before you can download them as source code. Perhaps you must forfeit your copyright when you upload a material? We have not investigated this thoroughly.

15.2 Desmos Graphs

Desmos provides interactive graphing applications. The following example was created by Ann Cary and made available via the "Share" function of Desmos. You can make your own Desmos graph, choose the "Share" icon, and the "Embed" option, to get a URL such as

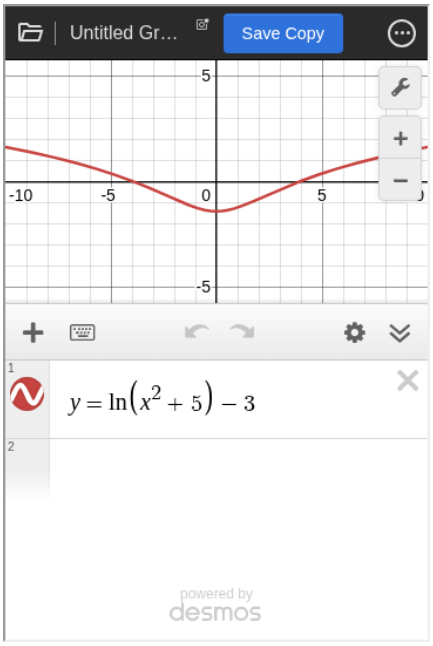
```
https://www.desmos.com/calculator/ttox1bvxku
```

You want to pick off the identifier on the end, in this case `ttox1bvxku`, then author

```
<interactive desmos="ttox1bvxku" width="60%" aspect="2:3" />
```

as we have done here.

The static image employed in the LaTeX version of this article was obtained by viewing the graph at the Desmos site (i.e., not the embedded version in the PreTeXt HTML version), and using the Share button to export a PNG image. In this case, we used a "Medium Rectangle" and "Thick" lines.



Standalone
In Context

Figure 15.2. Graph of $\ln(x^2 + 5) - 3$

Note that Desmos has extensive *Terms of Service* which include restrictions on commercial uses.

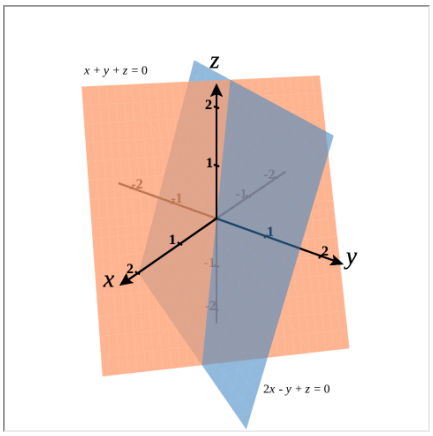
15.3 CalcPlot3D

CalcPlot3D is a Javascript application for creating, visualizing, and understanding plots of 3D surfaces. So it would be an ideal companion to a book on multivariate calculus, but should be useful in other courses of study.

To use it, start at the online app version. Create a plot and adjust the image to a viewpoint and scale if you like. Then, click the menu/hamburger icon in the upper-left and choose `File`. From here you can save a PNG image for the static version, but you also want to select `Encode View in URL`. Now your browser address bar is filled with a query string (*all* the stuff *after* the question-mark) that has all the information necessary to reproduce your plot (and view). Copy everything after the first question-mark to the `interactive/code` element. Be sure to replace any ampersands by `&` (see the Author's Guide for more about certain characters in URLs). Examine the source for the examples below to see how they are authored. The *Help Manual* for CalcPlot3D is also available off the menu/hamburger icon in the upper-left.

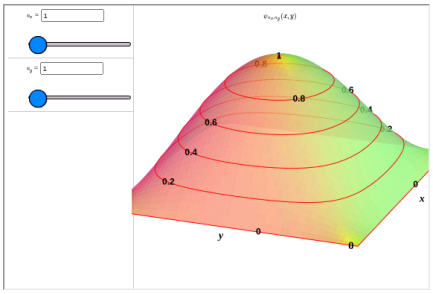
In Figure 15.3 grab the image with your mouse and rotate it in various directions. Then while the image has focus, press the `3` key (short for "3-D"), to get a 3D view, which will require some red-blue 3D glasses to fully appreciate. Press the key again to return to a regular view.

When using a version with controls (e.g. Figure 15.4), or the full application (e.g. Figure 15.5), specify an aspect ratio that is wider than it is tall. Start with `aspect="3:2"`, and perhaps fine-tune from there.



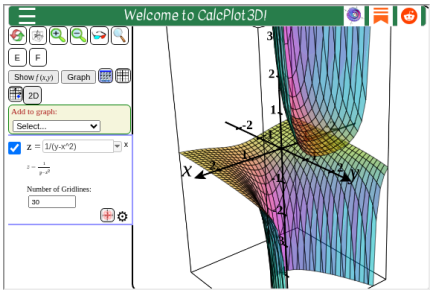
Standalone
In Context

Figure 15.3. Intersection of two planes (minimal embedding)



Standalone
In Context

Figure 15.4. Probability wavefunction with contours (includes controls)



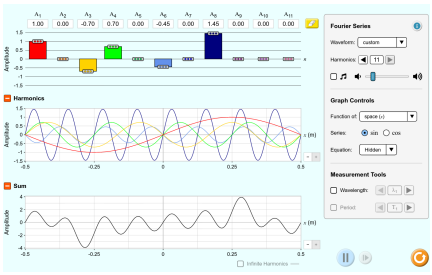
Standalone
In Context

Figure 15.5. Plot of $f(x,y) = \dfrac{1}{y-x^2}$ on $[-2,2] \times [-2,2]$ (full application)

15.4 IFrames from Servers

The `iFrame` versions of interactives can point to a network location, presuming the endpoint is reasonably well-behaved. If you are using this sytematically, let us know and perhpas it should become a more dedicated PreTeXt construction. See Subsection 14.9 for the local file version.

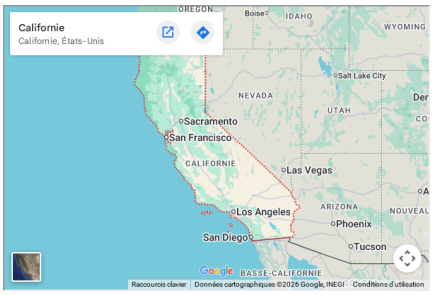
This example is from PhET Interactive Simulations.



Standalone
In Context

Figure 15.6. Fourier: Making Waves `iframe`

Anything that suggests you can **embed** an interactive widget via an iFrame is fair game for this feature. This is a Google Map of the state of California, for use in a French language document, from Julien Giol. The necessary URL is obtained by using the "Share" feature, and then the "Embed a map" option has HTML with the URL in a `@src` attribute.



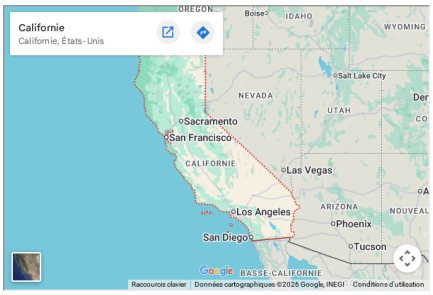
Standalone
In Context

Figure 15.7. California.

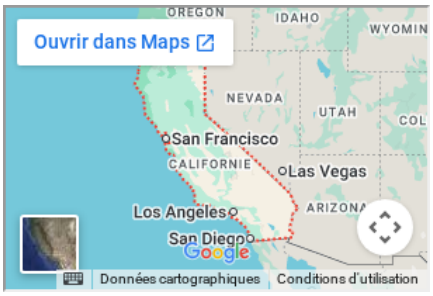
15.5 Layout of Interactives

In HTML, interactives are rendered in iframes, which can offer some layout challenges. The PreTeXt page has no direct control over the layout within the interactive. All it does is provide a fixed sized frame in which the

interactive content can display itself. The `@width` (a percentage) and `@aspect` (a width:height ratio like `2.1:1` OR a number like `1.5`) attributes on the interactive are the main tools to specify the dimensions. Below are three renderings of the same Google map using different widths/heights.



Standalone
In Context



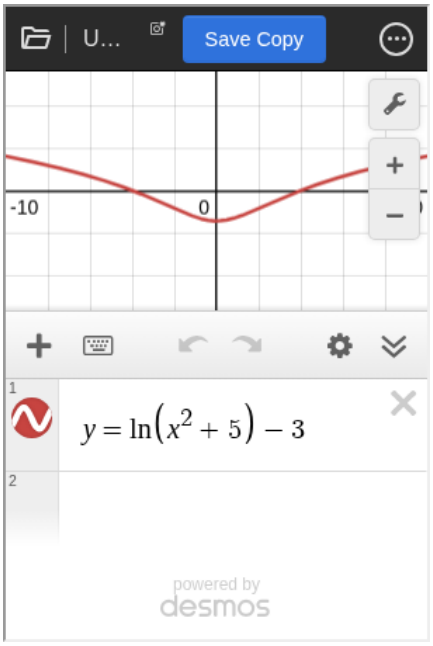
Standalone
In Context



Standalone
In Context

If you know a particular interactive is designed to display at a particular pixel size, you can can specify a `@design-width` like `300` to force exactly that

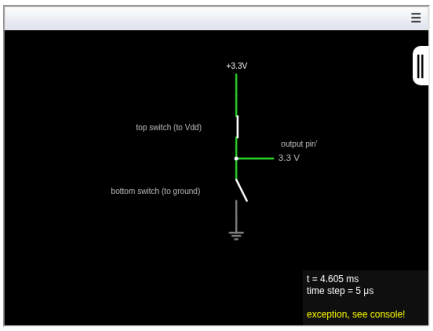
width. Combine that with `@aspect` to also force the height. The example below forces a size of 300x450.



Standalone
In Context

Figure 15.8. Graph of $ln(x^2 + 5) - 3$

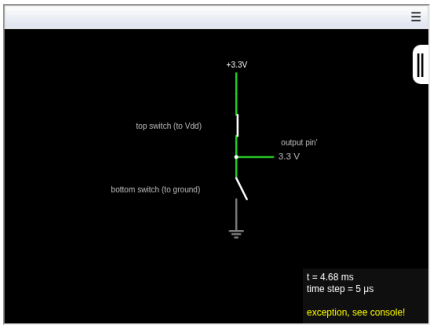
What happens if there is not enough space to display the requested frame? That depends on the `@resize-behavior` applied to the interactive. The default is `fixed-height`. In this case, the interactive's height will not change if the window becomes too small to display the full width of the interactive. Instead a horizontal scroll bar will appear. This can be seen in many of the samples in Section 14 and this CircuitJS sample:



Standalone
In Context

Figure 15.9. CircuitJS with `resize-behavior="fixed-height"`

The other option for `@resize-behavior` is `responsive`. In this case, the interactive's aspect ratio will be maintained. If the width gets smaller, so will the height. This can be seen in the version below.



Standalone
In Context

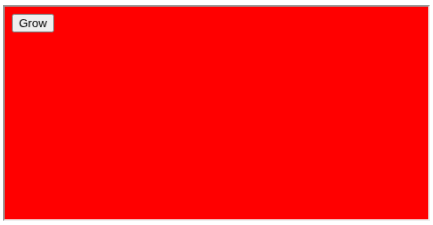
Figure 15.10. CircuitJS with `resize-behavior="responsive"`

The best behavior will be determined by the contents of the interactive. If the interactive knows how to dynamically size itself, `responsive` is likely the better option. If the interactive assumes that there is a certain amount of space and the contents will never resize themselves, `fixed-height` will likely work better.

In addition to specifying the resize-behavior in a particular interactive, you can apply a document level default value in your pub file. You can also specify a different default for a specific types of interactive. See the PreTeXt guide for documentation, or the `publication.xml` file of this Sample Article for an example of configuring defaults.

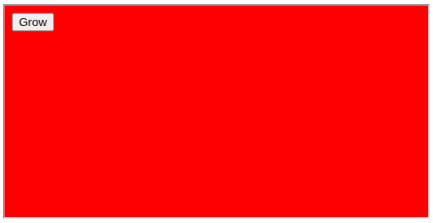
The code that generates content needs to make sure it does not overflow the available space or you will get scroll bars. Common culprits are `<canvas>` or `<svg>` elements that are not displayed as blocks or little bits of whitespace.

If an interactive wants to request a change to the amount of vertical space available to it, it can pass a message to the PreTeXt page asking for a `lti.frameResize`. (It is also possible to request a different amount of horizontal space, but doing so is not guaranteed to work and is much more likely to result in breaking the overall page layout.) The simple demo below uses that process to allow for resizing. A resize request should trump the original aspect ratio.



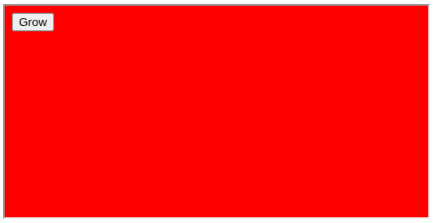
Standalone
In Context
Embed

Figure 15.11. Growable interactive fixed-height



Standalone
In Context
Embed

Figure 15.12. Growable interactive responsive with % width



Standalone
In Context
Embed

Figure 15.13. Growable interactive responsive and design-width

16 Interactive Exercises

Interactive components, *just* for testing, no commentary.

16.1 Faux Subsection

We use `<exercises>` divisions below, and need a `<subsection>` to feed the schema.

16.2 True/False

A True/False question.

1. True/False. Every vector space has finite dimension.

Hint. P_n, the vector space of polynomials with degree at most n, has dimension $n+1$ by 2.1. [Cross-reference is just a demo, content is not relevant.] What happens if we relax the defintion and remove the parameter n?

Answer. False.

Solution. False.

The vector space of all polynomials with finite degree has a basis, $B = \{1, x, x^2, x^3, ...\}$, which is infinte.

16.3 Multiple-Choice

Multiple-Choice problem

1. Multiple-Choice, Not Randomized, One Answer. What color is a stop sign?

A. Green
B. Red
C. White

Hint 1. What did you see last time you went driving?

Hint 2. Maybe go out for a drive?

Answer. B.

Solution.

A. *Incorrect.*
Green means "go!".
B. *Correct.*
Red is universally used for prohibited activities or serious warnings.
C. *Incorrect.*
White might be hard to see.

16.4 Parsons Problem, Math Proof

With some MathJax.

1. Parsons Problem, Mathematical Proof. Create a proof of the theorem: If n is an even number, then $n \equiv 0 \mod 2$.

- Click the heels of your ruby slippers together three times.
- Suppose n is even.
- Either:
  Then n is a prime number.
  Or:
  Then there exists an m so that $n = 2m$.
  Or:
  Then there exists an m so that $n = 2m + 1$.
- Thus $n \equiv 0 \mod 2$.
- So we have the displayed equation:

$$n = 2m + 0.$$

This is a superfluous second paragraph in this block.

Solution.

- Suppose n is even.
- Then there exists an m so that $n = 2m$.
- So we have the displayed equation:

$$n = 2m + 0.$$

This is a superfluous second paragraph in this block.

- Thus $n \equiv 0 \mod 2$.

16.5 Parsons Problem, Code

Programming Parsons problem, requiring indentation.

1. Parsons Problem, Programming. The Sieve of Eratosthenes computes prime numbers by starting with a finite list of the integers bigger than 1. The first member of the list is a prime and is saved/recorded. Then all multiples of that prime (which not a prime, excepting the prime itself!) are removed from the list. Now the first number remaining in the list is the next prime number. And the process repeats.

The code blocks below can be rearranged to form one of the many possible programs to implement this algorithm to compute a list of all the primes less than 250. [Ed. this version of this problem requires the reader to provide the necessary indentation.]

- `for nonprime in range(p, n, p):`
- Either:

  ```
  primes = []
  candidates = list(range(2,n))
  ```

  Or:

  ```
  candidates = []
  primes = list(range(2,n))
  ```
- `p = candidates[0]`
  `primes.append(p)`
- `print(primes)`
- `if nonprime in candidates:`
  `candidates.remove(nonprime)`
- `n = 250`
- `primes = candidates + [p]`
- `while candidates:`

Solution.

```
n = 250
primes = []
candidates = list(range(2,n))
while candidates:
    p = candidates[0]
    primes.append(p)
    for nonprime in range(p, n, p):
        if nonprime in candidates:
            candidates.remove(nonprime)
print(primes)
```

16.6 Matching

Events and dates.

1. Matching Problem, Dates. Match each event in United States history with the year it happened.

Louisiana Purchase	1823
Battle of Gettysburg	1886
Haymarket Riot	1803
Monroe Doctrine	1863

Solution.

- 1823
  - Monroe Doctrine
- 1886
  - Haymarket Riot
- 1803
  - Louisiana Purchase
- 1863
  - Battle of Gettysburg

16.7 Clickable Area

Words, not code.

1. Clickable Areas, "Regular" Text. Identify (by clicking, or by circling) all of the nouns in this quotation by Eleanor Roosevelt.

"The future belongs to those who believe in the beauty of their dreams."

Answer. Correct: *future*; *beauty*; *dreams*. Incorrect: ~~those~~; ~~their~~.

The incorrect words are pronouns.

Solution. "The *future* belongs to ~~those~~ who believe in the *beauty* of ~~their~~ *dreams*."

16.8 Old-Style Fillin-In

Do not use this as a model for new exercises. Just for backwards-compatibility.

1. Fill-In, String and Number Answers. Complete the following line of a Python program so that it will declare an integer variable `age` with an initial value of `5`.

_____ `age =` _____`;`

Solution. Complete the following line of a Python program so that it will declare an integer variable `age` with an initial value of `5`.

int `age =` _5_`;`

A variable of type `int` is appropriate for whole number ages.

An integer variable may be initialized to a value.

16.9 A Reading Question

1. Short Answer. This should be built with a text-box, *only* on a capable server (Runestone). So it can be answered

17 Dynamic Exercises

This section demonstrates the use of dynamic randomized exercises built upon the framework of Runestone components. These demonstration problems incorporate a library supporting mathematical expressions both for varying the content of the statement of the exercises as well as the checking of submitted answers. Then there is an example where a custom library is used for defining and parsing answer objects without using the built-in math library, and which also illustrates the use of a config structure passed in JSON format.

17 Dynamic Fill-In

The first problem illustrates revised markup for fill-in problems that don't involve randomization and use simple string and number comparison tests. Later exercises illustrate the use of dynamically generated mathematical expressions.

1. Fill-In, String and Number Answers. Complete the following line of a Python program so that it will declare an integer variable `age` with an initial value of `5`.

_____ `age =` _____;

Solution. Complete the following line of a Python program so that it will declare an integer variable `age` with an initial value of `5`.

int `age =` _5_;

2. Fill-In, Multiple Strings, Custom Checker. In order to apply the Integral Test to a sequence $\{a_n\}$, the function $a(n) = a_n$ must be _____, _____ and _____.

Solution. In order to apply the Integral Test to a sequence $\{a_n\}$, the function $a(n) = a_n$ must be _continuous_, _positive_ and _decreasing_.

3. Fill-In Formula (Dynamic). Find a formula for a cubic function $f(x)$ that has roots at $x = -5$, $x = -2$, and $x = 2$ and so that $f(0) = 5$.

$f(x) =$ _____

Solution 1. Find a formula for a cubic function $f(x)$ that has roots at $x = -5$, $x = -2$, and $x = 2$ and so that $f(0) = 5$.

$f(x) =$ _$-\frac{1}{4}(x+5)\,(x+2)\,(x-2)$_

Solution 2. Knowing the roots of a polynomial allows us to write down the formula of $f(x)$ in factored form,

$$f(x) = A\,(x+5)\,(x+2)\,(x-2)$$

with an unknown scaling multiple A.

When we evaluate $f(x)$ at $x = 0$ using this formula, we find

$$f(0) = -20A.$$

Since we also know $f(0) = 5$, we can write down the equation

$$A\,(x+5)\,(x+2)\,(x-2) = 5$$

and find that $A = -\frac{1}{4}$.

Consequently, we can write our function in the form

$$f(x) = -\frac{1}{4}(x+5)(x+2)(x-2).$$

4. Decompose the Function. Consider the function

$$h(x) = 4(-x-3)^3 + 5.$$

Find two nontrivial functions $f(x)$ and $g(x)$ so that $h(x) = f(g(x))$.
$f(x) =$ _____ and $g(x) =$ _____

Solution 1. Consider the function

$$h(x) = 4(-x-3)^3 + 5.$$

Find two nontrivial functions $f(x)$ and $g(x)$ so that $h(x) = f(g(x))$.
$f(x) = _4x^3 + 5_$ and $g(x) = _-1x - 3_$

Solution 2. Noticing that the expression $-1x - 3$ appears inside parentheses with a power, it makes sense to think of that as the inner function, defining $g(x) = -1x - 3$. The outer function describes what happens to that. If we imagined replacing the formula $-1x - 3$ with a box and then call that box our variable x, we find the outer function is given by $f(x) = 4x^3 + 5$.

This is not the only non-trivial composition. Can you find others?

5. Fill-In, Remaining Machinery. Let $n =$. Enter either n or its negative: _____.

Solution. Both and its negative are accepted; the automatic solution is declined in favor of this sentence.

6. Consider the matrix defined by

$$A = \begin{bmatrix} -5 & 12 & 75 \\ -3 & 7 & 44 \\ -32 & 77 & 481 \end{bmatrix}$$

Find the reduced row echelon form of A.
$\text{rref}(A) =$ _____
A matrix is entered by listing the rows separated by commas between brackets [and], and for each row listing entries separated by commas between additional brackets. The matrix $\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$ would be entered as `[ [1, 3, 5], [2, 4, 6] ]`.

Solution 1. Consider the matrix defined by

$$A = \begin{bmatrix} -5 & 12 & 75 \\ -3 & 7 & 44 \\ -32 & 77 & 481 \end{bmatrix}$$

Find the reduced row echelon form of A.
$\text{rref}(A) = _\begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & 5 \\ 0 & 0 & 0 \end{bmatrix}_$
A matrix is entered by listing the rows separated by commas between brackets [and], and for each row listing entries separated by commas between additional brackets. The matrix $\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$ would be entered as `[ [1, 3, 5], [2, 4, 6] ]`.

Solution 2. Row reduction of A gives

$$\text{rref}(A) = \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & 5 \\ 0 & 0 & 0 \end{bmatrix}$$

18 Interactive Coding

More interactive components, just for testing, no commentary.

18.1 ActiveCode

ActiveCode, Python program.

Listing 18.1. An interactive Python program, using *Runestone*

```
print("Hello, World!")
```

Listing 18.2. An interactive Python program without codelens.

```
print("Hello, World!")
```

18.2 CodeLens

A steppable Python program.

Listing 18.3. A Python program, stepable with CodeLens

```
print('Hello, World!')
```

18.3 Activity with An ActiveCode

Something to do with ActiveCode program.

Activity 18.1 Activity Coding Exercise. Similar to above, but now as a complete Python program inside an `<activity>`. This demonstrates the possibility to use any "project-like" block (`<project>`, `<activity>`, `<exploration>`, `<investigation>`), but not in the case when structured with `<task>`.

```
for i in range(10):
    print(i)
```

Answer. We're still not really sure.

18.4 YouTube

Video, observable on a Runestone server.



Standalone
In Context

19 Audio

2019-05-24: this is preliminary, and mostly based on the code for `<video>` so read the next section and mimic the style from there. But use an `<audio>` element and have the `@source` attribute point to an OGG, MP3, or WAV file. Plus, an `@aspect` attribute will be ignored.

We have not entirely decided how to handle the static version present in a PDF.

First in a `<figure>`, so it can be cross-referenced.

No static image provided via `@preview` attribute

Standalone
In Context

Figure 19.1. MP3 Piano Trill (`www.kozco.com/tech/soundtests.html`)

Now, naked, between a couple of paragraphs, with specified asymmetric margins, and a computed width.

No static image provided via `@preview` attribute

Standalone
In Context

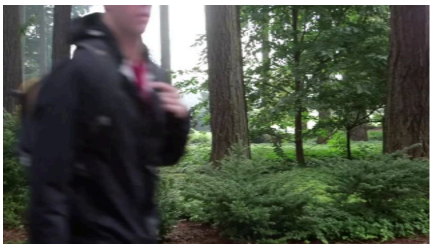
20 Video

First, a gratuitous reference to Exercise 11.2.3 about the derivative of a cosine.

You can specify a video by a filename if you host it as part of your document, or a URL giving a location on the Internet. There are a few extra features for YouTube and Vimeo videos, which are near the bottom of this page, so look there first if that meets your needs.

20.1 Video Files

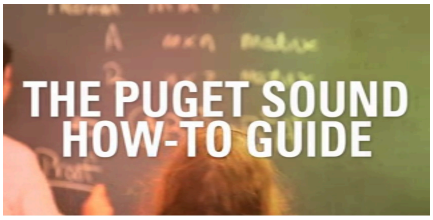
Embedded videos can make sense for a web version of your document. This is a video promoting the University of Puget Sound to potential new students, in WebM format. Support is limited to HTML5-capable browsers. The file format can be MP4, Ogg, or WebM, though this may vary depending upon the browser. Use a `<video>` element within a `<figure>` (numbered, captioned). The `@source` attribute in this first example *does not* include an extension, and so three possibilities above will be searched for preferentially (you need only provide the video in one format, but providing additional versions will increase the chances every browser will find a compatible format).



Standalone
In Context

Figure 20.1. University of Puget Sound Promotional Video

You can replace the "preview" image of a video with one of your own making. HTML refers to this as the **poster**, presumably because it advertises the video. The image you make should be of the same quality as the video, and with the same aspect-ratio, and is presumably one of the frames of the video. So a screenshot is the likely avenue. (Maybe we will have advice on how to do this easily, or see Issue #853.) On the `<video>` tag, include a `@preview` attribute which is the name of an image file, including a relative path. (JPEG or PNG formats are best. JPEG may be smaller for video screenshots, while PNG is lossless and so may work better for diagrams.) The next video has a preview/poster that is a fram part way into the introduction.



Standalone
In Context

Figure 20.2. University of Puget Sound Promotional Video

If you find the posters provided automatically for a video to be distracting or objectionable, you can cover them up by requesting a generic poster with the attribute-value pair: `preview = "generic"`.



Standalone
In Context

Figure 20.3. University of Puget Sound Promotional Video

You can use a very similar construction to point to a video hosted somewhere on the Internet, just use a complete URL for the `<source>` attribute. Note that if the URL has a query string (parameters following a question-mark), then any ampersands (&) will need to be escaped, so as to not confuse the XML processor (i.e. use `&amp`). Also, the question-mark character needs to *not* be URL-encoded (`%3F`), so presumably edit the URL to be the character. Here are several examples, the second one uses the `@start` and `@stop` attributes to limit the video to just the time between the 16-second and 30-second locations, and has asymmetric margins.



Standalone
In Context

Figure 20.4. *Sintel* trailer, by the Blender Foundation, licensed CC BY 3.0. The construction follows the "Video for Everybody" Test Page.



Standalone
In Context

Figure 20.5. *Sintel*, Controlled Start/End, Asymmetric Margins



Standalone
In Context

Figure 20.6. *Sintel*, Ogg container, `*.ogv` extension. Browsers have retired the Theora codec, so this one is expected *not* to play—it is here to exercise the markup, and to show what a reader sees when a format is no longer supported.



Standalone
In Context

Figure 20.7. *Sintel*, MP4 format



Standalone
In Context

Figure 20.8. *Sintel*, WebM format



Standalone
In Context

Figure 20.9. *Sintel*, no extension, so every format is offered and the browser chooses. Theora is no longer decoded, so the `*.ogv` source is passed over and the `*.webm` one is used—the fallback a list of `<source>` exists to provide.

Videos are assumed to have a 16:9 aspect ratio (width:height). If this is not the case, then you must specify the aspect ratio with either a ratio (e.g. 4:3) or as a number expressing the fraction width/height (e.g. 1.3333). Four decimal places should suffice for the latter. Note that you cannot *change* the aspect ratio, and you *must* supply the aspect ratio for any video that does not have

the default ratio. This is a technical requirement that allows us to smoothly scale the videos on small devices (try this page on your mobile phone!).

20.2 YouTube

YouTube videos may be embedded with only knowledge of a video's "ID" or a "playlist ID". A single video's YouTube ID is a string of eleven seemingly random characters that show up in the URL when you watch a video. For the Led Zeppelin performance below, the ID is `hAzdgU_kpGo`, which you might normally watch directly from the URL `www.youtube.com/watch?v=hAzdgU_kpGo`. As described just above, you need to specify a different aspect ratio if the video does not have a 16:9 aspect ratio.

Note: some of these videos may not play if viewed locally, and maybe not even if you start up a local web server (such as can be easily done with Python). So if you are testing new features, be careful about assuming videos from cloud services are not working properly.

If you have ever owned a drone, you sympathize with this guy. Way funnier than a cat video.



Standalone
In Context

Figure 20.10. First Drone Flight (1:28)

If you are only interested in a piece of the action, you can limit the video with `start` and `end` attributes in seconds. You might make those times clear in the caption for readers getting the link out of a PDF. Some videos may not respect these parameters.



Standalone
In Context

Figure 20.11. First Drone Flight (Splashdown, 0:54 to 1:12)

This next video comes with a default poster from YouTube featuring Robert Plant. We've replaced it with a poster featuring Jimmy Page.



Standalone
In Context

Figure 20.12. Kashmir (Live), Led Zeppelin. O2 Arena, London. December 10, 2007. (8:55)

And if you don't want to be reminded of Plant, Page, Bonham, or Jones, you can cover them up entirely.



Standalone
In Context

Figure 20.13. Kashmir (Live), Led Zeppelin. O2 Arena, London. December 10, 2007. (8:55)

Videos were first designed mostly on the assumption that they are wrapped in a `figure` with a `title` (which is distinct from a `caption`). But you can just place a video "naked" inbetween a couple of paragraphs. This is nice if you don't want the clutter of numbers, and/or if you plan to exclude videos from some edition of your document.



Standalone
In Context

A YouTube "playlist" can be built in one of two ways. You may specify the `youtube` attribute to be a space-separated list of several video IDs. Alternatively, you may set the `youtubeplaylist` attribute to a YouTube playlist ID.



Standalone
In Context

Figure 20.14. Individual IDs specified in an “itemized” playlist



Standalone
In Context

Figure 20.15. YouTube playlist ID specified in a “named” playlist

20.3 Vimeo

We support videos from Vimeo in much the same way as YouTube videos. Now the identifier is a long integer. For example, the video up next would normally be found at `https://vimeo.com/27246366`. But instead, you can embed the video with as little as `<video vimeo="27246366"/>`. As of 2019-05-18, we intend to support as many of the options described above as possible. Widths and heights (via the aspect ratio) will perform as expected. We have not investigated start and end times.



Standalone
In Context

Figure 20.16. “MOVE”, by Rick Mereki, `vimeo.com/rickmereki`

Now with an author-supplied poster.



Standalone
In Context

Figure 20.17. "MOVE", by Rick Mereki, `vimeo.com/rickmereki`

20.4 Captions and Subtitles

Experimental support for captions and subtitles first. Look at the source, which mimics the actual HTML. The words of the titles and/or subcaptions (there is a difference!) come from a file in Web Video Text Tracks (WEBVVT) format.



Standalone
In Context

Figure 20.18. *Sintel* with a single subtitle track

This video is identical to the previous one, except it tests the use of a default `<track>`. The `@default` attribute on `<track>` can be set to the value `yes` to make one set of captions the default (and only one!). The two `<track>` use different files, whose text says which is which, so switching tracks in the player's menu is visible. Track named `US English Two` will show as in-use at start-up.



Standalone
In Context

21 Cross-Referencing

Cross-references are easy, since that is a key reason for having a highly structured document. Here is a useful feature. Any `<xref>` will "know" what it points to, so you can let it provide the "type name" part of the cross-reference text, such as "Theorem" or "Section". You control this document-wide with the `@text` attribute of the `<cross-references>` element in your `<docinfo>`. The value `type-global`—the default, and what this article uses—supplies the type name automatically, so there is no need to write it into your source.

Moreover, the names themselves will change with the use of the one language-dependent file. And another bonus is that with an automatic type name, you get a *non-breaking* space between the name and the reference. A type name makes no sense for "provisional" cross-references, since there is no information about what they point to.

Here is a reference that suppresses its type with `text="global"`: 2.1. So you will just see a number that you can click on. Using `text="type-global"`—or simply accepting the document default—you would instead see "Theorem" prepended. Note that if you changed the theorem to a lemma, then that change would be reflected here automatically.

Whatever document-wide default you set with `<cross-references>`, there will still be instances where you want to override that choice. Simple: just say `text="type-global"` or `text="global"` as part of the `<xref>`. Each example below should look the same each time this article is processed, no matter the document-wide `<cross-references>` setting.

- No name ever: 4.1
- Always named: Corollary 4.1

You might also wish to provide a prefix to a cross-reference and have it incorporated into the text of what you would click on in an electronic version. So if you make an `xref` with some content, then that content will prefix the cross-reference within the clickable/pokeable text and be attached with a non-breaking space. This `xref` content totally overrides any prefix that might happen otherwise. So the name of an item (e.g. "corollary") could be replaced, and if you make a cross-reference with custom text, that will be the clickable also. An example:

- A grand result: Major Corollary 4.1
- A grand result: a nice corollary

Suppose you want to reference two theorems, so you might want to say something like "Theorems 4.6 and 5.2." With type names shown (the default), you can override the first `Theorem` by providing the content `Theorems` on the first `xref` and `text="global"` on the second `xref`. (With type names suppressed, you will also get what you want/expect.) Here is the test, which should look correct no matter the document-wide setting: Sections 21 and 22. (But notice that it is up to you to be certain the types of these targets do not change without you changing the content of the first `xref`. The "author-tools" mode and careful choices of `xml:id` strings can help avoid this trap.)

If you set the value of `@text` to `title`, then the title you assigned to the theorem will be used as the link for a cross-reference. Here is a the final example, which refers to a fundamental theorem by name The Fundamental Theorem of Calculus.

Cross-references to exercises with hard-coded numbers should respect the supplied number. Exercise 46.42a should reference problem 42a.

Here we form a list to test pointing at various structures. Each of the following should open a knowl in the HTML version, otherwise it will be a traditional hyperlink (if possible). Note that if a knowl opens, there will always be an "in-context" link which will take you to the actual location, should you have wished instead to just go there.

- Footnotes: Fermat allusion at 2.1.
- Citations: Judson's AATA with annotation at [J.3.1]
- Citations: Judson's AATA with a type name, which has zero effect [J.3.1]

- Citations: In a `<references>` division inside an appendix [J.2.1]
- Note: just the annotation of previous citation at J.3.1.1
- Examples: Mystery derivative at 4.2, or a question at 4.8.
- Definition-like: A mathematical statement with no proof 4.15.
- A numbered Note: 4.11
- A link to a `proposition` element, while this document has globally renamed `propositions` as "Conundrum"s, so this link should use the new name: Conundrum 33.1
- Theorems: Fundamental Theorem of Calculus, with proof at 2.1
- Proof: of second version of FTC at 4.1.1
- Figures: A plot with a derivative at 5.2.
- A Figure within a side-by-side panel, with its own number: 26.5
- A Table within a side-by-side panel, with a subnumber: 26.14(a)
- A Figure, containing a side-by-side with two sub-captioned images: 26.1
- Display Mathematics: single, first with no name: (4.1). Then with a type name: (4.1).
- Display Mathematics: multi-row, first with no name: (4.2). Then with a type name: (4.2). And two, with a plural form: Equations (4.1) and (4.2).
- You can cross-reference The Fundamental Theorem of Calculus via custom text of your choice.
- Display mathematics: an equation with "local" tag, which should not be used so very far away: (✶✶).
- You can author a cross-reference to a displayed equation with no number, but it will not be very satisfying. You should get a warning if you try.
- Exercises (divisional), a range, with plural form provided to override the type name: Exercises 11.2.1–11.2.3.
- Exercise (inline): with enclosed hint at 4.6
- A group of two exercises, with introduction, conclusion: Exercise Group 11.2.2–3
- Solution: A portion of an exercise, with its type name: Solution 46.42a.1
- Parts of a complicated exercise: Hint 46.12.2 Answer 46.12.1
- A task within a task of a divisional exercise: 12.10.1.b.i
- Three cross-references to individual exercises, but due to their location, they should have different "type names" in the cross-reference: in an `<exercises>` division, Exercise 41.2; in the narrative, Checkpoint 4.6; and in a `<worksheet>`, Worksheet Exercise 37.1.2.
- An item buried in nested ordered lists: Item 2.b.ii.C
- List item as knowls in HTML, including nested lists: 2, Item 2.b.ii
- A titled list: 12.3
- List item inside a named list, second color in rainbow list: Item 12.3:2
- Second color in rainbow list, but now as a local reference: Item 2
- An item in ordered list, but contained in an unordered list, hence without a number, so a cross-reference by number would be ambiguous. So we use a cross-reference which relies on custom text: No Number List Item
- Several examples of hybrid cross-references to list items within a named list can be found in, and adjacent to, List 12.4.
- Hybrid cross-references to tasks of exercises in a worksheet are shortened within the worksheet, as in the first and last exercises of Worksheet 37.6. From here, the same hybrid cross-references have full numbers: Task 37.6.3.e and Task (b) of Worksheet Exercise 37.6.3.
- An assemblage, which never has a number. A cross-reference now requires content in the `xref` element, with `text='custom'`: text to xref an assemblage

- A cross-reference to a list item in a description list, which has a title, but never a number: Central Processing Unit (CPU). Note that you need to include the attribute `text="title"` even if that is obvious from the situation. This requirement may be relaxed in a future refactoring of the cross-reference system.
- A very similar cross-reference to the previous one, but testing how final punctuation of titles is handled: Bus!.
- A cross-reference to a "paragraphs" subdivision, which never has a number (so comments above about description list items and titles applies here too): Structure
- A case within the proof of Claim 4.3: Case 3b: The inductive step
- A cross-reference to a description list item with a title containing math: Math x^2
- A cross-reference to an aside, by title necessarily, and with some formatting in the title: An Aside with a *Formatted* Title
- A cross-reference to an `objectives` block, with a type name. This demonstrates the number of the Objectives here, which is not shown in the original version since it is implicit: Objectives 4
- A cross-reference to an individual objective. This is authored as a list item, but displayed as an objective (singular) via a type name: Objective 4.1
- A cross-reference to the top-level element (e.g.`book`) will point to a summary page similar to a Table of Contents in HTML. For LaTeX output it will behave similarly, unless there is no Table of Contents, then it will go to the main title page: ToC or Title
- "Cross-references inside quotations previously lost track of their target, so this tests correcting that, not so much the cross-reference itself: Theorem 2.1"
- An activity with full details following: 4.3
- A `type-global` cross-reference to a second-level `task` within a project: Task 4.4.c.iii, the encompassing `project`: 4.4, and a `local` reference (c.iii).
- A subcaptioned named list: 26.23(b)
- This opens a knowl for an `example`. It has a solution, which is orginally presented as a hidden knowl. But since this version is a duplicate, the knowl for the solution is a file version, not an embedded version, and hence free from duplicating any unique identifaction like an HTML id. So we test its styling and function here: Example 4.7
- A cross-reference to a poem, where we need to use a title for the link text, since a `poem` is not numbered: The Charge of the Light Brigade
- A cross-reference to a `<references>` division in a subsection, which should not be numbered where born, but which has the number of its parent division in a cross-reference: References 11.4. And a cross-reference to a `<references>` division, which is the "main" bibliography in the back matter, and so is not numbered where born, nor in a cross-reference (which we must accomplish via it's title): References.
- A cross-reference to a `<solutions>` division in a subsection, which should not be numbered where born, but which has the number of its parent division in a cross-reference: Solutions 4.2.6. And a cross-reference to a `<solutions>` division, which should appear as an appendix both where born and as a cross-reference: Appendix B.
- A cross-reference to an `<exercises>` division in a subsection, which is the only such division in that subsection and therefore should not be numbered where born, but which has the number of its parent division

in a cross-reference: Exercises 4.2.3. In contrast we cross-reference an <exercises> division which is one of two inside a section, and therefore is numbered, when born and when cross-referenced, in continuity with the preceding subsections: Exercises 11.3. Also an <exercise> within an <exercises> which should have a cross-reference employing the number of the containing (unstructured) <section>: Exercise 41.2 which is in Exercises 41 which is in the (unstructured) Section 41.

- A custom cross-reference: Custom
- Cross-reference to <instructions> of an <interactive>: [cross-reference to target(s) "kinematics-instructions" missing or not unique]
- A hyperlink to a <subexercises> via its required title (no number is assigned): Hard Problems
- You can request the text to be a type, then a number, then a title: Theorem 2.1 The Fundamental Theorem of Calculus
- Asking for the text to be a type, then a number, then a title, when there is no title, is not a problem: Corollary 1
- A cross-reference to the <introduction> of a (structured) section, and to the <conclusion> of the same section. There is at most one of each per division, so each takes the number of its parent division, displayed only here in a cross-reference (never where born), and each should be a traditional hyperlink, never a knowl: Introduction 4 and Conclusion 4. In an HTML version chunked so the section becomes a summary page, these lead to the companions' own pages.

Cross-references to structural elements of the document will always take you there directly, since even in the HTML version these parts never get realized as knowls. You will find such links sprinkled through this document, but here is a type-named link to a subsubsection: Subsubsection 4.2.1.

Cross-references can be built into display mathematics, but they can only point to one item (i.e. a comma-delimited list of targets is not supported). Examples below should test the distinction in HTML output between a link that opens a knowl and a link that jumps to a larger chunk of content. Notice that display mathematics is entirely LaTeX syntax, no matter which output format you create. So if you do not use the automatic type name, you need to wrap non-math text in \text{} and perhaps use a tilde (~) as a non-breaking space (examine the source of this article).

$$x^2+y^2=z^2 \qquad \text{Theorem 2.1}$$
$$a^2+b^2=c^2 \qquad \text{Section 2}$$

Variations on the above include multiple xml:id as the value of a single ref attribute on an xref, in the form of a comma-separated list. In this case, only the numbers are links/knowls and the type name is based on the type of the first ref. Wrapping with brackets (citations) or parentheses (equations) is also controlled by the type of the first ref. And the detail attribute for a bibliographic reference is silently ignored. So you can do silly things like have a reference to a theorem within a list of equation numbers and there will be no error message. Handle with care. Spaces after commas in the list will migrate to the output as spaces, so if you don't have any, you won't get any.

- Four theorems, with spaces, with type names: Theorem 2.1, Theorem 32.1, Theorem 32.2, Theorem 32.3

- Two equations, no spaces, with type names: (4.1), (4.2)
- Two bibliographic items, no type name: [J.3.1, 2]

If you have a long list of items (such as homework exercises, not in an `exercisegroup`, or perhaps several chapters), you can get a cross-reference that prints as a range by using `xref` with two attributes `first` and `last`, which may contain a single `xml:id` each. As with multiple references, `first` will control the type name and other features.

- A range of exercises, with type names (this range appears "out-of-order" since the two `exercise` are numbered under two different schemes): Checkpoint 4.6–4.2.3.1
- A range of equations: (4.2)–(4.3)
- A system of equations, given as range from first to last: (7.1)–(7.2)
- A range of sections, hand-named to be plural: Sections 3–21
- A range of bibliographic items: [J.3.1–2]
- A range with custom text, which replaces the entire range, endpoints, separator, and any brackets (compare with the previous item): a pair of consecutive references

The `url` element may be used to link to a data file, either externally, or internally, if you want to make such an object available to a reader. A good example use case is a spreadsheet that might be part of an exercise, or contain data relevant to some discussion. First let us suppose the data resides somewhere on the Internet, then just use the complete address. Here is one from Microsoft: Sample Excel Spreadsheet.

For a link like the previous one, you might want to provide advice appropriate for your audience about using a context menu to download a file, or how to configure helper/viewer applications.

You can also provide a file yourself, but now it is your obligation to distribute the file with your document (HTML, PDF, etc.) and provide a relative link. This creates some complications, such as making sure an electronic PDF has the associated file in the same place relative to the PDF file. Of course, if you make a print PDF, this becomes impossible. Here is a test example anyway, which is highly likely to be broken in a PDF (including at the PreTeXt project site) unless you build this example on your own computer, locally. Here is a template from the Apache OpenOffice project, provided via the Public Documentation License (PDL): Running Statistics Template.

The next four paragraphs are each a single `<dataurl>` element. Explore the source and the output from different conversions. Strictly for testing as of 2022-11-04.

Foo Sample Excel Spreadsheet Bar

Foo Runners Template Bar

Foo Sample Excel Spreadsheet Bar

Foo Runners Template Bar

Testing of output positioning for xref's that are inside containers:

Table 21.1. Self-referential Xref In a table

A	B	C	D
Table 21.1	B	C	D

$\mathbb{Z}_2 \times \mathbb{Z}_2$

Theorem 2.1

$\mathbb{Z}_2 \times \mathbb{Z}_2$

Now we have two xref's to that same target that has a runestone component. Only the first one clicked should try to render the runestone.18.118.1

22 Internationalization

Supporting a multitude of possible characters, across many languages and across many output formats can be a challenge. One of our goals is to make this much easier for authors. Fortunately, the Unicode standard has led to improvements from the 7-bit ASCII standard of old.

Unicode Characters for HTML Output. First, we discuss HTML output. If you include Unicode characters in your PreTeXt source, they should survive just fine *en route* to a web browser or e-reader. Here are the caveats for HTML output:

- So that you can continue to get the best results with print and PDF output, use available empty elements for obscure characters, even if targeting HTML output, before resorting to a Unicode character. For example, use `<copyright/>` for the copyright symbol in text before resorting to the Unicode character `U+00A9`. It is a bit more work, but you will get better results with other conversions, even if you initially are only fascinated by HTML.
- How you actually enter Unicode characters into your source file is dependent on your editor and operating system, and is therefore outside the scope of our documentation. You can cut-and-paste characters and text from the source of our examples for initial testing and experimentation.
- Always, always identify your source as having Unicode characters by including the incantation

  ```
  <?xml version="1.0" encoding="UTF-8" ?>
  ```

  as the first line of your source file. (You *may* be able to accurately cut-and-paste this version here. But if the copy has non-standard characters in it, go back to the top of this source file for a copy.)
- Alan Wood's Unicode Resources has a plethora of samples of various groups of Unicode characters. If you, or your readers, are "missing" characters in a web browser, this is a good place to start testing the local setup.

Characters in LaTeX, PDF, print. The situation for LaTeX is a bit more complicated, since TeX pre-dates Unicode's widespread adoption.

This sample article is intended to work well, out-of-the-box, for authors just starting with PreTeXt. So we only include here examples that we know are likely to convert to PDF without any errors. For more extensive examples and experiments, we provide the sample document `examples/fonts/fonts-and-characters.xml`, so be aware of that example as you look to see what is possible.

Similarly, you should be able to process this sample article successfully with various LaTeX engines. We test regularly with `pdflatex` and `xelatex` and provide online sample PDF output of this document processed by `pdflatex`. In principle, you should be able to use `latex` (to produce a DVI), and possibly other (unsupported) engines, such as `lualatex`.

Once you get beyond the Latin alphabet, with accents common in Western Europe and the Western Hemisphere, you will almost assuredly need to restrict

your attention to producing PDF output with the `xelatex` engine. This is discussed and tested in `examples/fonts/fonts-and-characters.xml`.

Basic Latin, `U+0000`–`U+007F`. Unicode uses multiple 8-bit bytes to represent characters, and these are typically expressed in hexadecimal (base 16) notation. Using just a single byte, we can get 256 values, and the first 128 (hex `00` to `7F`) are the "usual" Latin characters with some values used as control codes. These 95 characters are the most basic, and will all render using `pdflatex` or `xelatex` with no special setup (and will render easily in HTML). `U+0000` to `U+001F` are control codes and not used here. `U+007F` is also a control code and so is excluded, while `U+0020` is a space, so appears invisible in the table. In the source we have authored each character by its escaped version using its Unicode number (in hexadecimal). So, for example, capital-B is authored as `B`.

Table 22.1. Basic Latin, Regular

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
002_		!	"	#	$	%	&	’	(	)	*	+	,	-	.	/
003_	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
004_	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
005_	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
006_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
007_	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	

Latin-1 Supplement, `U+0080`–`U+00FF`. Now we are interested in the next 128 possible bytes, (hex `80` to `FF`). The first 32 are again control codes and `U+00A0` is a non-breaking space, so is invisible, while `U+00AD` is a soft hyphen (which we have not implemented and so is excluded). We have taken care to see that the remainder will render using `pdflatex` or `xelatex` with no special setup (and HTML). In the source we have authored each character by its escaped version using its Unicode number (in hexadecimal). So, for example, a copyright symbol is authored as `©`.

Table 22.2. Latin-1 Supplement, Regular

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00A_		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
00B_	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
00C_	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
00D_	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
00E_	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
00F_	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Monospace, Basic Latin and Latin-1 Supplement, `U+0000`–`U+00FF`. A monospace font is critical for samples of keyboard input and to distinguish exact technical input from running commentary. We list here all of the reasonable characters from the first 256 Unicode code points. (We skip the same 65 control characters from above, and the soft hyphen.) These should

all render fine in HTML and when processed with `xelatex`, however our focus with this sample article for PDF output is the capabilities when processed with `pdflatex`. First, characters from `U+0000`–`U+007F`.

Table 22.3. Basic Latin, Monospace

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
002_		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
003_	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
004_	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
005_	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
006_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
007_	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	

Note that the single and double quotes are upright and dumb, not curly and smart: ' " ' " ' ". And a backtick is a backtick: ` ` `. The zero is distinguished from the capital "oh": 0 O 0 O 0 O. And the numeral one is slightly different from the lower-case "ell": 1 l 1 l 1 l. The hyphen should be short and not expanded into some other kind of dash: - - -. These characters should all cut/paste out of a PDF into a text editor with no conversion to other characters.

Now the remaining characters from `U+0080`–`U+00FF`. The `program` tag is implemented in LaTeX via the `listing` package and these characters require ad-hoc replacements for processing by `pdflatex`. (You can see the replacements in the preamble of the LaTeX source for this document.) The replacement mechanism provided by the `listing` package will cause the characters below to produce a LaTeX compilation error if processed by `pdflatex` and in a table cell in certain situations (which we have avoided in the table below). The only workaround in this case is to switch to `xelatex`.

Table 22.4. Latin-1 Supplement, Monospace

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00A_		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
00B_	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
00C_	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
00D_	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
00E_	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
00F_	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

The `pre` tag is implemented in LaTeX with the `fancyvrb` package. You can compare results here with the table above, lines here are rows above.

```
  ¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬   ® ¯
° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿
À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï
Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
à á â ã ä å æ ç è é ê ë ì í î ï
ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ
```

The `console` tag is also implemented with `fancyvrb`, with adjustments for the input lines. It will not look like it, but these are 8 such inputs, with similar results to above, but now bolded.

```
¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬   ® ¯
° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿
À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï
Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
à á â ã ä å æ ç è é ê ë ì í î ï
ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ
```

We take care to render the `U+0080`–`U+00FF` characters in Sage cells. This would allow some flexibility in comments and strings employed. The following is just a test of these characters in the `input` and `output` of a `sage` element. This is not functional code.

```
  ¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬   ® ¯
° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿
À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï
Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
à á â ã ä å æ ç è é ê ë ì í î ï
ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ
```

```
  ¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬   ® ¯
° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿
À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï
Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
à á â ã ä å æ ç è é ê ë ì í î ï
ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ
```

The table below has a single column, and each cell of the table has a string of 10 characters inside a `c` element. It is meant to test if the font is monospace in this situation.

Table 22.5. Alignment Test

0123456789
9876543210
iiiiiiiiii
mmmmmmmmmm

Again, more examples and more thorough explanations can be found in the sample: `examples/fonts/fonts-and-characters.xml`. Be aware that the nature of the more advanced sample is that it will likely produce many errors when processed with `pdflatex`. Adding `-interaction batchmode` or `-interaction nonstopmode` to the `pdflatex` command-line will sometimes be less painless than acknowledging each error. The more advanced sample will perform well when processed with `xelatex`.

23 Pre-Formatted Text

In Sage, if you wanted to build a matrix, then you would use the `matrix()` constructor. Here is the matrix of second partials of $f(x, y) = x^3 + 8x^2y^3 + y^4$,

as you would enter it in Sage. Notice that SR is the ring of symbolic expressions, Symbolic Ring.

```
var('x', 'y')
J = matrix(SR, [
    [6*x + 16*y^3, 48*x*y^2],
    [48*x*y^2, 48*x^2*y + 12*y^2]
    ])
```

That accomplished, Sage will easily and naturally provide a LaTeX representation of the matrix with the command latex(J).

```
\left(\begin{array}{rr}
16 \, y^{3} + 6 \, x & -48 \, x y^{2} \\
48 \, x y^{2} & 48 \, x^{2} y + 12 \, y^{2}
\end{array}\right)
```

The pre element surrounds text that should be preserved verbatim. It is like a special kind of paragraph, and can be used almost everywhere that a paragraph can be used. The realization of preformatted text should be robust enough that it can be cut from documents and pasted without any substitutions of "fancier" Unicode characters for generic ASCII characters. Try the "minus" sign on the 48 above to see if it does not become a dash, or the single quotes on the Sage variables.

For Sage input code, the first non-whitespace character sets the left margin, since legitimate Python code has no subsequent lines outdented. For pre-formatted code, the line with the *least* whitespace leading the line will determine the left margin. If preserving indentation is important, do not mix spaces and tabs. For syntax highlighting of text representing computer programs, or parts of them, see Section 24. Examine the source of the following example to help understand this paragraph.

```
        A normal line
                An indented line
An outdented line
```

Snippets should also be robust for cut/paste operations. For example, you should not get "curly" "smart" quote marks in verbatim text: this should have "dumb" quote marks. Here are a few characters that should migrate through LaTeX to a PDF unmolested: '"----"'

If you write a very long snippet of inline code (i.e. within a <c> element) it can impinge on the right margin, since very long words will not hypenate, unless you have a dash/hypen. Such as when you use words like pneumonoultramicroscopicsilicovolcanoconiosis, parastratiosphecomyia stratiosphecomyioides, floccinaucinihilipilification, or subdermatoglyphic. For output in LaTeX we get line-breaking, and perhaps word-spacing, but we do not get hyphenation and the font is fixed-width. So not always perfect. Consider other options like <cd> or <pre> below.

An intermediate type of verbatim text can be accomplished with the <cd> tag, short for "code display." It allows for larger chunks of verbatim text to show up in the middle of a paragraph, but with some vertical space above and below, and centered between the margins. It can be

```
authored as a single line
```

or if you wish to have multiple lines

```
there is the <cline> tag
meant to model the line tag
and short for "code line"
```

and you may even

```
use a single cline
```

if you like to have your source closely model the visual look of the output.

With the `@showspaces` attribute of `<cd>` set to `all` there will be a visual indication of every space character, which is nice if indentation is critical. For example,

```
there␣is␣the␣<cline>␣tag
␣␣␣␣meant␣to␣model␣the␣line␣tag
␣␣␣␣␣␣␣␣and␣short␣for␣"code␣line"
```

and as single line

```
authored␣as␣a␣single␣line
```

that is not structured with `<cline>` elements.

The `<pre>` tag is meant for use outside of paragraphs, but is otherwise very similar. The source may also be structured as a sequence of `<cline>` as in the next example, recycling content from above.

```
If you write a very long snippet of inline code (i.e. within
a <c> element) it can impinge on the right margin, since
very long words will not hypenate, unless you have a dash/hypen.
Such as when you use words like
pneumonoultramicroscopicsilicovolcanoconiosis,
parastratiosphecomyia stratiosphecomyioides,
floccinaucinihilipilification, or subdermatoglyphic. For output
in LaTeX we get line-breaking, and perhaps word-spacing, but we
do not get hyphenation and the font is fixed-width. So not always
perfect. Consider other options like <cd> or <pre> below.
```

We use a Unicode right arrow (Unicode Character 'RIGHTWARDS ARROW', U+2192) to sometimes indicate the truncation of long lines in a text file. It is available in our usual monospace font for LaTeX/PDF, but we include a use here in order to make certain that is always the case. Here: →.

24 Program Listings (with code in the title)

Sage cells can be used for Python examples, but Sage uses a mild amount of pre-parsing, so that might not be a wise decision, especially in instructional settings. We might implement Skulpt or Brython (in-browser Python) or the Python language argument to the Sage Cell Server. To see examples of authoring Sage cells, have a look at Section 3.

In the meantime, program listings, especially with syntax highlighting, is useful all by itself. The "R" language might not be a bad stand-in for pseudo-code, as it supports assignment with a left arrow and has fairly generic procedural syntax for control structures and data structures. Or maybe Pascal would be a good choice? Here is an example of R. Note in the source that the entire block of code is wrapped in a CDATA section due to the four left angle brackets. We do not recommend this technique for isolated problem characters, but it is a life-saver for situations like the XSLT code just following.

```
n_loops <- 10
x.means <- numeric(n_loops)  # create a vector of zeros for results
```

```
for (i in 1:n_loops){
    x <- as.integer(runif(100, 1, 7))  # 1 to 6, uniformly
    x.means[i] <- mean(x)
}
x.means
```

And some self-referential XSL:

```
<xsl:template match="biblio" mode="number">
    <xsl:apply-templates select="." mode="structural-number"/>
    <xsl:text>.</xsl:text>
    <xsl:number from="references" level="any" count="biblio"/>
</xsl:template>
```

Matlab is a commercial language for mathematics, while Octave in an open source version. The `@language` values of `matlab` and `octave` are somewhat interchangeable. Following is a very slighlty edited version of an example from "50 Basic Examples for Matlab".

```
a = [0:0.5:5]; % A Matlab comment here
b = 2*a.^2 + 3*a -5;
c = 1.2*a.^2+4*a-3;
subplot(1,2,1)
plot(a,b,'-or','MarkerFaceColor','g','LineWidth',2)
xlabel('X'); ylabel('Y'); legend('Curve ','Location','NorthWest')
subplot(1,2,2)
plot(a,c,'--ok','MarkerFaceColor','c','LineWidth',2)
xlabel('X'); ylabel('Y'); legend('Curve 2','Location','NorthWest')
```

You can write made-up pseudo-code, but you might explain to a reader what your symbols all mean. This routine takes the $m \times n$ marix A to reduced row-echelon form. Note that with no language specified, there is no special formatting or use of color. Note in the source the use of escaped characters for the three less-than symbols.

```
input m, n and A
r := 0
for j := 1 to n
   i := r+1
   while i <= m and A[i,j] == 0
       i := i+1
   if i < m+1
       r := r+1
       swap rows i and r of A (row op 1)
       scale A[r,j] to a leading 1 (row op 2)
       for k := 1 to m, k <> r
           make A[k,j] zero (row op 3, employing row r)
output r and A
```

Look in the `pretext-common.xsl` file to see the strings to use to identify languages. Always all-lowercase, no symbols, no punctuation.

Note that the above examples all have slightly different widths (theser are very evident in print with the frames). As 2-D atomic objects, to place them in the narrative requires the layout features of a `sidebyside` element. Then `width` and/or `margin` attributes will influence the width of the panel.

A `program` may also be nested inside a `listing`, which behaves similar to a `figure`. You can provide a `title`, and the listing will be numbered along with tables and figures. This then makes it possible to cross-reference the listing, such as Listing 24.1. It also removes the requirement of wrapping the `program` in a `sidebyside`. For technical reasons, the three examples above will not split

across a page break in PDF output, but the placement inside a `listing` will allow splits, as you should see in at least one example following.

Listing 24.1. C Version of "Hello, World!"

```
/* Hello World program */

#include<stdio.h>

main()
{
    printf("Hello, World!");
}
```

A <program> may include line numbers.

Listing 24.2. A static Java program with line numbers

```
import javax.swing.JFrame;  //Importing class JFrame
import javax.swing.JLabel;  //Importing class JLabel
public class HelloWorld {
    public static void main(String[] args) {
        JFrame frame = new JFrame();           //Creating frame
        frame.setTitle("Hi!");                 //Setting title frame
        frame.add(new JLabel("Hello, world!"));//Adding text to frame
        frame.pack();                          //Setting size to smallest
        frame.setLocationRelativeTo(null);     //Centering frame
        frame.setVisible(true);                //Showing frame
    }
}
```

A <program> may also include highlighted lines.

Listing 24.3. A static Java program with line numbers

```
import javax.swing.JFrame;  //Importing class JFrame
import javax.swing.JLabel;  //Importing class JLabel
public class HelloWorld {
    public static void main(String[] args) {
        JFrame frame = new JFrame();           //Creating frame
        frame.setTitle("Hi!");                 //Setting title frame
        frame.add(new JLabel("Hello, world!"));//Adding text to frame
        frame.pack();                          //Setting size to smallest
        frame.setLocationRelativeTo(null);     //Centering frame
        frame.setVisible(true);                //Showing frame
    }
}
```

Although a program should have a <code> element surrounding its code, we attempt to provide one when it is missing. This next sample tests that and intentionally has no leading or trailing newline inside the program.

```
print("Hello world")
```

Sometimes it is nice to author an inline fragment of a program and have the styling of its text match that of a <program>. To do this we can use the <pf> tag ("program fragment"). It is similar to a <c> element, but will be syntax highlighted according to either a `@language` or the default language for <programs>. Here is an example in C: `printf("The cost is $%f.2", money);`.

And another that is in Python and tests problematic characters: `print("& % $ # _ { } ~ ^ \")`.

If you are discussing algorithms in the abstract (or even concretely), you can set them off like a theorem, with a number, a title and a target for cross-references. Sometimes you claim an algorithm produces something in particular, or has certain properties, such as a theoretical run time, so a proof may be included. See the discussion just preceding about (limited) options for pseudo-code.

Algorithm 24.4 Sieve of Eratosthenes. *On input of a positive integer* `n` *this algorithm will compute all the prime numbers up to, and including,* `n`*. It was named for Eratosthenes of Cyrene (ca. 276 BC–ca. 195/194 BC) by Nicomachus (ca. 60–ca. 120 CE) in Introduction to Arithmetic. (Wikipedia, 2015)*

1. *Input:* `n`
2. *Form the list of all integers from* `2` *to* `n`
3. *Set* `p = 2`
4. *While* `p < sqrt(n)`
    1. *If present, remove from the list multiples* `2p, 3p, ...`
    2. *If* `p` *is now the last element of the list, stop*
    3. *Otherwise, set* `p` *to the element of the list immediately after current* `p`
5. *Output: the remaining elements of the list*

Proof. Any element removed is a non-trivial product of two integers and hence composite. So no prime is is ever removed from the list.

Each composite number is a multiple of some prime, and since no prime is ever removed, each composite will be removed. Hence the removed elements are precisely the set of composite numbers in the list and thus the remainder are precisely the primes on the list. ■

If you are writing about system-level software, you may need to write numbers in hexadecimal or binary. Here we use a numbered, displayed equation (mathematics) and LaTeX macros such as `\texttt` for a monospace text font, and `\;` for spacing/grouping the bits of the binary number.

$$\texttt{6C2A}_{16} = \texttt{0110\;1100\;0010\;1010}_2 \tag{24.1}$$

If you use these constructions repeatedly, then some LaTeX macros might be useful. It might also be beneficial for us to add some PreTeXt markup for such numbers used in a paragraph—send us a feature request.

Theorem 24.5. *This is a spurious theorem to break up the run of consecutive* `listing` *so we might test the effect.*

And this is a spurious paragraph to prove that the theorem beforehand, and the proof following, are distinct from one another.

Proof. This is a proof that is authored "detached." It is not associated with the theorem above in a way other than simply following it.[1]

Notice that this proof has a `@ref` attribute that points to the theorem being proved. In this way, the `<proof>` element doubles as an `<xref>`. So the `@text`

[1] Footnotes once behaved badly in a detached `<proof>`.

attribute is also available to fashion how the cross-reference appears. Options are basically the same as for <xref>, though some things are not possible, such as custom text. ■

A specialized version of a program listing is an interactive command/response session at a command-line, where differing fonts are used to differentiate the system prompt, the user's commands, and the system's reaction. A console session may be used by itself inside a sidebyside, or it can be wrapped in a listing to get a number and a caption. As elsewhere, you will need to escape ampersands and angle brackets (such as if you have a command using redirection), using &, <, and > in your source.

Listing 24.6. Console Session: int and float

```
pi@raspberrypi ~/progs/chap02 $ gcc -Wall -o intAndFloat intAndFloat.c
pi@raspberrypi ~/progs/chap02 $ ./intAndFloat
The integer is 19088743 and the float is 19088.742188
pi@raspberrypi ~/progs/chap02 $
```

Here is the plain version, some layout control. We simply place a small margin on the left (at 4% width).

```
pi@raspberrypi ~/progs/chap02 $ gcc -Wall -o intAndFloat intAndFloat.c
pi@raspberrypi ~/progs/chap02 $ ./intAndFloat
The integer is 19088743 and the float is 19088.742188
pi@raspberrypi ~/progs/chap02 $
```

If your console input exceeds more than one line, you can author it across several lines and your choice of line breaks will be reflected in the rendering. You can decide to indent lines after the first one for clarity, if desired. You can also decide if your audience needs line-continuation characters or not.

Listing 24.7. Console Session: int and float (multi-line input)

```
pi@raspberrypi ~/progs/chap02 $ gcc -Wall
    -o intAndFloat intAndFloat.c
pi@raspberrypi ~/progs/chap02 $ ./intAndFloat
The integer is 19088743 and the float is 19088.742188
pi@raspberrypi ~/progs/chap02 $
```

A <console> may specify a continuation symbol, as a prefix on every line but the first.

```
>>> for x in range(0:20):
   print(x)
   print("Excellent!")
>>> for x in range(0:20):
   print(x)
   print("Excellent!")
$
$
$
```

Notice in the HTML version of the above example that when you highlight all, or a portion, of the listing for a cut-and-paste that the prompts are not included.

Now a test of some listings (and pre-formatted text) in a block that likely has a background color in an HTML version.

Checkpoint 24.8 Code in a colored(?) container. Not really much of an exercise, but it should force a colored box.

Running a program in a console:

```
$ ./program
```

```
A pre
```

The next listing is just absurdity, to check various characters from LaTeX that are otherwise employed by the code supporting consoles, and some Latin-1 characters. We test each in a prompt, input, and output. We use `(*` and `*)` as sequences used to escape embedded LaTeX commands, so we test those also.

Listing 24.9. Console Session: problematic LaTeX characters

```
A backslash \ here  A backslash \ here
A backslash \ here
A begin group { here  A begin group { here
A begin group { here
An end group { here  An end group } here
An end group } here
An open escape sequence (* here  An open escape sequence (* here
An open escape sequence (* here
An end escape sequence *) here  An end escape sequence *) here
An end escape sequence *) here
Some quotation marks ` ' " here  Some quotation marks ` ' " here
Some quotation marks ` ' " here
The rest & % $ # _ ~ ^ of LaTeX  The rest & % $ # _ ~ ^ of LaTeX
The rest & % $ # _ ~ ^ of LaTeX
Latin-1: ÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ×ØÙÚÛÜÝÞß  Latin-1: ÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ×ØÙÚÛÜÝÞß
Latin-1: ÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ×ØÙÚÛÜÝÞß
```

A program can also have `<preamble>` and `<postamble>`. They are tested here mostly just for their handling of leading/trailing whitespace. Note that the indentation of the start/end tags for the subelements is intentionally chaotic, that should not affect the indentation of the output. (However, the actual contents of the various parts of the program must all be indented to a similar degree in the source.)

```
# meta: text in code with one authored blank line before and after
# meta: in the output there should be two blank lines above and three below
# Your code for bar()
```

An interactive rendering of the same program should produce the same visible code. Both samples should end in a newline. There will be a visible difference between how this the empty line created by that newline rendered in plain HTML (the empty line is not included) and Runestone (the empty line is visible).

```
# meta: text in code with one authored blank line before and after
# meta: in the output there should be two blank lines above and three below
# Your code for bar()
```

We conclude this section with a longer example of a program listing, an assembly language program from Bob Plantz, included to test a `listing` breaking across pages in PDF output.

Listing 24.10. A longer program listing

```
@ structPass2.s
@ Allocates two structs and assigns a value to each field
```

```
@ in each struct, then displays the values.
@ Bob Plantz - 6 July 2016

@ Constants for assembler
        .include "theTag_struct.s"  @ theTag struct defs.
        .equ    y,-28           @ y struct
        .equ    x,-16           @ x struct
        .equ    locals,28       @ space for the structs

@ Constant program data
        .section .rodata
        .align  2
displayX:
        .asciz        "x fields:\n"
displayY:
        .asciz        "y fields:\n"
dispAChar:
        .asciz        "         aChar = "
dispAnInt:
        .asciz        "         anInt = "
dispOtherChar:
        .asciz        "   anotherChar = "

@ The program
        .text
        .align  2
        .global main
        .type   main, %function
main:
        stmfd   sp!, {r4, fp, lr}   @ save caller's info
        add     fp, sp, #8      @ our frame pointer
        sub     sp, sp, #locals @ for the structs

@ fill the x struct
        add     r0, fp, #x      @ address of x struct
        mov     r1, #'1
        mov     r2, #456
        mov     r3, #'2
        bl      loadStruct

@ fill the y struct
        add     r0, fp, #y      @ address of y struct
        mov     r1, #'a
        mov     r2, #123
        mov     r3, #'b
        bl      loadStruct

@ display x struct
        add     r4, fp, #x        @ address of x struct
        ldr     r0, displayXaddr
        bl      writeStr
        ldr     r0, dispACharAddr @ display aChar
        bl      writeStr
        ldrb    r0, [r4, #aChar]
        bl      putChar
        bl      newLine
        ldr     r0, dispAnIntAddr @ display anInt
        bl      writeStr
        ldr     r0, [r4, #anInt]
        bl      putDecInt
```

```
        bl      newLine
        ldr     r0, dispOtherCharAddr @ display anotherChar
        bl      writeStr
        ldrb    r0, [r4, #anotherChar]
        bl      putChar
        bl      newLine

@ display y struct
        add     r4, fp, #y        @ address of y struct
        ldr     r0, displayXaddr
        bl      writeStr
        ldr     r0, dispACharAddr @ display aChar
        bl      writeStr
        ldrb    r0, [r4, #aChar]
        bl      putChar
        bl      newLine
        ldr     r0, dispAnIntAddr @ display anInt
        bl      writeStr
        ldr     r0, [r4, #anInt]
        bl      putDecInt
        bl      newLine
        ldr     r0, dispOtherCharAddr @ display anotherChar
        bl      writeStr
        ldrb    r0, [r4, #anotherChar]
        bl      putChar
        bl      newLine

        mov     r0, #0          @ return 0;
        sub     sp, fp, #8      @ restore sp
        ldmfd   sp!, {r4, fp, pc}   @ restore and return

        .align  2
@ addresses of messages
displayXaddr:
        .word   displayX
displayYaddr:
        .word   displayY
dispACharAddr:
        .word   dispAChar
dispAnIntAddr:
        .word   dispAnInt
dispOtherCharAddr:
        .word   dispOtherChar
```

25 Units of Measure

Units of measure can be given xml treatment too with the `quantity` element. In LaTeX, the `siunitx` package is loaded to achive unit handling. Since that package only offers SI units, some other common units will be added by PreTeXt in the preamble. In HTML, the capabilities of `siunitx` are simulated, weakly. Note that at present, you should not attempt to use the `quantity` element within a math environment.

The value of gravitational constant g is $9.8\,\mathrm{m/s^2}$. Force is measured in kg $\mathrm{m/s^2}$, also known as one N. A quantity with rather ridiculous units is $23\,\mu\mathrm{ha}^{23}/\,^{\circ}\mathrm{C\,s^2}$. One Hz is the same as $1/\mathrm{s}$. You can have a unitless quantity, like 42, which may help with consistency between such numbers and units in the LaTeX

output. Some non-SI units are available, such as the absurd °F ft lb/gal. The LaTeX command `\pi` is recognized within `mag` in conversions to HTML, which is consistent with the behavior with a conversion to LaTeX, for example there are 2\pi rad in a full circle. This is a similar quantity with multiple occurences of `\pi` to test a particular template used for HTML output. It is not meant to make any sense: 21\pi45\pi234\pi890 rad.

For a full list of the allowed units and prefixes, see `pretext-units.xsl`. If you have a need for more units, they need to be added to `pretext-units.xsl` in the section that deals with units which are not part of `siunitx` by default. Note that the `mag` element should come first, followed by the `unit` element, followed by the `per` element.

26 Side-By-Side Panels

The flow of a page is almost universally top-to-bottom. But at times you would like to go *across* a page, perhaps to compare items (identical content in two different languages), or to make good use of a page real estate by grouping several small items together (e.g. images). So the `<sidebyside>` tag is strictly a layout device, though it does convey some meaning by grouping certain objects together. A variety of different objects can be put side-by-side using the `sidebyside` element. Specifically, `figure`, `image`, `tabular`, `p`, `ol`, `ul`, `dl`, `pre`, `poem`, and more. The individual components of a `<sidebyside>` are generically called **panels**.

As a layout device, the `<sidebyside>` does not allow a `<caption>`, is never numbered, and therefore cannot be cross-referenced. You may cross-reference whatever element holds the `<sidebyside>`, and many of the panels may be cross-referenced individually.

As a first example, we have two single paragraphs, laid out with different widths, and slight margins on each side. The widths have been chosen experimentally to get roughly identical heights for the two paragraphs of varying length.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin lorem diam, convallis in nulla sed, accumsan fermentum urna. Pellentesque aliquet leo elit, ut consequat nunc dapibus ac. Sed lobortis leo tincidunt, vulputate nunc at, ultricies leo. Vivamus purus diam, tristique laoreet purus eget, mollis gravida sapien. Nunc vulputate nisl ac mauris hendrerit cursus. Sed vel molestie velit. Suspendisse sem sem, elementum at vehicula id, volutpat ac mi. Nullam ullamcorper fringilla purus in accumsan. Mauris at nunc accumsan orci dictum vulputate id id augue. Suspendisse at dignissim elit, non euismod nunc. Aliquam faucibus magna ac molestie semper. Aliquam hendrerit sem sit amet metus congue tempor. Donec laoreet laoreet metus, id interdum purus mattis vulputate. Proin condimentum vitae erat varius mollis. Donec venenatis libero sed turpis pretium tempor.

Praesent rutrum scelerisque felis sit amet adipiscing. Phasellus in mollis velit. Nunc malesuada felis sit amet massa cursus, eget elementum neque viverra. Integer sagittis dictum turpis vel aliquet. Fusce ut suscipit dolor, nec tristique nisl. Aenean luctus, leo et ornare fermentum, nibh dui vulputate leo, nec tincidunt augue ipsum sed odio. Nunc non erat sollicitudin, iaculis eros consequat, dapibus eros.

26.1 Figures with Numbers Side-By-Side

Figures, or other captioned items such as tables or listings, can be placed side-by-side using the `sidebyside` element. The figures will be captioned and numbered as if they were part of the vertical flow of the document. For example, see Figure 26.4 and Figure 26.5

However, if the `<sidebyside>` is placed inside another `<figure>`, then the outer figure gets an overall caption and a "regular" number, while the captions of the interior items will be labelled as (a), (b), (c), etc; for example, see the subfigures in Figure 26.1. You can also cross-reference the subfigures individually, for example: Figure 26.1(a).

The `sidebyside` tag can have an attribute `widths` that specifies a percentage width of the page for each panel of the layout. There are automatic margins by default, and any remaining width is divided evenly to space out the panels. When the `margins` attribute is given as `auto`, or in the default case, the margins provided each equal half of the inter-panel space.

With no attributes on the `sidebyside`, each panel is the same width and there is no inter-panel space and no margin. For a `sidebyside` with a single panel, with its width specified, the panel will be centered.

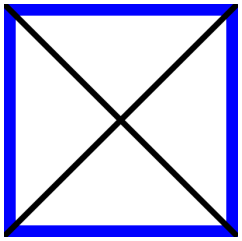


Figure 26.1(a).

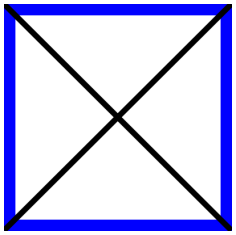


Figure 26.1(b).

Figure 26.1. Side-by-Side, with figures as children, automatic margin

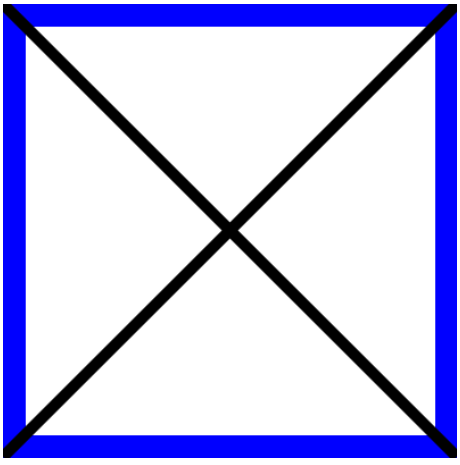


Figure 26.2(a). width=50%

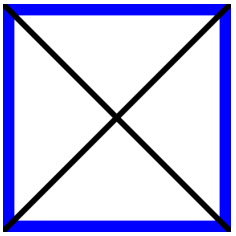


Figure 26.2(b). width=25%

Figure 26.2. Side-by-Side, with figures as children, margin set to zero

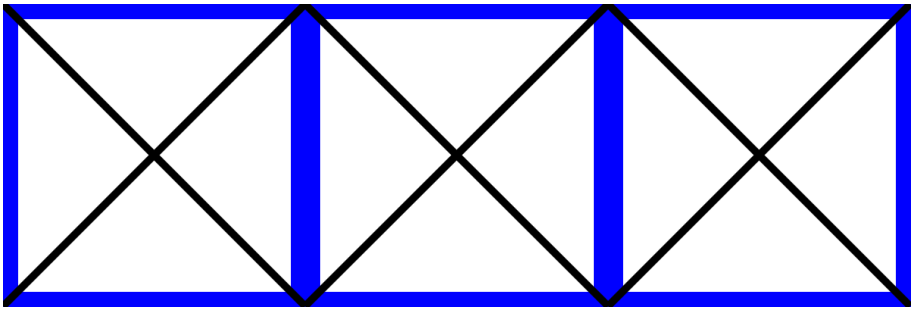


Figure 26.3(a). **Figure 26.3(b).** **Figure 26.3(c).**

Figure 26.3. Widths calculated automatically, all defaults

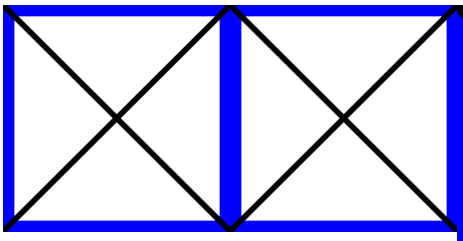


Figure 26.4. Interior figure

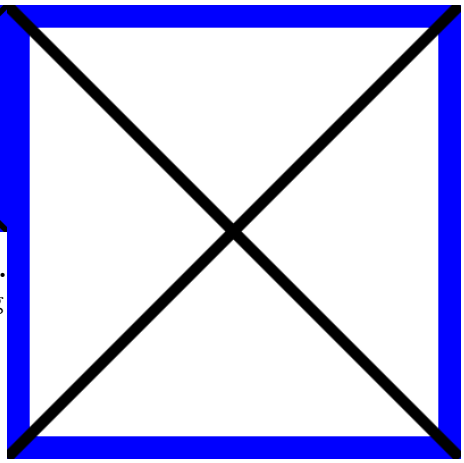
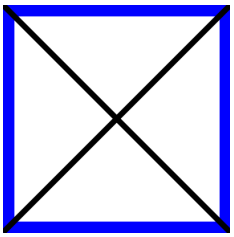


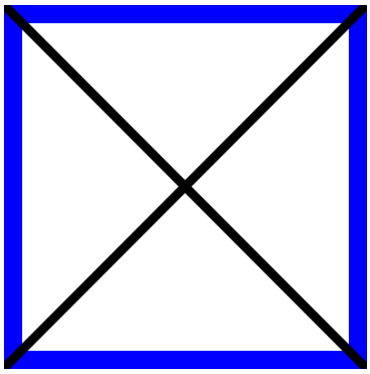
Figure 26.5. Regular numbering

Figure 26.6. Regular numbering

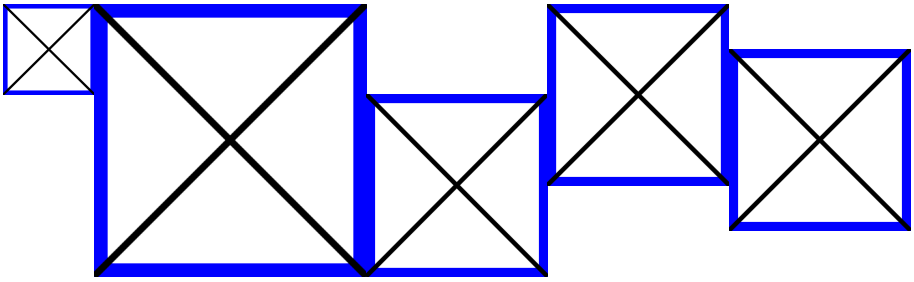
26.2 Images

We can use the `sidebyside` element to put `images` next to each other. These will illustrate a text, but with no captions or numbers, cannot be cross-referenced. This next example has `10%` margins, and the panels have widths `25%` and `40%`, leaving `15%` computed as the one inter-panel space.





Now we fine-tune with different widths (which add up to 100%). The five images have been given different vertical alignments, `top middle bottom top middle` via the `valigns` attribute.



If you want an overall caption to a group of images, but no sub-captions on your images, that is also straightforward. This example has no attributes specified. The overall `<figure>` may be cross-referenced, as Figure 26.7

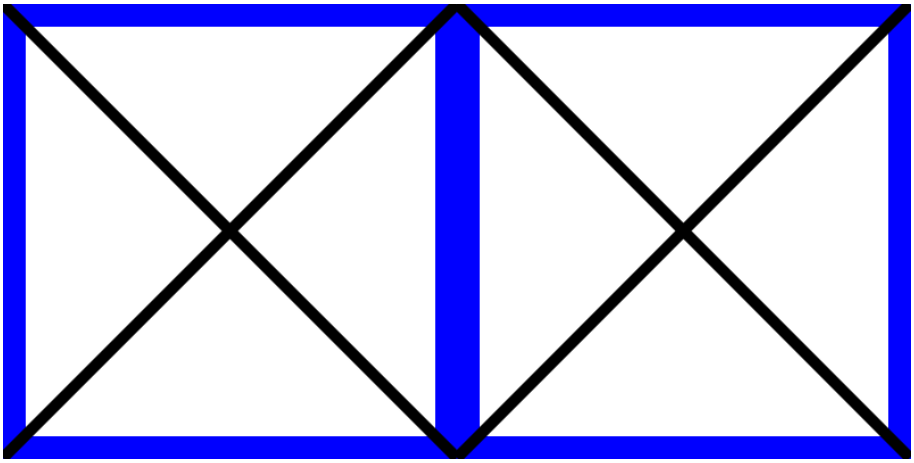


Figure 26.7. Two equally-spaced (identical) images

26.3 Common Side-By-Side Constructions

We have now seen at least one example of each of the four most common constructions involving `sidebyside`. Working from the exterior inward, we can go `figure`, `sidebyside`, `figure`, `X`, where `X` is some atomic (unnumbered) item we might use elsewhere in a PreTeXt document, the inner `figure` may be repeated with different objects `X`, and the `figure`s have captions. Each `figure` is independently optional, leading to the four combinations in this table. Note this applies to any captioned item used inside the `sidebyside`, but a `figure` is the most flexible.

Table 26.8. `sidebyside` and `figure` interactions

Outer Figure	Inner Figure	Effect
Absent	Absent	Layout only, no numbers nor captions
Absent	Present	Numbers and captions on figure(s)
Present	Absent	Number and overall caption
Present	Present	Number and overall caption, sub-numbers and captions on figure(s)

26.4 Vertical Alignment

Vertical alignment can be specified using the `valign` attribute which admits a space-separated list of `top`, `middle`, and `bottom`; the default is `top`.

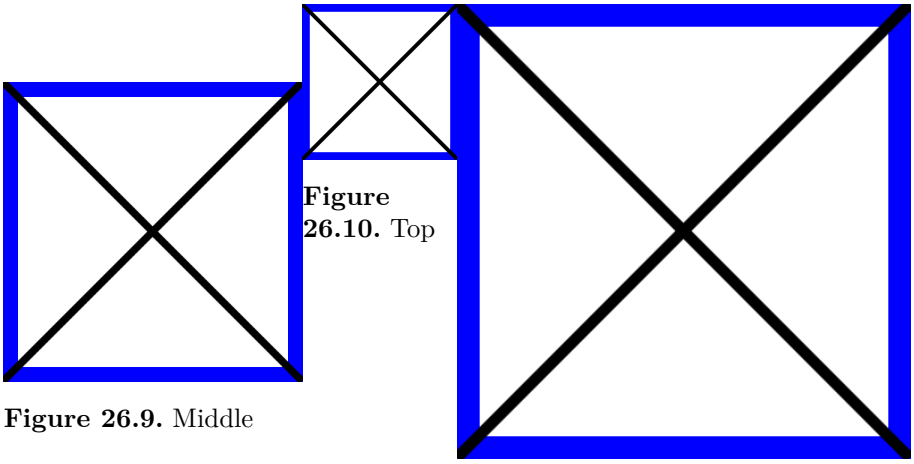
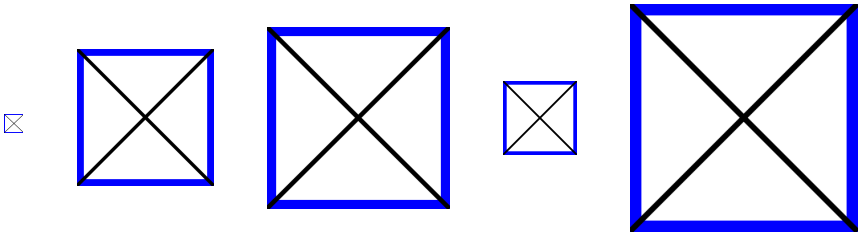


Figure 26.9. Middle

Figure 26.10. Top

Figure 26.11. Middle

The singular version of the attribute, `valign`, can provide the same alignment to each panel, here we use five different widths, but all with vertical alignment of `middle`.



26.5 Text Next to Text and Images

Text can be put next to other blocks of text using the `stack` element, which can contain multiple paragraphs using the `p` element (see Subsection 26.12). If only one paragraph is required, simply use the `p` element on its own.

here is some text here is some text

here is some text here is some text here is some text here is some text here

here is some text here is some text here is some text here is some text here

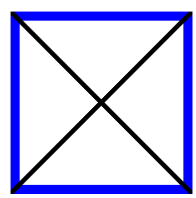
here is some text here is some text here is some text here is some text here

here is some text here is some text here is some text here is some text here

here is some text here is some text here is some text here is some text here

Similarly, text can be put next to images.

here is some text cross reference: Figure 26.12 and math: x^2



You can place text next to numbered figures, as shown below in Figure 26.12.

here is some text here is some text here is some text here is some text here is some text here is some text here is some text here is some text here is some text here is some text here is some text here is some text here is some text here is some text here is some text here is some text here is some text here is some text here is some text here is some text; cross reference: Figure 26.12 and math: x^2

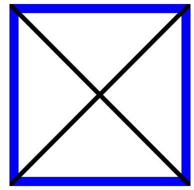
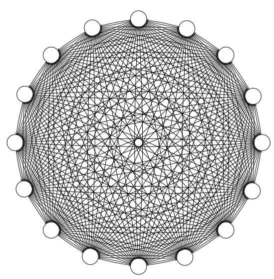


Figure 26.12. Text next to a figure

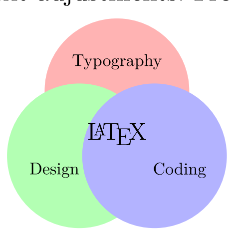
26.6 Image Formats, Side-by-Sides

Most of our demonstrations here use our square "blue cross" test image, which is provided as a PNG image. You may specify images by any of the methods described in the section on graphics, Section 10. The complete graph below is specified with no file extension, on the assumption that an SVG version exists for HTML output, and a PDF version exists for LaTeX output. The second is a JPEG image that we use elsewhere for a YouTube video, but recycle here as an image provided in that format. By default, they are aligned at their tops.





Here are two TikZ images, authored side-by-side. The first has had its geometric portions of the original scaled down to 75%, with the effect of increasing the text, relatively, so the application in a side-by-side panel with 25% width has legible text. We caption only the second panel, which has no text adjustments. From TeXample.net.



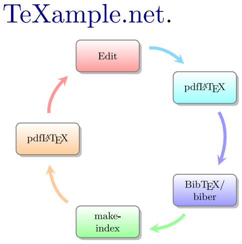


Figure 26.13. TeX Work Flow

Images by Stefan Kottwitz

- Venn Diagram
- Work Flow

26.7 Tables Side-By-Side

Tables can also be put side-by-side, as demonstrated below in Figure 26.14; naturally, subtables can be referenced as in Table 26.14(a).

Table 26.14(a). width=50%

1111	2222
aaaa	bbbb
AAAA	BBBB

Table 26.14(b). width=25%

1111	2222
aaaa	bbbb
AAAA	BBBB

Figure 26.14. Side-by-Side, with tables as children

Table 26.15(a). Table with automatic widths

1111	2222
aaaa	bbbb
AAAA	BBBB

Table 26.15(b). Table with automatic widths

1111	2222
aaaa	bbbb
AAAA	BBBB

Figure 26.15. Widths can be calculated automatically

If you put two `table` elements side-by-side without an enclosing `<figure>`, then they will use regular numbering; see Tables 26.16–26.18.

Table 26.16.

1111	2222
aaaa	bbbb
AAAA	BBBB

Table 26.17.

1111	2222
aaaa	bbbb
AAAA	BBBB

Table 26.18.

1111	2222
aaaa	bbbb
AAAA	BBBB

26.8 Tables Next to Figures

Tables and figures can go next to each other, as demonstrated in Table 26.19 and Figure 26.20, plus within an overall captioned figure, Figure 26.21.

Table 26.19. Table next to a Figure

1111	2222
aaaa	bbbb
AAAA	BBBB

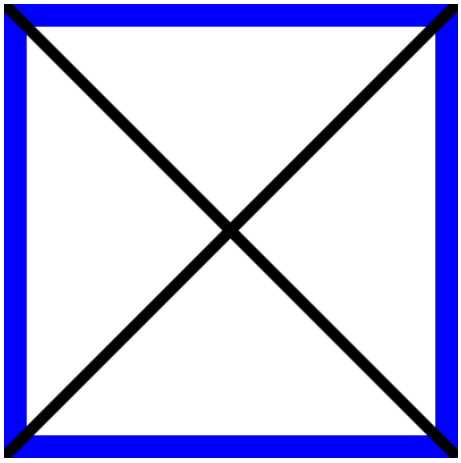


Figure 26.20. Figure next to a Table

Table 26.21(a). Table next to a Figure

1111	2222
aaaa	bbbb
AAAA	BBBB

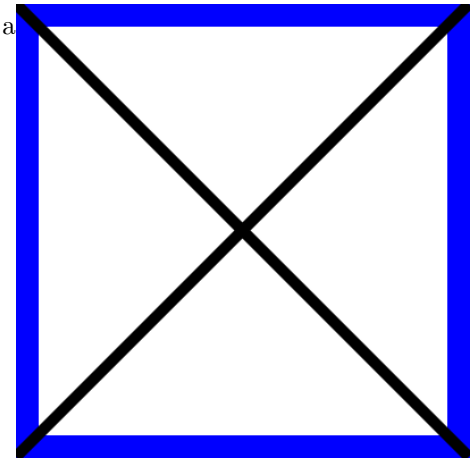


Figure 26.21(b). Figure next to a Table

Figure 26.21. Figure and Table, with overall caption, hence sub-captioned

26.9 Tables Next to Text

Tables can go next to blocks of text using the `<stack>` element (see Subsection 26.12).

Table 26.22. Table next to text

1111	2222
aaaa	bbbb
AAAA	BBBB

here is some text here is some text here is some text here is some text here is some text here

here is some text here is some text here is some text here is some text here is some text here

here is some text here is some text here is some text here is some text here is some text here

here is some text here is some text here is some text here is some text here is some text here

here is some text here is some text here is some text here is some text here is some text here

here is some text here is some text here is some text here is some text here is some text here

here is some text here is some text here is some text here is some text here is some text here

here is some text here is some text here is some text here is some text here is some text here

26.10 Tabular Next to Each Other

Four `tabular` elements inside a single `<sidebyside>` will result in no captions at all.

1111	2222
aaaa	bbbb
AAAA	BBBB
CCCC	DDDD

1111	2222
aaaa	bbbb
AAAA	BBBB

1111	2222
aaaa	bbbb
AAAA	BBBB

1111	2222
aaaa	bbbb
AAAA	BBBB

26.11 Lists in Side-by-Sides

A "regular" list normally belongs in a `p` but it can be placed unadorned into a panel of a side-by-side, as demonstrated below in Subsection 26.13. You can also put "named" lists into a panel, and then the title, introduction, conclusion, and caption will behave as expected, along with a number that might be used in a cross-reference (26.23(b)), or perhaps we might cross-reference by title, Color Shades.

List 26.23(a). Sea Life

Dr. Seuss again.

- One fish
- Two fish[1]
- Red fish
- Blue fish

List 26.23(b). Color Shades

1. Blue[2]
   (a) Light
   (b) Navy
   (c) Royal
2. Red
   (a) Maroon
   (b) Pink
   (c) Shocking

This ends our example.

Figure 26.23. Two named lists

These same two lists can individually be the panels of a `<sidebyside>`, where here vertical alignment on the `bottom` attempts to align the titles, which are placed below for panels of a `<sidebyside>`.

List 26.24. Sea Life

Dr. Seuss again.

- One fish[3]
- Two fish
- Red fish
- Blue fish

List 26.25. Color Shades

1. Blue
   (a) Light
   (b) Navy
   (c) Royal
2. Red[4]
   (a) Maroon
   (b) Pink
   (c) Shocking

This ends our example.

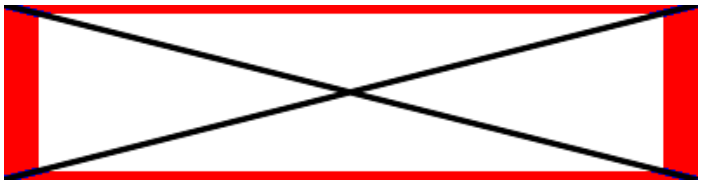
We also need to test a `sidebyside` in a list. The widths are now relative to the space given over to an indented item. Here we nest and nest and nest and nest to get a big, obvious indentation, and then include an image at 100% width and no margin. In your mind's eye, or with a ruler, check that the image spans all the way over to the right margin.

1. This is
   (a) a very

[2] in many shades
[1] Not fishes
[3] No more fishes
[4] a really nice color

i. wide

A. rectangle

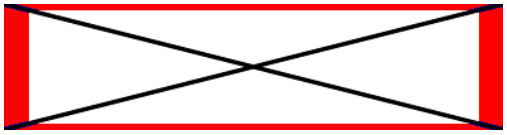


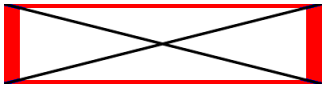
26.12 Stacking: Back to Vertical Flow

You might wish to mix disparate items within a panel, returning to a vertical flow within a panel. For example, you might want a diagram to the left and some paragraphs of commentary to the right. Or perhaps a photograph on one side and a list of bullet points to the other side. A `<stack>` is a container that can only be used to collect several items into a single panel of a `<sidebyside>`. You cannot point to it, but you can point to its contents as usual. Contents may be anything you could otherwise put into a `sidebyside` panel that does not have a `<caption>` or a `<title>`. In particular, these panels cannot be sub-numbered since the panel cannot be made into a `<figure>`.

Similar items can also be stacked, of course. Most importantly, a normal panel will accept a single paragraph. If you want several paragraphs, simply collect them in a `stack`.

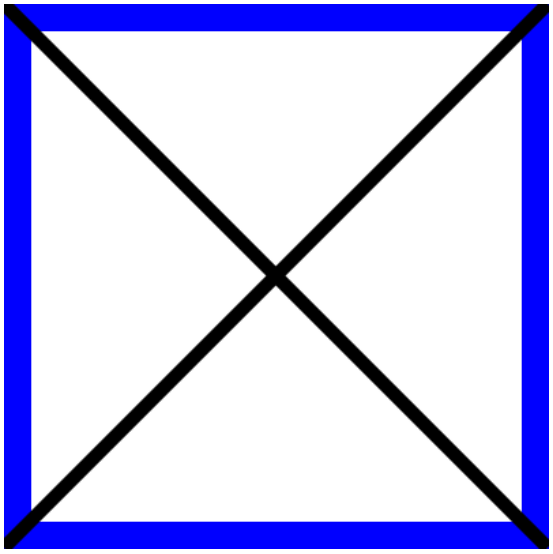
A simple sentence inside a single `<p>` as the first item in a stack.





A less simple sentence that will wrap inside the panel to make the right panel taller and allow us to experiment with sliding the left panel contents up and down, here it is placed in the `middle`.

We have an image to the left, as a regular panel (not a stack). In the right panel we stack a list of properties, followed by a descriptive paragraph. We middle-align the stack at the `bottom`, just as a demonstration (it would likely look better with `top` alignment).

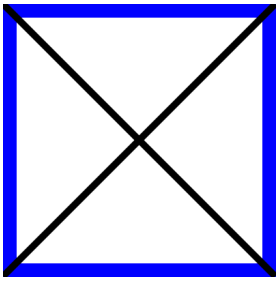


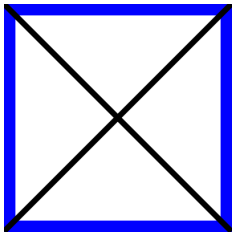
- Blue
- Square
- Geometric

The blue-ness of the border contrasts with the stark emptieness of the white interior, evoking images of blue skies and vast sandy deserts. The harsh black cross draws the viewer's attention to the exact center.

In LaTeX an image or a tabular can be used *within* a paragraph. Here we test a mixture of the three items to make sure they are properly separated in a conversion to LaTeX.

Paragraph one.





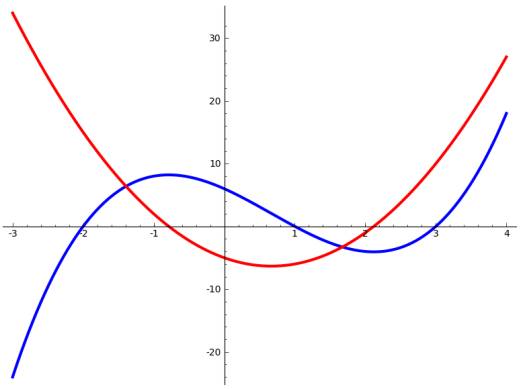
Paragraph two.

1111	2222
aaaa	bbbb
AAAA	BBBB

We imagine a `<sidebyside>` using a `<stack>` to enable constructions like a table of data in one panel, and maybe a plot with some text next to it.

In the toy example next, the list of data is rigid, so we have set the first panel width to `40%`, a value obtained experimentally to just contain the list. This allow us to set the second panel to a width of `58%`, and we use no margins. If you try to balance the heights of the two panels, this can become a bit of a zero-sum game. A wider second column means the text occupies fewer lines, but the wider image also creates a taller image, consuming more vertical space.

i	t_i	x_i	y_i
0	0.00	0.0000	0.50(
1	0.20	0.1000	0.48(
2	0.40	0.1960	0.45(
3	0.60	0.2872	0.42(
4	0.80	0.3731	0.40(
5	1.00	0.4536	0.37(
6	1.20	0.5293	0.35(
7	1.40	0.6011	0.34(
8	1.60	0.6707	0.3474
9	1.80	0.7402	0.3603
10	2.00	0.8123	0.3900



This set of values and this plot have nothing to do with each other. You'll recognize that they've been liberated from earlier in this work.

Step back and simply examine how the pieces all fit together within a `<figure>`.

Figure 26.26. Experimental results collected in a figure

The next paragraphs should start on a new page in PDF output if the standard publisher file is used.

Bully Pulpit. Remember that `<sidebyside>` has attributes that strongly influence layout. That is intentional. But to support a variety of output formats, it does not allow overly-precise control, and they be viewed as providing *hints* to an implementer of a conversion. So for example, do not expect `<sidebyside>` to function like a LaTeX `tabular` or an HTML `table`.

In particular, elements of two consecutive `<stack>` will not line up, unless perhaps you construct them identically. Consider a `<sbsgroup>` for something closer to putting items into rows.

26.13 Other Panels

Other elements may be placed within a `sidebyside` element. Pure lists first.

1. Footnotes: Fermat allusion at 2.1.
2. Examples: Mystery derivative at 4.2.
3. Definition-like: A mathematical statement with no proof 4.15.
4. Figures: An early plot, Figure 5.2.

- Footnotes: Fermat allusion at 2.1.
- Examples: Mystery derivative at 4.2.
- Definition-like: A mathematical statement with no proof 4.15.
- Figures: An early plot, Figure 5.2.

You can place *aligned* equations in paragraphs within a `sidebyside` element.

here is some text, and here is an equation that contains alignment.

$$\begin{aligned} f(x) &= x^2 + 3x + 2 \\ &= (x+2)(x+1) \end{aligned}$$

here is some text, and here is an equation that contains alignment.

$$\begin{aligned} f(x) &= x^2 + 3x + 2 \\ &= (x+2)(x+1) \end{aligned}$$

here is some text, and here is an equation that contains alignment.

$$\begin{aligned} f(x) &= x^2 + 3x + 2 \\ &= (x+2)(x+1) \end{aligned}$$

Pre-formatted text may be included by using the `pre` element. This content is horizontally-rigid, so as the author, you need to be sure to provide enough width for the panel to contain the content. It is easy to see the boundary of the panels when rendered in HTML since there is a background that fills the panel.

```
program HelloWorld;
begin
  WriteLn('Hello, world!');
end.
```

```
#include

int main()
{
    std::cout << "Hello, world!";
    return 0;
}
```

Figure 26.27. “Hello, World!” in Pascal and C++

```
graph1.txt
9
6 2
1 5
1 7
6 8
9 1
4 3
5 7
1 3
5 9
7 9
```

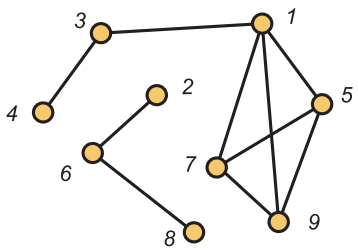


Figure 26.28. A graph defined by data (from Keller and Trotter's *Applied Combinatorics*)

26.14 Poems as Side-By-Side Panels

Poems may be panels of a side-by-side layout. Here we place some commentary alongside. See Section 29 for general information about poetry.

Fire and Ice

Some say the world will
end in fire,
Some say in ice.
From what I've tasted of
desire
I hold with those who
favor fire.
But if it had to perish
twice,
I think I know enough of
hate
To say that for destruction
ice
Is also great
And would suffice.

Robert Frost

You might have several things to say about a poem and you could use a sequence of paragraphs immediately adjacent.

This is a second paragraph of commentary.

Poems are not horizontally-rigid, but they are not perfectly horizontally-flexible either. The left copy of this next poem is in a panel roughly 2/3 the width of the page and fits there. The right copy has the first five lines and is in space about half the previous width, and you can see the lines being wrapped with obvious indentation. So you *can* constrain the width of a poem if you do not mind the additional indentation. (Recognize that this example is a bit extreme.)

Sonnet to Liberty

Not that I love thy children, whose dull eyes
See nothing save their own unlovely woe,
Whose minds know nothing, nothing care to know,
But that the roar of thy Democracies,
Thy reigns of Terror, thy great Anarchies,
Mirror my wildest passions like the sea,
And give my rage a brother! Liberty!
For this sake only do thy dissonant cries
Delight my discreet soul, else might all kings
By bloody knout or treacherous cannonades
Rob nations of their rights inviolate
And I remain unmoved-and yet, and yet,
These Christs that die upon the barricades,
God knows it I am with them, in some things.

Oscar Wilde

Sonnet to Liberty

Not that
I love thy
children,
whose dull
eyes
See nothing
save
their own
unlovely
woe,
Whose
minds
know
nothing,
nothing
care to
know,
But that
the roar
of thy
Democracies,
Thy reigns
of Terror,
thy great
Anarchies,
…

Oscar
Wilde

26.15 Side-By-Side Groups

A "side-by-side group," `<sbsgroup>`, is still in development. (Notably, subcaptions do not behave as expected.) It is a sequence of `sidebyside`, which may conceivably use the same margins, widths and vertical alignments for each horizontal run of panels. Attributes on the `sbsgroup` are global to the group's enclosed `sidebyside`, and will be used by each contained `sidebyside`. If attributes are present on an individual `sidebyside`, they override the global values. The next two examples demonstrate some of this behavior, in a limited way.

One. Two. Three.

Four. Five. Six.

Figure 26.29. Overall SBS Group

A `sidebyside` divides the width among its panels, so two figures that each need the full width, a wide plot or a panoramic photograph, say, cannot share a row. For these, a `figure` may hold a `stack` of two or more `figure`: the outer `figure` carries the number and the overall caption, and each interior `figure` is a subfigure with its own caption and a letter, as in Figure 26.30. A cross-

reference may reach the whole, or a single subfigure such as Figure 26.30(b). An interior `figure` is restricted just as a `figure` occupying a panel: no layout elements, no media elements.

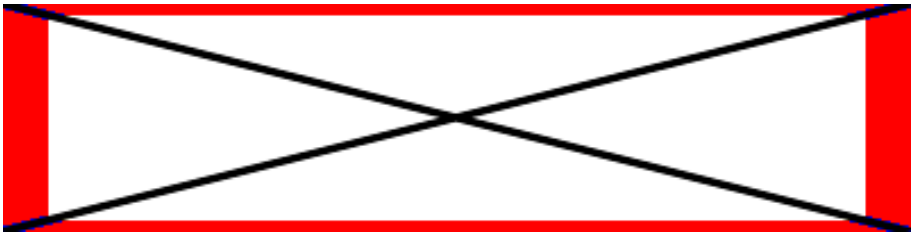


Figure 26.30(a). A wide rectangle, full width

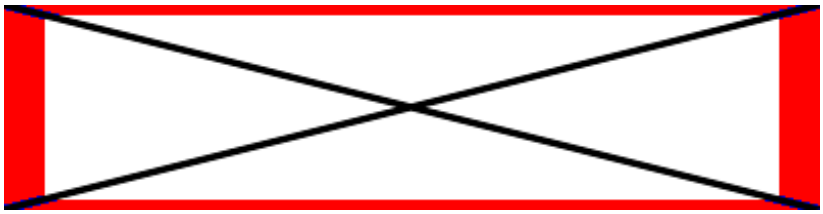


Figure 26.30(b). A wide rectangle, nine-tenths width

Figure 26.30. Two figures, stacked

A long poem, when placed into a `sidebyside` will not fit onto a physical page and will not break across pages. With a `sbsgroup` you can put each stanza (say) into its own `sidebyside` and place something (commentary) next to it. We include the `title` with the first stanza and the `author` with the last stanza. This device can also be useful to attach commentary to specific stanzas.

The Stolen Child

Where dips the rocky highland
Of Sleuth Wood in the lake,
There lies a leafy island
Where flapping herons wake
The drowsy water-rats;
There we've hid our faery vats,
Full of berries
And of reddest stolen cherries.
Come away, O human child!
To the waters and the wild
With a faery, hand in hand,
For the world's more full of
weeping than you
can understand.

Some commentary on Stanza One.

Where the wave of moonlight
glosses
The dim grey sands with light,
Far off by furthest Rosses
We foot it all the night,
Weaving olden dances,
Mingling hands and mingling
glances
Till the moon has taken flight;
To and fro we leap
And chase the frothy bubbles,
While the world is full of troubles
And is anxious in its sleep.
Come away, O human child!
To the waters and the wild
With a faery, hand in hand,
For the world's more full of
weeping than you
can understand.

Some commentary on Stanza Two.

Where the wandering water gushes
From the hills above Glen-Car,
In pools among the rushes
That scarce could bathe a star,
We seek for slumbering trout
And whispering in their ears
Give them unquiet dreams;
Leaning softly out
From ferns that drop their tears
Over the young streams.
Come away, O human child!
To the waters and the wild
With a faery, hand in hand,
For the world's more full of
weeping than you
can understand.

Some commentary on Stanza Three.

Away with us he's going,
The solemn-eyed:
He'll hear no more the lowing
Of the calves on the warm hillside
Or the kettle on the hob
Sing peace into his breast,
Or see the brown mice bob
Round and round the oatmeal-
chest.
For he comes, the human child,
To the waters and the wild
With a faery, hand in hand,
From a world more full of weeping
than he
can understand.

Some commentary on Stanza Four.

William Butler Yeats

The main rationale for `sbsgroup` is to layout a grid of items, and by placing the layout parameters on the `sbsgroup` element, the items can line up across `sidebyside` and subcaptioning can run across the whole group. So, for example, if you have images to compare by placing in a grid, then making them all the same size, or of the same aspect ratio, can help with the overall consistency.

This example has three `sidebyside`, each with four `figure` containing an identical `image`. Since the images are identical and the `width` is set to 20% they should all line up nicely with little effort. Since the default for margins is automatic, the remaining 20% of the overall width will be used for three inter-panel spaces of 5% and two margins of 2.5% each. Note the numbering of these as independent figures. We have left the captions empty for reasons of space, but you could add more information. Note that in print, a page break is allowed between any two of the `sidebyside` and cannot be suppressed.

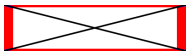


Figure 26.31.

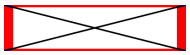


Figure 26.32.



Figure 26.33.

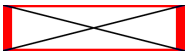


Figure 26.34.

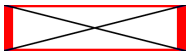


Figure 26.35.

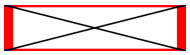


Figure 26.36.



Figure 26.37.

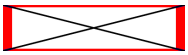


Figure 26.38.



Figure 26.39.



Figure 26.40.



Figure 26.41.



Figure 26.42.

We recycle the prior `sbsgroup` but now put it in its own overall figure. That will allow a caption for the whole group, and will cause the twelve figures to be subcaptioned. Except the subcaptioning is not implemented. Soon.

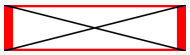


Figure 26.43(a).

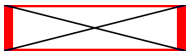


Figure 26.43(b).



Figure 26.43(c).

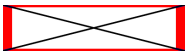


Figure 26.43(d).

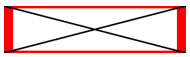


Figure 26.43(e).

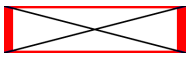


Figure 26.43(f).



Figure 26.43(g).

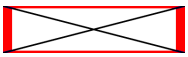


Figure 26.43(h).



Figure 26.43(i).



Figure 26.43(j).



Figure 26.43(k).



Figure 26.43(l).

Figure 26.43. Twelve images, arranged in a grid

One more test. We override the spacing and vertical alignments of the middle `sidebyside`. Note that it is easy to make a panel so skinny that even the smallest possible caption does not fit in the width.

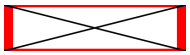


Figure 26.44.

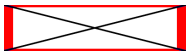


Figure 26.45.



Figure 26.46.

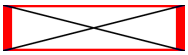


Figure 26.47.

Figure 26.48.

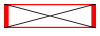


Figure 26.49.



Figure 26.50.

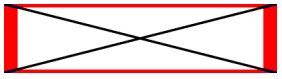


Figure 26.51.

Figure 26.52. **Figure 26.53.** **Figure 26.54.** **Figure 26.55.**

Now an `<sbsgroup>` that should start on a new page if this sample article is built to PDF using the distribution publisher file.



Figure 26.56.



Figure 26.57.



Figure 26.58.



Figure 26.59.



Figure 26.60.



Figure 26.61.



Figure 26.62.



Figure 26.63.

The following is a <sbsgroup> full of "operation" tables. Once upon a time it was rather cramped vertically in HTML output, but Andrew Scholer improved the spacing at GitHub #2387. The example is from Valerio Monti.

$$\begin{array}{cc} + & [0]_1 \\ [0]_1 & [0]_1 \end{array} \qquad \begin{array}{cc} \cdot & [0]_1 \\ [0]_1 & [0]_1 \end{array}$$

$$\begin{array}{ccc} + & [0]_2 & [1]_2 \\ [0]_2 & [0]_2 & [1]_2 \\ [1]_2 & [1]_2 & [0]_2 \end{array} \qquad \begin{array}{ccc} \cdot & [0]_2 & [1]_2 \\ [0]_2 & [0]_2 & [0]_2 \\ [1]_2 & [0]_2 & [1]_2 \end{array}$$

$$\begin{array}{cccc} + & [0]_3 & [1]_3 & [2]_3 \\ [0]_3 & [0]_3 & [1]_3 & [2]_3 \\ [1]_3 & [1]_3 & [2]_3 & [0]_3 \\ [2]_3 & [2]_3 & [0]_3 & [1]_3 \end{array} \qquad \begin{array}{cccc} \cdot & [0]_3 & [1]_3 & [2]_3 \\ [0]_3 & [0]_3 & [0]_3 & [0]_3 \\ [1]_3 & [0]_3 & [1]_3 & [2]_3 \\ [2]_3 & [0]_3 & [2]_3 & [1]_3 \end{array}$$

$$\begin{array}{ccccc} + & [0]_4 & [1]_4 & [2]_4 & [3]_4 \\ [0]_4 & [0]_4 & [1]_4 & [2]_4 & [3]_4 \\ [1]_4 & [1]_4 & [2]_4 & [3]_4 & [0]_4 \\ [2]_4 & [2]_4 & [3]_4 & [0]_4 & [1]_4 \\ [3]_4 & [3]_4 & [0]_4 & [1]_4 & [2]_4 \end{array} \qquad \begin{array}{ccccc} \cdot & [0]_4 & [1]_4 & [2]_4 & [3]_4 \\ [0]_4 & [0]_4 & [0]_4 & [0]_4 & [0]_4 \\ [1]_4 & [0]_4 & [1]_4 & [2]_4 & [3]_4 \\ [2]_4 & [0]_4 & [2]_4 & [0]_4 & [2]_4 \\ [3]_4 & [0]_4 & [3]_4 & [2]_4 & [1]_4 \end{array}$$

Figure 26.64. Tabelle delle operazioni per $1 \leq n \leq 4$

26.16 Testing a Side-By-Side First

A <sidebyside> that appears first within some other container can wreak havoc in LaTeX output. Below we have this situation twice, once in an <activity>, then in an <example>, then in a <paragraphs>.

Here is text block 1

Here is text block 2

Example 26.65.

Here is text block 1

Here is text block 2

▲

And a <sbsgroup> in similar circumstances.

Example 26.66.

Here is text block 1

Here is text block 2

Here is text block 3

Here is text block 4

▲

First Child of a Paragraphs.

A B	α β
C D	γ δ

26.17 Testing Styling of Related Elements

This subsection has non-side-by-side structures, to aid with the effects of styling decisions across the range of possibilities. First a `figure` with a `caption` holding a scaled image and a cross-reference for knowl testing: Figure 26.67.

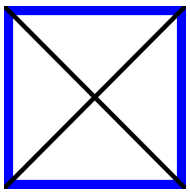


Figure 26.67. A traditional figure

27 Side-by-Side Gallery

This subsection attempts to survey all the possible items that can be placed into a `sidebyside` element, in various combinations. While intended to be exhaustive across contents, it does not test all possibilities, and is not meant to be instructive (see Section 26 for that). The layout is identical for each `sidebyside`, 5% margins, panel widths of 40% and 45%, leaving 5% for the space between the panels. The vertical alignment is left at the default, `top`.

We begin with "simpler" atomic items. If necessary, comments follow each.

Vestibulum sit amet est non lacus accumsan iaculis aliquam nec leo. Maecenas placerat consequat quam, a lobortis odio convallis vitae. Curabitur sagittis, risus non suscipit pulvinar, enim tortor posuere purus, id dignissim sapien sapien non dui. Vestibulum ultrices, enim a ornare consectetur, nisl est iaculis arcu, eget scelerisque nunc magna a nisl. Vestibulum vestibulum ante sit amet ex vulputate, eu facilisis sapien tempor.

Aliquam dui nisi, pharetra id enim vel, imperdiet laoreet risus. Nunc convallis elit eu erat imperdiet tincidunt. Sed eget augue et nunc mollis tempor. Suspendisse luctus elit non lorem scelerisque, nec lacinia lectus dictum.

Vivamus ut orci nisl. Donec eleifend ultricies tortor, a pellentesque neque dignissim in. Praesent maximus, augue eu pretium auctor, dolor quam feugiat augue, ut vulputate nunc eros vitae massa. Phasellus quis ante quis est venenatis dapibus eget luctus ipsum.

Figure 27.1. Single `<p>` (left), `<stack>` (right)

1. Blue
2. Red
3. Green
4. Purple
5. Violet
6. Brown

- Vestibulum sit amet est non lacus accumsan iaculis aliquam nec leo. Maecenas placerat consequat quam, a lobortis odio convallis vitae.

  Curabitur sagittis, risus non suscipit pulvinar, enim tortor posuere purus, id dignissim sapien sapien non dui.
- Vestibulum ultrices, enim a ornare consectetur, nisl est iaculis arcu, eget scelerisque nunc magna a nisl.

  Vestibulum vestibulum ante sit amet ex vulputate, eu facilisis sapien tempor.

Figure 27.2. An <ol> with simple items, a <ul> with items with paragraphs

```
n_loops <- 10
x.means <- numeric(n_loops)
for (i in 1:n_loops){
    x <- as.integer(runif(100, 1, 7))
    x.means[i] <- mean(x)
}
x.means
```

```
pi@rpi ~$ gcc -o intAndFloat intAndFloat.c
pi@rpi ~$ ./intAndFloat
19088743 (integer) and 19088.742188 (float)
pi@rpi ~$
```

Figure 27.3. A <program> and a <console>

Note that these two chunks of verbatim text will very likely exceed the right side of a too-skinny panel. We have severly edited these two examples from previous appearances just to fit here.

To A Friend Whose Work Has Come To Nothing

Now all the truth
is out,
Be secret and take
defeat
From any brazen
throat,
For how can you
compete,
Being honour bred,
with one
Who, were it
proved he lies,
Were neither
shamed in his own
Nor in his
neighbours' eyes?
Bred to a harder
thing
Than Triumph,
turn away
And like a laughing
string
Whereon mad
fingers play
Amid a place of
stone,
Be secret and
exult,
Because of all
things known
That is most
difficult.

William Butler Yeats

Organism	Classification
Trout	Fish
Monkey	Mammal
Crow	Bird
Crimini	Fungus
Bee	Insect

Figure 27.4. An `<poem>` and a `<tabular>`

A `tabular` can exceed the width of its panel in print, while in `HTML` it may reflow individual cells to stay within a panel, depending on their contents.

```
Vestibulum sit amet est non
    lacus accumsan iaculis
aliquam nec leo. Maecenas
placerat consequat quam, a
lobortis odio convallis
vitae.
```

```
Vestibulum sit amet est non
    lacus accumsan iaculis
aliquam nec leo. Maecenas
    placerat consequat quam,
a lobortis odio convallis
        vitae.
```

Figure 27.5. A `<pre>`, and a `<pre>` employing `<cline>`

Be aware that the lines of `pre` can spill outside of its panel without any word-wrapping. So you may need to vary panel widths or rearrange line breaks manually. Page width is a scarce resource.

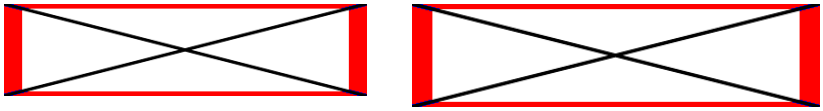


Figure 27.6. An identical `<image>`, twice

Images will scale to fill their panel's width. We provide no services to change the aspect ratio of your images, that is your responsibility to accomplish elsewhere. This rectangular image will have slightly different widths, and so will be slightly deeper in the right panel (at a 45:40 ratio). Remember, vertical alignment is at the top.

Now we turn to "captioned" items: `figure`, `table`, `listing`, and the anomalous "named list", `list`, whose future is uncertain. We test subcaptions here. Note that many different atomic items can go in a figure, and largely they will behave in a `sidebyside` much as they do when placed in a panel all by themselves (i.e. captionless).

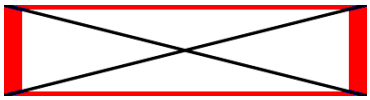


Figure 27.7(a). A Rectangular Test Image

Table 27.7(b). Classifying Organisms

Organism	Classification
Trout	Fish
Monkey	Mammal
Crow	Bird
Crimini	Fungus
Bee	Insect

Figure 27.7. A `<figure>` and a `<table>`

Listing 27.8(a). A statistical computation

```
n_loops <- 10
x.means <- numeric(n_loops)
for (i in 1:n_loops){
    x <- as.integer(runif(100, 1, 7))
    x.means[i] <- mean(x)
}
x.means
```

List 27.8(b). Colors Again

We have named list of colors.

1. Blue
2. Red
3. Green
4. Purple
5. Violet
6. Brown

That was nice.

Figure 27.8. A `<listing>` and a `<list>`

Now let's test the alignment of various items that might be placed with an image in a figure. We want asymptote links and long descriptions close to the image. The caption for figures should be aligned.

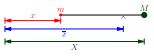


Figure 27.9(a). Asymptote Lever Demonstration

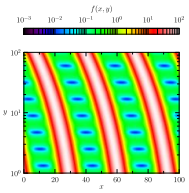


Figure 27.9(b). Asymptote Contour Plot

Figure 27.9. Two Asympote images side-by-side

28 Open Problems

Like for mathematical research. An `openproblem`, `openquestion`, or `openconjecture` is numbered as a block, sharing the overall block counter by default, though a publisher can give them a counter of their own.

Beyond a `statement`, an open problem can be followed by discussion appendages such as `status`, `discussion`, `opinion`, `suggestion`, or `context`, and, like a `project`, it may be divided into `task`s.

Open Problem 28.1 ([2]) Solve the Riemann Hypothesis[1] *and* provide a short proof of Fermat's Last Theorem.

Open Question 28.2 (Goldbach, [1, Chapter 3]) Is every even integer greater than two the sum of two primes?

Status 1. Verified by Oliveira e Silva for every even integer below 4×10^{18}. The ternary version—every odd number above five is a sum of three primes—was settled by Helfgott in 2013, but the binary question is the one our seminar is after.

Discussion 2. The circle method predicts on the order of $n/(\log n)^2$ representations of a large even n, comfortably positive; the stubborn obstruction is controlling the minor-arc contribution, where we have made no real progress.

Open Conjecture 28.3 Sum-free Sets of Squares.
This is the current focus of a project with our collaborators. Write $S_N = \{1, 4, 9, \dots, N^2\}$ for the first N perfect squares, and call a set **sum-free** when it contains no solution of $a + b = c$. We conjecture the following, listed in increasing order of difficulty.

> **(a)** There is an absolute constant $c > 0$ and, for every large N, a sum-free subset $A \subseteq S_N$ with $|A| \geq (1/2 + c)\, N$.
>
> **Status 1**. For every $N \leq 60$ the largest sum-free subset we have found has exactly the size of the odd squares, so we cannot yet exhibit a single configuration that provably beats that baseline.
>
> **Discussion 2**. The odd squares $\{1, 9, 25, \dots\}$ are already sum-free, since a sum of two of them is congruent to 2 modulo 4 while no square is; this gives density $1/2$, so the whole content of the conjecture is the gain c.

[1] Footnotes were once incomplete on open problems.

(b) The limiting density $d = \lim_{N \to \infty} \frac{1}{N} \max_{A} |A|$ exists.

Suggestion 1. One of our collaborators proposes a Fekete-style subadditivity argument to force the limit to exist, even before we can name its value.

(c) Replacing the squares by the cubes $\{1, 8, 27, \ldots\}$ yields the same limiting density d.

Even the bare existence claim of the second part is open; we would be glad of either a proof or a numerical counterexample.

29 Poetry

There is support for poems via the `poem` tag, which can contain a `title`, `author` and multiple `stanza`, each containing multiple `line`. See the source of the following poem for an example of the exact arrangement. Note how the first quote crosses two `line` elements and how this is handled in the source. There are many very flexible options for horizontal alignment and indentation. Further extensive examples, constructed by Jahrme Risner, are available in the example Humanities document.

The Charge of the Light Brigade

Half a league, half a league,
Half a league onward,
All in the valley of Death
Rode the six hundred.
“Forward, the Light Brigade!
Charge for the guns!” he said:
Into the valley of Death
Rode the six hundred.

“Forward, the Light Brigade!”
Was there a man dismay’d?
Not tho’ the soldier knew
Someone had blunder’d:
Theirs not to make reply,
Theirs not to reason why,
Theirs but to do and die:
Into the valley of Death
Rode the six hundred.

Alfred Lord Tennyson

Ken Levasseur, who teaches at UMass-Lowell, has limericks in his Applied Discrete Structures textbook. When he reported that they were unable to be the target of a cross-reference, Karl-Dieter Crisman penned the following limerick.

CS students studying in Lowell
Required their books to have soul.
Along came their teacher

Who asked for this feature:
A poem that lives in a knowl.

Karl-Dieter Crisman

And when yours truly tried to joke about poetry on GitHub CLI #182, back came:

There once was a maintainer named Rob
Who told bad jokes while on the job
While they were lame
You could say the same
Of Steven's limericks that cause you to sob

Steven Clontz

30 Literate Programming

There is support for literate programming using the `<fragment>` tag. It should always contain an `@@xml:id` or a `@@filename` attribute, and may contain `<code>` segments as well as `<fragref>` segments referencing other fragments. The final code is to be assembled starting from a fragment with a filename attribute and traversing the tree of fragment references.

For example here is the gcd algorithm in Python. Note that indentation is hard to get right at the moment.

⟨1 The GCD algorithm⟩ ≡
Root of file: `gcd.py`

```
def gcd(a, b):
    while(b):
```

⟨The key loop part 2 [203]⟩

```
return a
```

The key part is the inner part of the loop:

⟨2 The key loop part⟩ ≡

```
a, b = b, a % b
```

This double assignment changes both a and b.

31 Atomic Objects

Some PreTeXt objects are relatively indivisable and are used as components of other structures. We call them **atomic**, even if the term is not perfect. A good example is `<image>` (next, 31.1). This section is arranged according to these objects and tests the various ways they can be employed.

We frequently include some nonsense text inside short intervening paragraphs to test spacing and establish margins.

31.1 <image>

An <image> can be placed in five different ways:

1. all by itself, as a peer of <p> typically, with layout control,
2. inside a <figure>, earning a number and caption,
3. inside a <sidebyside>, with size and layout configured,
4. inside a <figure> inside a <sidebyside>, with size and layout configured, with a number and caption, and
5. inside a <figure> inside a <sidebyside> inside a <figure>, with size and layout configured, with a number and caption, but now sub-numbered ((a), (b), (c),…).

Examples of each, and more.

All by itsef, with no layout specified, so showing the default size and placement. Vivamus in congue massa. Morbi condimentum ac magna at accumsan. Vestibulum ac augue eu lorem semper gravida.



Width set at 40%, so equal margins and thus centered. Aenean faucibus augue tellus, et sollicitudin tortor finibus non. Maecenas semper dolor quis diam placerat, iaculis sollicitudin augue finibus. Vestibulum facilisis ligula lectus, ac tristique nisl aliquet non.



Asymmetric margins of 20% and 40% given, implying 40% width, equal to previous instance. Vivamus suscipit diam eget mi cursus viverra.



As a plain component of a `<sidebyside>`. Widths here are 20% and 30%, margins and gaps are automatic, default alignment on top edges. Nulla pharetra imperdiet elit, in sodales nibh blandit ultricies. Maecenas efficitur ac felis ut pharetra.





Inside a `<figure>` with no adjustments, so default behavior. Note how a `<figure>` occupies the entire width of the page, so then does the caption.



Figure 31.1. New Zealand Landscape

Inside a <figure> with asymmetric (large) margins of 30% and 60%. Quisque finibus augue sit amet facilisis fringilla. Aenean faucibus augue tellus, et sollicitudin tortor finibus non.



Figure 31.2. New Zealand Landscape

Inside figures inside a <sidebyside>. Same widths as previous <sidebyside> but alignment on bottoms of the panels, to partially align captions. Note how the captions are constrained in width by the width of the panels of the side-by-side.



Figure 31.3. NZ Landscape



Figure 31.4. New Zealand Terrascape

Identical code to previous example, but now wrapped in an overall <figure>, which has its own caption and number, leaving the interior figures to be sub-numbered. Cross-references use the full number: Figure 31.5(b).



Figure 31.5(a). NZ Landscape



Figure 31.5(b). New Zealand Terrascape

Figure 31.5. Amalgamation of Scapes

For LaTeX, in some circumstances it is desirable to print the image on the next line, but backed up by some amount. This “top-aligns” the image with a number of some sort off to the left. The following are tests for this behavior. Here is a list.

1. 
2. 

A `rotation="n"` attribute applied to a bare image will rotate the image by n°. The vertical space adjusts to accomodate the rotated image in the latex version but not in the html version.



Figure 31.6(a). `rotate="180"`



Figure 31.6(b). `rotate="15"`

Figure 31.6. Rotated Images

For pdf output destined for print, i.e. when the publication file entry `latex/@print="yes"`, a `@landscape="yes"` attribute applied to a `<figure>`, `<table>`, `<list>` or `<listing>` will cause the object to be rotated 90° and presented on its own page. Placement of the float is determined by LaTeX and multipage objects are not supported.

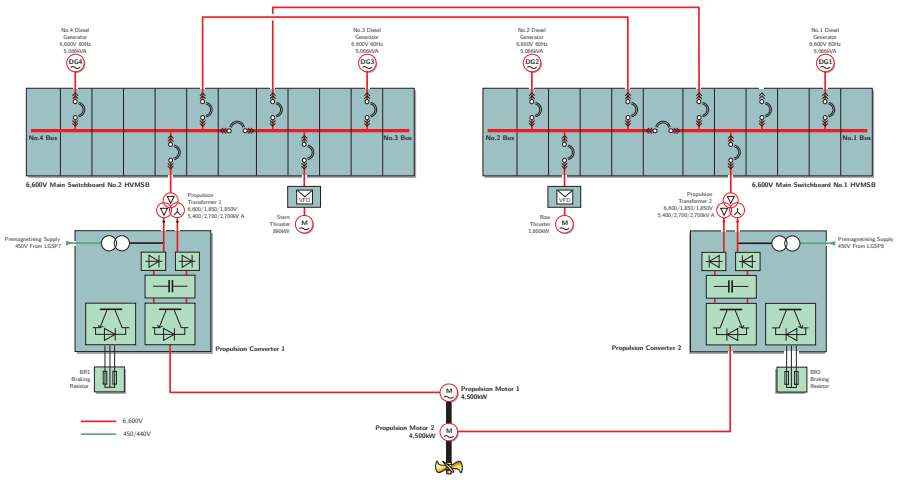


Figure 31.7. This landscape figure will be rotated so the long edge is vertical, and will appear on its own page in *print PDF* output.



Figure 31.8(a). Quack

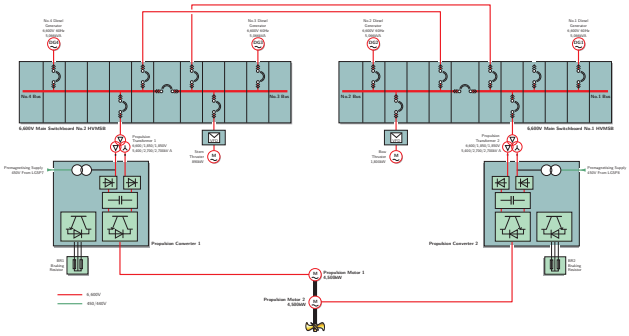


Figure 31.8(b). Propulsion System

Figure 31.8. Wide figure containing a sidebyside containing a rotated image. This will be rotated and appear on its own page in **print PDF** output.

Checkpoint 31.9.





31.1 Exercises

1.





The following images are grouped two per row.





31.2 <video>

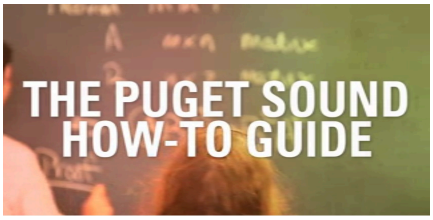
A <video> can be placed in two different ways:

1. all by itself, as a peer of <p> typically, with layout control, and
2. inside a <figure>, earning a number and caption.

Examples of each, and more.

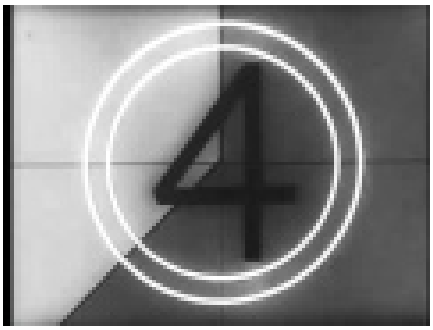
Videos can be realized in many forms, and can come from a variety of sources. See Section 20 for tests of some of that variety. Here we are testing placement within surroundings and testing the schema for location. But we do have two videos in each test, one provided as a local file and one embedded from a service.

All by itsef, with no layout specified, so showing the default size and placement. Vivamus in congue massa. Morbi condimentum ac magna at accumsan. Vestibulum ac augue eu lorem semper gravida.



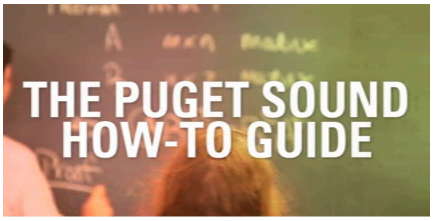
Standalone
In Context

Vestibulum facilisis ligula lectus, ac tristique nisl aliquet non. Quisque ornare felis arcu. Vivamus suscipit diam eget mi cursus viverra.



Standalone
In Context

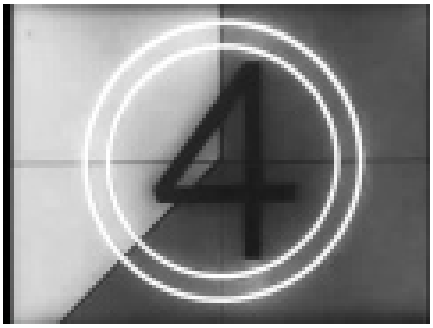
Width set at 40%, so equal margins and thus centered. Aenean faucibus augue tellus, et sollicitudin tortor finibus non. Maecenas semper dolor quis diam placerat, iaculis sollicitudin augue finibus. Vestibulum facilisis ligula lectus, ac tristique nisl aliquet non.



Standalone
In Context

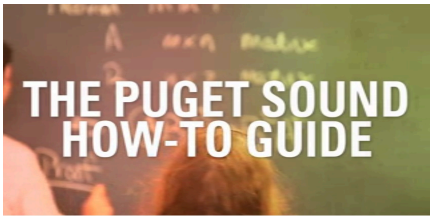
Vestibulum facilisis ligula lectus, ac tristique nisl aliquet non. Quisque ornare felis arcu. Vivamus suscipit diam eget mi cursus viverra.

This next video should start on a new page if the distribution publication file is in use.



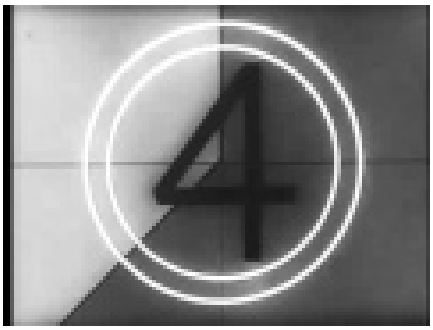
Standalone
In Context

Asymmetric margins of 20% and 40% given, implying 40% width, equal to previous instance. Vivamus suscipit diam eget mi cursus viverra.



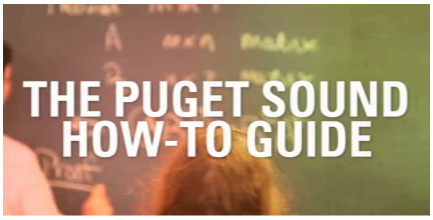
Standalone
In Context

Vestibulum facilisis ligula lectus, ac tristique nisl aliquet non. Quisque ornare felis arcu. Vivamus suscipit diam eget mi cursus viverra.



Standalone
In Context

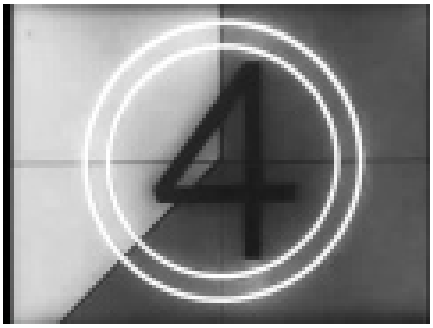
Inside a `<figure>` with no adjustments, so default behavior. Note how a `<figure>` occupies the entire width of the page, so then does the caption.



Standalone
In Context

Figure 31.10. University of Puget Sound Promotional Video

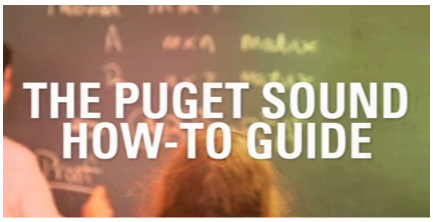
Vestibulum facilisis ligula lectus, ac tristique nisl aliquet non. Quisque ornare felis arcu. Vivamus suscipit diam eget mi cursus viverra.



Standalone
In Context

Figure 31.11. Pre-Roll Countdown

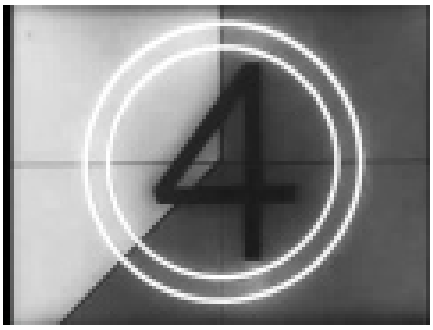
Inside a `<figure>` with asymmetric (large) margins of 30% and 60%. Quisque finibus augue sit amet facilisis fringilla. Aenean faucibus augue tellus, et sollicitudin tortor finibus non.



Standalone
In Context

Figure 31.12. University of Puget Sound Promotional Video

Vestibulum facilisis ligula lectus, ac tristique nisl aliquet non. Quisque ornare felis arcu. Vivamus suscipit diam eget mi cursus viverra.



Standalone
In Context

Figure 31.13. Pre-Roll Countdown

31.3 `<program>`, `<console>`

A `<program>` and/or `<console>` can be placed in at least six different ways:

1. all by itself, as a peer of `<p>` typically, with layout control
2. inside a `<listing>`, earning a number and caption, with layout control
3. inside a `<sidebyside>`, with size and layout configured
4. inside a `<sidebyside>`, with size and layout configured, and inside a `<figure>`
5. inside a `<sidebyside>`, with size and layout configured, with each inside a `<listing>`, earning different numbers

6. inside a <figure> inside a <sidebyside> inside a <listing>, with size and layout configured, with a number and title, but now sub-numbered ((a), (b), (c),…).

Examples of each, and more.

Programs can be realized in many forms, and can come from a variety of sources. See Section 20 for tests of some of that variety. Here we are testing placement within surroundings and testing the schema for location. But we do have two videos in each test, one provided as a local file and one embedded from a service.

All by itsef, with no layout specified, so showing the default size and placement. Vivamus in congue massa. Morbi condimentum ac magna at accumsan. Vestibulum ac augue eu lorem semper gravida.

```
n_loops <- 10
x.means <- numeric(n_loops)  # create a vector of zeros for results
for (i in 1:n_loops){
    x <- as.integer(runif(100, 1, 7))  # 1 to 6, uniformly
    x.means[i] <- mean(x)
}
x.means
```

Now a program with shorter lines, with no layout control.

```
/* Hello World program */

#include<stdio.h>

main()
{
    printf("Hello, World!");
}
```

And a <console> element, also with no layout control.

```
pi@raspberrypi ~/progs/chap02 $ gcc -o intAndFloat intAndFloat.c
pi@raspberrypi ~/progs/chap02 $ ./intAndFloat
The integer is 19088743 and the float is 19088.742188
pi@raspberrypi ~/progs/chap02 $
```

Now similar examples, but with layout control: margins and width.

A <program> with a @width attribute, so centered and with equal margins. Note how the lines word wrap due to the smaller width.

```
n_loops <- 10
x.means <- numeric(n_loops)  # create a vector of zeros for results
for (i in 1:n_loops){
    x <- as.integer(runif(100, 1, 7))  # 1 to 6, uniformly
    x.means[i] <- mean(x)
}
x.means
```

A <program> with short lines, so significant, and asymmetric margins, which experimentally do not induce any word-wrapping.

```
/* Hello World program */

#include<stdio.h>

main()
{
    printf("Hello, World!");
}
```

A longer <console>, with margins so significant the appearance is ill-advised.

```
pi@raspberrypi ~/progs/chap02 $ gcc -Wall -o intAndFloat intAndFloat.c
pi@raspberrypi ~/progs/chap02 $ ./intAndFloat
The integer is 19088743 and the float is 19088.742188
pi@raspberrypi ~/progs/chap02 $
```

Two <listing>, with <title>, and no layout control.

Listing 31.14. Hello, World! in C

```
/* Hello World program */

#include<stdio.h>

main()
{
    printf("Hello, World!");
}
```

Listing 31.15. A console session on a Raspberry Pi

```
pi@raspberrypi ~/progs/chap02 $ gcc -Wall -o intAndFloat intAndFloat.c
pi@raspberrypi ~/progs/chap02 $ ./intAndFloat
The integer is 19088743 and the float is 19088.742188
pi@raspberrypi ~/progs/chap02 $
```

Same two <listing>, but now with layout control on the <program> and <console>.

Listing 31.16. Hello, World! in C

```
/* Hello World program */

#include<stdio.h>

main()
{
    printf("Hello, World!");
}
```

Listing 31.17. A console session on a Raspberry Pi

```
pi@raspberrypi ~/progs/chap02 $ gcc -Wall -o intAndFloat intAndFloat.c
pi@raspberrypi ~/progs/chap02 $ ./intAndFloat
The integer is 19088743 and the float is 19088.742188
pi@raspberrypi ~/progs/chap02 $
```

This <sidebyside> gives each panel a 30% width. The remaining 10% is apportioned for margins and separation.

```
/* Hello World program */

#include<stdio.h>

main()
{
    printf("Hello, World!");
}
```

```
pi@raspberrypi ~/progs/chap02 $ gcc -Wall -o intAndFloat intAndFloat.c
pi@raspberrypi ~/progs/chap02 $ ./intAndFloat
The integer is 19088743 and the float is 19088.742188
pi@raspberrypi ~/progs/chap02 $
```

```
n_loops <- 100
x.means <- vector(length=n_loops)  # create a vector of zero
for (i in 1:n_loops) {
    x <- as.integer(runif(100, 1, 7))  # 1 to 6, unifo
    x.means[i] <- mean(x)
}
x.means
```

This is the same three-panel <sidebyside>, but now inside of a <figure>, earning a number and a <caption>.

```
/* Hello World program */

#include<stdio.h>

main()
{
    printf("Hello, World!");
}
```

```
pi@raspberrypi ~/progs/chap02 $ gcc -Wall -o intAndFloat intAndFloat.c
pi@raspberrypi ~/progs/chap02 $ ./intAndFloat
The integer is 19088743 and the float is 19088.742188
pi@raspberrypi ~/progs/chap02 $
```

```
n_loops <- 1000
x.means <- vector(length=n_loops)  # create a vector of zero
for (i in 1:n_loops) {
  x <- as.integer(runif(100, 1, 7))  # 1 to 6, unifo
  x.means[i] <- mean(x)
}
x.means
```

Figure 31.18. Some Code Samples

Finally, a smaller <program> and a smaller <console>, each inside a <listing>, as the two panels of a <sidebyside> with no margins, and slightly different widths (to control word-wrapping). The panels have been aligned vertically so their captions align.

Listing 31.19. Hello!

```
/* Hello World program */

#include<stdio.h>

main()
{
    printf("Hello, World!");
}
```

Listing 31.20. Raspberry Pi

```
$ gcc -Wall -o intAndFloat intAndFloat.c
$ ./intAndFloat
The integer is 19088743 and the float is 19088.742188
$
```

And again, the two-panel <sidebyside> of <listing>, but now inside a <figure> that has a number and a title. And then the <listing> are sub-numbered as (a) and (b).

Listing 31.21(a). Hello!

```
/* Hello World program */

#include<stdio.h>

main()
{
    printf("Hello, World!");
}
```

Listing 31.21(b). Raspberry Pi

```
$ gcc -Wall -o intAndFloat intAndFloat.c
$ ./intAndFloat
The integer is 19088743 and the float is 19088.742188
$
```

Figure 31.21. Two Code Listings

31.4 <tabular>

A <tabular> can be placed in six different ways:

1. all by itself, as a peer of <p> typically, with no layout control and hence with a "natural width," and centered
2. all by itself, as a peer of <p> typically, with explicit layout control,
3. inside a <table>, earning a number and title,
4. inside a <sidebyside>, with size and layout configured,
5. inside a <table> inside a <sidebyside>, with size and layout configured, with a number and title, and

6. inside a <table> inside a <sidebyside> inside a <figure>, with size and layout configured, with a number and title, but now sub-numbered ((a), (b), (c),…).

Examples of each, and more.

A <tabular> realized by LaTeX for PDF/print will normally be as wide as necessary to hold the content, without word-wrapping the content of any cell that is not explicitly authored that way. This is the most rigid of the content types we call "planar." So for PreTeXt output as LaTeX, when you explicitly constrain the width to be less than the "natural width" (including use as a panel of a <sidebyside>, or even setting margins) the table will be scaled *down* in width, which can result in an apparent font size very much smaller than that of the surrounding text. Note that we do not ever scale a tabular up to be wider with a larger font size. Note also that if there is no attempt to control the space for the table (no layout control, not in a <sidebyside>) then no scaling is attempted at all and the table may be wider than the text and protrude into the right margin. For more, see the three examples at: Table 31.26 , Table 31.27, Table 31.28. Generally, much of the commentary and testing here is about LaTeX/PDF/print. While for HTML output the cells will usually automatically word-wrap to fit in the available space, without adjusting the font size. Some might like this behavior and some might not.

Data in a table form can be placed in amongst a series of paragraphs. With no layout control, it will occupy its "natural width" and be centered.

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

The same effect can be had by specifying that the @width attribute have the value auto, but do not specify any @margins. We test multiple footnotes in a <tabular>, not included in a <table>.

State[a]	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223[b]	163,696	1850

[a] Only from the West Coast.

[b] Wow! That is as big as many countries.

In amongst a run of paragraphs (or similar) a <tabular> can be placed with layout control. For LaTeX output, this will scale the table to fit within the explicit, or implicit, width. This can result in obvious differences in the apparent font size. We first have a @width that is experimentally similar to the natural width, with asymetric margins. Then a narrow width, and a wide width, as an illustration.

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Narrow. 45% width. 20% margin left, 35% margin right.

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Wide. 97% width. 1% margin left, 2% right.

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Naturally, a `<tabular>` can be placed inside a `<table>`, earning a number and a title.

Table 31.22. Natural Width

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

A little narrower, but still centered by default.

Table 31.23. Width of 60%, automatic centering

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Very narrow, asymmetric margins.

Table 31.24. Width of 30%, 30% left margin, 40% right margin

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Wider than necessary, asymmetric margins.

Table 31.25. Width of 90%, 8% left margin, 2% right margin

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

The next table is purposely much too wide. In Table 31.26 we make no attempt to control the width, and so it will extend into the margins. In Table 31.27 we have simple added the attribute `width="auto"`. This attempt to use layout control will cause an automatic reduction in width and a smaller apparent font size. Adjusting margins providing an explicit percentage width, or placing the tabular as a panel of `<sidebyside>` will have the same effect. In Table 31.28 we have set the width explicity to 100% and so it should be identical to the automatic width case just prior.

Table 31.26. Tabular too wide, no layout control

State	Population	Area (sq. mi.)	Statehood (Year)	Capitol City	Largest City
Washington	7,614,893	71,362	1889	Olympia	Seattle
Oregon	4,217,737	98,381	1859	Salem	Portland
California	39,512,223	163,696	1850	Sacramento	Los Angeles

Table 31.27. Tabular too wide, scale to automatic width

State	Population	Area (sq. mi.)	Statehood (Year)	Capitol City	Largest City
Washington	7,614,893	71,362	1889	Olympia	Seattle
Oregon	4,217,737	98,381	1859	Salem	Portland
California	39,512,223	163,696	1850	Sacramento	Los Angeles

Table 31.28. Tabular too wide, scale to 100% width

State	Population	Area (sq. mi.)	Statehood (Year)	Capitol City	Largest City
Washington	7,614,893	71,362	1889	Olympia	Seattle
Oregon	4,217,737	98,381	1859	Salem	Portland
California	39,512,223	163,696	1850	Sacramento	Los Angeles

Now into `<sidebyside>` in various ways and with various sizes. First, two `<tabular>` as panels with widths at 60% and 30%. Note that in LaTeX/PDF/print the tabular of functional values does not need the full 30% width, so it is at its natural size and centered within its panel.

State	Population	Area (sq. mi.)
Washington	7,614,893	71,362
Oregon	4,217,737	98,381
California	39,512,223	163,696

x	f(x)
3	9
5	25

Let's do that again, but with widths experimentally set to make font sizes match (approximately).

State	Population	Area (sq. mi.)	Statehood (Year)	x	$f(x)$
Washington	7,614,893	71,362	1889	3	9.734
Oregon	4,217,737	98,381	1859	5	2.175
California	39,512,223	163,696	1850		

Same tabular, which fills roughly 80% by itself, packed into a single <sidebyside> with just a 2% gap, and no side margins.

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Natural widths, but now as a pair of tables.

Table 31.29. West Coast

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Table 31.30. Function Values

x	$f(x)$
3	9.734
5	2.175

Finally, as two individual <table>, grouped and laid out via a <sidebyside>, and collected as a <figure>. Which causes sub-numbering of the two enclosed <table>.

Table 31.31(a). West Coast

State	Population	Area (sq. mi.)	Statehood (Year)
Washington	7,614,893	71,362	1889
Oregon	4,217,737	98,381	1859
California	39,512,223	163,696	1850

Table 31.31(b). Function Values

x	$f(x)$
3	9.734
5	2.175

Figure 31.31. Geography and Mathematics

32 Advanced Numbering

This section demonstrates the numbering patterns used throughout PreTeXt. There are five subsections. Two intermediate subsections each have two subsubsections. This creates a total of seven divisions that are leaves of the document tree. In each leaf we have placed two numbered theorems, for a total of fourteen. There is no real content, this is just a demonstration.

Use values of `0` through `3` for the `numbering.theorems.level` parameter to see how these numbers change accordingly. It is easiest to compare if you use `chunk.level < 2` so the theorems all land on the same page if you are previewing in HTML.

32.1 One

A document leaf.

Theorem 32.1 First Theorem. (Cauchy) *No statement.*

Theorem 32.2 Second Theorem. (Bunyakovsky) *No statement.*

32.2 Two

Further subdivided.

32.2.1 Uno

A document leaf.

Theorem 32.3 First Theorem! (Schwarz) *No statement.*

Theorem 32.4 Second Theorem? (Inequality) *No statement.*

32.2.2 Dos

A document leaf.

Theorem 32.5 First Theorem? *No statement.*

Theorem 32.6 Second Theorem! *No statement.*

32.3 Three

A document leaf.

Theorem 32.7 First Theorem. *No statement.*

Theorem 32.8 Second Theorem. *No statement.*

32.4 Four

Further subdivided. We include two theorems as numbered items in the introduction to test their numbers, which should always be logical.

Theorem 32.9 Good Numbered Theorem One. *No statement.*

Theorem 32.10 Good Numbered Theorem Two. *No statement.*

32.4.1 Uno

A document leaf.

Theorem 32.11 First Theorem. *No statement.*

Theorem 32.12 Second Theorem. *No statement.*

32.4.2 Dos

A document leaf.

Theorem 32.13 First Theorem. *No statement.*

Theorem 32.14 Second Theorem. *No statement.*

Conclusion now. We include two theorems as numbered items in the conclusion to test their numbers, which are sometimes totally illogical and are inconsistent across output formats. To see the effect, set the level for numbering theorems to `3`. See this GitHub Issue #139 for details.

Theorem 32.15 Bad Numbered Theorem One. *No statement.*

Theorem 32.16 Bad Numbered Theorem Two. *No statement.*

32.5 Five

A document leaf.

Theorem 32.17 First Theorem. *No statement.*

Theorem 32.18 Second Theorem. *No statement.*

32.6 Theorems in This Section

We have a lot of theorems in this section, so we illustrate including an automatic list of these here. We use the `elements` attribute to limit the list to `theorem` elements, and we use the `scope` attribute to limit the list to this `section`. You can use an introductory `p` like this one, or not. The list gets no title or visual separation, so use the usual subdivision elements to make that happen. The `elements` attribute can be a space-delimited list of many different elements. This list should not include the Fundamental Theorem of Calculus, Theorem 2.1. See a slightly different example in Appendix F.

Theorem 32.1 First Theorem
Theorem 32.2 Second Theorem
Theorem 32.3 First Theorem!
Theorem 32.4 Second Theorem?
Theorem 32.5 First Theorem?
Theorem 32.6 Second Theorem!
Theorem 32.7 First Theorem
Theorem 32.8 Second Theorem
Theorem 32.9 Good Numbered Theorem One
Theorem 32.10 Good Numbered Theorem Two
Theorem 32.11 First Theorem
Theorem 32.12 Second Theorem
Theorem 32.13 First Theorem
Theorem 32.14 Second Theorem
Theorem 32.15 Bad Numbered Theorem One
Theorem 32.16 Bad Numbered Theorem Two
Theorem 32.17 First Theorem
Theorem 32.18 Second Theorem

32.7 Definitions in This Article

Here is a similar list, but for `definition` elements across the whole article. This list also serves as a regression test. 10.2 contains a PreFigure diagram whose source uses a `definition` element of its own, in the PreFigure XML namespace. An earlier version of `<list-of>` matched descendants by local name only, ignoring namespaces, and so would catch PreFigure definitions and then fail to produce numbers for them. The list below correctly contains only the PreTeXt definitions.
Definition 2.2
Definition 8.32 A title
Definition 8.39
Definition 8.40 Titled, Credited, and Cited
Definition 8.49 A Statement Closing with a Description List
Definition 12.1 Group
Definition 38.1 Definition of the Derivative

32.8 A Title with] a Right Bracket

LaTeX has trouble with brackets that end up inside optional arguments, so this subsection title is only a check on the defense against that. And now an `<exercise>` with a title that could really be a problem.

Checkpoint 32.19 A Right Brace } and a Right Bracket]. The right brace is is used as a grouping character in LaTeX so this is just a test of its behavior in titles.

Hint. A faux hint to get this exercise to migrate into a `<solutions>`.

32.9 A Title with } a Right Brace

And now a right brace in a division title.

32.10 A Title with $a]b$ a Math Right Bracket

And now a right bracket within math in a division title.

We do not test a right brace within math, since it should be escaped, as is normal LaTeX practice.

32.11 Just an Exercise

Checkpoint 32.20 An Extraneous Exercise. This exercise is here just as a test of the `<solutions>` division coming next. So it is serving a purpose, even if it is not apparent.

Hint. A hint, so this exercise looks identical in structure to the one in the previous subsection.

32.12 Solutions

This is a `<solutions>` division, which will be a peer of the other `<subsection>` in this `<section>`. The default behavior is to look to the parent division (a

<section> here) and collect all the hints, answers, and solutions from every <exercise> (and friends) inside this containing division. (There are just two, similar inline <exercise>.)

But instead of the default, we employ a @scope attribute to *define* the parent division of the exercises whose solutions will be shown. In this example we specify the <subsection> that is two back, the one which tests brackets in titles.

32.8 · A Title with] a Right Bracket

Checkpoint 32.19 A Right Brace } and a Right Bracket].

Hint. A faux hint to get this exercise to migrate into a <solutions>.

33 Customizations

33.1 Renaming Document Parts, Plus This Is A Really Long Title So That We Can Test How Well It Reacts To The Right Margin And Wraps Around To Form A Couple Of Lines, Plus How It Sits Relative To The Number Of The Subsection

"Names" for various parts of a document are determined exactly once for each language, ensuring consistency and saving you the bother of always typing them in.

However, you may want to have "Conundrum"s in your document and you have no use for any "Proposition"s. So you can repurpose the proposition tag to render a different name. Or you might have a Lab Manual and want to rename subsection as "Activity". See the docinfo portion of this sample article to see how this is done, in concert with the example below. Note that you may provide versions for different languages by specifying a @xml:lang attribute.

Conundrum 33.1 (Smith) *Aah, this is confusing!*

Important Notes. If you are renaming many parts of your document, then you may not understand the design philosophy of PreTeXt. In particular, you should not be doing a wholesale shuffle of part, chapter, section, etc. This feature is intended for very limited use and is *not considered best practice*.

This feature could also be abused to provide a comprehensive suite of translations into a language not yet supported. If so, please contact us about moving your translations into PreTeXt for the benefit of all. Thanks.

33.2 Customizing Phrases

There is a facility for providing alternate text for small or short phrases, or other components of a paragraph. Here we just provide some tests. Each is inside of a block quote to identify it clearly.

We have two auxiliary files of custom elements, so you need to adjust the publisher file to specify the second one during testing. First, a very simple string as the variation.

This is an article about alligators.

Now a string which is partially text and partially simple markup.

We like to write with *feeling*, since it is more fun.

And a mildly more complicated structure (a list) as the variable text.

Some of our favorite colors are

- Red
- Blue
- Green

A cross-reference to test, since context is critical.

See also 2.1.

The URLs used as a replacement have the `@visual` attribute which is also managed by the assembly pre-processor to provide a parenthetical for print PDF production. So this is a good test of the organization of the multiple passes employed by the pre-processor.

A URL that should have a parenthetical in the print PDF version: example.com.

34 MyOpenMath Interactive Problems

This is a *test* with two inline exercises containing MyOpenMath (MOM) problems. None of this is in the schema, and all of it is subject to change.

Checkpoint 34.1 Negative Numbers and Exponents. This is an introduction, providing articulation between the MOM problem and the contents of the text. For example, you might cross-reference a result or example given previously.

Try to evaluate $-(-100)^{-1/2}$ without using a calculator. Enter your answer as an integer or reduced fraction (no decimals). Enter DNE if the number is not real.

The value is _____

Solution.

- DNE

Checkpoint 34.2 A Statistical Test. This is an introduction, providing articulation between the MOM problem and the contents of the text. For example, you might cross-reference a result or example given previously.

You wish to determine if there is a linear correlation between the two variables at a significance level of $\alpha = 0.001$. You have the following bivariate data set.

x	y
34.3	86.5
42	45.6
35.5	46.7
42.2	48
0.7	138.9
78.6	-64.7
43.5	50.6
68.2	-45
50.2	17.4
39.1	64.9
78.3	-105.8
30.2	47.5
52.7	19.4
40.8	51
21.5	72.1
57	30.6
67	-40.2
58.3	-9.7
30.8	90.9
45.5	34.3
3.3	168.3
43	23.1
34.8	29.2
30.9	104.8
13.1	101.6
60.9	-68.9
30.9	64.2
41.4	60.3
35.3	20.8
49.8	38.7
39	59.5
31.9	92.5
18.9	113.9
53.6	17.6
59	-40.4

download link provided here: Download CSV
What is the correlation coefficient for this data set?

$r =$ _____

(report answer accurate to at least 3 decimal places)

To find the p-value for a correlation coefficient, you need to convert to a t-score: $t = r \cdot \sqrt{\frac{n-2}{1-r^2}}$ This t-score is from a t-distribution with $n-2$ degrees of freedom.

What is the p-value for this correlation coefficient?

p-value = _____

(report answer accurate to at least 4 decimal places)

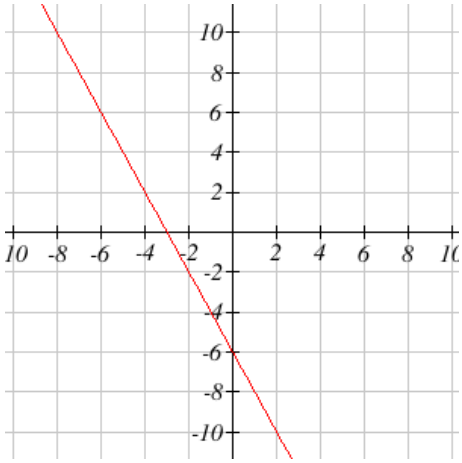
Your final conclusion is that...

1. There is sufficient sample evidence to support the claim that there is a statistically significant correlation between the two variables.
2. There is insufficient sample evidence to support the claim the there is a correlation between the two variables.

Solution.

- -0.928
- 0
- A: There is sufficient sample evidence to support the claim that there is a statistically significant correlation between the two variables.

Checkpoint 34.3 With an image. This is an introduction, providing articulation between the MOM problem and the contents of the text. For example, you might cross-reference a result or example given previously.



Find the equation of the line in the form $y = mx + b$.

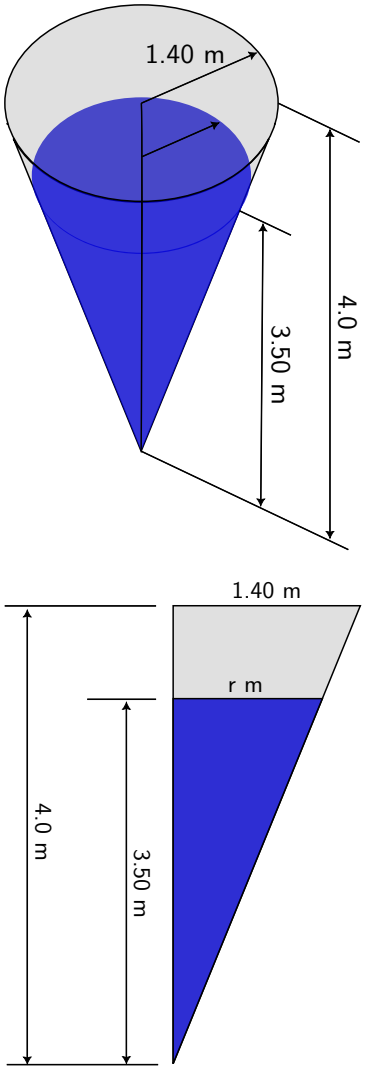
Answer: $y =$ _____

Solution.

- $y = -2x - 6$

Checkpoint 34.4 Geometry of a Cone. This problem has two figures. They come from the MOM server as SVG files. These image files are then converted to PDF for static formats like PDF, and into PNG for formats like Kindle.

The right circular conical tank shown has height 4.00 m and top radius 1.40 m. It is filled with water to a height of 3.50 m. The cross section is a right triangle with height 4.00 m and top side 1.40 m. Inside is another right triangle with height 3.50 m.



What is the volume of water? _____ m
How much water must be added to fill the tank? _____ m^3

Solution.

- 5.55
- 2.66

35 STACK Interactive Questions

These are interactive exercises provided by the STACK Assessment system. In HTML output you will need to ask for a "new" question to make each question render initially.

1. STACK integration question. An introduction to a test question, providing context within the larger document.

Find

$$\int (x-3)^6 dx$$

Answer. $\frac{(x-3)^7}{7} + c$

Solution.
We can either do this question by inspection (i.e. spot the answer) or in a more formal manner by using the substitution

$$u = (x - 3).$$

Then, since $\frac{d}{dx}u = 1$ we have

$$\int (x-3)^6 dx = \int u^6 du = \frac{u^7}{7} + c = \frac{(x-3)^7}{7} + c.$$

2. STACK multiple choice question. This is a multiple choice question where multiple answers can (in principle) be correct.

Differentiate $\sin(2 \cdot x)$ with respect to x.

(1) $2 \cdot \cos(2 \cdot x)$
(2) $\sin(2 \cdot x)$
(3) $-\frac{\cos(2 \cdot x)}{2}$
(4) $\cos(2 \cdot x) + c$
(5) None of these

Answer. (1)

3. STACK matrix input question. This is a question where the input is a matrix.

Type in $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

Answer. $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

4. STACK question with generated plot. This is a question that generates a plot that must be rendered in the question text.

What is the function plotted here:

Answer. $(x-1)^2 + 2$

5. STACK question with advanced typesetting and embedded image. This question comes from the HELM project. It features an embedded image in its solution, as well as advanced typesetting and multi-line latex.

Exercise

In triangle OAB the point P divides AB internally in the ratio $m : n$.

If $\overrightarrow{OA} = \underline{a}$ and $\overrightarrow{OB} = \underline{b}$, then find an expression for $\overrightarrow{OP}$ in terms of $\underline{a}$ and $\underline{b}$.

You might find it helpful to draw a diagram showing this triangle.

$\overrightarrow{OP} =$ _____

Answer. $\frac{(b-a)m}{n+m} + a$

Solution.
The figure below shows the triangle with the vector $\overrightarrow{OP}$ marked with a dotted line within.

From the diagram we can see $\overrightarrow{OP} = \overrightarrow{OA} + \overrightarrow{AP}$. We are told $\overrightarrow{OA} = \underline{a}$ and so we need to find $\overrightarrow{AP}$. We know that P divides $\overrightarrow{AB}$ in the ratio $m : n$ so

$$\overrightarrow{AP} = \frac{m}{m+n}\left(\overrightarrow{AB}\right),$$

as

$$\overrightarrow{AB} = \underline{b} - \underline{a}$$

we find

$$\overrightarrow{AP} = \frac{m}{m+n}(\underline{b} - \underline{a})$$

and therefore,

$$\overrightarrow{OP} = \underline{a} + \frac{m}{m+n}(\underline{b} - \underline{a}).$$

Note: You can arrive at an equivalent expression by starting from $\overrightarrow{OP} = \overrightarrow{OB} + \overrightarrow{BP}$.

36 Ancillaries

Once your content is in place, you can begin thinking about various useful derivative works. A natural example for a textbook is an "Instructor's Version". Various switches for hints, answers, and solutions to exercises would allow you to include more of these for the use of just an instructor. Here we also demonstrate the `<commentary>` element. It is similar in many ways to a `<paragraphs>` in that it can be placed within any division and must be titled. The main difference is that it is not displayed by default, so you must set the string parameter `commentary` to the value `yes`. Other distinctions are:

- Since it is elective, you need to be careful about cross-references to and from a `<commentary>`. It is highly likely that you will want to make cross-references *within* a `<commentary>` *pointing to* other portions of your text, and this is always a good idea. You will want to avoid making cross-references *to* a `<commentary>` from other parts of the text, with the exception of a cross-reference that originates *within* some `<commentary>`.
- Numbered items are prohibited within a `<commentary>`, such as a `<figure>` or a `<theorem>`. Doing so would disrupt consecutive numbering in different versions, with or without, `<commentary>` included. Numbered equations are not prohibited in the schema, but should definitely be avoided anyway.

After some nonsense text in a paragraph, there is a `<commentary>` with two paragraphs. For the online version of this sample article, we have enabled commentaries. But if you are experimenting yourself, you will want to be aware if you are enabling these or not.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam odio orci, ullamcorper eget quam et, viverra tristique magna. Integer auctor arcu a sapien

pulvinar elementum. Mauris porta, nulla id molestie dignissim, urna dolor rutrum ligula, eu elementum odio nisl sed libero. Nulla nec libero sem.

Sed justo ex, efficitur dictum risus nec, eleifend consequat nibh. Proin rutrum mi id metus viverra blandit. In vel ligula a nibh aliquam pellentesque. Duis placerat purus et ligula sollicitudin, sodales consectetur ante viverra.

Mauris fringilla nulla arcu, sagittis ultrices quam malesuada eleifend. Proin tristique elit eu bibendum tincidunt. Donec commodo lorem in magna egestas, vitae malesuada velit ornare. Pellentesque finibus neque in venenatis tristique. In id blandit est, in euismod urna. Donec commodo sagittis ligula, in venenatis nulla porttitor in. Donec nec tortor sit amet felis posuere ultricies. Suspendisse euismod quis ex eu placerat.

37 Worksheets

This is a section full of worksheets. Each is a division of its own, via the `<worksheet>` element. This is an optional `<introduction>` to the current `<section>`. In practice you might want to rip out all the worksheets of an entire book and bundle them up as an "activity book."

If you make PDF output you will notice an increased amount of control over layout. Also, if the publication file elects LaTeX draft mode, then there will be visual indicators of prescribed whitespace.

37.1 A Geometric Prelude (without authored pages)

Objectives 37.1.

- Practice visualizing vector addition
- Use vectors without explicit coordinates

This two-page worksheet was generously donated to the sample article by Dave Rosoff at a CuratedCourses workshop in August 2018. It has the default (skinny) margins.

It was known to Euclid, and probably earlier, that the midpoints of the sides of any quadrilateral all lie in the same plane (even if the vertices of the quadrilateral do not). In fact, these midpoints are the vertices of a parallelogram, as pictured in Figure 37.1.

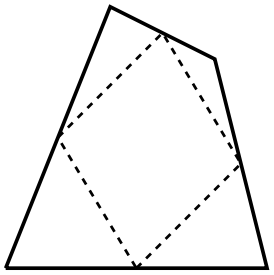


Figure 37.1. The midpoints of the sides of a quadrilateral are the vertices of a parallelogram.

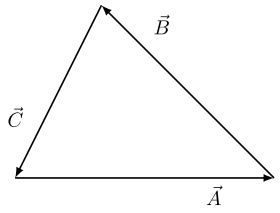


Figure 37.2. The sides of a triangle presented as vectors.

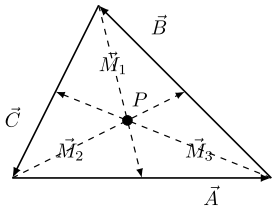


Figure 37.3. The medians of the triangle are $\vec{M}_1$, $\vec{M}_2$, and $\vec{M}_3$.

In this exercise, we'll use vectors to show that the medians of any triangle (Figure 37.2) intersect at a point. Recall that medians are the lines connecting the vertices of the triangle to the midpoints of their opposite edges, as in the figure. We'll do this in a few steps.

1. What is the value of $\vec{A} + \vec{B} + \vec{C}$?

Figure 37.3 from the previous page is reproduced for your convenience.

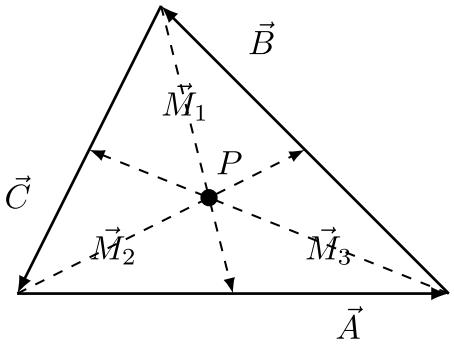


Figure 37.4. The medians of the triangle are $\vec{M}_1$, $\vec{M}_2$, and $\vec{M}_3$.

2. Show that $\vec{M}_1 + \vec{M}_2 + \vec{M}_3 = 0$.

Hint. Use Worksheet Exercise 37.1.1.

3. To show that the point P exists (as the common intersection of the $\vec{M}_i$), show that

$$\vec{A} + \frac{2}{3}\vec{M}_3 = \frac{2}{3}\vec{M}_2 = \quad .$$

4. If you have time, try to devise a vector proof of Euclid's result presented at the beginning of the workshop. Recall that a **parallelogram** is a four-sided polygon whose opposite sides are parallel.

It's possible to do interesting things with vector arithmetic in a coordinate-free way: we didn't specify an origin, or any entries of any vectors in the examples.

37.2 A Geometric Prelude (with authored pages)

Objectives 37.2.

- Practice visualizing vector addition
- Use vectors without explicit coordinates

This two-page worksheet was generously donated to the sample article by Dave Rosoff at a CuratedCourses workshop in August 2018. It has the default (skinny) margins.

It was known to Euclid, and probably earlier, that the midpoints of the sides of any quadrilateral all lie in the same plane (even if the vertices of the quadrilateral do not). In fact, these midpoints are the vertices of a parallelogram, as pictured in Figure 37.1.

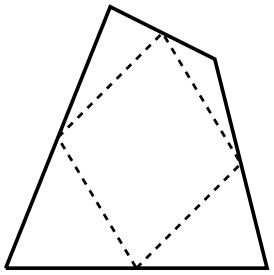


Figure 37.5. The midpoints of the sides of a quadrilateral are the vertices of a parallelogram.

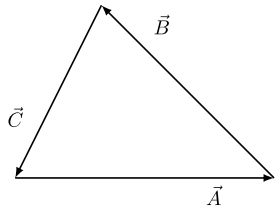


Figure 37.6. The sides of a triangle presented as vectors.

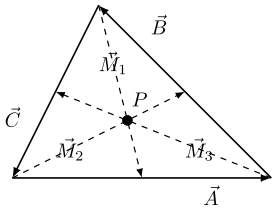


Figure 37.7. The medians of the triangle are $\vec{M}_1$, $\vec{M}_2$, and $\vec{M}_3$.

In this exercise, we'll use vectors to show that the medians of any triangle (Figure 37.2) intersect at a point. Recall that medians are the lines connecting the vertices of the triangle to the midpoints of their opposite edges, as in the figure. We'll do this in a few steps.

1. What is the value of $\vec{A} + \vec{B} + \vec{C}$?

Figure 37.3 from the previous page is reproduced for your convenience.

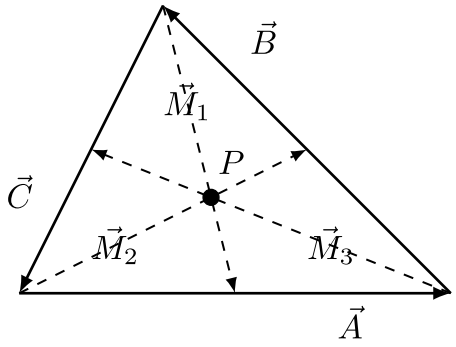


Figure 37.8. The medians of the triangle are $\vec{M}_1$, $\vec{M}_2$, and $\vec{M}_3$.

2. Show that $\vec{M}_1 + \vec{M}_2 + \vec{M}_3 = 0$.

Hint. Use Worksheet Exercise 37.1.1.

3. To show that the point P exists (as the common intersection of the $\vec{M}_i$), show that

$$\vec{A} + \frac{2}{3}\vec{M}_3 = \frac{2}{3}\vec{M}_2 = \quad .$$

4. If you have time, try to devise a vector proof of Euclid's result presented at the beginning of the workshop. Recall that a **parallelogram** is a four-sided polygon whose opposite sides are parallel.

It's possible to do interesting things with vector arithmetic in a coordinate-free way: we didn't specify an origin, or any entries of any vectors in the examples.

37.3 Networks Worksheet (no authored pages)

Basic laws for electrical circuits. This two-page worksheet was generously donated to the sample article by Virgil Pierce at a CuratedCourses workshop in August 2018. It has default (skinny) left and right margins, but we have specified longer top and bottom margins, with the top being the larger of the two.

Theorem 37.9 Ohms Law. The current through a resistor is proportional to the ratio of the *Voltage* to the *Resistance*

$$I = \frac{V}{R}$$

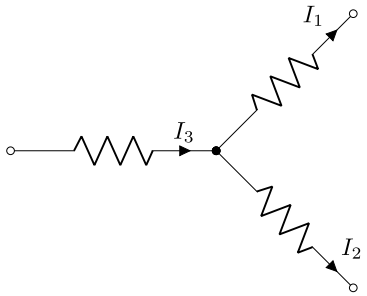
Or for our purposes

$$IR = V$$

Theorem 37.10 Kirchoffs Current Law. The sum of the currents in a network meeting at a point is zero.

$$\sum_{k=1}^{n} I_k = 0$$

Example 37.11 Kirchoff's Current Law. For the circuit below $I_1 + I_2 = I_3$.



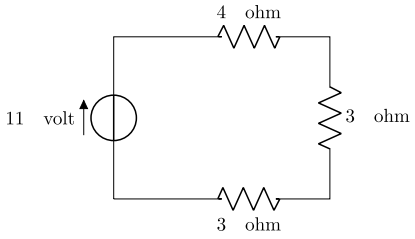
▲

Theorem 37.12 Kirchoffs Voltage Law. The sum of the voltages around any closed circuit (or subcircuit) is zero.

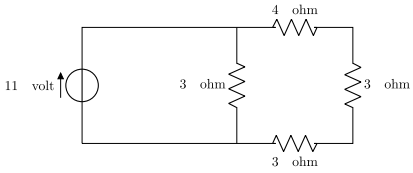
$$\sum_{k=1}^{n} V_k = 0$$

Kirchoffs Current Law and Kirkoffs Voltage Law combined with Ohms Law gives for any circuit of resistors and sources a linear system that may (or may not) determine the currents.

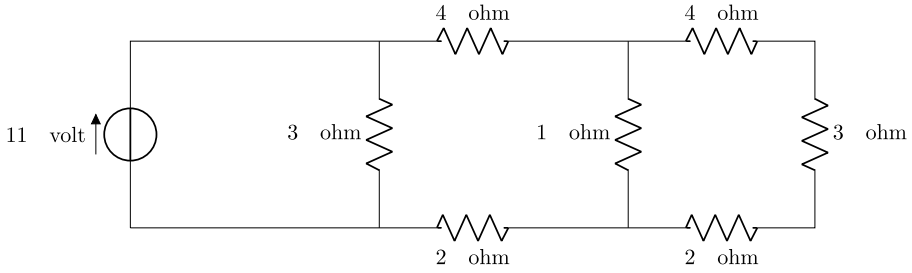
1. For the simple network pictured, calculuate the amperage in each part of the network by setting up a system of linear equations for the amperages.



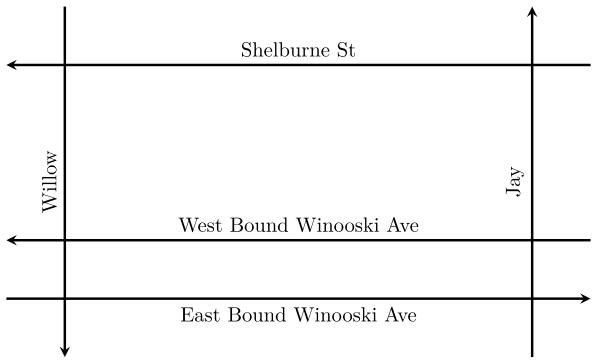
2. Compare it with a parallel circuit network. Calculate the amperage in each part of the network by setting up a system of linear equations for the amperages.



3. Now for a more complicated network. Calculate the amperage in each part of the network by setting up a system of linear equations for the amperages.



4. Now generalize these ideas to a context outside of electrical circuits. Consider the network of streets given in the diagram (with one-way directions as indicated).



A traffic engineer counts the hourly flow of cars into and out of this network at the entrances. They get (EB = East Bound; WB = West Bound):

Table 37.13. Estimated hourly traffic flow for the road network

	EB Winooski	**WB Winooski**	**Shelburne St**	**Willow**	**Jay**
into	50	400	0	10	50
out of	55	390	20	15	30

Use a variable for each segment inside of the network and set up a system of linear equations restricting the flow. Solve the system. Note that you should not get a unique solution as traffic should be able to flow through the network in various ways.

37.4 Networks Worksheet (with authored pages)

Basic laws for electrical circuits. This two-page worksheet was generously donated to the sample article by Virgil Pierce at a CuratedCourses workshop in August 2018. It has default (skinny) left and right margins, but we have specified longer top and bottom margins, with the top being the larger of the two.

Theorem 37.14 Ohms Law. The current through a resistor is proportional to the ratio of the *Voltage* to the *Resistance*

$$I = \frac{V}{R}$$

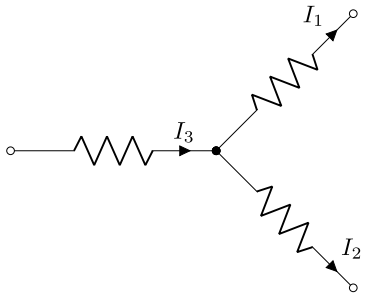
Or for our purposes

$$IR = V$$

Theorem 37.15 Kirchoffs Current Law. The sum of the currents in a network meeting at a point is zero.

$$\sum_{k=1}^{n} I_k = 0$$

Example 37.16 Kirchoff's Current Law. For the circuit below $I_1 + I_2 = I_3$.



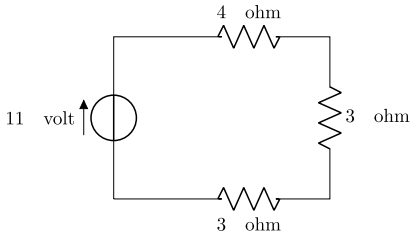
▲

Theorem 37.17 Kirchoffs Voltage Law. The sum of the voltages around any closed circuit (or subcircuit) is zero.

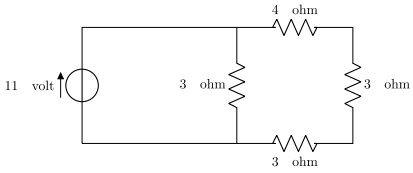
$$\sum_{k=1}^{n} V_k = 0$$

Kirchoffs Current Law and Kirkoffs Voltage Law combined with Ohms Law gives for any circuit of resistors and sources a linear system that may (or may not) determine the currents.

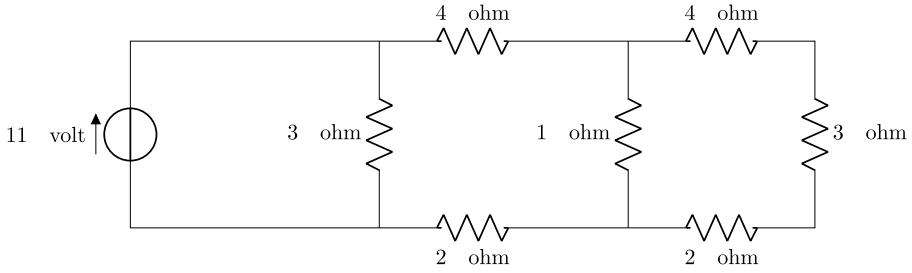
1. For the simple network pictured, calculuate the amperage in each part of the network by setting up a system of linear equations for the amperages.



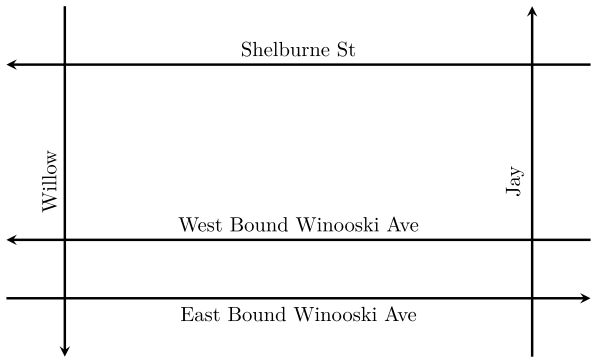
2. Compare it with a parallel circuit network. Calculate the amperage in each part of the network by setting up a system of linear equations for the amperages.



3. Now for a more complicated network. Calculate the amperage in each part of the network by setting up a system of linear equations for the amperages.



4. Now generalize these ideas to a context outside of electrical circuits. Consider the network of streets given in the diagram (with one-way directions as indicated).



A traffic engineer counts the hourly flow of cars into and out of this network at the entrances. They get (EB = East Bound; WB = West Bound):

Table 37.18. Estimated hourly traffic flow for the road network

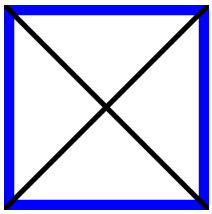
	EB Winooski	**WB Winooski**	**Shelburne St**	**Willow**	**Jay**
into	50	400	0	10	50
out of	55	390	20	15	30

Use a variable for each segment inside of the network and set up a system of linear equations restricting the flow. Solve the system. Note that you should not get a unique solution as traffic should be able to flow through the network in various ways.

37.5 Worksheet

This is a mock one-page worksheet for testing purposes. We have specified an overall margin just slightly less than the default.

1. This `<exercise>` is a panel of a `<sidebyside>`, and the image below is not a panel. So it is sized by its own `@width` of 50%, relative to the available width where it sits.



2. Integer sagittis dictum turpis vel aliquet. Fusce ut suscipit dolor, nec tristique nisl. Aenean luctus, leo et ornare fermentum, nibh dui vulputate leo, nec tincidunt augue ipsum sed odio. Nunc non erat sollicitudin, iaculis eros consequat, dapibus eros.

Hint. This is a hint. You can hide it in the print preview before printing.

Answer. Here is an answer. Also hidable.

Solution. This is a rather long solution. When you hide or show it in the print preview, the workspace should be adjusted (the page will reload).

Integer sagittis dictum turpis vel aliquet. Fusce ut suscipit dolor, nec tristique nisl. Aenean luctus, leo et ornare fermentum, nibh dui vulputate leo, nec tincidunt augue ipsum sed odio. Nunc non erat sollicitudin, iaculis eros consequat, dapibus eros.

A two-line paragraph interspersed to check on spacing, breaks and all that.

3. A full-width exercise. Praesent rutrum scelerisque felis sit amet adipiscing. Phasellus in mollis velit. Nunc malesuada felis sit amet massa cursus, eget elementum neque viverra.

Integer sagittis dictum turpis vel aliquet. Fusce ut suscipit dolor, nec tristique nisl. Aenean luctus, leo et ornare fermentum, nibh dui vulputate leo, nec tincidunt augue ipsum sed odio. Nunc non erat sollicitudin, iaculis eros consequat, dapibus eros.

Solution. This is a sample solution for a full-width exercise. Integer sagittis dictum turpis vel aliquet. Fusce ut suscipit dolor, nec tristique nisl. Aenean luctus, leo et ornare fermentum, nibh dui vulputate leo, nec tincidunt augue ipsum sed odio. Nunc non erat sollicitudin, iaculis eros consequat, dapibus eros.

Another two-line paragraph interspersed to check on spacing, breaks and all that.

4. Praesent rutrum scelerisque felis sit amet adipiscing. Phasellus in mollis velit. Nunc malesuada felis sit amet massa cursus, eget elementum neque viverra.

5. Integer sagittis dictum turpis vel aliquet. Fusce ut suscipit dolor, nec tristique nisl. Aenean luctus, leo et ornare fermentum, nibh dui vulputate leo, nec tincidunt augue ipsum sed odio. Nunc non erat sollicitudin, iaculis eros consequat, dapibus eros.

6. Praesent rutrum scelerisque felis sit amet adipiscing. Phasellus in mollis velit. Nunc malesuada felis sit amet massa cursus, eget elementum neque viverra.

Activity 37.1 A Mock Activity. The problem, as we see it.

Hint. A worksheet could have hints, no? But no spacing. Note row below has widths set to balance the heights.

7. Praesent rutrum scelerisque felis sit amet adipiscing. Phasellus in mollis velit. Nunc malesuada felis sit amet massa cursus, eget elementum neque viverra.

8. Integer sagittis dictum turpis vel aliquet. Fusce ut suscipit dolor, nec tristique nisl. Aenean luctus, leo et ornare fermentum, nibh dui vulputate leo, nec tincidunt augue ipsum sed odio. Nunc non erat sollicitudin, iaculis eros consequat, dapibus eros.

9. Praesent rutrum scelerisque felis sit amet adipiscing. Phasellus in mollis velit. Nunc malesuada felis sit amet massa cursus, eget elementum neque viverra.

37.6 Dot products and projection

1. Let $\vec{v}_1 = (-4, 1)$, $\vec{v}_2 = (2, 2)$, $\vec{v}_3 = (1, 2, 3)$, $\vec{v}_4 = (-2, 1, 0)$. Find the values of the following expressions:

(a) $\vec{v}_1 \cdot \vec{v}_2 =$

(b) $\vec{v}_3 \cdot \vec{v}_4 =$

(c) $\|\vec{v}_1\| =$

(d) $\|\vec{v}_4\| =$

(e) Based on Task (a) and Task (b), are any of these vectors perpendicular to each other? _____

2. The vectors $\vec{a} = (3, 9)$ and $\vec{b} = (4, 2)$ are pictured below. Derive the formula for projection on a line and use it to find the projection of $\vec{a}$ on the line spanned by $\vec{u}$. Also compute the length of the residual vector.[1]

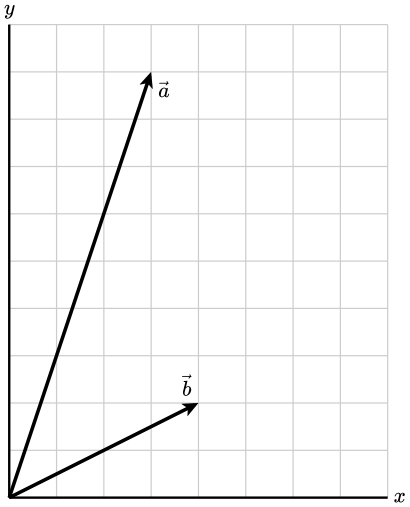


Figure 37.19. Two vectors $\vec{a}$ and $\vec{b}$.

[1] Here is a footnote.

3. Consider the vector equation

$$m \begin{bmatrix} 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 3 \\ 7 \end{bmatrix}.$$

(a) Check that there is no solution m that makes the equation true.

(b) Use projection to find the best approximation $\hat{m}$.

(c) Compute $\hat{m} \begin{bmatrix} 2 \\ 5 \end{bmatrix}$.

(d) Compute the residual vector.

(e) Compute the length of the residual vector and explain what it means.

4. Consider the system of equations

$$3t = 5$$
$$2t = 9.$$

(a) Write the system in vector form.[2]

(b) Find the best estimate, $\hat{t}$, of t using projection, as in Task (b) of Worksheet Exercise 3.

(c) Compute the length of the residual vector, and compare with Task 3.e.

[2] Another footnote

37.7 Worksheet

Activity 37.2. Just a simple activity here.

Activity 37.3. Here is a second activity.

37.8 Worksheet

Activity 37.4. This is going to be an activity with tasks.

(a) Here is the first task.

Solution. Here is a solution for the first task. It should get indented the same amount as the task.

(b) Here is the second task.

(c) Here is the third task.

This is a conclusion that comes after the last task.

37.9 Worksheet

This is a mock worksheet for testing the use of `@workspace` on an `<exercise>` in a worksheet.
Do things to the following.

1. Apple

2. Banana

3. Cherry

4. Durian

5. Elderberry (with workspace override)

6. Fig

7. Guava

8. Habanero

9. I can't think of an "I" fruit.

10. Jackfruit

Now, do things to the following, again.

11. Apple

12. Banana

13. Cherry

14. Durian

15. Elderberry (with workspace override)

16. Fig

17. Guava

18. Habanero

19. I can't think of an "I" fruit.

20. Jackfruit

37.10 Worksheet with an exercise with tasks and subtasks

Page breaking between tasks in a single exercise can be tricky. This is a test for that.

1. Here are multiple tasks with just enough workspace to cause a page break after the third one.

(a) This task has two short subtasks that should be indented under it.

(i) The first subtask.

(ii) The second subtask.

Conclusion for the first task.

(b) The second task.

(c) The third task.

(d) The fourth task.

Conclusion for the entire exercise.

2. An exercise without any tasks but some workspace.

37.11 Worksheet with <paragraphs>

This is a test of the <paragraphs> element within a worksheet. It should be able to contain exercises that have workspace, and we should be able to insert page breaks even inside of paragraphs.

Warm up.

1. This is a warm-up exercise. It should have a small workspace and be on the first page of the worksheet.

More exercises.

2. This is a second exercise. It should have a larger workspace and be on the second page of the worksheet.

3. This is a third exercise with some tasks. The workspace should be such that there will be a page break between tasks.

(a) Do the first thing.

(b) Do the second thing.

(c) Do the third thing.

A third division. This is a paragraph that starts the third <paragraphs> division.

4. This is an exercise in the third paragraphs division inside a side-by-side.

5. This is a second exercise in the third paragraphs division. It should have a larger workspace and be on the fourth page of the worksheet.

6. This is a third exercise in the third paragraphs division. It should have a larger workspace and be on the fifth page of the worksheet.

38 Handouts

Like worksheets, a **handout** is a division that is inteded to be printed for use in a classroom. In HTML output, you get the same print preview and page layout as you do with worksheets; in PDF, these will start on a new page with possibly different margins than the rest of the document.

Unlike worksheets, handouts do not have a special class of exercises or activities (exercises in a handout are treated like an *inline* exercise). The other main distinction is that a handout lets workspace be specified on pretty much any block or paragraph element, not just exercises, tasks, and project-like elements. This allows them to be used in the creation of guided notes containing some premade content with lots of space for students to fill in details during class.

38.1 Derivative Rules

Rules for specific types of functions.

Constant functions	$(k)' = 0$
Power functions	$(x^n)' = nx^{n-1}$
Exponential functions	$(a^x)' = \ln(a)a^x$ (for $a > 0$) $(e^x)' = e^x$
Logarithmic functions	$(\ln(x))' = \dfrac{1}{x}$
Trigonometric functions	$(\sin(x))' = \cos(x)$. $(\cos(x))' = -\sin(x)$. $(\tan(x))' = \sec^2(x) = \frac{1}{\cos^2(x)}$. $(\arcsin(x))' = \frac{1}{\sqrt{1-x^2}}$. $(\arctan(x))' = \frac{1}{1+x^2}$.

Rules for combinations of functions.

Constant multiples	$(k \cdot f(x))' = k \cdot f'(x)$
Sum and difference	$(f(x) \pm g(x))' = f'(x) \pm g'(x)$
Products (the product rule)	$(f(x) \cdot g(x))' = f'(x)g(x) + f(x)g'(x)$
Quotients (the quotient rule)	$\left(\dfrac{f(x)}{g(x)}\right)' = \dfrac{f'(x)g(x) - f(x)g'(x)}{(g(x))^2}$
Compositions (the chain rule)	$(f(g(x)))' = f'(g(x)) \cdot g'(x)$

38.2 Guided Notes: Derivatives of sums

Today we will explore how to take the derivative of the sum of two functions. For example, if

$$f(x) = x^2 + 3x,$$

what is $f'(x)$? What two functions is this the sum of? What are the derivatives of each of those functions?

To be sure of the derivative of the sum, we should use the definition of the derivative.

Definition 38.1 Definition of the Derivative. The derivative of a function $f(x)$ at any point x is defined as... ◆

Now let's apply this definition to the function $f(x) = x^2 + 3x$. We have:
What should the general rule be?

Theorem 38.2. *For any two differentiable functions $f(x)$ and $g(x)$, the derivative of their sum $h(x) = f(x) + g(x)$ is given by:*

Proof. Let $f(x)$ and $g(x)$ be differentiable functions and let $h(x) = f(x) + g(x)$. Then by the limit definition of the derivative, ■

Example 38.3. Find the derivative of $f(x) = x^5 + e^x$. ▲

Example 38.4. Find the derivative of $f(x) = \sqrt{x} + x^3 + 7x$. ▲

Example 38.5. Find the derivative of $f(x) = 5x^4$. ▲

38.3 A handout with a long list

It is common for a handout, especially a set of lecture notes, to consist of a long list of bullets, possibly more than would fit on a page. This handout should test that such a handout can have its items split between pages even if they belong to the same paragraph.

- This is the first item in a long list. It should be on the first page of the handout.
- This is the second item in a long list. It should be on the first page of the handout.
- This is the third item in a long list. It should be on the second page of the handout.

  Maybe we have a second paragraph in the same list item. This is really just to help fill up space. Maybe I should have used some of that standard space filler, but I'm just typing now because I can't remember what is call.
- I suppose I could just start copy/pasting now.
- Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
- Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
- Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
- Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
- Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.[1] Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

[1] A late footnote

- Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
- Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
- See, I figured it out!

And that's the end of the list. Did this go onto a second page?

39 Projects as Printouts

A project-like element outside of any worksheet or handout may still carry `workspace`, on itself or on its tasks. When it does, HTML output gives it the same print preview icon a worksheet gets, and previewing it lays out just that project across pages with the workspace expanded to fill them. In PDF such a project is typeset as an ordinary block, with no extra room.

The third project below has no `workspace` anywhere, so it gets no icon.

Project 39.1 A Project With Workspace On Its Tasks. Show all of your reasoning in the space provided.

(a) Find the derivative of $f(x) = x^2$.

Hint. Use the power rule.

Solution. We have $f'(x) = 2x$.

(b) Now find the derivative of $g(x) = x^3$, and explain the pattern you see.

Compare your two answers with a neighbor.

Project 39.2 A Project With Workspace On Itself. Explain, in a paragraph, why the workspace requested here is not visible on screen, but does appear in the print preview.

Project 39.3 A Project With No Workspace. This project should not have a print preview icon.

40 Worksheet Solution Testing

Solutions to projects and worksheets can both be selected independently for the auto-generated solutions divisions. What happens when a worksheet contains a project? Here we explore the various combinations.

40.1 Subsection with only exercise

Checkpoint 40.1. This is an exercise's statement.

Answer. This is an exercise's answer.

Solution. This is the exercise's solution.

40.2 Subsection with only project

Project 40.1. This is a project's statement.

Answer. This is a project's answer.

Solution. This is the project's solution.

40.3 Subsection with both exercise and project

Checkpoint 40.2. This is an exercise's statement.

Answer. This is an exercise's answer.

Solution. This is the exercise's solution.

Project 40.2. This is a project's statement.

Answer. This is a project's answer.

Solution. This is the project's solution.

40.4 Worksheet with only exercise

1. This is an exercise's statement.

Answer. This is an exercise's answer.

Solution. This is the exercise's solution.

40.5 Worksheet with only project

Project 40.3. This is a project's statement.

Answer. This is a project's answer.

Solution. This is the project's solution.

40.6 Worksheet with both exercise and project

1. This is an exercise's statement.

Answer. This is an exercise's answer.

Solution. This is the exercise's solution.

Project 40.4. This is a project's statement.

Answer. This is a project's answer.

Solution. This is the project's solution.

40.7 Inline answers only

40.1 · Subsection with only exercise

Checkpoint 40.1.

Answer. This is an exercise's answer.

40.3 · Subsection with both exercise and project

Checkpoint 40.2.

Answer. This is an exercise's answer.

40.8 Project answers only

40.2 · Subsection with only project

Project 40.1.

Answer. This is a project's answer.

40.3 · Subsection with both exercise and project

Project 40.2.

Answer. This is a project's answer.

40.5 · Worksheet with only project

Project 40.3.

Answer. This is a project's answer.

40.6 · Worksheet with both exercise and project

Project 40.4.

Answer. This is a project's answer.

40.9 Worksheet answers only

40.4 · Worksheet with only exercise

1.

Answer. This is an exercise's answer.

40.6 · Worksheet with both exercise and project

1.

Answer. This is an exercise's answer.

40.10 Inline answers, project statement

40.1 · Subsection with only exercise

Checkpoint 40.1.

Answer. This is an exercise's answer.

40.2 · Subsection with only project

Project 40.1. This is a project's statement.

40.3 · Subsection with both exercise and project

Checkpoint 40.2.

Answer. This is an exercise's answer.

Project 40.2. This is a project's statement.

40.5 · Worksheet with only project

Project 40.3. This is a project's statement.

40.6 · Worksheet with both exercise and project

Project 40.4. This is a project's statement.

40.11 Inline answers, worksheet statement

40.1 · Subsection with only exercise

Checkpoint 40.1.

Answer. This is an exercise's answer.

40.3 · Subsection with both exercise and project

Checkpoint 40.2.

Answer. This is an exercise's answer.

40.4 · Worksheet with only exercise

1. This is an exercise's statement.

40.6 · Worksheet with both exercise and project

1. This is an exercise's statement.

40.12 Project answers, worksheet statement

40.2 · Subsection with only project

Project 40.1.

Answer. This is a project's answer.

40.3 · Subsection with both exercise and project

Project 40.2.

Answer. This is a project's answer.

40.4 · Worksheet with only exercise

1. This is an exercise's statement.

40.5 · Worksheet with only project

Project 40.3.

Answer. This is a project's answer.

40.6 · Worksheet with both exercise and project

1. This is an exercise's statement.

Project 40.4.

Answer. This is a project's answer.

40.13 Inline answers, project statement, worksheet solution

40.1 · Subsection with only exercise

Checkpoint 40.1.

Answer. This is an exercise's answer.

40.2 · Subsection with only project

Project 40.1. This is a project's statement.

40.3 · Subsection with both exercise and project

Checkpoint 40.2.

Answer. This is an exercise's answer.

Project 40.2. This is a project's statement.

40.4 · Worksheet with only exercise

1.

Solution. This is the exercise's solution.

40.5 · Worksheet with only project

Project 40.3. This is a project's statement.

40.6 · Worksheet with both exercise and project

1.

Solution. This is the exercise's solution.

Project 40.4. This is a project's statement.

41 Exercises, One Subsection

This `<section>` of the sample article demonstrates an "unstructured division." There are no `<subsection>`, you are just reading the first two paragraphs, followed by some nonsense text. Then there is a *single* `<exercises>` division. Note that this division is not numbered (since it is unique within the `<section>`). And a cross-reference to one of the contained `<exercise>` will be numbered as a member of the `<section>`, Exercise 41.4.

If you use the unstructured form of a division, *and* have both inline and divisional exercises, there is a potential to form ambiguous cross-references. To wit, check that 41.2 and 41.2 are really different exercises (which you are *unable to do* if you are reading this in print!). The solution is to include the type of exercise in the reference, which will assist everybody, but especially your print readers: Checkpoint 41.2 and Exercise 41.2.

Compare this section with the similar Section 43, next. The following text is mostly nonsense, just for testing purposes.

Checkpoint 41.1 Inline One. Aliquam vitae risus placerat, pellentesque leo vitae, iaculis ante. Praesent ac odio eget mi bibendum eleifend ac eget metus. Morbi in dolor et diam accumsan mattis. Aenean elementum pulvinar efficitur. Etiam viverra ut tellus quis consequat. Phasellus sit amet nisl a ligula pharetra tempus id in elit. Maecenas congue quam eu purus fermentum pretium. Fusce pellentesque ultricies arcu, egestas sollicitudin erat condimentum non. Integer non velit at dolor dictum aliquam et rhoncus mauris. Sed nec nibh id nunc convallis tincidunt ut at ligula. Etiam elementum nisl eu erat dapibus rhoncus.

Pellentesque nec condimentum ligula, quis interdum mauris. Ut sed urna lacinia, aliquam arcu id, faucibus nisi. Suspendisse potenti. Curabitur in erat ultricies, condimentum mi nec, vehicula mauris. Duis faucibus risus fermentum velit hendrerit, non laoreet massa maximus. Donec bibendum elit ac lectus lobortis luctus. Ut finibus, dolor ut euismod tristique, ligula tortor tempus arcu, finibus semper purus erat ut ligula. Aenean accumsan ut ante vel euismod.

Checkpoint 41.2 Inline Two. Ut porttitor neque a pharetra euismod. Vivamus ut metus pretium, placerat massa tempor, condimentum metus. Phasellus vestibulum iaculis turpis non posuere. Vestibulum quis aliquet neque. Donec nec metus iaculis, laoreet massa vitae, suscipit tellus. Etiam et ultrices quam, quis pretium ligula. In ut cursus metus. Aenean volutpat quam odio, quis tempus dolor egestas eget. Nunc fringilla lobortis nunc, ut interdum lorem posuere sed. Sed sodales risus a laoreet venenatis. Nunc sodales tempor mollis. Nam sollicitudin velit sed ex viverra feugiat. Nunc consectetur mi vitae urna sollicitudin malesuada. Fusce eget risus lectus. Mauris augue velit, vestibulum vitae tempus sit amet, porttitor eget turpis.

Theorem 41.3 Major Result. Vivamus tortor tortor, lobortis et sem vel, accumsan placerat libero. Sed eget metus non magna accumsan efficitur a non turpis. Curabitur maximus arcu ipsum, eget vestibulum nulla mollis ac. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Duis vel eleifend risus. Morbi hendrerit tellus eget nibh imperdiet, ac mollis nisl sagittis. Ut commodo pharetra leo. Suspendisse consequat velit eget velit condimentum feugiat.

Checkpoint 41.4 Inline Three. Suspendisse lacinia mattis risus, eget viverra urna dictum eu. Maecenas ut sem in turpis egestas varius nec at ipsum. Praesent bibendum nisi et turpis congue, a pellentesque felis tempor. Vivamus non dolor in risus interdum mattis. In tempus iaculis velit, sit amet rhoncus tellus aliquam convallis. Sed ut tellus id ipsum blandit convallis sed eget tortor. Nunc leo felis, scelerisque vel ante porta, volutpat rhoncus neque. Mauris convallis, felis at aliquam aliquet, felis ipsum semper mi, vitae auctor purus ante non erat. Ut nec felis mi.

41 Exercise Collection

1. Drill One. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras congue urna nulla. Aliquam eget euismod tellus. Maecenas nibh libero, venenatis a laoreet in, tempor sit amet sem. Morbi sit amet justo tempor velit auctor placerat. Maecenas nec lobortis orci. Aenean dictum enim lacus, ac blandit lacus elementum nec. Mauris porttitor neque volutpat tincidunt sollicitudin. Cras porta lectus ac facilisis tempor. Suspendisse in velit nisl. Sed convallis leo at nunc aliquet fermentum. Pellentesque feugiat at ex sed elementum. In porta vulputate ipsum sit amet consectetur.

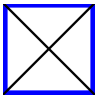


Figure 41.5. A figure in an `<exercise>` in an *unstructured* division to test that it is numbered as if the containing `<exercises>` is not present



Figure 41.6(a). First Panel, subcaptioned



Figure 41.6(b). Second Panel, subcaptioned

Figure 41.6. A side-by-side in a figure in an `<exercise>` in an *unstructured* division to test that it is numbered as if the containing `<exercises>` is not present and to test the numbering of the panels in a solutions manual

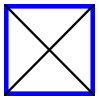


Figure 41.7. First Panel, not subcaptioned

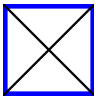


Figure 41.8. Second Panel, not subcaptioned

2. Drill Two. Aliquam sagittis ex at magna porttitor, quis scelerisque ligula malesuada. Vestibulum vitae mauris id nisi pretium ornare a id lectus. Nam suscipit magna id sem ultricies vestibulum. Nulla facilisi. Duis venenatis lectus massa, ac mollis nisi suscipit non. Sed et sapien vulputate, imperdiet nibh vel, lobortis nisl. Curabitur sagittis justo nibh. Nulla vulputate sodales justo at efficitur. Cras justo augue, mollis vitae nulla sed, vestibulum tincidunt ante. Vestibulum et lorem lectus. Maecenas accumsan lacus a nisi euismod rutrum eu dapibus justo. Etiam scelerisque a odio a euismod. Pellentesque vestibulum pellentesque nisi, a tempus erat cursus ac. Morbi ut auctor lectus.

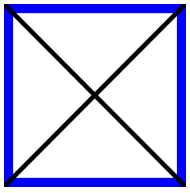


Figure 41.9. Testing numbering, figure in a `<exercise>` in an un-numbered `<exercises>`

3. Drill Three. Nam congue ex nec justo iaculis maximus. Vestibulum lobortis magna sed urna auctor, vel dignissim massa posuere. In sed venenatis elit. Vivamus congue gravida tempus. Nunc quis fermentum nisi. Nullam hendrerit lorem et tellus semper, sit amet scelerisque purus lobortis. Quisque sollicitudin quis neque eu suscipit. Praesent volutpat justo quis magna vehicula molestie.

4. Challenging One. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras congue urna nulla. Aliquam eget euismod tellus. Maecenas nibh libero, venenatis a laoreet in, tempor sit amet sem. Morbi sit amet justo tempor velit auctor placerat. Maecenas nec lobortis orci. Aenean dictum enim lacus, ac blandit lacus elementum nec. Mauris porttitor neque volutpat tincidunt sollicitudin. Cras porta lectus ac facilisis tempor. Suspendisse in velit nisl.

Sed convallis leo at nunc aliquet fermentum. Pellentesque feugiat at ex sed elementum. In porta vulputate ipsum sit amet consectetur.

An introduction to an `exercisegroup`. This is here to attempt to interrupt the flow of the counting from this division to the next.

5. Challenging Two. Aliquam sagittis ex at magna porttitor, quis scelerisque ligula malesuada. Vestibulum vitae mauris id nisi pretium ornare a id lectus. Nam suscipit magna id sem ultricies vestibulum. Nulla facilisi. Duis venenatis lectus massa, ac mollis nisi suscipit non. Sed et sapien vulputate, imperdiet nibh vel, lobortis nisl. Curabitur sagittis justo nibh.

Nulla vulputate sodales justo at efficitur. Cras justo augue, mollis vitae nulla sed, vestibulum tincidunt ante. Vestibulum et lorem lectus. Maecenas accumsan lacus a nisi euismod rutrum eu dapibus justo. Etiam scelerisque a odio a euismod. Pellentesque vestibulum pellentesque nisi, a tempus erat cursus ac. Morbi ut auctor lectus.

6. Challenging Three. Nam congue ex nec justo iaculis maximus. Vestibulum lobortis magna sed urna auctor, vel dignissim massa posuere. In sed venenatis elit. Vivamus congue gravida tempus. Nunc quis fermentum nisi. Nullam hendrerit lorem et tellus semper, sit amet scelerisque purus lobortis. Quisque sollicitudin quis neque eu suscipit. Praesent volutpat justo quis magna vehicula molestie.

7. Impossible One. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras congue urna nulla. Aliquam eget euismod tellus. Maecenas nibh libero, venenatis a laoreet in, tempor sit amet sem. Morbi sit amet justo tempor velit auctor placerat. Maecenas nec lobortis orci. Aenean dictum enim lacus, ac blandit lacus elementum nec. Mauris porttitor neque volutpat tincidunt sollicitudin. Cras porta lectus ac facilisis tempor. Suspendisse in velit nisl. Sed convallis leo at nunc aliquet fermentum. Pellentesque feugiat at ex sed elementum. In porta vulputate ipsum sit amet consectetur.

8. Impossible Two. Aliquam sagittis ex at magna porttitor, quis scelerisque ligula malesuada. Vestibulum vitae mauris id nisi pretium ornare a id lectus. Nam suscipit magna id sem ultricies vestibulum. Nulla facilisi. Duis venenatis lectus massa, ac mollis nisi suscipit non. Sed et sapien vulputate, imperdiet nibh vel, lobortis nisl. Curabitur sagittis justo nibh. Nulla vulputate sodales justo at efficitur. Cras justo augue, mollis vitae nulla sed, vestibulum tincidunt ante. Vestibulum et lorem lectus. Maecenas accumsan lacus a nisi euismod rutrum eu dapibus justo. Etiam scelerisque a odio a euismod. Pellentesque vestibulum pellentesque nisi, a tempus erat cursus ac. Morbi ut auctor lectus.

9. Impossible Three. Nam congue ex nec justo iaculis maximus. Vestibulum lobortis magna sed urna auctor, vel dignissim massa posuere. In sed venenatis elit. Vivamus congue gravida tempus. Nunc quis fermentum nisi. Nullam hendrerit lorem et tellus semper, sit amet scelerisque purus lobortis. Quisque sollicitudin quis neque eu suscipit. Praesent volutpat justo quis magna vehicula molestie.

41 More Reading

Left intentionally blank, just checking sectioning.

[1] David C. Lay, *Subspaces and Echelon Forms*. The College Mathematics Journal, January 1993, **24** no. 1, 57–62.

42 Exercise Section, Structured

A collection of exercises, in a top-level `<exercises>` division, structured with `<subexercises>`.

Easy Exercises

1. Arithmetic. Compute $3 + 4$.

Answer. 7

2. Trigonometry. Compute $\sin(\pi/4)$.

Hard Problems

3. Number Theory. Prove Fermat's Last Theorem.

4. Millenial. Find general solutions to the Navier-Stokes equation.

With an Exercise Group

This is an exercise group, and this is its introduction.
And a second line for the introduction.

5. One. $6 - 5$

Hint. Subtract.

6. Two. $1 + 1$

Here we conclude the group.
And a second line for the conclusion.

7. Outside exercisegroup, inside subexercises. 6+5

43 Exercises, Multiple Subsections

This `<section>` of the sample article demonstrates a "structured division." You are reading the introduction to the division, then there is a faux `<subsection>`, followed by three `<exercises>` divisions. Note that the three are numbered as if they are also fellow`<subsection>`. And a cross-reference to one of the contained `<exercise>` will be numbered use the number of the `<subsection>`, Exercise 43.3.1.

Compare this section with the similar Section 41, previous. The following text is mostly nonsense, just for testing purposes.

43.1 Faux Subsection

Checkpoint 43.1 Inline One. Aliquam vitae risus placerat, pellentesque leo vitae, iaculis ante. Praesent ac odio eget mi bibendum eleifend ac eget metus. Morbi in dolor et diam accumsan mattis. Aenean elementum pulvinar efficitur. Etiam viverra ut tellus quis consequat. Phasellus sit amet nisl a ligula pharetra tempus id in elit. Maecenas congue quam eu purus fermentum pretium. Fusce pellentesque ultricies arcu, egestas sollicitudin erat condimentum non. Integer non velit at dolor dictum aliquam et rhoncus mauris. Sed nec nibh id nunc convallis tincidunt ut at ligula. Etiam elementum nisl eu erat dapibus rhoncus.

Pellentesque nec condimentum ligula, quis interdum mauris. Ut sed urna lacinia, aliquam arcu id, faucibus nisi. Suspendisse potenti. Curabitur in erat ultricies, condimentum mi nec, vehicula mauris. Duis faucibus risus fermentum velit hendrerit, non laoreet massa maximus. Donec bibendum elit ac lectus lobortis luctus. Ut finibus, dolor ut euismod tristique, ligula tortor tempus arcu, finibus semper purus erat ut ligula. Aenean accumsan ut ante vel euismod.

Checkpoint 43.2 Inline Two. Ut porttitor neque a pharetra euismod. Vivamus ut metus pretium, placerat massa tempor, condimentum metus. Phasellus vestibulum iaculis turpis non posuere. Vestibulum quis aliquet neque. Donec nec metus iaculis, laoreet massa vitae, suscipit tellus. Etiam et ultrices quam, quis pretium ligula. In ut cursus metus. Aenean volutpat quam odio, quis tempus dolor egestas eget. Nunc fringilla lobortis nunc, ut interdum lorem posuere sed. Sed sodales risus a laoreet venenatis. Nunc sodales tempor mollis. Nam sollicitudin velit sed ex viverra feugiat. Nunc consectetur mi vitae urna sollicitudin malesuada. Fusce eget risus lectus. Mauris augue velit, vestibulum vitae tempus sit amet, porttitor eget turpis.

Theorem 43.3 Major Result. Vivamus tortor tortor, lobortis et sem vel, accumsan placerat libero. Sed eget metus non magna accumsan efficitur a non turpis. Curabitur maximus arcu ipsum, eget vestibulum nulla mollis ac. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Duis vel eleifend risus. Morbi hendrerit tellus eget nibh imperdiet, ac mollis nisl sagittis. Ut commodo pharetra leo. Suspendisse consequat velit eget velit condimentum feugiat.

Checkpoint 43.4 Inline Three. Suspendisse lacinia mattis risus, eget viverra urna dictum eu. Maecenas ut sem in turpis egestas varius nec at ipsum. Praesent bibendum nisi et turpis congue, a pellentesque felis tempor. Vivamus non dolor in risus interdum mattis. In tempus iaculis velit, sit amet rhoncus tellus aliquam convallis. Sed ut tellus id ipsum blandit convallis sed eget tortor. Nunc leo felis, scelerisque vel ante porta, volutpat rhoncus neque. Mauris convallis, felis at aliquam aliquet, felis ipsum semper mi, vitae auctor purus ante non erat. Ut nec felis mi.

43.2 Drill Exercises

1. Drill One. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras congue urna nulla. Aliquam eget euismod tellus. Maecenas nibh libero, venenatis a laoreet in, tempor sit amet sem. Morbi sit amet justo tempor velit auctor placerat. Maecenas nec lobortis orci. Aenean dictum enim lacus, ac blandit lacus elementum nec. Mauris porttitor neque volutpat tincidunt sollicitudin. Cras porta lectus ac facilisis tempor. Suspendisse in velit nisl.

Sed convallis leo at nunc aliquet fermentum. Pellentesque feugiat at ex sed elementum. In porta vulputate ipsum sit amet consectetur.

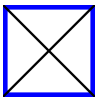


Figure 43.5. A figure in an `<exercise>` in a *structured* division to test that it is numbered with consideration of the containing `<exercises>`

2. Drill Two. Aliquam sagittis ex at magna porttitor, quis scelerisque ligula malesuada. Vestibulum vitae mauris id nisi pretium ornare a id lectus. Nam suscipit magna id sem ultricies vestibulum. Nulla facilisi. Duis venenatis lectus massa, ac mollis nisi suscipit non. Sed et sapien vulputate, imperdiet nibh vel, lobortis nisl. Curabitur sagittis justo nibh. Nulla vulputate sodales justo at efficitur. Cras justo augue, mollis vitae nulla sed, vestibulum tincidunt ante. Vestibulum et lorem lectus. Maecenas accumsan lacus a nisi euismod rutrum eu dapibus justo. Etiam scelerisque a odio a euismod. Pellentesque vestibulum pellentesque nisi, a tempus erat cursus ac. Morbi ut auctor lectus.

3. Drill Three. Nam congue ex nec justo iaculis maximus. Vestibulum lobortis magna sed urna auctor, vel dignissim massa posuere. In sed venenatis elit. Vivamus congue gravida tempus. Nunc quis fermentum nisi. Nullam hendrerit lorem et tellus semper, sit amet scelerisque purus lobortis. Quisque sollicitudin quis neque eu suscipit. Praesent volutpat justo quis magna vehicula molestie.

43.3 Challenging Exercises

1. Challenging One. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras congue urna nulla. Aliquam eget euismod tellus. Maecenas nibh libero, venenatis a laoreet in, tempor sit amet sem. Morbi sit amet justo tempor velit auctor placerat. Maecenas nec lobortis orci. Aenean dictum enim lacus, ac blandit lacus elementum nec. Mauris porttitor neque volutpat tincidunt sollicitudin. Cras porta lectus ac facilisis tempor. Suspendisse in velit nisl. Sed convallis leo at nunc aliquet fermentum. Pellentesque feugiat at ex sed elementum. In porta vulputate ipsum sit amet consectetur.

An introduction to an `exercisegroup`. This is here to attempt to interrupt the flow of the counting from this division to the next.

2. Challenging Two. Aliquam sagittis ex at magna porttitor, quis scelerisque ligula malesuada. Vestibulum vitae mauris id nisi pretium ornare a id lectus. Nam suscipit magna id sem ultricies vestibulum. Nulla facilisi. Duis venenatis lectus massa, ac mollis nisi suscipit non. Sed et sapien vulputate, imperdiet nibh vel, lobortis nisl. Curabitur sagittis justo nibh.

Nulla vulputate sodales justo at efficitur. Cras justo augue, mollis vitae nulla sed, vestibulum tincidunt ante. Vestibulum et lorem lectus. Maecenas accumsan lacus a nisi euismod rutrum eu dapibus justo. Etiam scelerisque a odio a euismod. Pellentesque vestibulum pellentesque nisi, a tempus erat cursus ac. Morbi ut auctor lectus.

3. Challenging Three. Nam congue ex nec justo iaculis maximus. Vestibulum lobortis magna sed urna auctor, vel dignissim massa posuere. In

sed venenatis elit. Vivamus congue gravida tempus. Nunc quis fermentum nisi. Nullam hendrerit lorem et tellus semper, sit amet scelerisque purus lobortis. Quisque sollicitudin quis neque eu suscipit. Praesent volutpat justo quis magna vehicula molestie.

43.4 Impossible Exercises

1. Impossible One. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras congue urna nulla. Aliquam eget euismod tellus. Maecenas nibh libero, venenatis a laoreet in, tempor sit amet sem. Morbi sit amet justo tempor velit auctor placerat. Maecenas nec lobortis orci. Aenean dictum enim lacus, ac blandit lacus elementum nec. Mauris porttitor neque volutpat tincidunt sollicitudin. Cras porta lectus ac facilisis tempor. Suspendisse in velit nisl. Sed convallis leo at nunc aliquet fermentum. Pellentesque feugiat at ex sed elementum. In porta vulputate ipsum sit amet consectetur.

2. Impossible Two. Aliquam sagittis ex at magna porttitor, quis scelerisque ligula malesuada. Vestibulum vitae mauris id nisi pretium ornare a id lectus. Nam suscipit magna id sem ultricies vestibulum. Nulla facilisi. Duis venenatis lectus massa, ac mollis nisi suscipit non. Sed et sapien vulputate, imperdiet nibh vel, lobortis nisl. Curabitur sagittis justo nibh. Nulla vulputate sodales justo at efficitur. Cras justo augue, mollis vitae nulla sed, vestibulum tincidunt ante. Vestibulum et lorem lectus. Maecenas accumsan lacus a nisi euismod rutrum eu dapibus justo. Etiam scelerisque a odio a euismod. Pellentesque vestibulum pellentesque nisi, a tempus erat cursus ac. Morbi ut auctor lectus.

3. Impossible Three. Nam congue ex nec justo iaculis maximus. Vestibulum lobortis magna sed urna auctor, vel dignissim massa posuere. In sed venenatis elit. Vivamus congue gravida tempus. Nunc quis fermentum nisi. Nullam hendrerit lorem et tellus semper, sit amet scelerisque purus lobortis. Quisque sollicitudin quis neque eu suscipit. Praesent volutpat justo quis magna vehicula molestie.

43.5 Layout challenge

A multicolumn exercisegroup with layout challenges. The first two should be on one row. The next two should be on one row until the hint is opened. Then one that takes up an entire row. The final row should always have two exercises.

Aside Column test. An aside in a multi-column container.

$$\begin{aligned} 2x + \ y &= 10 \\ x \ &= \ 6 \\ x + 3y &= 13 \end{aligned}$$

2.

$$\begin{aligned} 2x + \ y &= 10 \\ x \ &= \ 6 \\ x + 3y &= 13 \end{aligned}$$

3.

$$\begin{aligned} 2x + \ y &= 10 \\ x \ &= \ 6 \\ x + 3y &= 13 \end{aligned}$$

4.

$$\begin{aligned} 2x + \ y &= 10 \\ x \ &= \ 6 \\ x + 3y &= 13 \end{aligned}$$

Hint (OPEN ME!).

$$\begin{aligned} 3u + 1v + 3w + 2x + y \ &+ 3z = 10 \\ 2u + 2v - 2w + x \ &+ z \ = 6 \\ 1u - 3v + 1w + x \ &+ 3y + 2z = 13 \end{aligned}$$

5.

$$\begin{aligned} 200t + 3u + 1v + 3w + 2x + \ y + 3z &= 10 \\ + 2u + 2v - 2w \ + x \ + \ z &= \ 6 \\ 100t + 1u - 3v + 1w \ + x + 3y + 2z &= 13 \end{aligned}$$

6.

$$\begin{aligned} 2x + \ y &= 10 \\ x \ &= \ 6 \\ x + 3y &= 13 \end{aligned}$$

7.

$$\begin{aligned} 2x + \ y &= 10 \\ x \ &= \ 6 \\ x + 3y &= 13 \end{aligned}$$

Here we conclude the group of exercises.

And a second line for the conclusion.

43.6 More Reading

Left intentionally blank, just checking sectioning.

[1] David C. Lay, *Subspaces and Echelon Forms*. The College Mathematics Journal, January 1993, **24** no. 1, 57–62.

44 Bibliography Markup Styles

44.1 The Three Styles

PreTeXt supports three styles of markup for the `<biblio>` entries of a `<references>` division. To compare them, the same twelve references appear three times in the divisions that follow, once in each style. The references are deliberately varied: books and journal articles, a chapter and a conference paper, a thesis, a report, a software system, and a web page; with one, two, and three authors, as well as organizational authors.

A **raw** entry (`type="raw"`) is free-form mixed content: the author supplies the order, the punctuation, and almost all of the formatting, while PreTeXt styles only a few recognized elements, such as an italicized `<title>`, a bold `<volume>`, a parenthesized `<year>`, and a `<number>` prefixed by "no."

A **BibTeX-style** entry (`type="bibtex"`) uses structured fields, rendered in the order written, with punctuation supplied automatically. The result is more uniform than raw markup, but the field set is small, so details with no dedicated field—an edition, a URL, a thesis designation, a report number—are tucked into a free-form `<pubnote>`.

A **CSL-style** entry records structured data only—names, dates, titles, and the like—using markup modeled on the CSL-JSON data format, in a fixed canonical order. The order of presentation and the punctuation are then supplied by PreTeXt, or, when a CSL style is configured in the publication file, by the `citeproc` processor. This is the recommended style as of 2025-08-13.

44.2 Raw Entries

The twelve references as `type="raw"` entries.

[1] Thomas W. Judson, *Abstract Algebra: Theory and Applications*, Orthogonal Publishing, Ann Arbor, MI, 2022.

[2] David C. Lay, *Subspaces and Echelon Forms*, The College Mathematics Journal **24**, no. 1 (1993), 57–62.

[3] J. B. Conrey and D. W. Farmer, *Mean values of L-functions and symmetry*, International Mathematics Research Notices **17** (2000), 883–908.

[4] David S. Dummit and Richard M. Foote, *Abstract Algebra*, 3rd edition, John Wiley & Sons, Hoboken, NJ, 2004.

[5] The PreTeXt Project, *The PreTeXt Guide*, `https://pretextbook.org/doc/guide/html/`, accessed June 1, 2026.

[6] Jane Q. Author, "Finite Fields and Their Applications", in *Topics in Modern Algebra*, edited by John Smith, Academic Press, San Diego, CA, 2010, 45–98.

[7] A. Lovelace, C. Babbage, and G. Boole, "A Fast Algorithm for Matrix Factorization", in *Proceedings of the International Congress of Mathematicians*, 2018, 1123–1140.

[8] Terence Tao, *Three Regularity Results in Harmonic Analysis*, Ph.D. thesis, Princeton University, 1996.

[9] The Sage Developers, *SageMath, the Sage Mathematics Software System (Version 10.3)*, 2024, `https://www.sagemath.org`.

[10] National Institute of Standards and Technology, *Digital Signature Standard (DSS)*, FIPS PUB 186-4, U.S. Department of Commerce, 2013.

[11] Gilbert Strang, *Introduction to Linear Algebra*, 5th edition, Wellesley–Cambridge Press, Wellesley, MA, 2016, 584 pages, ISBN 978-0-9802327-7-6.

[12] L. N. Trefethen, "A Brief Note on Spectral Methods", SIAM Review **50** no. 1 (2008), p. 67, doi:10.1137/060657716, ISSN 0036-1445.

44.3 BibTeX-Style Entries

The twelve references as `type="bibtex"` entries.

[1] Thomas W. Judson, *Abstract Algebra: Theory and Applications*, Orthogonal Publishing, Ann Arbor, MI, (2022).

[2] David C. Lay, *Subspaces and Echelon Forms*, The College Mathematics Journal **24** no. 1 (1993) pp. 57-62.

[3] J. B. Conrey and D. W. Farmer, *Mean values of L-functions and symmetry*, International Mathematics Research Notices **17** (2000) pp. 883-908.

[4] David S. Dummit and Richard M. Foote, *Abstract Algebra*, John Wiley & Sons, Hoboken, NJ, (2004) 3rd edition.

[5] The PreTeXt Project, *The PreTeXt Guide*, https://pretextbook.org/doc/guide/html/, accessed June 1, 2026.

[6] Jane Q. Author, John Smith (ed.), *Finite Fields and Their Applications*, Topics in Modern Algebra, Academic Press, San Diego, CA, (2010) pp. 45-98.

[7] A. Lovelace, C. Babbage, and G. Boole, *A Fast Algorithm for Matrix Factorization*, Proceedings of the International Congress of Mathematicians, (2018) pp. 1123-1140.

[8] Terence Tao, *Three Regularity Results in Harmonic Analysis*, Ph.D. thesis, Princeton University, (1996).

[9] The Sage Developers, *SageMath, the Sage Mathematics Software System (Version 10.3)*, (2024) https://www.sagemath.org.

[10] National Institute of Standards and Technology, *Digital Signature Standard (DSS)*, U.S. Department of Commerce, (2013) FIPS PUB 186-4.

[11] Gilbert Strang, *Introduction to Linear Algebra*, Wellesley-Cambridge Press, Wellesley, MA, (2016) 584, 5th edition; ISBN 978-0-9802327-7-6.

[12] L. N. Trefethen, *A Brief Note on Spectral Methods*, SIAM Review **50** no. 1 (2008) 67, doi:10.1137/060657716; ISSN 0036-1445.

44.4 CSL-Style Entries

The twelve references as CSL-style entries.

[1] Judson, Thomas W., *Abstract Algebra: Theory and Applications* (2022), Orthogonal Publishing, Ann Arbor, MI.

[2] Lay, David C., *Subspaces and Echelon Forms*, The College Mathematics Journal **24** no. 1 (1993) 57-62.

[3] Conrey, J. B. and D. W. Farmer, *Mean values of L-functions and symmetry*, International Mathematics Research Notices **17** (2000) 883-908.

[4] Dummit, David S. and Richard M. Foote, *Abstract Algebra*, 3rd edition (2004), John Wiley & Sons, Hoboken, NJ.

[5] The PreTeXt Project, *The PreTeXt Guide*, accessed 2026-06-01 `https://pretextbook.org/doc/guide/html/`.

[6] Author, Jane Q., Smith, John (ed.), *Finite Fields and Their Applications*, Topics in Modern Algebra, Lectures on Algebra 12 **1**, 2 volumes, chapter 3 (2010) 45-98, Academic Press, San Diego, CA.

[7] Lovelace, A.; C. Babbage and G. Boole, *A Fast Algorithm for Matrix Factorization*, Proceedings of the International Congress of Mathematicians, International Congress of Mathematicians (2018) 1123-1140, Institutional Repository item 4471, accepted manuscript.

[8] Tao, Terence, *Three Regularity Results in Harmonic Analysis*, Ph.D. thesis (1996), Princeton University.

[9] The Sage Developers, *SageMath, the Sage Mathematics Software System*, version 10.3 (2024) `https://www.sagemath.org`.

[10] National Institute of Standards and Technology, *Digital Signature Standard (DSS)* no. FIPS PUB 186-4 (2013), U.S. Department of Commerce.

[11] Strang, Gilbert, *Introduction to Linear Algebra*, 5th edition (2016), 584 pages, Wellesley-Cambridge Press, Wellesley, MA, ISBN 978-0-9802327-7-6.

[12] Trefethen, L. N., *A Brief Note on Spectral Methods*, SIAM Review **50**(1) (2008) 67, doi:10.1137/060657716, ISSN 0036-1445.

45 Runestone Components

One of each interactive Runestone component, each with a moderately complicated set of attributes and features, so this article can serve as a regression test for the whole collection. The final chapter of the sample book is the comprehensive tour, with many variants of each component; here one representative apiece suffices. Fill-in exercises, in both the dynamic and the legacy styles, are exercised thoroughly at Section 17 and so are not repeated here. Note that many of these components render, and function, in ordinary offline HTML; a Runestone server adds grading and persistence.

Checkpoint 45.1 True/False, a True Statement. The square of a real number is never negative.

Hint. Consider the three cases: negative, zero, positive.

Answer. True.

Solution. True.
Squaring preserves the sign of zero and erases the sign of everything else.

Checkpoint 45.2 Multiple Choice, Several Correct, Randomized. Which of the following are prime numbers?

A. 2
B. 1
C. 17

Answer. A, C.

Solution.

A. *Correct.*
The only even prime.
B. *Incorrect.*
A unit, by convention not a prime.
C. *Correct.*
Indeed, no divisors but 1 and itself.

Checkpoint 45.3 Parsons, Vertical, Partial Ordering. Arrange the blocks to define a function and then call it. There is more than one correct order, a paired distractor, and an unneeded block.

- Either:
  ```
  n = int(input())
  ```
  Or:
  ```
  n = input()
  ```
- `print(double(n))`
- ```
  def double(x):
      return 2 * x
  ```
- `import antigravity`
  ```

Solution.

```
def double(x):
    return 2 * x
n = int(input())
print(double(n))
```

Checkpoint 45.4 Parsons, Horizontal, Reused Blocks. Assemble a familiar proverb. Two of the blocks are each used twice.

| early | to | bed | and | early | to | rise |

Solution. early to early to bed and early to rise

Checkpoint 45.5 Cardsort, with Distractors. Sort each function by its growth rate. One category has no members, and one function fits no category.

2^n	Polynomial
n^2	Exponential
$n \log n$	Constant
$100n^3$	

Solution.

- Polynomial
  - n^2
  - $100n^3$
- Exponential
  - 2^n
- Constant
- (Uncategorized)
  - $n \log n$

Checkpoint 45.6 Matching, Premises with References. Match each derivative rule to its name. One rule matches two names.

$(fg)' = f'g + fg'$	Chain Rule
$(f \circ g)' = (f' \circ g) \cdot g'$	Product Rule
	Composite Rule

Solution.

- Product Rule
  - $(fg)' = f'g + fg'$
- Chain Rule
  - $(f \circ g)' = (f' \circ g) \cdot g'$
- Composite Rule
  - $(f \circ g)' = (f' \circ g) \cdot g'$

Checkpoint 45.7 Clickable Areas, Code. Click every line that mutates the list `data`.

```
data = [3, 1, 2]
data.sort()
```

```
total = sum(data)
data.append(total)
```

Answer. Correct: `data.sort()`; `data.append(total)`. Incorrect: `total = sum(data)`.

Reading a list is not mutating it.

Checkpoint 45.8 Select, an A/B Experiment. Runestone-only: an A/B experiment (named `sample-article-regression`) with Checkpoint 45.1 as (A) and Checkpoint 45.2 as (B).

Checkpoint 45.9 Short Answer, with Attachment. Describe, in a sentence or two, the difference between a sequence and a series. Attach a photograph of your handwritten scratch work.

Checkpoint 45.10 ActiveCode, Decorated. Run the program, then modify the highlighted line so it counts down instead.

```
for i in range(5):
    print(i)
```

Checkpoint 45.11 CodeLens, with a Checkpoint. Step the visualization and answer the question it poses.

```
a = 5
b = a + 2
c = a * b
```

Query. Which topic in this article did you find most challenging?

1. The Fundamental Theorem
2. Displayed mathematics
3. Table calisthenics

Data: `regression.csv`

```
x,y
1,1
2,4
3,9
```

Checkpoint 45.12 A Dual Exercise (Experimental). An exercise about array-based list insertion goes here, in an output with no interaction.

46 Exercises, Top-Level

This `<exercises>` of the sample article is a peer of all the preceding `<section>` and is the only such `<exercises>`. As such, it is not numbered, and contains only `<exercise>`, but for this `<introduction>` you are reading. The `<exercises>` contained within will be numbered in cross-references according to the enclosing division, in this case the entire article and so without any qualification, to wit, Exercise 46.4.

1. Drill One. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras congue urna nulla. Aliquam eget euismod tellus. Maecenas nibh libero, venenatis a laoreet in, tempor sit amet sem. Morbi sit amet justo tempor

velit auctor placerat. Maecenas nec lobortis orci. Aenean dictum enim lacus, ac blandit lacus elementum nec. Mauris porttitor neque volutpat tincidunt sollicitudin. Cras porta lectus ac facilisis tempor. Suspendisse in velit nisl. Sed convallis leo at nunc aliquet fermentum. Pellentesque feugiat at ex sed elementum. In porta vulputate ipsum sit amet consectetur.

2. Drill Two. Aliquam sagittis ex at magna porttitor, quis scelerisque ligula malesuada. Vestibulum vitae mauris id nisi pretium ornare a id lectus. Nam suscipit magna id sem ultricies vestibulum. Nulla facilisi. Duis venenatis lectus massa, ac mollis nisi suscipit non. Sed et sapien vulputate, imperdiet nibh vel, lobortis nisl. Curabitur sagittis justo nibh. Nulla vulputate sodales justo at efficitur. Cras justo augue, mollis vitae nulla sed, vestibulum tincidunt ante. Vestibulum et lorem lectus. Maecenas accumsan lacus a nisi euismod rutrum eu dapibus justo. Etiam scelerisque a odio a euismod. Pellentesque vestibulum pellentesque nisi, a tempus erat cursus ac. Morbi ut auctor lectus.

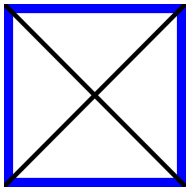


Figure 46.1. Testing numbering in a `<exercises>` without a number

3. Drill Three. Nam congue ex nec justo iaculis maximus. Vestibulum lobortis magna sed urna auctor, vel dignissim massa posuere. In sed venenatis elit. Vivamus congue gravida tempus. Nunc quis fermentum nisi. Nullam hendrerit lorem et tellus semper, sit amet scelerisque purus lobortis. Quisque sollicitudin quis neque eu suscipit. Praesent volutpat justo quis magna vehicula molestie.

4. Challenging One. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras congue urna nulla. Aliquam eget euismod tellus. Maecenas nibh libero, venenatis a laoreet in, tempor sit amet sem. Morbi sit amet justo tempor velit auctor placerat. Maecenas nec lobortis orci. Aenean dictum enim lacus, ac blandit lacus elementum nec. Mauris porttitor neque volutpat tincidunt sollicitudin. Cras porta lectus ac facilisis tempor. Suspendisse in velit nisl. Sed convallis leo at nunc aliquet fermentum. Pellentesque feugiat at ex sed elementum. In porta vulputate ipsum sit amet consectetur.

An introduction to an `exercisegroup`. This is here to attempt to interrupt the flow of the counting from this division to the next.

> **5. Challenging Two.** Aliquam sagittis ex at magna porttitor, quis scelerisque ligula malesuada. Vestibulum vitae mauris id nisi pretium ornare a id lectus. Nam suscipit magna id sem ultricies vestibulum. Nulla facilisi. Duis venenatis lectus massa, ac mollis nisi suscipit non. Sed et sapien vulputate, imperdiet nibh vel, lobortis nisl. Curabitur sagittis justo nibh.
>
> Nulla vulputate sodales justo at efficitur. Cras justo augue, mollis vitae nulla sed, vestibulum tincidunt ante. Vestibulum et lorem lectus. Maecenas accumsan lacus a nisi euismod rutrum eu dapibus justo. Etiam scelerisque a odio a euismod. Pellentesque vestibulum pellentesque nisi, a tempus erat cursus ac. Morbi ut auctor lectus.

6. Challenging Three. Nam congue ex nec justo iaculis maximus. Vestibulum lobortis magna sed urna auctor, vel dignissim massa posuere. In sed venenatis elit. Vivamus congue gravida tempus. Nunc quis fermentum nisi. Nullam hendrerit lorem et tellus semper, sit amet scelerisque purus lobortis. Quisque sollicitudin quis neque eu suscipit. Praesent volutpat justo quis magna vehicula molestie.

7. Impossible One. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras congue urna nulla. Aliquam eget euismod tellus. Maecenas nibh libero, venenatis a laoreet in, tempor sit amet sem. Morbi sit amet justo tempor velit auctor placerat. Maecenas nec lobortis orci. Aenean dictum enim lacus, ac blandit lacus elementum nec. Mauris porttitor neque volutpat tincidunt sollicitudin. Cras porta lectus ac facilisis tempor. Suspendisse in velit nisl. Sed convallis leo at nunc aliquet fermentum. Pellentesque feugiat at ex sed elementum. In porta vulputate ipsum sit amet consectetur.

8. Impossible Two. Aliquam sagittis ex at magna porttitor, quis scelerisque ligula malesuada. Vestibulum vitae mauris id nisi pretium ornare a id lectus. Nam suscipit magna id sem ultricies vestibulum. Nulla facilisi. Duis venenatis lectus massa, ac mollis nisi suscipit non. Sed et sapien vulputate, imperdiet nibh vel, lobortis nisl. Curabitur sagittis justo nibh. Nulla vulputate sodales justo at efficitur. Cras justo augue, mollis vitae nulla sed, vestibulum tincidunt ante. Vestibulum et lorem lectus. Maecenas accumsan lacus a nisi euismod rutrum eu dapibus justo. Etiam scelerisque a odio a euismod. Pellentesque vestibulum pellentesque nisi, a tempus erat cursus ac. Morbi ut auctor lectus.

9. Impossible Three. Nam congue ex nec justo iaculis maximus. Vestibulum lobortis magna sed urna auctor, vel dignissim massa posuere. In sed venenatis elit. Vivamus congue gravida tempus. Nunc quis fermentum nisi. Nullam hendrerit lorem et tellus semper, sit amet scelerisque purus lobortis. Quisque sollicitudin quis neque eu suscipit. Praesent volutpat justo quis magna vehicula molestie.

10. An Exercise in a Section. Exercises can appear in a "section" of their own. You need to give the section a title, even if it seems obvious what to call it. Individual exercises may have titles, as you choose. Problem: How should we hide solutions?

Solution. Maybe a global switch should be used to suppress solutions, while a separate processing regime could use them as part of a solutions manual.

42a. An Exercise with a Hard-Coded Problem Number. Compute the definite integral $\int_2^4 x^2\,dx$, not as an approximate value from a Riemann sum, but as an exact value based of the limit by using the Fundamental Theorem.

Solution. An antiderivative of x^2 is $F(x) = x^3/3$, so by the FTC,

$$\int_2^4 x^2\,dx = F(4) - F(2) = \frac{1}{3}\big(4^3 - 2^3\big) = \frac{56}{3}\text{!?!}$$

This is indeed an exciting result, but we are mostly interested in seeing that the sentence-ending punctuation is absorbed properly into the displayed equation.

And sentence-ending punctuation is also placed carefully when display math finishes with a LaTeX `cases` environment. To wit, the absolute value can be defined by

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0. \end{cases}$$

12. Can you prove Corollary 4.1 directly? If not consider that a problem could have several parts, which should be formatted as a second-level list, since the problems normally get numbered at the top level.

1. Why is this result a Corollary?
2. Could you interchange the Theorem and Corollary?

Hint 1 (MVT). Consider the definite integral as an area function and employ the Mean Value Theorem.

Hint 2 (Motivator). Think harder!

Answer (Helpful).

1. It follows easily.
2. Yes.

Solution. We could prove either result first, then obtain the other as an easy consequence.

A. Notation

This is some notation introduced in the article. PreTeXt allows the "notation list" generator anywhere, so we have this paragraph to test spacing above and below the table. We'll say that again.

This is some notation introduced in the article. PreTeXt allows the "notation list" generator anywhere, so we have this paragraph to test spacing above and below the table. We'll say that again.

Symbol	Description	Location
$\int_a^b f(x)\,dx$	definite integral of $f(x)$	Paragraph
$\int f(x)\,dx$	indefinite integral of $f(x)$	Definition 2.2
ρ	this symbol could be used for lots of things, but we are just trying to make a super-long description to get it to wrap within the column where it belongs, which is sometimes set to a fixed width to accomodate really complicated explanations	Paragraph
$\mathbb{Z}_n$	(ring of) integers modulo n	Paragraph
∇	gradient operator	Paragraph

This is some notation introduced in the article. PreTeXt allows the "notation list" generator anywhere, so we have this paragraph to test spacing above and below the table. We'll say that again.

This is some notation introduced in the article. PreTeXt allows the "notation list" generator anywhere, so we have this paragraph to test spacing above and below the table. We'll say that again.

Checkpoint A.1. List two pieces of notation that appear in this appendix.

B. Solutions to Selected Exercises

This is an introduction, where you might explain that this division of the back matter contains various hints, answers, solutions of inline exercises, divisional exercises, project-like blocks, worksheet exercises, and/or reading questions. See the source to see just how this solutions division was built.

4 · An Interesting Corollary
4.2 · A Pedagogical Note about Subsection 4.1
4.2.1 · Symbolic and Numerical Integrals

Checkpoint 4.6 Essay Question: Compare and Contrast.

Hint. Start writing!

4.2.2 · Advice

Checkpoint 4.9 An Inline Exercise.

Hint. A good hint.

Answer. 42.

Solution. If your exercise feels like proving a theorem, then you might want to make some comments, but also clearly delineate which part of the solution is a the complete proof.

Proof. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin lorem diam, convallis in nulla sed, accumsan fermentum urna. Pellentesque aliquet leo elit, ut consequat nunc dapibus ac. Sed lobortis leo tincidunt, vulputate nunc at, ultricies leo. Vivamus purus diam, tristique laoreet purus eget, mollis gravida sapien. Nunc vulputate nisl ac mauris hendrerit cursus. Sed vel molestie velit. Suspendisse sem sem, elementum at vehicula id, volutpat ac mi. Nullam ullamcorper fringilla purus in accumsan. Mauris at nunc accumsan orci dictum vulputate id id augue. Suspendisse at dignissim elit, non euismod nunc. Aliquam faucibus magna ac molestie semper. Aliquam hendrerit sem sit amet metus congue tempor. Donec laoreet laoreet metus, id interdum purus mattis vulputate. Proin condimentum vitae erat varius mollis. Donec venenatis libero sed turpis pretium tempor.

Praesent rutrum scelerisque felis sit amet adipiscing. Phasellus in mollis velit. Nunc malesuada felis sit amet massa cursus, eget elementum neque viverra. Integer sagittis dictum turpis vel aliquet. Fusce ut suscipit dolor, nec tristique nisl. Aenean luctus, leo et ornare fermentum, nibh dui vulputate leo, nec tincidunt augue ipsum sed odio. Nunc non erat sollicitudin, iaculis eros consequat, dapibus eros. ■

Activity 4.3 Hints, Answers, Solutions.

Answer. The result, but no help in getting there.

Checkpoint 4.12 A very structured exercise.

(c) A title of a task that has a subtask with an `<answer>` for the Solutions.

(i) A task with a title and an `<answer>` for the Solutions.

Answer. With a proof.

Proof. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis. ■

And a bit more to say.

(ii)

Answer. Right! In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(d)

Hint. One hint. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Answer 1. First answer. In interdum suscipit ullamcorper.

Answer 2. Second answer. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Solution. At last, the solution. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Project 4.4 A very structured project.

(c)

(i)

Answer. With a proof.

Proof. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis. ■

And a bit more to say.

(ii)

Answer. Right! In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(d)

Answer 1. First answer. In interdum suscipit ullamcorper.

Answer 2. Second answer. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Project 4.5.

Answer. Here's the answer we are looking for.

4.2.3 · Exercises

1.

Solution. This solution will migrate to a list of solutions in the backmatter. We include a `sidebyside` as a test.

This is a skinny paragraph which should be just 30% of the width.

And another skinny paragraph which should also be just 30% of the width.

4.6 · Introductions and Conclusions

4.6.2 · Test Two

Checkpoint 4.16.

Answer. An answer so there is something to appear in a `<solutions>`.

· What Did You Learn?

1.

Answer. An answer so there is something to appear in a `<solutions>`.

· Exercises

1.

Answer. An answer so there is something to appear in a `<solutions>`.

11 · Further Reading
11.3 · More Exercises

6.

Hint. Addition is associative.

Answer. 12

Solution. First, add 3 and 4 to get 7, then add 5 to arrive at 12.

8.

Answer. 15

11.

Hint. Addition is associative.

Answer. 12

Solution. First, add 3 and 4 to get 7, then add 5 to arrive at 12.

Proof. A simple argument. ■

And a bit more.

13.

Answer. 15

16.

Hint. Addition is associative.

Answer. 12

Solution. First, add 3 and 4 to get 7, then add 5 to arrive at 12.

18.

Answer. 15

21.

Hint. Addition is associative.

Answer. 12

Solution. First, add 3 and 4 to get 7, then add 5 to arrive at 12.

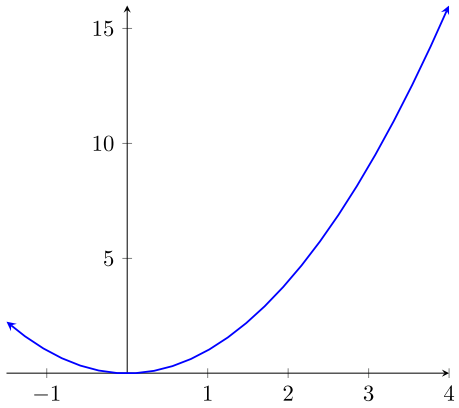
23.

Answer. 15

25.

Answer.

x	$f(x)$
0	0
1	1
2	4
3	9



16 · Interactive Exercises

16.2 · True/False

1. True/False.

Hint. P_n, the vector space of polynomials with degree at most n, has dimension $n+1$ by 2.1. [Cross-reference is just a demo, content is not relevant.] What happens if we relax the defintion and remove the parameter n?

Answer. False.

Solution. False.

The vector space of all polynomials with finite degree has a basis, $B = \{1, x, x^2, x^3, ...\}$, which is infinte.

16.3 · Multiple-Choice

1. Multiple-Choice, Not Randomized, One Answer.

Hint 1. What did you see last time you went driving?

Hint 2. Maybe go out for a drive?

Answer. B.

Solution.

A. *Incorrect.*
Green means "go!".
B. *Correct.*
Red is universally used for prohibited activities or serious warnings.
C. *Incorrect.*
White might be hard to see.

16.4 · Parsons Problem, Math Proof

1. Parsons Problem, Mathematical Proof.

Solution.

- Suppose n is even.
- Then there exists an m so that $n = 2m$.
- So we have the displayed equation:

$$n = 2m + 0.$$

  This is a superfluous second paragraph in this block.
- Thus $n \equiv 0 \mod 2$.

16.5 · Parsons Problem, Code

1. Parsons Problem, Programming.

Solution.

```
n = 250
primes = []
candidates = list(range(2,n))
while candidates:
    p = candidates[0]
    primes.append(p)
    for nonprime in range(p, n, p):
        if nonprime in candidates:
            candidates.remove(nonprime)
print(primes)
```

16.6 · Matching

1. Matching Problem, Dates.

Solution.

- 1823
  - Monroe Doctrine
- 1886
  - Haymarket Riot
- 1803
  - Louisiana Purchase
- 1863

◦ Battle of Gettysburg

16.7 · Clickable Area

1. Clickable Areas, "Regular" Text.

Answer. Correct: *future*; *beauty*; *dreams*. Incorrect: ~~those~~; ~~their~~.

The incorrect words are pronouns.

Solution. "The *future* belongs to ~~those~~ who believe in the *beauty* of ~~their~~ *dreams*."

16.8 · Old-Style Fillin-In

1. Fill-In, String and Number Answers.

Solution. Complete the following line of a Python program so that it will declare an integer variable `age` with an initial value of `5`.

int `age =` _5_;

A variable of type `int` is appropriate for whole number ages.

An integer variable may be initialized to a value.

17 · Dynamic Exercises

· Dynamic Fill-In

1. Fill-In, String and Number Answers.

Solution. Complete the following line of a Python program so that it will declare an integer variable `age` with an initial value of `5`.

int `age =` _5_;

2. Fill-In, Multiple Strings, Custom Checker.

Solution. In order to apply the Integral Test to a sequence $\{a_n\}$, the function $a(n) = a_n$ must be _continuous_, _positive_ and _decreasing_.

3. Fill-In Formula (Dynamic).

Solution 1. Find a formula for a cubic function $f(x)$ that has roots at $x = -5$, $x = -2$, and $x = 2$ and so that $f(0) = 5$.

$f(x) =$ _$-\frac{1}{4}(x+5)(x+2)(x-2)$_

Solution 2. Knowing the roots of a polynomial allows us to write down the formula of $f(x)$ in factored form,

$$f(x) = A(x+5)(x+2)(x-2)$$

with an unknown scaling multiple A.

When we evaluate $f(x)$ at $x = 0$ using this formula, we find

$$f(0) = -20A.$$

Since we also know $f(0) = 5$, we can write down the equation

$$A(x+5)(x+2)(x-2) = 5$$

and find that $A = -\frac{1}{4}$.

Consequently, we can write our function in the form

$$f(x) = -\frac{1}{4}(x+5)(x+2)(x-2).$$

4. Decompose the Function.

Solution 1. Consider the function

$$h(x) = 4(-x-3)^3 + 5.$$

Find two nontrivial functions $f(x)$ and $g(x)$ so that $h(x) = f(g(x))$.

$f(x) = _4x^3 + 5_$ and $g(x) = _-1x - 3_$

Solution 2. Noticing that the expression $-1x - 3$ appears inside parentheses with a power, it makes sense to think of that as the inner function, defining $g(x) = -1x - 3$. The outer function describes what happens to that. If we imagined replacing the formula $-1x - 3$ with a box and then call that box our variable x, we find the outer function is given by $f(x) = 4x^3 + 5$.

This is not the only non-trivial composition. Can you find others?

5. Fill-In, Remaining Machinery.

Solution. Both and its negative are accepted; the automatic solution is declined in favor of this sentence.

6.

Solution 1. Consider the matrix defined by

$$A = \begin{bmatrix} -5 & 12 & 75 \\ -3 & 7 & 44 \\ -32 & 77 & 481 \end{bmatrix}$$

Find the reduced row echelon form of A.

$$\text{rref}(A) = _\begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & 5 \\ 0 & 0 & 0 \end{bmatrix}_$$

A matrix is entered by listing the rows separated by commas between brackets [and], and for each row listing entries separated by commas between additional brackets. The matrix $\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$ would be entered as `[ [1, 3, 5], [2, 4, 6] ]`.

Solution 2. Row reduction of A gives

$$\text{rref}(A) = \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & 5 \\ 0 & 0 & 0 \end{bmatrix}$$

18 · Interactive Coding
18.3 · Activity with An ActiveCode

Activity 18.1 Activity Coding Exercise.

Answer. We're still not really sure.

32 · Advanced Numbering
32.8 · A Title with] a Right Bracket

Checkpoint 32.19 A Right Brace } and a Right Bracket].

Hint. A faux hint to get this exercise to migrate into a `<solutions>`.

32.11 · Just an Exercise

Checkpoint 32.20 An Extraneous Exercise.

Hint. A hint, so this exercise looks identical in structure to the one in the previous subsection.

34 · MyOpenMath Interactive Problems

Checkpoint 34.1 Negative Numbers and Exponents.

Solution.

- DNE

Checkpoint 34.2 A Statistical Test.

Solution.

- -0.928
- 0
- A: There is sufficient sample evidence to support the claim that there is a statistically significant correlation between the two variables.

Checkpoint 34.3 With an image.

Solution.

- $y = -2x - 6$

Checkpoint 34.4 Geometry of a Cone.

Solution.

- 5.55
- 2.66

35 · STACK Interactive Questions

1. STACK integration question.

Answer. $\frac{(x-3)^7}{7} + c$

Solution.
We can either do this question by inspection (i.e. spot the answer) or in a more formal manner by using the substitution

$$u = (x - 3).$$

Then, since $\frac{d}{dx}u = 1$ we have

$$\int (x-3)^6 dx = \int u^6 du = \frac{u^7}{7} + c = \frac{(x-3)^7}{7} + c.$$

2. STACK multiple choice question.

Answer. (1)

3. STACK matrix input question.

Answer. $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

4. STACK question with generated plot.

Answer. $(x-1)^2 + 2$

5. STACK question with advanced typesetting and embedded image.

Answer. $\frac{(b-a)m}{n+m} + a$

Solution.
The figure below shows the triangle with the vector $\overrightarrow{OP}$ marked with a dotted line within.

From the diagram we can see $\overrightarrow{OP} = \overrightarrow{OA} + \overrightarrow{AP}$. We are told $\overrightarrow{OA} = \underline{a}$ and so we need to find $\overrightarrow{AP}$. We know that P divides $\overrightarrow{AB}$ in the ratio $m : n$ so

$$\overrightarrow{AP} = \frac{m}{m+n}\left(\overrightarrow{AB}\right),$$

as

$$\overrightarrow{AB} = \underline{b} - \underline{a}$$

we find

$$\overrightarrow{AP} = \frac{m}{m+n}(\underline{b} - \underline{a})$$

and therefore,

$$\overrightarrow{OP} = \underline{a} + \frac{m}{m+n}(\underline{b} - \underline{a}).$$

Note: You can arrive at an equivalent expression by starting from $\overrightarrow{OP} = \overrightarrow{OB} + \overrightarrow{BP}$.

37 · Worksheets
37.1 · A Geometric Prelude (without authored pages)

2.

Hint. Use Worksheet Exercise 37.1.1.

37.2 · A Geometric Prelude (with authored pages)

2.

Hint. Use Worksheet Exercise 37.1.1.

37.5 · Worksheet

2.

Hint. This is a hint. You can hide it in the print preview before printing.

40 · Worksheet Solution Testing
40.1 · Subsection with only exercise

Checkpoint 40.1.

Answer. This is an exercise's answer.

Solution. This is the exercise's solution.

40.2 · Subsection with only project

Project 40.1.

Answer. This is a project's answer.

40.3 · Subsection with both exercise and project

Checkpoint 40.2.

Answer. This is an exercise's answer.

Solution. This is the exercise's solution.

Project 40.2.

Answer. This is a project's answer.

40.5 · Worksheet with only project

Project 40.3.

Answer. This is a project's answer.

40.6 · Worksheet with both exercise and project

Project 40.4.

Answer. This is a project's answer.

42 · Exercise Section, Structured

Easy Exercises

1. Arithmetic.

Answer. 7

With an Exercise Group

5. One.

Hint. Subtract.

43 · Exercises, Multiple Subsections
43.5 · Layout challenge

4.

Hint (OPEN ME!).

$$\begin{aligned} 3u + 1v + 3w + 2x + y \ &+ 3z = 10 \\ 2u + 2v - 2w + x \ &+ z \ = 6 \\ 1u - 3v + 1w + x \ &+ 3y + 2z = 13 \end{aligned}$$

45 · Runestone Components

Checkpoint 45.1 True/False, a True Statement.

Hint. Consider the three cases: negative, zero, positive.

Answer. True.

Solution. True.
Squaring preserves the sign of zero and erases the sign of everything else.

Checkpoint 45.2 Multiple Choice, Several Correct, Randomized.

Answer. A, C.

Solution.

A. *Correct.*
The only even prime.
B. *Incorrect.*
A unit, by convention not a prime.
C. *Correct.*
Indeed, no divisors but 1 and itself.

Checkpoint 45.3 Parsons, Vertical, Partial Ordering.

Solution.

```
def double(x):
    return 2 * x
n = int(input())
print(double(n))
```

Checkpoint 45.4 Parsons, Horizontal, Reused Blocks.

Solution. early to early to bed and early to rise

Checkpoint 45.5 Cardsort, with Distractors.

Solution.

- Polynomial
  - n^2
  - $100n^3$
- Exponential
  - 2^n
- Constant
- (Uncategorized)
  - $n \log n$

Checkpoint 45.6 Matching, Premises with References.

Solution.

- Product Rule
  - $(fg)' = f'g + fg'$
- Chain Rule
  - $(f \circ g)' = (f' \circ g) \cdot g'$
- Composite Rule
  - $(f \circ g)' = (f' \circ g) \cdot g'$

Checkpoint 45.7 Clickable Areas, Code.

Answer. Correct: `data.sort()`; `data.append(total)`. Incorrect: `total = sum(data)`.

Reading a list is not mutating it.

46 · Exercises, Top-Level

10. An Exercise in a Section.

Solution. Maybe a global switch should be used to suppress solutions, while a separate processing regime could use them as part of a solutions manual.

42a. An Exercise with a Hard-Coded Problem Number.

Solution. An antiderivative of x^2 is $F(x) = x^3/3$, so by the FTC,

$$\int_2^4 x^2\,dx = F(4) - F(2) = \frac{1}{3}\left(4^3 - 2^3\right) = \frac{56}{3}!?!$$

This is indeed an exciting result, but we are mostly interested in seeing that the sentence-ending punctuation is absorbed properly into the displayed equation.

And sentence-ending punctuation is also placed carefully when display math finishes with a LaTeX `cases` environment. To wit, the absolute value can be defined by

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0. \end{cases}$$

12.

Hint 1 (MVT). Consider the definite integral as an area function and employ the Mean Value Theorem.

Hint 2 (Motivator). Think harder!

Answer (Helpful).

1. It follows easily.
2. Yes.

Solution. We could prove either result first, then obtain the other as an easy consequence.

And a conclusion to this solutions division, which may not be readily apparent as distinct from the final division's worth of solutions, but since it is not prefixed with a number, it may be different enough.

C. Solutions to a Single Exercises Division

Sometimes you may want the solutions to exercises within a single `<exercises>` division (or similar, like a `<worksheet>`). Default behavior in this situation can produce two titles, when just one would be sufficient. Here we have used the `@scope` attribute to point to a specific `<exercises>`, and provided a (single) very specific `<title>`.

11.3 · More Exercises

6.

Hint. Addition is associative.

Solution. First, add 3 and 4 to get 7, then add 5 to arrive at 12.

11.

Hint. Addition is associative.

Solution. First, add 3 and 4 to get 7, then add 5 to arrive at 12.

Proof. A simple argument. ■

And a bit more.

16.

Hint. Addition is associative.

Solution. First, add 3 and 4 to get 7, then add 5 to arrive at 12.

21.

Hint. Addition is associative.

Solution. First, add 3 and 4 to get 7, then add 5 to arrive at 12.

D. All Solutions that are Answers

The back matter solutions appendix requests every `<answer>`, and only `<answer>`, of all types of exercises: inline, divisional, worksheet, reading questions, and projects. Now, as observed by Bruce Yoshiwara, it gets a bit tedious to see the "Answer" heading over and over in print, when every entry has that heading. So we squelch it for you in print/PDF output. (For HTML output we use knowls and need to have something to click on. But perhaps for EPUB we should be more careful?) Note that you might want to use a `<title>`, or an `<introduction>`, that explains which component of the exercises is being displayed, so there is no confusion.

4 · An Interesting Corollary
4.2 · A Pedagogical Note about Subsection 4.1
4.2.2 · Advice

Checkpoint 4.9 An Inline Exercise.

Answer. 42.

Activity 4.3 Hints, Answers, Solutions.

Answer. The result, but no help in getting there.

Checkpoint 4.12 A very structured exercise.

(c) A title of a task that has a subtask with an `<answer>` for the Solutions.

(i) A task with a title and an `<answer>` for the Solutions.

Answer. With a proof.

Proof. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis. ■

And a bit more to say.

(ii)

Answer. Right! In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(d)

Answer 1. First answer. In interdum suscipit ullamcorper.

Answer 2. Second answer. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Project 4.4 A very structured project.

(c)

(i)

Answer. With a proof.

Proof. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis. ■

And a bit more to say.

(ii)

Answer. Right! In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

(d)

Answer 1. First answer. In interdum suscipit ullamcorper.

Answer 2. Second answer. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit amet. Nulla at neque nisi. Integer vel porta felis.

Project 4.5.

Answer. Here's the answer we are looking for.

4.6 · Introductions and Conclusions
4.6.2 · Test Two

Checkpoint 4.16.

Answer. An answer so there is something to appear in a `<solutions>`.

· What Did You Learn?

1.

Answer. An answer so there is something to appear in a `<solutions>`.

· Exercises

1.

Answer. An answer so there is something to appear in a `<solutions>`.

11 · Further Reading
11.3 · More Exercises

6.

Answer. 12

8.

Answer. 15

11.

Answer. 12

13.

Answer. 15

16.

Answer. 12

18.

Answer. 15

21.

Answer. 12

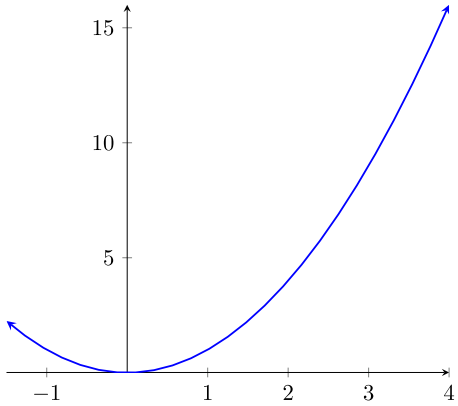
23.

Answer. 15

25.

Answer.

x	$f(x)$
0	0
1	1
2	4
3	9



16 · Interactive Exercises
16.2 · True/False

1. True/False.

Answer. False.

16.3 · Multiple-Choice

1. Multiple-Choice, Not Randomized, One Answer.

Answer. B.

16.7 · Clickable Area

1. Clickable Areas, "Regular" Text.

Answer. Correct: *future*; *beauty*; *dreams*. Incorrect: ~~those~~; ~~their~~.
The incorrect words are pronouns.

18 · Interactive Coding
18.3 · Activity with An ActiveCode

Activity 18.1 Activity Coding Exercise.

Answer. We're still not really sure.

35 · STACK Interactive Questions

1. STACK integration question.

Answer. $\frac{(x-3)^7}{7} + c$

2. STACK multiple choice question.

Answer. (1)

3. STACK matrix input question.

Answer. $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

4. STACK question with generated plot.

Answer. $(x-1)^2+2$

5. STACK question with advanced typesetting and embedded image.

Answer. $\frac{(b-a)m}{n+m}+a$

37 · Worksheets
37.5 · Worksheet

2.

Answer. Here is an answer. Also hidable.

40 · Worksheet Solution Testing
40.1 · Subsection with only exercise

Checkpoint 40.1.

Answer. This is an exercise's answer.

40.2 · Subsection with only project

Project 40.1.

Answer. This is a project's answer.

40.3 · Subsection with both exercise and project

Checkpoint 40.2.

Answer. This is an exercise's answer.

Project 40.2.

Answer. This is a project's answer.

40.4 · Worksheet with only exercise

1.

Answer. This is an exercise's answer.

40.5 · Worksheet with only project

Project 40.3.

Answer. This is a project's answer.

40.6 · Worksheet with both exercise and project

1.

Answer. This is an exercise's answer.

Project 40.4.

Answer. This is a project's answer.

42 · Exercise Section, Structured

Easy Exercises

1. Arithmetic.

Answer. 7

45 · Runestone Components

Checkpoint 45.1 True/False, a True Statement.

Answer. True.

Checkpoint 45.2 Multiple Choice, Several Correct, Randomized.

Answer. A, C.

Checkpoint 45.7 Clickable Areas, Code.

Answer. Correct: `data.sort()`; `data.append(total)`. Incorrect: `total = sum(data)`.

Reading a list is not mutating it.

46 · Exercises, Top-Level

12.

Answer (Helpful).

1. It follows easily.
2. Yes.

E. All Solutions that are Answers to Odd Exercises

This should be identical to the previous collection of answers, *except* only exercises with an even number are reported (via the `@admit` attribute).

4 · An Interesting Corollary
4.2 · A Pedagogical Note about Subsection 4.1
4.2.2 · Advice

Checkpoint 4.9 An Inline Exercise.

Answer. 42.

Activity 4.3 Hints, Answers, Solutions.

Answer. The result, but no help in getting there.

Project 4.5.

Answer. Here's the answer we are looking for.

4.6 · Introductions and Conclusions
4.6.2 · Test Two

· What Did You Learn?

1.

Answer. An answer so there is something to appear in a `<solutions>`.

· Exercises

1.

Answer. An answer so there is something to appear in a `<solutions>`.

11 · Further Reading
11.3 · More Exercises

11.

Answer. 12

13.

Answer. 15

21.

Answer. 12

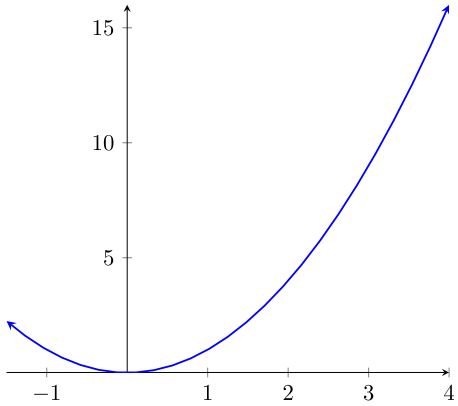
23.

Answer. 15

25.

Answer.

x	$f(x)$
0	0
1	1
2	4
3	9



16 · Interactive Exercises
16.2 · True/False

1. True/False.

Answer. False.

16.3 · Multiple-Choice

1. Multiple-Choice, Not Randomized, One Answer.

Answer. B.

16.7 · Clickable Area

1. Clickable Areas, "Regular" Text.

Answer. Correct: *future*; *beauty*; *dreams*. Incorrect: ~~those~~; ~~their~~.
The incorrect words are pronouns.

18 · Interactive Coding
18.3 · Activity with An ActiveCode

Activity 18.1 Activity Coding Exercise.

Answer. We're still not really sure.

35 · STACK Interactive Questions

1. STACK integration question.

Answer. $\frac{(x-3)^7}{7} + c$

3. STACK matrix input question.

Answer. $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

5. STACK question with advanced typesetting and embedded image.

Answer. $\frac{(b-a)m}{n+m} + a$

40 · Worksheet Solution Testing

40.1 · Subsection with only exercise

Checkpoint 40.1.

Answer. This is an exercise's answer.

40.2 · Subsection with only project

Project 40.1.

Answer. This is a project's answer.

40.4 · Worksheet with only exercise

1.

Answer. This is an exercise's answer.

40.5 · Worksheet with only project

Project 40.3.

Answer. This is a project's answer.

40.6 · Worksheet with both exercise and project

1.

Answer. This is an exercise's answer.

42 · Exercise Section, Structured

Easy Exercises

1. Arithmetic.

Answer. 7

45 · Runestone Components

Checkpoint 45.1 True/False, a True Statement.

Answer. True.

Checkpoint 45.7 Clickable Areas, Code.

Answer. Correct: `data.sort()`; `data.append(total)`. Incorrect: `total = sum(data)`.

Reading a list is not mutating it.

F. List of Results

We had an automatic list of theorems for just one section, back in Subsection 32.6. Here we expand to include `corollary` in our space-delimited list of `elements` and we request `divisions` (headings) at each `subsection` and `section`. The default `scope` is the entire document, which is appropriate here in the backmatter. There are many subsections with no results, so we set the `empty` attribute to `no` to suppress them, though this is the default behavior (`yes` being the other option to see divisions with no list items). These lists are most valuable if you are in the practice of giving items titles.

Section 2 The Fundamental Theorem

Theorem 2.1 The Fundamental Theorem of Calculus

Section 4 An Interesting Corollary

Subsection 4.1 Second Version of FTC

Corollary 4.1

Section 8 Grouping Samples

Subsection 8.3 Theorem-like Blocks

Theorem 8.14 A title
Theorem 8.15 A title
Theorem 8.16 A title
Theorem 8.17 A title
Theorem 8.18 A title
Corollary 8.19 A title

Subsection 8.6 Attributed Blocks

Theorem 8.33 Every Attribution
Corollary 8.35

Subsection 8.10 Blocks Closing with a List or a Display

Theorem 8.50 A Proof Closing with a List

Section 24 Program Listings (with code in the title)

Theorem 24.5

Section 32 Advanced Numbering

Subsection 32.1 One

Theorem 32.1 First Theorem
Theorem 32.2 Second Theorem

Subsection 32.2 Two

Theorem 32.3 First Theorem!
Theorem 32.4 Second Theorem?
Theorem 32.5 First Theorem?
Theorem 32.6 Second Theorem!

Subsection 32.3 Three

Theorem 32.7 First Theorem
Theorem 32.8 Second Theorem

Subsection 32.4 Four

Theorem 32.9 Good Numbered Theorem One
Theorem 32.10 Good Numbered Theorem Two
Theorem 32.11 First Theorem
Theorem 32.12 Second Theorem
Theorem 32.13 First Theorem
Theorem 32.14 Second Theorem
Theorem 32.15 Bad Numbered Theorem One
Theorem 32.16 Bad Numbered Theorem Two

Subsection 32.5 Five

Theorem 32.17 First Theorem
Theorem 32.18 Second Theorem

Section 37 Worksheets

Theorem 37.9 Ohms Law
Theorem 37.10 Kirchoffs Current Law
Theorem 37.12 Kirchoffs Voltage Law
Theorem 37.14 Ohms Law
Theorem 37.15 Kirchoffs Current Law

Theorem 37.17 Kirchoffs Voltage Law

Section 38 Handouts

Theorem 38.2

Section 41 Exercises, One Subsection

Theorem 41.3 Major Result

Section 43 Exercises, Multiple Subsections

Subsection 43.1 Faux Subsection

Theorem 43.3 Major Result

G. Lists of Exercises

Since `<exercise>` come in several flavors, we use **pseudo-elements** to specify the distinct types. These are the names a publication file uses for the same flavors. There are not many reading questions—but here is a list of all of them, by section, using an `exercise-reading` pseudo-element.

Section 4 An Interesting Corollary

Reading Question 4.2.4.1
Reading Question 4.2.4.2
Reading Question 4.6.2.1
Reading Question 4.6.2.2

Section 16 Interactive Exercises

Reading Question 16.9.1 Short Answer

And now a list of all the inline exercises, and including the title of every `<section>`, even if there is no inline exercise contained inside it. The pseudo-element is `exercise-inline`.

Section 1 Introduction

Section 2 The Fundamental Theorem

Section 3 Computing Integrals with Sage ()

Section 4 An Interesting Corollary

Checkpoint 4.6 Essay Question: Compare and Contrast

Checkpoint 4.9 An Inline Exercise
Checkpoint 4.12 A very structured exercise
Checkpoint 4.16

Section 5 Some Facts and Figures

Section 6 Some Advanced Ideas

Section 7 Mathematics

Section 8 Grouping Samples

Section 9 Entering Text in Paragraphs, Titles, Captions

Checkpoint 9.1

Section 10 Graphics

Section 11 Further Reading

Section 12 List Calisthenics

Section 13 Table Calisthenics

Section 14 Interactive Elements, Authored in Javascript

Checkpoint 14.2
Checkpoint 14.10 Graph Planarity
Checkpoint 14.13 Changing Secant Lines

Section 15 Interactive Elements, Server & Layouts

Section 16 Interactive Exercises

Section 17 Dynamic Exercises

Section 18 Interactive Coding

Section 19 Audio

Section 20 Video

Section 21 Cross-Referencing

Section 22 Internationalization

Section 23 Pre-Formatted Text

Section 24 Program Listings (with code in the title)

Checkpoint 24.8 Code in a colored(?) container

Section 25 Units of Measure

Section 26 Side-By-Side Panels

Section 27 Side-by-Side Gallery

Section 28 Open Problems

Section 29 Poetry

Section 30 Literate Programming

Section 31 Atomic Objects

Checkpoint 31.9

Section 32 Advanced Numbering

Checkpoint 32.19 A Right Brace } and a Right Bracket]
Checkpoint 32.20 An Extraneous Exercise

Section 33 Customizations

Section 34 MyOpenMath Interactive Problems

Checkpoint 34.1 Negative Numbers and Exponents
Checkpoint 34.2 A Statistical Test
Checkpoint 34.3 With an image
Checkpoint 34.4 Geometry of a Cone

Section 36 Ancillaries

Section 37 Worksheets

Section 38 Handouts

Section 39 Projects as Printouts

Section 40 Worksheet Solution Testing

Checkpoint 40.1
Checkpoint 40.2

Section 41 Exercises, One Subsection

Checkpoint 41.1 Inline One
Checkpoint 41.2 Inline Two
Checkpoint 41.4 Inline Three

Section 43 Exercises, Multiple Subsections

Checkpoint 43.1 Inline One
Checkpoint 43.2 Inline Two
Checkpoint 43.4 Inline Three

Section 44 Bibliography Markup Styles

Section 45 Runestone Components

Checkpoint 45.1 True/False, a True Statement
Checkpoint 45.2 Multiple Choice, Several Correct, Randomized
Checkpoint 45.3 Parsons, Vertical, Partial Ordering
Checkpoint 45.4 Parsons, Horizontal, Reused Blocks
Checkpoint 45.5 Cardsort, with Distractors
Checkpoint 45.6 Matching, Premises with References
Checkpoint 45.7 Clickable Areas, Code
Checkpoint 45.8 Select, an A/B Experiment
Checkpoint 45.9 Short Answer, with Attachment
Checkpoint 45.10 ActiveCode, Decorated

Checkpoint 45.11 CodeLens, with a Checkpoint
Checkpoint 45.12 A Dual Exercise (Experimental)
Checkpoint A.1

H. A Structured Appendix

A deeply-structured appendix for testing purposes.

H.1 A Subsection in an Appendix

Inside a subsection.

H.1.1 A Subsubsection in a Subsection in an Appendix

Nearly terminal.

Paragraphs in a Subsubsection in a Subsection in an Appendix. The `paragraphs` element can go in any division, but does not get a number.

I. Index

There is an index manufactured at the end of the back matter. So we are talking about it here, rather than within the index, which is an impossibility. It contains some sample entries, and is not meant to be comprehensive. Look at the source of this XML file, searching on `<idx>`, to see how they are written. They may be placed inside of a a variety of structures, and their location greatly influences the cross-references produced in the HTML version of the index.

The LaTeX version of the index is more traditional, using page numbers to reference locations. A newer package is used to create the index, and so there is no extra intermediate step required to process the index. The one downside of this convenience is that index entries may not be placed in the back colophon (which is the only subdivision that may follow the index).

There is an index entry about multicolumn lists which spans more than one page. This requires doubly-linked index entries, the first has the index content and points to the `xml:id` of the second. The second is an empty element, but points back to the `xml:id` of the first entry. So each has a marker and a reference, which allows the span of the index topic to cut across XML boundaries in the source. This is the mechanism to produce a page range in the LaTeX index. See the source of this article for syntax details.

Bully Pulpit: Index Headings. Professionals do not capitalize the headings (entries) of an index, unless it is a proper noun (name, place, etc.). We do not provide any enforcement of this advice, nor any assistance. It is your responsibility to provide quality source material in this regard.

Note. Most all of the index entries below to page 2 (PDF output) are just from a suite of non-sensical tests. These are harder to recognize in the HTML output.

J. Multiple References

J.1 Multiple Specialized References

You might want to have lists of references, in the back, but with multiple such lists. Make an `<appendix>` to hold them, give it some structure (for an `<article>`, a leading `<subsection>`, such as the one you are reading right now), then follow with multiple `<references>` divisions. A typical citation will then look like: [J.3.2].

2025-05-23: currently testing citations to CSL-style references in the back matter. These should go eventually somewhere besides right where the references are.

- Judson: [1]
- Lay: [2]
- Doe: [4]
- Doe, later: [3]
- Conrey/Farmer: [5]
- D'Arcus: [6]
- Two, authored in-order: [5, 6]
- Two, authored out-of-order: [6, 5]
- Three, authored reverse-order: [2, 1, 4]

J.2 General References

[1] Gilbert Strang, *The Fundamental Theorem of Linear Algebra*, The American Mathematical Monthly November 1993, **100** no. 9, 848–855.

J.3 Specialized References

[1] Tom Judson, *Abstract Algebra: Theory and Applications*.
Another online, open-source offering.

[2] Gilbert Strang, *The Fundamental Theorem of Linear Algebra*, The American Mathematical Monthly November 1993, **100** no. 9, 848–855.

Glossary

A glossary may have a `<headnote>`, perhaps with some explanation. This glossary is placed in the back matter. Placement as a specialized division is another option, see Glossary 4.2.5.

bar. A part of **foobar**. See foobar.

foobar. A synonym for the acronym FUBAR.

References

[1] Judson, Thomas, *Abstract Algebra: Theory and Applications*.

[2] Lay, David C., *Subspaces and Echelon Forms*, The College Mathematics Journal **24** no. 1 (1993) 57-62.

[3] Doe, John, *Still Anonymous, But now Ibid* (1970).

[4] Doe, John, *His Anonymous Life* (1965).

[5] Conrey, J. B. and D. W. Farmer, *Mean values of L-functions and symmetry*, Internat. Math. Res. Notices **17** (2000) 883-908.

[6] D'Arcus, Bruce, *Boundaries of Dissent: Protest and State Power in the Media Age* (2006), Routledge, New York `http://www.test01.com`.

Index

An index may begin with a `<headnote>` like this one. Indexing advice holds that a headnote is only necessary when the index has some unusual feature a reader should know in advance—perhaps how symbols are ordered, or that persons and topics are interfiled. This index has no such feature; this headnote is a demonstration. For a fuller discussion of this index, and of authoring index entries generally, see Appendix I.

$\int f(x)\,dx$, 92

A. *See* F; G; H
  B, 2
    C, 2 (*see also* A; D)
activity, 11
alert, 25
Archimedian Spiral, 124
aside, 12
assemblage, 24
asymptote graphics language, 71
attributes
  doctest, 3
  ncbi attribute, 58

biological names, 58
  genus, 58
  ncbi attribute, 58
  species, 58
  taxon, 58, 58
bug
  in-context broken, 161

A bug test, sorted as “c”
  b
    c one, 3
    c three, 3
  x
    c two, 3
c, 3
CalcPlot3D, 137
canceling a term, 39
`cancelto` macro, 39
sorted as if “Cat”, 2
CAYLEY GRAPH, 2, 2
Cayley graph, 2, 2
cayley graph, 2, 2
Charge of the Light Brigade
  second stanza, 202
Charge of the Light Brigade, The, Tennyson, 202
colors
  shades, 185, 185
conundrum
  repurposed from proposition, 225, 225
cross-reference, 157
  type name, 157

D3.js, 120
derivative
  cosine, 85
description list, 94, 95
Desmos graphs, 137
doctest, 3

em
  emphasis, 25
embedded interactive elements
  authored, 118
  server, 135
even numbers, 144
exercise group
  two derivatives, 84
exploration, 11

figure
  holding a paragraph, 83
fill-in blank, 85
Fire and Ice, Frost, 190
footnotes, 26
Fundamental Theorem of Calculus, 1
  Corollary, 6

gas, 5
gas masks, 5
gas production, 5
gas works, 5
gasoline, 5
gastritis, 5
genus, 58
Geogebra
  material, 136
  server, 136
glossary, 19, 318
Group, 91
group
  definition, 91

image, 178
  LaTeX image, 66

PreFigure image, 63
indefinite integral, 2
integral
indefinite integral, 2
interactive elements
authored, 118
layout, 135
server, 135
internationalization, 163
investigation, 11

JessieCode, 126
JSXGraph, 123

keyboard keys, 55
knowl
nested, 159

LaTeX image, 66
Led Zeppelin video, 154, 155
link
external, url, 56, 151
list, 99
description, 94, 95
multicolumn, 27
named, 98
ordered, 88, 88
unordered, 88, 88
listing, 168

matching US dates, 145
math in lists $\frac{d}{dx}(\sin(x)) = \cos(x)$, 92
matrix, 166
Maxwell's equations, 31
Mermaid diagrams, 74
mixed-content *emphasized*, 2

name, 225
named list, 98
ncbi attribute, 58
numbering, 221
numerical. *See* Sage; Sage integration
numerical integration
Sage
cell (*see* Sage)
numerics, 2. *See also* Sage

ordered list, 88, 88

package
siunitx, 175
panels, 176
paragraph
initial list, 91
normal, 100
opens with list, 100
poem, 190, 202
PreFigure image, 63
prime numbers, 145
problem, 11
program listing, 168
project, 11, 11
PSTricks, 71

QR code, 66
question, 11
sorted as if "Quorum", 2
quotations, 51

R, 3
ρ–fibers, 2
reading questions, 18
reference
external, 162
external, url, 56, 151
references
within a section, 88
rename an environment, 11
conundrum, 225
rho, a test, 30
Runestone components, 282

Sage
integration, 2
cell, 2
numerical, 2
interacts, 127
Sage cell
with a title, 5
Sage cells
linking, 22
Sage plots, 77
scientific names. *See* biological names
Sieve of Eratosthenes, 145
siunitx package, 175
solutions
all answers, 302
all even answers, 307
as an appendix, 288
for a division, 19
for one group of exercises, 301
Sonnet to Liberty, Wilde, 191
species, 58
STACK, 229
Stolen Child, The, Yeats, 192
stop signs, 144

strange colors, 27, 27
structured-content *emphasized*, 2
styling words
  alert, 25
  delete, 25
  em, 25
  emphasis, 25
  insert, 25
  stale, 25

table, 182
task, 11
taxon, 58
test: buried in theorem/statement/
  p, 1

Underdown, Jason, 39
Unicode, 163
units, 175
  Z (***sort as*** A), 2
  A (***sort as*** Z), 2
unordered list, 88, 88
url, 162

vector space, 143
`verbatim text`, use sortby, 2
videos
  YouTube, 154

x as a variable, 2, 2
X, 2. *See also* F; G; H
  Y, 2
    Z (*see* A; D)

YouTube videos, 154

Colophon

This article was authored in, and produced with,
PreTeXt. It is typeset with the Latin Modern font.