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.

2 The Fundamental Theorem

There is a remarkable theorem:

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

\[ \int_{a}^{b} f(x) \, dx = F(b) - F(a) \]

**Proof.** Left to the reader. ■

You will find almost nothing about all this in the article [2], nor in the book [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 (((some textbook about the FTC))) or we might want to reference a later (((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

---

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

\[
\text{numerical_integral}(\sin(x)^2, (0, 2))
\]

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

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

\[
\text{integral}(\sin(x)^2, x)
\]

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

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.

\[
\text{integral}(\sin(x)^2, (x, 0, 2))
\]

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

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.

\[
\text{integral}(e^{x^2}, x)
\]

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).

\[
\text{random()}
\]

\(0.11736021338650582\)
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 \( \epsilon = 0.006 \).

\[
12 + 0.01*\text{random()}
\]

12.005

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 `&lt;` as we have done in the source for this example. (See the discussion in Subsection 8.1.) As a reader you learn that the “Evaluate” button for a pre-loaded Sage cell will indicate the language in use.

```r
ruth <- c(22, 25, 34, 41, 41, 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, `&lt;` several times (see Subsection 8.1).

```r
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)
```

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. (The linking seems a bit buggy, as it repeats the boxplot in the output, as of 2016-06-13). 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. 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.

Titled Sage Cells.

\[
\text{integral}(\sin(x)^2, x)
\]

\[
\frac{1}{2}x - \frac{1}{4}\sin(2x)
\]

You can place Sage cells inside of a paragraphs if you want to give them a title, but no numbers, etc. Their surrounding box sometimes gets clobbered in \LaTeX output if they are the first piece of content, so we test that here also.

4 An Interesting Corollary

Objectives: Fundamental Structures

This is an \texttt{<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.
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 \texttt{type-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.
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
\[
\frac{d}{dx} \int_a^x f(t) dt = f(x).
\] (4.1)

Proof. We simply take the indicated derivative, applying Theorem 2.1 at (4.2)
\[
\frac{d}{dx} \int_a^x f(t) dt = \frac{d}{dx} (F(x) - F(a))
\] (4.2)
\[
= \frac{d}{dx} F(x) - \frac{d}{dx} F(a)
\]  
\[
= f(x) - 0 = f(x).
\] (4.3)

Alternate 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.
\[
\frac{d}{dx} \int_a^x f(t) dt = \frac{d}{dx} (F(x) - F(a))
\] (4.4)
\[
= \frac{d}{dx} F(x) - \frac{d}{dx} F(a)
\]  
\[
= f(x) - 0 = f(x)
\]

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.

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).
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.

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

\((\Leftarrow)\) 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.

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

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

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

Exact 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. ■

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

This subsection has a title in the source, but it is empty. That’s OK, but not advisable since titles get used lots of places (such as page headers and the table of contents).

4.2.3 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

\(^1\)Which I think sometimes students lose sight of.
not conflating the notation for two very different things until the Fundamental Theorem exposes them as being highly related.

**Example 4.5 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 4.6 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 examples, 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 4.7 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.** What was the question?

**Example 4.8 An Example of with $\int e^x \, dx$ 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 `--stringparam numbering.projects.level 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 26.1).

**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).

This is quite the activity upcoming. This is a prelude authored within the activity element, but visually just prior.

**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.

This was quite the activity just now. This is a *postlude* authored within the *activity* element, but visually just after.

**Note 4.9 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.

**An Aside.** 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.

The next block is a project, demonstrating the use of the *task* element to structure its parts. You are reading the *prelude* now. The project has lots of nonsense words, so we can test spacing the nested 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.


**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.


(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

In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id
vestibulum magna. Nulla blandit dui metus, malesuada mollis sapien

(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. In interdum suscipit ullamcorper. Morbi sit amet male-
suada augue, id vestibulum magna. Nulla blandit dui metus, male-
suada mollis sapien ullamcorper sit amet. Nulla at neque nisi. In-
teger vel porta felis.

(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, male-
suada 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, male-
suada mollis sapien ullamcorper sit amet. Nulla at neque nisi.
Integer vel porta felis.

The conclusion of the structured subtask. In interdum suscipit ul-
łamcorper. Morbi sit amet malesuada augue, id vestibulum magna.
Nulla blandit dui metus, malesuada mollis sapien ullamcorper sit

(iv) A simple task as the last subtask. In interdum suscipit ullamcorper.
Morbi sit amet malesuada augue, id vestibulum magna. Nulla blan-
dit 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.


**Answer 1.** First answer. In interdum suscipit ullamcorper.


**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.

This postlude appears visually outside the project, but is authored within, to make clear its attachment to 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.

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

**Technology 4.10 Sample Use of Sage.** This would be a good place to talk about Sage, including a cell or two.

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

```
4*x^3
```

But you might want to describe how to use some other calculator, or maybe some numerical method.
4.2.4 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 4.2.4.1.

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.

4.2.5 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 /grave.ts1 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 4.2.5.1. 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>.

4.2.6 Glossary

A glossary may have an <introduction>, perhaps with some explanation.

bar. A part of foobar. See foobar.

foobar. A synonym for the acronym fubar.

And for symmetry, a <conclusion>.

4.2.7 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.
Checkpoint 4.4 Essay Question: Compare and Contrast. Hint. Start writing!

4.2.3 · Advice

Checkpoint 4.7 An Inline Exercise. Hint. A good hint.

Answer. 42.

Solution. What was the question?

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.

Project 4.4 A very structured project.


Answer 1. First answer. In interdum suscipit ullamcorper.


4.2.4 · Exercises
4.2.4.1. 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 4.11 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, \texttt{<theorem>}, \texttt{<corollary>}, \texttt{<lemma>}, \texttt{<algorithm>}, \texttt{<proposition>}, \texttt{<claim>}, \texttt{<fact>}, and \texttt{<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 \texttt{<axiom>}, \texttt{<conjecture>}, \texttt{<principle>}, \texttt{<heuristic>}, \texttt{<hypothesis>}, and \texttt{<assumption>} are functionally the same, barring a proof (since they would never have one!). Definitions are an exception, as it is natural to place \texttt{<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, 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 \texttt{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.

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.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uno</td>
<td>Dos</td>
<td>Tres</td>
</tr>
</tbody>
</table>

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

**Assemblages containing $\mu \forall \tau \ll = 0\kappa$.**

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

### 4.6 Introductions and Conclusions

**An Introductory Introduction.** 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. A title is optional (and probably not advisable). 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 subdivision 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 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.

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 incomplete cross-references for testing purposes only: (((a first incomplete 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: 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) \(\frac{d}{dx}x^n = nx^{n-1}\)
  - (b) \(\frac{d}{dx}e^x = e^x\)
  - (c) \(\frac{d}{dx}\cos(x) = -\sin(x)\)

- Antiderivatives
  - i) \(\int x^n \, dx = \frac{x^{n+1}}{n+1}\) if \(n \neq -1\)
  - ii) \(\int e^x \, dx = e^x\)
  - iii) \(\int \sin(x) \, dx = -\cos(x)\)

**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](image)

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

---

1. First test footnote
2. Second test footnote
3. Third test footnote
4. Fourth test footnote
5. Fifth test footnote
6. Sixth test footnote
7. Seventh test footnote
examples, one with short snippets and lots of columns, the other with lots of text in paragraphs.


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 tempor in vitae metus. Aliquam arcu turpis, mattis non libero eu, lacinia feugiat turpis. Phasellus rhoncus lacinia lacus facilisis ullamcor-
Ut nec sem vitae ipsum inter-

dum vestibulum sit amet sed

velit. Aliquam tempor nibh vi-
tae augue pulvinar, at ultricies

urna commodo. Donec in porta
lectus, ac sagittis felis. Vestibu-

ulum tincidunt quis metus facil-
isis luctus. In lobortis lacus
vel ornare vehicula. Duis ali-

quet, ligula semper sodales adip-

iscing, augue nibh ornare ante,
quis pulvinar justo mi eget mi.

Mauris varius imperdiet vehic-

ula. Duis dignissim magna quis
velit mattis, in cursus lectus ve-

hicula. Morbi quis tempus felis,

ut gravida nisi.

Vivamus eu commodo est,

pretium fringilla dolor. Cur-

abitur vel sollicitudin libero. In-

teger sit amet auctor felis. Maec-

enas sagittis erat at ante feu-

giat, in tincidunt ligula pretium.

Integer eget auctor ipsum, quis

volutpat felis. Morbi id dignis-

sim eros. Suspendisse aliquet

pulvinar lorem gravida egestas.

Cum sociis natoque penatibus et

magnis dis parturient montes,

nascetur ridiculus mus. Prae-

sent nec massa dui. Suspendisse

convallis lacus sit amet adip-

iscing varius. Suspendisse tem-

pus diam vitae justo ornare, in

condimentum metus pharetra.

Curabitur sem dolor, auctor

vitae sagittis vestibulum, po-

suere imperdiet metus. Etiam

pretium lacus urna, vel auc-

tor diam tincidunt non. Etiam

viverra sodales iaculis.

• Sed varius leo urna. Phasellus

tempus mollis ultricies. Cur-

abitur non neque aliquet, fa-

cilisis tortor in, sodales dui.

Donec hendrerit ultricies nulla
mollis rhoncus. In vel lobortis

est. Vestibulum consectetur la-
cus vel sem dignissim vestibul-

um. 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 cu-
bilia Curae; In et libero id turpis

pharetra faucibus. Integer con-

sequat dignissim 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 mau-

ris, porta in nulla eget, biben-
dum 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 fe-
lis a enim blandit, at venenatis
diam congue. Nulla augue diam,

egestas eget fermentum nec, po-

suere eget risus. Praesent eges-
tas nulla eros, eget accumsan au-

gue euismod vel. Pellentesque

pellentesque non erat vitae po-

suere. Curabitur lacus arcu, var-

ius sed risus ut, ullamcorper tin-
cidunt lorem. Sed et lacus dig-
nissim, tincidunt nisl ac, portti-
tor sapien.
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,

\[
L(y')(s) = sL(y)(s) - y(0) = sY(s) - y(0)
\]

\[
L(y'')(s) = s^2L(y)(s) - sy(0) - y'(0) = s^2Y(s) - sy(0) - y'(0).
\]

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 amount 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.

\[
L(y')(s) = sL(y)(s) - y(0) = sY(s) - y(0) \tag{6.1}
\]

\[
L(y'')(s) = s^2L(y)(s) - sy(0) - y'(0) = s^2Y(s) - sy(0) - y'(0). \tag{6.2}
\]

And so it follows that,

\[
L(y')(s)^{++} = sL(y)(s) - y(0) = sY(s) - y(0) \tag{6.3}
\]

\[
L(y'')(s)^5 = s^2L(y)(s) - sy(0) - y'(0) = s^2Y(s) - sy(0) - y'(0). \tag{6.4}
\]

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

\[
L(y')(s) = sL(y)(s) - y(0) = sY(s) - y(0)
\]

\[
L(y'')(s) = s^2L(y)(s) - sy(0) - y'(0) = s^2Y(s) - sy(0) - y'(0).
\]

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

\[
L(y')(s)^{++} = sL(y)(s) - y(0) = sY(s) - y(0)
\]

\[
L(y'')(s)^5 = s^2L(y)(s) - sy(0) - y'(0) = s^2Y(s) - sy(0) - y'(0).
\]

Tables can get quite complex. Simple ones are simpler, such as this example of numerical computations for Euler’s method.

<table>
<thead>
<tr>
<th>(i)</th>
<th>(t_i)</th>
<th>(x_i)</th>
<th>(y_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00</td>
<td>0.0000</td>
<td>0.5000</td>
</tr>
<tr>
<td>1</td>
<td>0.20</td>
<td>0.1000</td>
<td>0.4800</td>
</tr>
<tr>
<td>2</td>
<td>0.40</td>
<td>0.1960</td>
<td>0.4560</td>
</tr>
<tr>
<td>3</td>
<td>0.60</td>
<td>0.2872</td>
<td>0.4295</td>
</tr>
<tr>
<td>4</td>
<td>0.80</td>
<td>0.3731</td>
<td>0.4027</td>
</tr>
<tr>
<td>5</td>
<td>1.00</td>
<td>0.4536</td>
<td>0.3783</td>
</tr>
<tr>
<td>6</td>
<td>1.20</td>
<td>0.5293</td>
<td>0.3591</td>
</tr>
<tr>
<td>7</td>
<td>1.40</td>
<td>0.6111</td>
<td>0.3480</td>
</tr>
<tr>
<td>8</td>
<td>1.60</td>
<td>0.6707</td>
<td>0.3474</td>
</tr>
<tr>
<td>9</td>
<td>1.80</td>
<td>0.7402</td>
<td>0.3603</td>
</tr>
<tr>
<td>10</td>
<td>2.00</td>
<td>0.8123</td>
<td>0.3900</td>
</tr>
</tbody>
</table>

Table 6.1: Euler’s approximation for Duffing’s Equation with \(h = 0.2\)
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. For a complete list, see the MathJax Supported LaTeX commands. We load the AMSsymbols library and the library for extensible arrows, extpfeil.

7.1 Basic Mathematics

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

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

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

$$\nabla \times \vec{B} - \frac{1}{c} \frac{\partial \vec{E}}{\partial t} = \frac{4\pi}{c} \vec{j}$$
$$\nabla \cdot \vec{E} = 4\pi \rho$$

$$\nabla \times \vec{E} + \frac{1}{c} \frac{\partial \vec{B}}{\partial t} = \vec{0}$$
$$\nabla \cdot \vec{B} = 0$$

A small test that the extensible arrows library is included properly:

$$A \xrightarrow{\Phi+\Psi+\Theta} \text{bijection} B$$

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

Chris Hughes has made available “slanted”, or “beveled”, or “nice” fractions. To wit, we mean fractions such as: $\frac{3}{8}$. Use the built-in \sfrac{}{} macro in your mathematics to achieve this presentation.

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

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

7.2 Displayed Mathematics

Multi-line displays of mathematics are achieved with the md tag (“math display”), and the variant that produces numbers on each line, mdn (“math display numbered”), 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 \texttt{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 AMSSmath \LaTeX\ package’s \texttt{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 \texttt{\&amp} as one way specify these, but the escape sequence \texttt{\&amp}; 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 (\texttt{\amp\, =}), or for a column of explanations or commentary, just prior to the \texttt{\text{}} macro. Note that this scenario suggests always having an odd number of ampersands in each \texttt{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 \texttt{\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 \texttt{m} tag will be ineffective).

\[
\frac{dx}{dt} = x^2 - 4x - y + 4 \\
\frac{dy}{dt} = x^3 - y \\
\frac{dw}{dt} = z^3 - w \\
\frac{dz}{dt} = z^2 - 4z - w + 4
\]

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 \texttt{\&alignment} attribute can override automatic detection. We use a simple \LaTeX\ macro to demonstrate setting \texttt{alignment="align"} to override the use of a \texttt{gather} environment and use a \texttt{align} environment instead. Example:

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

The AMSSmath \LaTeX\ package’s \texttt{alignat} environment is a third variant of alignment. It never happens automatically, you need to ask for it with \texttt{alignment="alignat"}. It is very similar to \texttt{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:

\[
\frac{dx}{dt} = x^2 - 4x - y + 4 \\
\frac{dy}{dt} = x^3 - y \\
\frac{dw}{dt} = z^3 - w \\
\frac{dz}{dt} = z^2 - 4z - w + 4z, w\ version.
\]

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 \texttt{mrow}
with the greatest number of ampersands (and how many there are),

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

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.

\[
A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.
\]

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:

\[
\begin{align*}
2x + y + 3z &= 10 \\
x + z &= 6 \\
x + 3y + 2z &= 13.
\end{align*}
\]

Beautiful.

A long equation, to check layout on various screen sizes. This is Weil’s “explicit formula” for the Riemann \( \zeta \)-function:

\[
\sum_{\gamma} S_{-}(\gamma) = \frac{\log Q}{\pi} \hat{S}_-(0) + \frac{1}{2\pi} \sum_{j=1}^{d} \Re \left\{ \int_{-\infty}^{\infty} \frac{\Gamma(t)}{\Gamma(t/2 + \mu_j)} S_-(t) dt \right\} - \frac{d}{2\pi} \hat{S}_-(0) \log \pi.
\]

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.

\[
\begin{align*}
x^2 + y^2 &= z^2 \\
a^2 + b^2 &= c^2 \\
\alpha^2 + \beta^2 &= \gamma^2 \\
m^2 + n^2 &= p^2 \\
x^2 + y^2 &= z^2 \\
a^2 + b^2 &= c^2 \\
\alpha^2 + \beta^2 &= \gamma^2 \\
m^2 + n^2 &= p^2 \\
x^2 + y^2 &= z^2 \\
a^2 + b^2 &= c^2
\end{align*}
\]
\[ \alpha^2 + \beta^2 = \gamma^2 \]
\[ m^2 + n^2 = p^2 \]
\[ x^2 + y^2 = z^2 \]
\[ a^2 + b^2 = c^2 \]
\[ \alpha^2 + \beta^2 = \gamma^2 \]
\[ m^2 + n^2 = p^2 \]
\[ x^2 + y^2 = z^2 \]
\[ a^2 + b^2 = c^2 \]
\[ \alpha^2 + \beta^2 = \gamma^2 \]
\[ m^2 + n^2 = p^2 \]
\[ x^2 + y^2 = z^2 \]
\[ a^2 + b^2 = c^2 \]
\[ \alpha^2 + \beta^2 = \gamma^2 \]
\[ m^2 + n^2 = p^2 \]
\[ x^2 + y^2 = z^2 \]
\[ a^2 + b^2 = c^2 \]
\[ \alpha^2 + \beta^2 = \gamma^2 \]
\[ m^2 + n^2 = p^2 \]
\[ x^2 + y^2 = z^2 \]
\[ a^2 + b^2 = c^2 \]
\[ \alpha^2 + \beta^2 = \gamma^2 \]
\[ m^2 + n^2 = p^2 \]
\[ x^2 + y^2 = z^2 \]
\[ a^2 + b^2 = c^2 \]
\[ \alpha^2 + \beta^2 = \gamma^2 \]
\[ m^2 + n^2 = p^2 \]
\[ x^2 + y^2 = z^2 \]
\[ a^2 + b^2 = c^2 \]
\[ \alpha^2 + \beta^2 = \gamma^2 \]
\[ m^2 + n^2 = p^2 \]

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.

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

This is a poorly-authored paragraph 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 use macros from a \LaTeX package and there is a MathJax extension of the same name which implements the same macros, then you may use these 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{1}{}`.

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

Look at the source of this article to see the package name being supplied in a \texttt{latex-preamble/package} element within the \texttt{docinfo} section. That is the only setup required to make the macro usable in \LaTeX and HTML output.

The packages appear before the author-supplied macros, so you can use macros from the packages as building blocks for document-specific macros. We cannot guarantee there will be no conflicts between additional packages and those in use normally, or added in the future. So use at your own risk.

### 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 sup-
ported 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.

\[ \alpha \text{ is not equal to } \beta \]
\[ \alpha \neq \beta \]

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.

\[ y \rightarrow \frac{\sin(0)}{0} \rightarrow "0" \]

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.

\[ \zeta(1) = \sum_{n=1}^{\infty} \frac{1}{n} \stackrel{\text{"=}\text{}}{=} \prod_{p} \left( \frac{1}{1-1/p} \right) = \prod_{p} \left( \frac{1}{1-p^{-1}} \right) \]

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,

\[ A = B + C \quad \text{Corollary 4.1} \]
\[ = D + E \quad \text{The Fundamental Theorem of Calculus.} \]

### 7.5 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{}, only. (Remember, you can have just one \mrow.) The behavior is identical within an \md{} or \mdn{}. 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
Cross-references to these tagged equations happens in the usual way and should behave as expected.

\[ c^2 = a^2 + b^2 \]  
\[ z^2 = x^2 + y^2 \]

Here is a local cross-reference: (**) We test another farther away in Section 17, contrary to our advice above.

8 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 8.1 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.

8.1 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 &lt; 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 &amp;.

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 \&lt; and \&amp; 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 describe mathematical expressions. XML has three other escape sequences \&gt;, \&apos;, and \&quot; 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 8.1 How was \&amp; authored? Work it out, and then check the source here for the answer.
8.2 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 24 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.

- “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”
N.B. We have taken no special care to protect against interactions of the actual quote characters (described above) in LaTeX with themselves, or with the grouping tags.

8.3 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.

8.4 Characters, Symbols, and Constructions

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. Here we test many of these. Read the Author’s Guide for tags and more detail.

© ® ª ™ SM … · ~ % © ¶ § × /

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

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

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. The command line stringparam `emdashespace` can be set to `none` or `thin` to control this.

### 8.5 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 `\&\#x20B1;`. 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>
<thead>
<tr>
<th>Sign</th>
<th>Unicode</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>U+0024</td>
<td>dollar</td>
</tr>
<tr>
<td>¢</td>
<td>U+00A2</td>
<td>cent</td>
</tr>
<tr>
<td>£</td>
<td>U+00A3</td>
<td>sterling</td>
</tr>
<tr>
<td>₤</td>
<td>U+00A4</td>
<td>currency</td>
</tr>
<tr>
<td>¥</td>
<td>U+00A5</td>
<td>yen</td>
</tr>
<tr>
<td>ℓ</td>
<td>U+0192</td>
<td>florin</td>
</tr>
<tr>
<td>B</td>
<td>U+0E3F</td>
<td>baht</td>
</tr>
<tr>
<td>C</td>
<td>U+20A1</td>
<td>colon</td>
</tr>
<tr>
<td>£</td>
<td>U+20A4</td>
<td>lira</td>
</tr>
<tr>
<td>N</td>
<td>U+20A6</td>
<td>naira</td>
</tr>
<tr>
<td>₪</td>
<td>U+20A9</td>
<td>won</td>
</tr>
<tr>
<td>d</td>
<td>U+20AB</td>
<td>dong</td>
</tr>
<tr>
<td>€</td>
<td>U+20AC</td>
<td>euro</td>
</tr>
<tr>
<td>P</td>
<td>U+20B1</td>
<td>peso</td>
</tr>
<tr>
<td>G</td>
<td>U+20B2</td>
<td>guarani</td>
</tr>
</tbody>
</table>

**Table 8.2: Supported Currency**

### 8.6 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>
<thead>
<tr>
<th>Name</th>
<th>Icon</th>
<th>Name</th>
<th>Icon</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrow-down</td>
<td>arrow-down</td>
<td>arrow-left</td>
<td>arrow-left</td>
</tr>
<tr>
<td>arrow-up</td>
<td>file-save</td>
<td>gear</td>
<td></td>
</tr>
<tr>
<td>menu</td>
<td>wrench</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 8.3: User-Interface Icons**

Nominations of new icons must

- Have a Unicode character representation.
- Be in the HTML/CSS/JS Font Awesome catalog.
- Be in the LaTeX fontawesome 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 8.4 Icons, xelatex, and Fonts.** When processing a LaTeX file with xelatex the Font Awesome icons are expected to be in a system font whose name is Font Awesome. This is not a filename, and installing the LaTeX fontawesome package into your LaTeX installation does not mean you have made this font 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.

8.7 URLs, such as http://example.com

An internet URL can contain a limited subset of keyboard characters. So we include 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 attribute, but you may need to provide XML escape sequences, see 8.1). 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. Three versions: one with the content as normal text, one with the characters as verbatim text, and then as a URL with the same string of characters as the address and the clickable, viewable part.

```
ABCDEFHGIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789%-_.~/?#\[]@!$&'()*+,;=
```

The source of the four above examples can be instructive.

- Four ampersands need to be authored as \&; two href attributes and two strings of verbatim text.
- Two ampersands are authored as <ampersand \slash{>}: two strings of normal text.

When a url has no content, then its href attribute is displayed as the text, automatically in a typewriter font.

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

We are not fans of footnotes, they are totally unstructured.\footnote{Carleson’s Theorem} A URL in a footnote migrates around, and so care must be taken.\footnote{http://example.com/ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789%58-._~:/?#\[]@!$&'()*+,;=} This paragraph has two footnotes, one with a real URL from Jesse Oldroyd, another with a fake URL from the above suite (the fourth one). For good measure, we repeat the URL found in the first footnote: Carleson’s Theorem. And we include a no-content version of the same link: https://en.wikipedia.org/wiki/Carleson%27s_theorem.

8.8 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 Cyclops kolensis. 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

\footnote{Carleson’s Theorem}
\footnote{http://example.com/ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789%58-._~:/?#\[]@!$&'()*+,;=}
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 *Drosophila miranda* which could be used to construct a link to further information via id number or even further information via just the name.

8.9 Verbatim in titles, \&\texttt{\textbackslash 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 with \texttt{--stringparam toc.level 2}.

8.10 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 8.1).

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

**Univariate Polynomial Ring in t over Integer Ring**

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

```plaintext
R = ZZ['u']
```

```plaintext
u = R.gen(0)
```

```plaintext
(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 \texttt{<!\[CDATA[ ]]>}. It should be 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 misunderstood as a solution better remedied by reading Subsection 8.1 again.

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

\[
\begin{align*}
  a^2 + b^2 &< c^2 \\
  c^2 &> a^2 + b^2
\end{align*}
\]

8.11 Jupyter Notebook, Markdown, MathJax, Delimiters

A Jupyter notebook allows a mix of HTML (our logistical preference for a conversion) and Markdown (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 \$

33
8.12 \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: \# $ % ^ & _ { } \sim \backslash. And again: X#X$X%X&X_X{X}X\sim X\backslash, but smashed up tight to intermediate characters.

In a verbatim presentation: \# $ % ^ & _ { } \sim \backslash.

And X#X$X%X^X&X_X{X}X\sim X\backslash. (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 \textbackslash newpage, will fail since the leading backslash will be caught and converted to \textbacks/l.Varash. (See? It just happened twice.) For technical reasons we want to particularly test \textbackslash, \textbraceleft, and \textbraceright.

Four “\TeX ligatures”, --, ---, /grave.Var/grave Var, and /quotesingle.Var/quotesingle Var, 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 \textbackslash{} and \textbackslash{} 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).

Runsofbackticks(accentgrave)shouldnotbecomesmartleftdoublequotes 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 hyphenate anywhere. PreTeXXt 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 \textsolidus 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\text/test/of/some/short/words/that/go/off/the/end/of/a/line/A\text/test/of/some/short/words/that/go/off/the/end/of/a/line/A\text/test/of/some/short/words/that/go/off/the/end/of/a/line.A\text/test/of/some/short/words/that/go/off/the/end/of/a/line.A\text/test/of/some/short/words/that/go/off/the/end/of/a/line.A\text/test/of/some/short/words/that/go/off/the/end/of/a/line.
8.13 HTML and accidental mathematics

We render mathematics in web pages with the fantastic MathJax Javascript library. Simplifying just a bit, it recognizes \( \text{\LaTeX} \) syntax within a page, takes control of that text, and replaces it with nice fonts and formatting. Now, if you write about \( \text{\LaTeX} \) 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: \( \sqrt{x^2} \).

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

9 Graphics

Mathbook XML supports several languages for describing diagrams and pictures with human-readable source code (i.e. plain text), rather than using a “paint” program. Any \( \text{\LaTeX} \) macros used in the rest of your document may be employed in the \( \text{\LaTeX} \)-standalone or Asymptote diagrams (with Sage graphics coming next?).

9.1 \( \text{\LaTeX} \) images

There are several graphics engine packages that a \( \text{\LaTeX} \) document can employ. Code from these packages renders diagrams automatically as part of normal processing of \( \text{\LaTeX} \) files. For HTML output the mbx 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 \texttt{docinfo/latex-image-preamble}, which is also where global package settings should be made. If any ampersands occur in your \( \text{\LaTeX} \) code you should use the \texttt{\amp} macro pre-defined by PreTeXt. These first examples are from the \texttt{TeXample.net} site.
The next example began life in Sketch, which will output TikZ code (though the code has been edited by hand for readability).

The pgfplots package was included in `docinfo/latex-image-preamble`. Here, it is used. Also, here we demonstrate using \texttt{\textbackslash amp} where you would normally use an ampersand in \LaTeX. There are known issues with \texttt{xelatex} processing any gradient shading in \texttt{tikz}. To (successfully) create the gradient shading in the 3D image here, you may need to use \texttt{pdflatex} until \LaTeX developers resolve this issue.
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. In the example below data is the directory and hodgkin-huxley-data.dat is the file with the data points. You must place the file in a subdirectory (it cannot reside next to your source file), but that directory may have subdirectories if you have many such files and want to organize them that way. Then the --include command-line argument to the mbx script will manage the external files properly as it creates individual image files.

It is still your responsibility to be sure this directory of external data files follows your \LaTeX output to whatever directory you use to convert to a PDF and is in the right location for the relative path given in the XML source. The discussion above only applies to generating individual image files, such as you would need for the HTML output.

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
\pdfpdflatex --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.

9.2 Placing Images without a Caption

To place an image without a caption, use the <sidebyside> layout element,
containing just a single <image>. There is no way to add a caption, and the
item will not be numbered. You cannot cross-reference it, nor will it appear
in a knowl in HTML output. You will get a bit of vertical separation for the
transitions to/from horizontal layout. Use margins=auto on the <sidebyside>
to center the image—this should become the default behavior. A variety of
other elements may be placed in a similar manner. See Section 22 to learn
more about the <sidebyside> layout element.

9.3 Asymptote

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 http://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.

And a colorful contour plot with logarithmic scale. Again, from the Asymptote
documentation.

This SVG image includes a PNG image for the gradient. It needs some
work and is temporarily unavailable.
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 identical 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.

\[ \mathcal{X} = \int_{0}^{1} x\delta(x) \, dx \]

And finally, an example of a 3-D graph (from the documentation again). This SVG image includes a PNG image for the gradient. It needs some work and is temporarily unavailable.
9.4 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 \textit{must} return a Sage \texttt{Graphics} object (or 3D plot). The \texttt{mbx} script will isolate this last line, use it as the RHS of an assignment statement, and the Sage \texttt{.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. For visualizations of 3D plots, Sage will only produce Portable Network Graphics (PNG) files, which can be included in HTML pages or \LaTeX output.

Pay careful attention to the requirement that the last line of your code be a graphics object. In particular, while \texttt{show()} might appear to do the right thing, it evaluates to Python’s \texttt{None} object and that is just what you will get. The code for Figure 9.10 illustrates creating two graphics objects and combining them into an expression on the last line that evaluates to a graphics object.
The following examples are from the Sage Tour. We package them into a sidebyside layout element, see Section 22.

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.

Figure 9.10: Two Sage plots on one set of axes

Figure 9.11: A Sage multi-graph of a sentence

Figure 9.12: Sage polynomial approximations of \( f(x) = \frac{1}{(1 + 25x^2)} \)

Figure 9.13: A Sage implicitly defined 3D surface
9.5 Images from External Sources

If you have raster images (photographs, etc) then they are specified with complete filenames, as above in Figure 5.2. 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 for the \LaTeX output, and was converted to an SVG for use with the HTML output.

![Image](image-source)

**Figure 9.14:** Complete graph on 16 vertices, from www.texample.net

9.6 Copies of Images

Sometimes you want to use the same image more than once. For a raster image, you can just point to it again. You are free to wrap it in a `figure` or `sidebyside` and change the caption. It will get a new number as a new figure, and you will need to assign a unique `@xml:id` attribute. Maybe appending `-copy=2`, or similar, will be helpful.

![Image](image-source)

**Figure 9.15:** Copy of raster image, in a figure, so now numbered and captioned

If you have a figure generated from source code (such as in TikZ) you can
place the code into a file and \texttt{xinclude} it twice. The \texttt{mbx} script will make two copies if you include the source twice, but with different names (generated from the different \texttt{@xml:id}). Furthermore, with the \texttt{@parse} attribute on the \texttt{xinclude} element set to \texttt{text} you are absolved from escaping the dangerous XML characters (see Subsection 8.1) and definitely have no need for the often mis-understood \texttt{CDATA} mechanism. We demonstrate this with the sample book, since it is all set up with the \texttt{xinclude} mechanism. See the two plots of the 8-th roots of unity in the complex numbers section of the chapter on cyclic groups.

### 9.7 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 \texttt{mbx}. 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 \texttt{@source} attribute should include the complete relative path with the subdirectory. Files generated by the \texttt{mbx} script will be specified in the output using the relative directory \texttt{images}, which can be changed using the \texttt{directory.images} stringparam. There is no reason author-provided files cannot also be placed in this same directory (presuming no duplicate names).

<table>
<thead>
<tr>
<th>MBX Element</th>
<th>Specification</th>
<th>\LaTeX/Print</th>
<th>HTML</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>image/@source</td>
<td>full relative path, w/ extension</td>
<td>directly included</td>
<td>directly included</td>
<td>author-provided</td>
</tr>
<tr>
<td>image/@source</td>
<td>full relative path, w/o extension</td>
<td>presumes PDF</td>
<td>presumes SVG</td>
<td>author-provided</td>
</tr>
<tr>
<td>image/latex-image-code</td>
<td>\LaTeX-compatible source</td>
<td>directly included</td>
<td>SVG via \texttt{mbx}</td>
<td>e.g. tikz, pgfplots, xypic</td>
</tr>
<tr>
<td>image/sageplot</td>
<td>Sage code</td>
<td>PDF via \texttt{mbx}</td>
<td>SVG via \texttt{mbx}</td>
<td>PNG for 3-D</td>
</tr>
<tr>
<td>image/asymptote</td>
<td>Asymptote code</td>
<td>PDF via \texttt{mbx}</td>
<td>SVG via \texttt{mbx}</td>
<td></td>
</tr>
</tbody>
</table>

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

### 9.8 Caption Testing

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

Hey!

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 [11.4.2, 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.

**Two Derivative Problems** In the next two problems compute the indicated derivative.

Use a sidebyside element to insert a relevant image, or tabular, or other un-numbered item that does not fit in a sentence.

![Graph](image)

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
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 \(<\text{exercisegroup}>\) has a title and no \(<\text{introduction}>\), which once resulted in some aberrant formatting in \(\LaTeX\) output.

Two More Derivative Problems

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 \(<\text{exercisegroup}>\): 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, 4 characters): \( \sin(\blank) \)
- Inside inline math (default, 10 characters): \( \sin(\blank\blank) \)
- Inside exponents and subscripts (2 characters each). 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): \( \tan(\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank\blank) \)
- So use inside a displayed equation

\[
16 \log \blank
\]

like this one.

- Inside the second line of a multi-line display:

\[
y = x^7 x^8 = x^{\blank}
\]

11.3 More Exercises

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. $3 + 4 + 5$

7. $5 + 6$

8. Add seven to eight.

   **Hint.** Addition is associative.

   **Answer.** 12

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

9. $9 + 10$

10. $1 + 2$

11. $3 + 4 + 5$

12. $5 + 6$

13. Add seven to eight.

   **Answer.** 15

14. $9 + 10$

15. $1 + 2$

16. $3 + 4 + 5$

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 \(\text{LaTeX}\) 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.

11.4 References

These items are here to test basic formatting of references.

   An online, open-source offering.  

This is a conclusion, which has not been used very much in this sample. Did you see the the second reference above has a short annotation? So you can make annotated bibliographies easily.

12 List Calisthenics

12.1 Lists, Generally

Use \texttt{ol} to make an ordered list, and \texttt{ul} to make an unordered (bulleted) list. In both cases, use \texttt{li} for each entry. If an entry contains more than one paragraph, then each must be wrapped in \texttt{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 a description list. Note in the source the optional use of a paragraph (\texttt{p}) for the list items of the list of colors.

1. First.
2. Second.
3. Third.

- Red
- 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. The second-level defaults (lower-case Latin) are formatted slightly different in \LaTeX{} versus HTML. The HTML style is not easy to adjust, but you can specify the \LaTeX{} version to match if it is important. 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. 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. Level 4, third.
         
      iii. Level 3, third.
      
   (c) 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 MBX (disc, circle, square, disc). Again, the default order for Markdown/ Jupyter (disc, square, circle, circle) is different than for \LaTeX{} and HTML (disc, circle, square, disc)

- Level 1, first.
- Level 1, second.
   
   o Level 2, first.
   
   o Level 2, second.
      
      ■ Level 3, first.
      
      ■ Level 3, second.
      
      • Level 4, first.
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.
               o Level 6, first.
               o Level 6, second.
               o 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. Note the adornments of the labels will be rendered in LaTeX, but not in HTML.

*I* Level 1, first.

*II* Level 1, second.
   ++A Level 2, first.
   ++B Level 2, second.
      1) Level 3, first.
      2) Level 3, second.
      3) Level 3, third.
   ++C Level 2, third.

*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.

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.

\textbf{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 \textbf{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).

\textbf{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. in a two column list look good.

(b) Two ducks. (d) Four items.

(c) Three items. Plus a few more words to check that long entries  (e) Another long entry that simultaneouly 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.
- Tres item.
- Quattro items.

And a paragraph after that list so that spacing can be checked.

12.4 Description Lists

Use d/l 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.

**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.)

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. Conversion to \LaTeX{} ignores this attribute, and conversion to HTML only reacts to narrow (medium is the default, and wide behaves identically).

**Red** The color of the sun at sunset.

**Blue** The color of a clear sky.

**Aqua** The color of shallow tropical waters.

**Math** $x^2$ Sorry, not a color but testing titles with math in them.
12.5 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.

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.

List 12.2: Colors of the Rainbow

This next list is used for testing cross-references to it. See Section 17.

This is the introduction to this named list, which references an item within, via the hybrid `@text` attribute: Item B.c.

- 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 12.3: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

**List 12.3:** A named list of targets

This is a paragraph just outside the preceding named list, which references an item within, via the hybrid `@text` attribute: Item 12.3:B.c.

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.

### 12.6 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.

### 12.7 Exercises (with lists)
1. This exercise should have several parts, and labels should follow the defaults for second-level lists (since the exercise is numbered according to the top-level default).
   (a) Exercise 1, first part.
   (b) Exercise 1, second part.
   i. Exercise 1, second part, first refinement.
   (c) Exercise 1, third part.

2. 

<table>
<thead>
<tr>
<th>1111, 2222</th>
<th>3333</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaa</td>
<td>bbbb,ccc</td>
</tr>
<tr>
<td>AAAA</td>
<td>BBBB</td>
</tr>
</tbody>
</table>

Table 12.4: Table Alignment Example

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 subsidiary parts of exercises. Exercise 1, third part: 12.7.1.c. Exercise 1, second part, first refinement: 12.7.1.b.i.

12.8 References (with lists in Annotations)

Here is the annotation and an ordered list as part of that annotation.
   (a) Book 1, first part.
   (b) Book 1, second part.
   (c) Book 1, third part.

13 Table Calisthenics

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.

<table>
<thead>
<tr>
<th>Red</th>
<th>Green</th>
<th>Yellow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue</td>
<td>White</td>
<td>Pink</td>
</tr>
</tbody>
</table>

Table 13.1: Some Colors

Note that tables may be constructed using the \LaTeX\ Complex Table Editor tool online at latex-tables.com and then exported in PreTeXt syntax. Tables can be used many ways. We describe long division of polynomials by using vertical and horizontal borders on individual entries of a table. The division lines are slightly thicker than the subtraction lines. This is a good example of the typical abuse of tables for horizontal and vertical layout. Also indicative of this abuse is that it might make more sense to call this a “Figure,” not a “Table”.

54
\[
x + 2 \begin{array}{c}
x^2 - 3x - 8 \\
x^2 + 2x \\
-5x - 8 \\
-5x - 10 \\
= 2
\end{array}
\]

Table 13.2: Polynomial Long Division

The next table describes how to construct tables via the \texttt{tabular} element. The \texttt{table} element may be used to enclose the raw table, so as to associate a caption and get vertical separation with horizontal centering.

The \texttt{tabular} element contains a sequence of \texttt{row} elements, and must contain at least one. Each \texttt{row} contains a sequence of \texttt{cell} elements and must have the same number in each row (accounting for the use of the \texttt{colspan} attribute). The contents of the \texttt{cell} elements are the text to appear in entries of the table.

A sequence of \texttt{col} elements may optionally be used. But if one appears, then there must be the right number for the width of the table. They are empty elements always, and just carry information about their respective column.

Where the body of the table below has an entry, it means the attribute may be used on the element, and affects the range of the tabular described by the element. Employment of an attribute on elements to the right in the table will supersede use on elements to the left. Generally, every cell has right and bottom borders, but only cells at the left side of the table have a left border and only cells across the top have a top border. Only one cell has four borders.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Elements</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>\text{tabular}</td>
<td>\text{col}</td>
</tr>
<tr>
<td>top</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>left</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>bottom</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>right</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>halign</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>halign</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>header</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>footer</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>valign</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>colspan</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>rowspan</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>width</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>colors</td>
<td>p</td>
<td>p</td>
</tr>
</tbody>
</table>

Table 13.3: Tabular Elements and Attributes (p = planned)

Bully Pulpit: Vertical Rules in Tables. One of the goals of PreTeXt is to gently guide authors towards good choices in the design of their documents, even if we do not claim to know it all ourselves. Take a close look at that table about tables. What’s missing? No vertical rules. Try living without them, you will not really miss them. If you think you need to divide a table into two halves, maybe you really need two tables (and then see the “side-by-side” capabilities, Section 22).

In the documentation for his excellent \LaTeX package, \texttt{booktabs}, Simon
Fear gives two rules for what he calls “formal tables”: (1) Never, ever use vertical rules, and (2) Never use double rules. We have resisted the temptation to enforce the former and have provided an alternative to the second (three thicknesses). He refers to using tables for layout as creating “tableau.” If you are finicky about the look of your work, the first three pages of the documentation is recommended reading.

That all said, we now give several examples in order to stress and demonstrate our code.

An example of aligning table cells’ contents horizontally. See the source for comments.

<table>
<thead>
<tr>
<th>1234567890</th>
<th>1234567890</th>
<th>1234567890</th>
<th>1234567890</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>Second</td>
<td>Third</td>
<td>Fourth</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 13.4: Horizontal Alignment Example

Example from above, but now with horizontal rules, plus an extra row to test the bottom border. See the source for comments.

<table>
<thead>
<tr>
<th>1234567890</th>
<th>1234567890</th>
<th>1234567890</th>
<th>1234567890</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>Second</td>
<td>Third</td>
<td>Fourth</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 13.5: Horizontal Rules Example

For a table without a caption, create a `<tabular>` and place it inside a `<sidebyside>`. This will allow control over the horizontal placement, but without a caption, there is no number, and the tabular cannot be cross-referenced.

Table 13.6: Vertical Rules Example

Same example as before, but now with vertical rules. See the source for comments.

<table>
<thead>
<tr>
<th>1234567890</th>
<th>1234567890</th>
<th>1234567890</th>
<th>1234567890</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>Second</td>
<td>Third</td>
<td>Fourth</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 13.7: Progressively Thicker Rules Example

<table>
<thead>
<tr>
<th>1111</th>
<th>2222</th>
<th>3333</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaa</td>
<td>bbbb</td>
<td>cccc</td>
</tr>
<tr>
<td>AAAA</td>
<td>BBBB</td>
<td>CCCC</td>
</tr>
</tbody>
</table>
Table 13.8: Column Span Example

Example 13.9 Example Environment with Leading Table.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5+6+7</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 13.10: Column Spans, No \texttt{col} Elements, Nine Columns

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 \texttt{colspan} attributes. □

A bare minimum table (one row with one cell) to test edge cases:

\begin{center}
\begin{tabular}{|c|}
\hline
One \\
\hline
\end{tabular}
\end{center}

Table 13.11: One entry table

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 \texttt{p} children. And such cells should \textit{only} have \texttt{p} children. The width of a paragraph cell is determined by a \texttt{width} attribute on the corresponding \texttt{col} (as a percentage). If no \texttt{width} is specified (or there isn’t even a \texttt{col} in the first place) then \texttt{xsltproc} will abort. 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 \texttt{colspan} greater than 1, although cells with \texttt{colspan} greater than 1 that are above or below a paragraph cell will behave. Setting \texttt{width} on a \texttt{col} that has no paragraph cells may produce unexpected results. A \texttt{valign} for the parent \texttt{row} (or the ambient \texttt{tabular}) can control vertical alignment (top, middle, or bottom). A paragraph cell’s \texttt{halign} attribute (left, center, right, or justify) controls how the text is justified. Cells inherit \texttt{halign} from \texttt{row}, \texttt{col}, and \texttt{tabular} in that order of preference. In a non-paragraph cell where \texttt{halign}='justify', the horizontal alignment will match the behavior of \texttt{halign}='left'.

57
The duration of 9,192,631,770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the cesium-133 atom.

The time it takes you to say the phrase “differential calculus”.

The length of one episode of a premium cable television show.

Table 13.12: Time Units

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>
<thead>
<tr>
<th>Unit</th>
<th>Stands For</th>
<th>Definition</th>
<th>Roughly</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>second</td>
<td>the duration of 9,192,631,770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the cesium-133 atom</td>
<td>the time it takes you to say the phrase “differential calculus”</td>
</tr>
<tr>
<td>min</td>
<td>minute</td>
<td>exactly 60 seconds</td>
<td>how long it takes to microwave a full dinner plate from the refrigerator</td>
</tr>
<tr>
<td>h</td>
<td>hour</td>
<td>exactly 3600 seconds; exactly 60 minutes</td>
<td>the length of one episode of a premium cable television show</td>
</tr>
</tbody>
</table>

Table 13.13: Dr. Seuss lines

This is a table torture test with many combinations of `halign`, `valign`, `colspan`, `<p>` children, and `<line>` children.
And now a `<sidebyside>` with a `<table>` and a `<tabular>` to check that width is scaled appropriately. See Section 22 to learn about `<sidebyside>`s.
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.

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 13.15: Some text from the US Constitution

Tables are formed in \texttt{\LaTeX} output with copious use of the \texttt{\textbackslash multicol} macro to override more global alignment settings, and to spread the content of one cell across several columns. However, sometimes \texttt{\LaTeX}’s special characters have behaved badly in this situation. So the table below, two items per row, is just designed for \texttt{\LaTeX} testing. But of course, it should still render fine in other formats. The three test cases are from \ref{sec:multicolumn}, but without 50 alphabetic characters and 8 digits, which should not be problems in this context. 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 \textit{forced} to be wrapped in a \texttt{\textbackslash multicol} by specifying their horizontal alignment. The second column’s entries \textit{will not} be wrapped in a \texttt{\textbackslash multicol}. 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 \textit{Subsection} 8.11.)

<table>
<thead>
<tr>
<th></th>
<th>\url{<a href="http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=%7D">http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=}</a></th>
<th>\url{<a href="http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=%7D">http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=}</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>\url{<a href="http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=%7D">http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=}</a></td>
<td>\url{<a href="http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=%7D">http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=}</a></td>
</tr>
<tr>
<td>2</td>
<td>\url{<a href="http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=%7D">http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=}</a></td>
<td>\url{<a href="http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=%7D">http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=}</a></td>
</tr>
<tr>
<td>3</td>
<td>\url{<a href="http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=%7D">http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=}</a></td>
<td>\url{<a href="http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=%7D">http://example.com/09az%58-._~:/?#$!&amp;'(++)++,=}</a></td>
</tr>
</tbody>
</table>

Table 13.16: Problematic Table Cells for \texttt{\LaTeX}

14 Interactive Elements

Authored in Javascript

When outputting Web page versions, it is possible to embed a variety of dynamic interactive elements. In a \texttt{\LaTeX}/PDF version, these will necessarily need to be replaced by some static substitute, such as a screenshot. See \textit{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

merganser.math.gvsu.edu/david/linear.algebra/ula/index.html

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.

![Figure 14.1: A simple eigenvector demonstration](image)

**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 $2 \times 2$ 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?

![Figure 14.3: Eigenvector demonstration](image)

The next example has ten `<slate>` elements communicating with each other, and arranged with the layout features of a `<sidebyside>` (see Section 22).
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 14.4: Affine Transformations**

**Figure 14.5: 1-Dimensional Kinematics**

### 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 at 3231298 with a GPL license. A similar demonstration, only using an HTML5 canvas is at bl.ocks.org/mbostock/3231307.

![Figure 14.6: Force layout and collision detection](image)

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.

This is adapted from a block by shimizu at e6209de87cddde38dadbb746feaf3a3 with a GPL license.

![Figure 14.7: Graph Layout](image)

**Checkpoint 14.8 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 at 7aa53c7bf3e411238ac8eaf280bd6581, provided 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 data collected from the Twitter analysis is contained in a JSON file, mention_network.json, and where the script loads that file, it needs a path relative to the HTML file where the interactive is viewed.

![Figure 14.9: Tweet mentions within the Welsh Assembly](image)

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 change in \( x \), denoted here as \( h \), is reported when the “Show secant line” option is ticked. The \( h \) is being rendered as mathematics by the Javascript MathJax library. However, this cannot be accomplished with \$...\$ nor by \( \cdot(\cdot\cdot) \), but instead by using <script> tags that MathJax recognizes. So the employment of MathJax here is accomplished with

\[
\text{<script type="math/tex">h = \</script>}
\]
Checkpoint 14.11 Changing Secant Lines. When discussing the derivative as a limit, we think of the point of tangency as being fixed (the green point in 14.10) and the “other” point defining the secant line as changing (the red point in 14.10). 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 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 example is taken from the JSXGraph example wiki. The code could be written in 7 lines. Width is 80% and aspect ratio is 4:3.

Figure 14.12: 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.

![Figure 14.13: Infinity, from the JSXGraph Showcase](image)

Here are the two new examples. They have been included in a sidebyside layout element with equal widths (see Section 22) so they can be placed horizontally across the page. They are not wrapped as figures, so cannot be cross-referenced. These are again from the example wiki, the left being Fermat’s Spiral and the right being a demonstration of B-splines.

![Figure 14.14: Piecewise Function](image)

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.

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 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, wiki.sagemath.org/interact/calculus, specifically here.

Figure 14.15: Numerical integrals using the midpoint rule

14.6 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 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 wiki.geogebra.org/en/Reference:GeoGebra_Apps_API, but do not include the ggbApplet. or applet. used in examples to prefix the function that part of the code will be provided automatically by PreTeXt.

Jack Green created an applet on the Classroom Resources site with ID D4s2v4ft, which you may view at 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 wiki.geogebra.org/en/Reference:GeoGebra_Apps_API. It is important to use one command per line.

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. We modify David’s applet in a few ways. Remove the modifications to see the original.

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.
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.

Lastly, you may just wish to build something from scratch using GeoGebra API. Note that for accessibility 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.

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 guess (or measure with an on-screen ruler), the aspect ratio and use that in the <interactive> element.

www.geogebra.org/material/iframe/id/KGn2d4Qd

Figure 15.1: Right Triangle Paradox

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/ttox1bvxxu

You want to pick off the identifier on the end, in this case ttox1bvxxu, then author

<interactive desmos="ttox1bvxxu" 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.
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, find the online app version at

http://www.monroecc.edu/faculty/paulseeburger/calcnsf/CalcPlot3D/

Create a plot and adjust the image to a viewpoint and scale you like. Then, click the menu 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.

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.

Figure 15.3: Intersection of two planes (minimal embedding)
Figure 15.4: Probability wavefunction with contours (includes controls)

Figure 15.5: Plot of $f(x, y) = \frac{1}{y - x^2}$ on $[-2, 2] \times [-2, 2]$ (full application)

15.4 Wolfram CDF

You can embed interactive demonstrations created in Wolfram’s **Computable Document Format** so that they will be played with the Wolfram CDF Player™. Once you create and save a demonstration, you want to determine the UUID that is the identifier of your demonstration. For example, Figure 15.7 is identified by 9fa2acff-c809-4b7f-a73b-c59ace36affc. This identifier is enough to create the PreTeXt to embed the demonstration. See [https://reference.wolfram.com/language/howto/DeployInteractiveContentInTheWolframCloud.html](https://reference.wolfram.com/language/howto/DeployInteractiveContentInTheWolframCloud.html) for information about creating your demonstration.

[http://www.wolfram.com/cdf/adopting-cdf/deploying-cdf/web-delivery-cloud.html](http://www.wolfram.com/cdf/adopting-cdf/deploying-cdf/web-delivery-cloud.html) explains hosting CDF files at the Wolfram Cloud, and is the source of Figure 15.7. You can learn about powering your CDF with Wolfram Cloud Credits at [https://www.wolfram.com/cloud-credits/](https://www.wolfram.com/cloud-credits/). CDF is a public format, and the FreeCDF™ license is a variant of a Creative Commons BY-SA license, see [http://www.wolfram.com/cdf/adopting-cdf/licensing-options.html](http://www.wolfram.com/cdf/adopting-cdf/licensing-options.html).

The first example here (Figure 15.6) was developed by Itai Seggev, a Senior Kernel Developer at Wolfram.

(2018-04-02) These behave as expected in Chrome, but perhaps not in Firefox. Testing welcome.
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 videos, which are near the bottom of this page, so look there first if that meets your needs.

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 either a `<figure>` (numbered, captioned) or a `<sidebyside>` for more precise control. 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).
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 https://github.com/rbeezer/mathbook/issues/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.)

![Image](source-image.png)

**Figure 16.2:** University of Puget Sound Promotional Video

If you find the posters distracting or objectionable, you can cover them up by requesting a generic poster with the attribute-value pair: `preview = "generic"`.

![YouTube Video](source-video.png)

**Figure 16.3:** 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 two 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.
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 https://www.youtube.com/watch?v=hAzdgU_kpGo. Screen real estate is determined by specifying an optional @width attribute as a percentage, and aspect ratio is preserved on the assumption of HD video (16:9). (Enhancements will include other aspect ratios.)

If you have ever owned a drone, you sympathize with this guy. Way funnier than a cat video.

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.
Videos may be placed “standalone” in a `<sidebyside>`, but are designed mostly on the assumption that they are wrapped in a `figure` with a `title` (which is distinct from a `caption`). This video comes with a default poster featuring Robert Plant. We’ve replaced it with Jimmy Page.

And if you don’t want to be reminded of Plant, Page, Bonham, or Jones, you can cover them up.

We can pack two videos side-by-side, with a lot of horizontal control, using two panels in the `sidebyside` element. We have simply chose not to provide a caption (overall, or separately) as an illustration. The sizes are purposely a bit odd. See Section 22 for much more on side-by-side panels. These videos come from the “Topic” and “VEVO” areas of YouTube (respectively) and both have start/end times.
These next two videos are evenly spaced, one from YouTube, one from a source file hosted by the author. Now with separate captions, but identical margins (through very different choices of layout parameters than in the preceding pair of videos).

![Drone Flight](image1.jpg) ![UPS Promo](image2.jpg)

**Figure 16.10:** Drone Flight  **Figure 16.11:** UPS Promo

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 `youtubep/playlist` attribute to a YouTube playlist ID.

https://www.youtube.com/watch_videos?video_ids=6tuVHMjWORFvI
https://www.youtube.com/p/playlist?playlist=PLGRecdYLYDnDx5NEkS3KH-ZtC

**Figure 16.12:** Individual IDs specified in an “itemized” playlist  **Figure 16.13:** YouTube playlist ID specified in a “named” playlist

Videos may be embedded, or popped-out to play in a new window or tab (at greater width), or a link will give the reader the option to choose either style of playback. The automatic pop-out option requires a static thumbnail image. (For YouTube, these images can be obtained automatically with the `mbx` script.)

![Embedded Only](image3.jpg) ![Pop-Out Only, w/ Start/End](image4.jpg) ![User Selection](image5.jpg)

**Figure 16.14:** YouTube Play Location Options (`play-at` attribute)

Each of the three options above may have a generic play button placed over the one provided by the video.
17 Cross-Referencing

Cross-references are easy, since that is a key reason for having a highly structured document. Here is a useful feature if you elect to use it. Any `<xref>` will “know” what it points to, so you can let it provide the “naming” part of the cross-reference text. You can turn this on globally with the command-line parameter `autoname` set to 'yes'. If you do that, you will see most of the names in this document doubled, since the names are written into the source already in most places outside of this section. Try it and see: use `--stringparam autoname 'yes'` as an argument to `xsltproc`

Moreover, the names themselves will change with the use of the one language dependent file. And another bonus is that with an autoname, you automatically get a non-breaking space between the name and the reference. The autoname switch makes no sense for “provisional” cross-references, since there is no information about what they point to.

Here is a reference that has no indication of its type in the source: 2.1. So by default you will just see a number that you can click on. If you use the `text="type-global"` switch then you should see “Theorem” prepended. Note that if you changed the theorem to a lemma, then that change would be reflected here automatically when autonaming is in effect.

If you set the autonaming behavior globally, or accept the default behavior, 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 how
the global autoname is set.

- 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 \texttt{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 \texttt{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 a title as the clickable, then that text
can be replaced 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 global autonaming on, you can
override the first \texttt{Theorem} by providing the content \texttt{Theorems} on the first \texttt{xref}
and \texttt{text=“global”} on the second \texttt{xref}. (With global autonaming off, you will
also get what you want/expect.) Here is the test, which should look correct
no matter what the global switch is: Sections 17 and 18. (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 \texttt{xref}. The “author-tools” mode and careful
choices of xml:id strings can help avoid this trap.)

One final twist. If you say \texttt{text=“title”}, then the title you assigned to
the theorem will prefix the number. Here is a the final example, which should
always refer 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 32.42a 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 [1]
- Citations: Judson’s AATA with autoname that should have zero effect
  [1]
- Citations: In a \texttt{<references>} division inside an appendix [G.2.1]
- Note: just the annotation of previous citation at 1.1
- Examples: Mystery derivative at 4.2, or a question at 4.6.
- Definition-like: A mathematical statement with no proof 4.11.
- A numbered Note: 4.9
- 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 26.1
- 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: 22.5
- A Table within a side-by-side panel, with a subnumber: 22.14(a)
- A Figure, containing a side-by-side with two sub-captioned images: 22.1
- Display Mathematics: single, first with no name: (4.1). Then with an autoname: (4.1).
- Display Mathematics: multi-row, first with no name: (4.2). Then with an autoname: (4.2). And two, with a plural form: Equations (4.1) and (4.2).
- 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 autonaming: Exercises 11.2.1–11.2.3.
- Exercise (inline): with enclosed hint at 4.4
- A group of two exercises, with introduction, conclusion: Exercise Group 11.2.2–3
- Solution: An autonamed portion of an exercise: Solution 32.42a.1
- Parts of a complicated exercise: Hint 32.12.2 Answer 32.12.1
- A subsidiary part of an exercise: 12.7.1.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 30.2; in the narrative, Checkpoint 4.4; and in a <worksheet>, Worksheet Exercise 29.2.2.
- 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: 12.2
- List item inside a named list, second color in rainbow list: Item 12.2:2
- 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 ambiguous: Item
- Same item in ordered list, but contained in an unordered list, now with a cross-reference by title with a content override: 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 12.3.

• An assemblage, which never has a number. A cross-reference now requires content in the xref element, with text='title': 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: An Aside

• A cross-reference to an objectives block, with an autoname. 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 an autoname: 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

• An interactive program inside a program listing, to test if the Javascript will execute properly within a knowl: 20.8

• A cross-reference to a block quotation (which is never numbered): Quote by Dr Seuss

• 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: 22.23(b)
• A cross-reference to a paragraph (p) in the statement of an exercise: paragraph. Notice that such a paragraph has no number and no title, so you need to (a) use text='title', and (b) provide custom text for the title, as an override that becomes the content of the link. You may wish to provide very explicit location information for hardcopy print. Notice that this is not a cross-reference to the exercise, just a small portion of it.

• This opens a knowl for an example. It has a solution, which is originally 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 identification like an HTML id. So we test its styling and function here: Example 4.5

• 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 4.2.7. 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 4.2.4. 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.

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 an autonamed 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 autoname facility, 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).

\begin{align*}
x^2 + y^2 &= z^2 & \text{Theorem 2.1} \\
a^2 + b^2 &= c^2 & \text{Section 2}
\end{align*}

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 autonaming attribute 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.

- Three theorems, with spaces, autonamed: Theorem 2.1, Theorem 25.1,
Theorem 25.3
- Two equations, no spaces, autonamed: (4.1), (4.2)
- Two bibliographic items, no autoname: [1, 2]

If you have a long list of items (such as homework exercises, not in an
exercise group, 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
autonaming and other features.

- A range of exercises, autonamed (this range appears “out-of-order” since
  the two exercise are numbered under two different schemes): Check-
  point 4.4–4.2.4.1
- 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 3–17
- A range of bibliographic items: [1–2]

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 some-
where 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 appro-
priate 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 dis-
tribute 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 Pre-
TeXt 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.

18 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 \TeX, PDF, print. The situation for \TeX 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 \TeX 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 \&amp;#x0042;.

Table 18.1: Basic Latin, Regular

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 pdf/tex or xetex 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 \&amp;#x00A9;.

Table 18.2: Latin-1 Supplement, Regular

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 xetex, 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 18.3: Basic Latin, Monospace

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 ₀ ₀ ₀ ₀ ₀. And the numeral one is slightly different from the lower-case “ell”: ₁ ₁ ₁ ₁ ₁ ₁. 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 \texttt{listing} package and these characters require ad-hoc replacements for processing by \texttt{pdflatex}. (You can see the replacements in the preamble of the \LaTeX{} source for this document.) The replacement mechanism provided by the \texttt{listing} package will cause the characters below to produce a \LaTeX{} compilation error if processed by \texttt{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 \texttt{xelatex}.

```
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>00A_\texttt{\textbackslash i \textbackslash f}</td>
<td>nych \texttt{\textbackslash y}</td>
<td>\texttt{\textasciicircum}</td>
<td>\texttt{\textasciitilde}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00B_\texttt{\textdagger} \texttt{\textdagger}</td>
<td>\texttt{\textasciitilde}</td>
<td>\texttt{\textdagger}</td>
<td>\texttt{\textdagger}</td>
<td>\texttt{\textdagger}</td>
<td>\texttt{\textdagger}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 00C_ \texttt{\textbackslash a \textbackslash a \textbackslash a \textbackslash a \textbackslash a \textbackslash a \textbackslash e \textbackslash c \textbackslash e \textbackslash e \textbackslash e \textbackslash e \textbackslash i \textbackslash i \textbackslash i \textbackslash i \textbackslash i}
| 00D_ \texttt{\textbackslash d \textbackslash n \textbackslash o \textbackslash o \textbackslash o \textbackslash o \textbackslash o \times \textbackslash o \textbackslash u \textbackslash u \textbackslash u \textbackslash u \textbackslash y \textbackslash p \textbackslash b}
| 00E_ \texttt{\textbackslash d \textbackslash n \textbackslash o \textbackslash o \textbackslash o \textbackslash o \textbackslash o \times \textbackslash o \textbackslash u \textbackslash u \textbackslash u \textbackslash u \textbackslash y \textbackslash p \textbackslash b}
```

Table 18.4: Latin-1 Supplement, Monospace

The \texttt{pre} tag is implemented in LaTeX with the \texttt{fancyvrb} package. You can compare results here with the table above, lines here are rows above.

```
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>00A_\texttt{\textbackslash i \textbackslash f}</td>
<td>nych \texttt{\textbackslash y}</td>
<td>\texttt{\textasciicircum}</td>
<td>\texttt{\textasciitilde}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00B_\texttt{\textdagger} \texttt{\textdagger}</td>
<td>\texttt{\textasciitilde}</td>
<td>\texttt{\textdagger}</td>
<td>\texttt{\textdagger}</td>
<td>\texttt{\textdagger}</td>
<td>\texttt{\textdagger}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 00C_ \texttt{\textbackslash a \textbackslash a \textbackslash a \textbackslash a \textbackslash a \textbackslash a \textbackslash e \textbackslash c \textbackslash e \textbackslash e \textbackslash e \textbackslash e \textbackslash i \textbackslash i \textbackslash i \textbackslash i \textbackslash i \textbackslash i}
| 00D_ \texttt{\textbackslash d \textbackslash n \textbackslash n \textbackslash n \textbackslash n \textbackslash o \textbackslash o \textbackslash o \textbackslash o \textbackslash o \times \textbackslash o \textbackslash u \textbackslash u \textbackslash u \textbackslash u \textbackslash u \textbackslash u \textbackslash p \textbackslash b}
| 00E_ \texttt{\textbackslash d \textbackslash n \n o \n o \n o \n o \n o \times \textbackslash o \u u \u u \u y \p b}
```

The \texttt{console} tag is also implemented with \texttt{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.

```
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>00A_\texttt{\textbackslash i \textbackslash f}</td>
<td>nych \texttt{\textbackslash y}</td>
<td>\texttt{\textasciicircum}</td>
<td>\texttt{\textasciitilde}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00B_\texttt{\textdagger} \texttt{\textdagger}</td>
<td>\texttt{\textasciitilde}</td>
<td>\texttt{\textdagger}</td>
<td>\texttt{\textdagger}</td>
<td>\texttt{\textdagger}</td>
<td>\texttt{\textdagger}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 00C_ \texttt{\textbackslash a \textbackslash a \textbackslash a \textbackslash a \textbackslash a \textbackslash a \textbackslash e \textbackslash c \textbackslash e \textbackslash e \textbackslash e \textbackslash e \textbackslash i \textbackslash i \textbackslash i \textbackslash i \textbackslash i \textbackslash i}
| 00D_ \texttt{\textbackslash d \textbackslash n \n o \n o \n o \n o \n o \times \textbackslash o \u u \u u \u y \p b}
| 00E_ \texttt{\textbackslash d \textbackslash n \n o \n o \n o \n o \n o \times \textbackslash o \u u \u u \u u \u y \p b}
```

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 \texttt{sage} element. This is not functional code.

```
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>00A_\texttt{\textbackslash i \textbackslash f}</td>
<td>nych \texttt{\textbackslash y}</td>
<td>\texttt{\textasciicircum}</td>
<td>\texttt{\textasciitilde}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00B_\texttt{\textdagger} \texttt{\textdagger}</td>
<td>\texttt{\textasciitilde}</td>
<td>\texttt{\textdagger}</td>
<td>\texttt{\textdagger}</td>
<td>\texttt{\textdagger}</td>
<td>\texttt{\textdagger}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 00C_ \texttt{\textbackslash a \textbackslash a \textbackslash a \textbackslash a \textbackslash a \textbackslash a \textbackslash e \textbackslash c \textbackslash e \textbackslash e \textbackslash e \textbackslash e \textbackslash i \textbackslash i \textbackslash i \textbackslash i \textbackslash i \textbackslash i}
| 00D_ \texttt{\textbackslash d \textbackslash n \n o \n o \n o \n o \n o \times \textbackslash o \u u \u u \u u \u y \p b}
| 00E_ \texttt{\textbackslash d \textbackslash n \n o \n o \n o \n o \n o \times \textbackslash o \u u \u u \u u \u y \p b}
```

86
The table below has a single column, and each cell of the table has a string of 10 characters inside a \texttt{c} element. It is meant to test if the font is monospace in this situation.

```
0123456789
9876543210
iiiiiiiiii
mmmmmmmm
```

Table 18.5: Alignment Test

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 \texttt{pdflatex}. Adding \texttt{-interaction batchmode} or \texttt{-interaction nonstopmode} to the \texttt{pdflatex} command-line will sometimes be less painless than acknowledging each error. The more advanced sample will perform well when processed with \texttt{xelatex}.

## 19 Pre-Formatted Text

In Sage, if you wanted to build a matrix, then you would use the \texttt{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 \texttt{SR} is the ring of symbolic expressions, Symbolic Ring.

```python
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 \texttt{latex(J)}.

```
\left(\begin{array}{rr}
16, y^3 + 6 \backslash, \ x & -48 \ backslash, \ x \ y^2 \\
48 \ backslash, \ x \ y^2 & 48 \ backslash, \ x^2 y + 12 \ y^2 \end{array}\right)
```

The \texttt{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 \textit{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 20. Examine the source of the following example to help understand this paragraph.

\begin{verbatim}
A normal line
An indented line
An outdented line
\end{verbatim}

Snippets should also be robust for cut/paste operations. For example, you should not get “curly” “smart” quote marks in verbatim text: this \textit{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 \texttt{<c>} element) it can impinge on the right margin, since very long words will not hyphenate, unless you have a dash/hyphen. Such as when you use words like pneumonomultramicroscopicsilicovolcanonconiosis, parastratiosphecomyia stratiosphecomyioides, floccinaucinhibilipilification, 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 \texttt{<cd>} or \texttt{<pre>} below.

An intermediate type of verbatim text can be accomplished with the \texttt{<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

\begin{verbatim}
there is the \texttt{<cline>} tag
meant to model the line tag
and short for "code line"
\end{verbatim}

and you may even

\begin{verbatim}
use a single cline
\end{verbatim}

if you like to have your source closely model the visual look of the output.

The \texttt{<pre>} tag is meant for use outside of paragraphs, but is otherwise very similar. The source may also be structured as a sequence of \texttt{<cline>} as in the next example, recycling content from above.

If you write a very long snippet of inline code (i.e. within a \texttt{<c>} element) it can impinge on the right margin, since very long words will not hyphenate, unless you have a dash/hyphen. Such as when you use words like pneumonomultramicroscopicsilicovolcanonconiosis, parastratiosphecomyia stratiosphecomyioides, floccinaucinhibilipilification, 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 \texttt{<cd>} or \texttt{<pre>} below.
20 Program Listings

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.

```r
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:

```xml
<xsl:template match="biblio"
  mode="number">
  <xsl:apply-templates select="." 
    mode="structural-number" />
  <xsl:text>.</xsl:text>
</xsl:template>
```

You can write made-up pseudo-code, but you might explain to a reader what your symbols all mean. This routine takes the \( m \times n \) matrix \( A \) 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.

```plaintext
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
```

89
Look in the `mathbook-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 (these 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 `caption`, and the listing will be numbered along with tables and figures. This then makes it possible to cross-reference the listing, such as Listing 20.1. 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.

```c
/* Hello World program */
#include<stdio.h>
main()
{
    printf("Hello, World!");
}
```

Listing 20.1: C Version of “Hello, World!”

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 20.2  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 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{\texttt} for a monospace text font, and \; for spacing/grouping the bits of the binary number.

\begin{equation}
6C2A_{16} = 0110\ 1100\ 0010\ 1010_2
\end{equation}

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 20.3** This is a spurious theorem to break up the run of consecutive listing so we might test the effect.

**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.

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 \texttt{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 &\lt; and &\gt; in your source.

```
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 $
```

**Listing 20.4:** Console Session: int and float

Here is the plain version, placed inside a \texttt{sidebyside} for layout control. We simply place a small margin on the left (at 4%).

```
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.

```
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 $
```

**Listing 20.5:** Console Session: int and float (multi-line input)

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.

The next listing is just absurdity, to check 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 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.

@ 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 locality,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, #locality @ 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:
Listing 20.7: A longer program listing

In HTML output, a program can be interactive. This is an example program provided by Python Tutor.

```python
# From "Teaching with Python" by John Zelle
def happy:
    print("Happy_Birthday_to_you!")
def sing(P):
    happy()
happy()
    print("Happy_Birthday_dear_" + P + "!")
happy()

# main
sing("Fred")
```

Listing 20.8: An interactive Python program, using Python Tutor

21 Units of Measure

Units of measure can be given XML treatment too with the `quantity` element. In `\LaTeX`, the `siunitx` package is loaded to achieve unit handling. Since that package only offers SI units, some other common units will be added by MBX 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 m/s$^2$. Force is measured in kg m/s$^2$, also known as one N. A quantity with rather ridiculous units is $23$ µha$^{23}$°C s$^2$. One Hz is the same as 1. 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 $\frac{\text{Ft lb gal}}{\text{g}}$. 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.

For a full list of the allowed units and prefixes, see `mathbook-units.xsl`. If you have a need for more units, they need to be added to `mathbook-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.

22 Side-By-Side Panels

Introduction. 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.


22.1 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 22.4 and Figure 22.5.

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 22.1. You can also cross-reference the subfigures individually, for example: Figure 22.1(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 22.1: Side-by-Side, with figures as children, automatic margin](image)

(a) width=50%  
(b) width=25%

Figure 22.2: Side-by-Side, with figures as children, margin set to zero

![Figure 22.3: Widths calculated automatically, all defaults](image)

(a)  
(b)  
(c)

Figure 22.3: Widths calculated automatically, all defaults


22.2 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 22.7.
22.3 Common Side-By-Side Constructions

We have now seen at least one example of each of the four most common constructions involving \textit{sidebyside}. Working from the exterior inward, we can go \texttt{figure}, \texttt{sidebyside}, \texttt{figure}, $X$, where $X$ is some atomic (unnumbered) item we might use elsewhere in a PreTeXt document, the inner \texttt{figure} may be repeated with different objects $X$, and the \texttt{figures} have captions. Each \texttt{figure} is independently optional, leading to the four combinations in this table. Note this applies to any captioned item used inside the \texttt{sidebyside}, but a \texttt{figure} is the most flexible.

<table>
<thead>
<tr>
<th>Outer Figure</th>
<th>Inner Figure</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absent</td>
<td>Absent</td>
<td>Layout only, no numbers nor captions</td>
</tr>
<tr>
<td>Absent</td>
<td>Present</td>
<td>Numbers and captions on figure(s)</td>
</tr>
<tr>
<td>Present</td>
<td>Absent</td>
<td>Number and overall caption</td>
</tr>
<tr>
<td>Present</td>
<td>Present</td>
<td>Number and overall caption, sub-numbers and captions on figure(s)</td>
</tr>
</tbody>
</table>

\textbf{Table 22.8: sidebyside and figure interactions}

22.4 A Single Captionless Image

Sometimes you may want to include a single \texttt{image} or \texttt{table} in an example or exercise without a caption. This formerly required a \texttt{sidebyside}, but here we are testing a more direct markup. The first has width 10\%, 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).

Results will be incorrect for the latter if the widths of the left margin and the panel do not add to 100%. In general, in the case of a single panel, the left margin, right margin, and panel width should all add up to 100%.

Of course, asymmetric margins may be used with several panels. The following is designed to leave a 25% gap between the two panels.

### 22.5 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`.

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`. 
22.6 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 22.13). If only one paragraph is required, simply use the p element on its own.

Similarly, text can be put next to images.

You can place text next to numbered figures, as shown below in Figure 22.12.
22.7 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 9**. 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.

22.8 Tables Side-By-Side

Tables can also be put side-by-side, as demonstrated below in **Figure 22.14**; naturally, subtables can be referenced as in **Table 22.14(a)**.
Table 22.14: Side-by-Side, with tables as children

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaa</td>
<td>bbbb</td>
</tr>
<tr>
<td>AAAA</td>
<td>BBBB</td>
</tr>
</tbody>
</table>

(a) width=50%

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaa</td>
<td>bbbb</td>
</tr>
<tr>
<td>AAAA</td>
<td>BBBB</td>
</tr>
</tbody>
</table>

(b) width=25%

**Figure 22.14:** Side-by-Side, with tables as children

Table 22.15: Widths can be calculated automatically

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaa</td>
<td>bbbb</td>
</tr>
<tr>
<td>AAAA</td>
<td>BBBB</td>
</tr>
</tbody>
</table>

(a)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaa</td>
<td>bbbb</td>
</tr>
<tr>
<td>AAAA</td>
<td>BBBB</td>
</tr>
</tbody>
</table>

(b)

**Figure 22.15:** 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 22.16–22.18.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaa</td>
<td>bbbb</td>
</tr>
<tr>
<td>AAAA</td>
<td>BBBB</td>
</tr>
</tbody>
</table>

Table 22.16

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaa</td>
<td>bbbb</td>
</tr>
<tr>
<td>AAAA</td>
<td>BBBB</td>
</tr>
</tbody>
</table>

Table 22.17

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaa</td>
<td>bbbb</td>
</tr>
<tr>
<td>AAAA</td>
<td>BBBB</td>
</tr>
</tbody>
</table>

Table 22.18

22.9 Tables Next to Figures

Tables and figures can go next to each other, as demonstrated in Table 22.19 and Figure 22.20, plus within an overall captioned figure, Figure 22.21.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaa</td>
<td>bbbb</td>
</tr>
<tr>
<td>AAAA</td>
<td>BBBB</td>
</tr>
</tbody>
</table>

**Figure 22.19:** Table next to a Figure

**Figure 22.20:** Figure next to a Table

**Figure 22.20:** Figure next to a Table

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaa</td>
<td>bbbb</td>
</tr>
<tr>
<td>AAAA</td>
<td>BBBB</td>
</tr>
</tbody>
</table>

Table 22.19: Table next to a Figure

**Figure 22.20:** Figure next to a Table
Table 22.21: Figure and Table, with overall caption, hence sub-captioned

22.10 Tables Next to Text

Tables can go next to blocks of text using the <stack> element (see Subsection 22.13).

Table 22.22: Table next to text

22.11 Tabular Next to Each Other

Four tabular elements inside a single <sidebyside> will result in no captions at all.
22.12 Lists in Side-by-Sides

A “regular” list normally belongs in a \texttt{p} but it can be placed unadorned into a panel of a side-by-side, as demonstrated below in Subsection 22.14. 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 (22.23(b)), or perhaps we might cross-reference by title, \texttt{A List of Colors}.

\begin{center}
\textbf{A List of Colors}
\end{center}

Dr. Seuss again.

\begin{itemize}
\item One fish
\item Two fish
\item Red fish
\item Blue fish
\end{itemize}

\begin{itemize}
\item 1. Blue
\hspace{1em} (a) Light
\hspace{1em} (b) Navy
\hspace{1em} (c) Royal
\item 2. Red
\hspace{1em} (a) Maroon
\hspace{1em} (b) Pink
\hspace{1em} (c) Shocking
\end{itemize}

This ends our example.

\begin{itemize}
\item (a) Sea Life
\item (b) Color Shades
\end{itemize}

\textbf{Figure 22.23:} Two named lists

We also need to test a \texttt{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

\begin{itemize}
\item (a) a very wide
\hspace{1em} i. wide
\end{itemize}

A. rectangle

22.13 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 \texttt{<stack>} is a container that can only be used to collect several items into a single panel of a \texttt{<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 \texttt{sidebyside} panel that does not have a \texttt{caption} or a \texttt{title}. In particular, these panels cannot be sub-numbered since the panel cannot be made into a \texttt{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 \texttt{stack}.

A simple sentence inside a single \texttt{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 emptiness 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 \textit{within} a paragraph. Here we test a mixture of the three items to make sure they are properly separated in a conversion to \LaTeX. 
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.

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

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 22.24:** Experimental results collected in a figure

**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 `\text{La}\text{T}eX\text{ }\text{table}` or an `HTML\text{ }\text{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.
22.14 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 4.11.
4. 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.

\[ f(x) = x^2 + 3x + 2 \]
\[ = (x + 2)(x + 1) \]

here is some text, and here is an equation that contains alignment.

\[ f(x) = x^2 + 3x + 2 \]
\[ = (x + 2)(x + 1) \]

here is some text, and here is an equation that contains alignment.

\[ f(x) = x^2 + 3x + 2 \]
\[ = (x + 2)(x + 1) \]

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.

```cpp
#include

program HelloWorld;
begin
  WriteLn('Hello, world!');
end.

int main()
{
  std::cout << "Hello, world!";
  return 0;
}
```

**Figure 22.25:** “Hello, World!” in Pascal and C++
Figure 22.26: A graph defined by data (from Keller and Trotter’s *Applied Combinatorics*)

### 22.15 Poems as Side-By-Side Panels

Poems may be panels of a side-by-side layout. Here we place some commentary alongside. See Section 24 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,
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

22.16 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 22.27: Overall SBS Group

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.

William Butler Yeats

The main rationale for \texttt{sbsgroup} is to layout a grid of items, and by placing the layout parameters on the \texttt{sbsgroup} element, the items can line up across \texttt{sidebyside} and sub-captioning 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 \texttt{sidebyside}, each with four \texttt{figure} containing an identical \texttt{image}. Since the images are identical and the \texttt{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 \texttt{sidebyside} and cannot be suppressed.

We recycle the prior \texttt{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 sub-captioned. Except the subcaptioning is not implemented. Soon.
Figure 22.40: Twelve images, arranged in a grid

One more test. We override the spacing and vertical alignments of the middle \textit{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 22.41 Figure 22.42 Figure 22.43 Figure 22.44

Figure 22.45 Figure 22.46 Figure 22.47 Figure 22.48

Figure 22.49 Figure 22.50 Figure 22.51 Figure 22.52

22.17 Testing a Side-By-Side First

A \texttt{<sidebyside>} that appears first within some other container can wreak havoc in \LaTeX\ output. Below we have this situation twice, once in an \texttt{<activity>},
then in an \texttt{<example>}

\textbf{Activity 22.1}
Here is text block 1            Here is text block 2

\textbf{Example 22.53}
Here is text block 1            Here is text block 2

\hfill $\square$

And a \texttt{<sbsgroup>} in similar circumstances.

\textbf{Example 22.54}
Here is text block 1            Here is text block 2
Here is text block 3            Here is text block 4

\hfill $\square$

22.18 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 \texttt{figure} with a \texttt{caption} holding a scaled image and a cross-reference for knowl testing: Figure 22.55.

112
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 22 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.


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 23.1: 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 23.2: An <ol> with simple items, a <ul> with items with paragraphs

```r
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
```

Figure 23.3: A <pre> 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 severely 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

Figure 23.4: 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.

<table>
<thead>
<tr>
<th>Organism</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trout</td>
<td>Fish</td>
</tr>
<tr>
<td>Monkey</td>
<td>Mammal</td>
</tr>
<tr>
<td>Crow</td>
<td>Bird</td>
</tr>
<tr>
<td>Crimini</td>
<td>Fungus</td>
</tr>
<tr>
<td>Bee</td>
<td>Insect</td>
</tr>
</tbody>
</table>

Vestibulum sit amet est non lacus accumsan iaculis aliquam nec leo. Maecenas placerat consequat quam, a lobortis odio convallis vitae.

Figure 23.5: 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 23.6: 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).

<table>
<thead>
<tr>
<th>Organism</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trout</td>
<td>Fish</td>
</tr>
<tr>
<td>Monkey</td>
<td>Mammal</td>
</tr>
<tr>
<td>Crow</td>
<td>Bird</td>
</tr>
<tr>
<td>Crimini</td>
<td>Fungus</td>
</tr>
<tr>
<td>Bee</td>
<td>Insect</td>
</tr>
</tbody>
</table>

(a) A Rectangular Test Image (b) Classifying Organisms

**Figure 23.7:** A `<figure>` and a `<table>`

```r
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
```

1. Blue
2. Red
3. Green
4. Purple
5. Violet
6. Brown

That was nice.

(a) A statistical computation (b) Colors Again

**Figure 23.8:** A `<listing>` and a `<list>`

Now we have some more interactive elements

Videos can be placed quite compactly for HTML output, but we display a fair amount of information for a YouTube video in print, and therefore two videos side-by-side gets pretty crowded. The examples above have the bare minimum amount of information attached (not in an overarching figure), and the bare amount which which is displayed in print. We could relax our common spacing to make it a bit better. For other examples see Section 16.
Sometime Around Midnight

YouTube: https://www.youtube.com/watch?v=UYPoMjr6-Ao

(a) Sometime Around Midnight, Airborne Toxic Event, 2009

Figure 23.9: A <sidebyside> with one large panel

We make an exception to our common layout and put one YouTube video, with a start and end time into a single panel of a <sidebyside> as a figure with a title and a caption that is rendered as a sub-caption. Read about “side-by-side” groups (<sbsgroup>) and experiment with stacking several sub-captioned videos into an overall captioned figure (Subsection 22.16).

24 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:
Their’s not to make reply,
Their’s not to reason why,
Their’s 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

25 Advanced Numbering

This section demonstrates the numbering patterns used throughout PreTeXt. There are five subsections. Two intermediate subsections each have two sub-subsections. 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.

25.1 One
A document leaf.

Theorem 25.1 First Theorem. (Cauchy) No statement.

Theorem 25.2 Second Theorem. (Bunyakovsky) No statement.

25.2 Two
Further subdivided.

25.2.1 Uno
A document leaf.

Theorem 25.3 First Theorem! (Schwarz) No statement.

Theorem 25.4 Second Theorem? (Inequality) No statement.

25.2.2 Dos
A document leaf.

Theorem 25.5 First Theorem? No statement.

Theorem 25.6 Second Theorem! No statement.

25.3 Three
A document leaf.
Theorem 25.7  First Theorem.  *No statement.*

Theorem 25.8  Second Theorem.  *No statement.*

25.4 Four

Further subdivided. We include two theorems as numbered items in the introduction to test their numbers, which should always be logical.

Theorem 25.9  Good Numbered Theorem One.  *No statement.*

Theorem 25.10  Good Numbered Theorem Two.  *No statement.*

25.4.1 Uno

A document leaf.

Theorem 25.11  First Theorem.  *No statement.*

Theorem 25.12  Second Theorem.  *No statement.*

25.4.2 Dos

A document leaf.

Theorem 25.13  First 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, use --stringparam numbering.thereoms.level 3 in the xsltproc invocation. See this GitHub issue for details.

Theorem 25.15  Bad Numbered Theorem One.  *No statement.*

Theorem 25.16  Bad Numbered Theorem Two.  *No statement.*

25.5 Five

A document leaf.

Theorem 25.17  First Theorem.  *No statement.*

Theorem 25.18  Second Theorem.  *No statement.*

25.6 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 D.

Theorem 25.1  First Theorem

(Continued on next page)
Theorem 25.2 Second Theorem
Theorem 25.3 First Theorem!
Theorem 25.4 Second Theorem?
Theorem 25.5 First Theorem?
Theorem 25.6 Second Theorem!
Theorem 25.7 First Theorem
Theorem 25.8 Second Theorem
Theorem 25.9 Good Numbered Theorem One
Theorem 25.10 Good Numbered Theorem Two
Theorem 25.11 First Theorem
Theorem 25.12 Second Theorem
Theorem 25.13 First Theorem
Theorem 25.14 Second Theorem
Theorem 25.15 Bad Numbered Theorem One
Theorem 25.16 Bad Numbered Theorem Two
Theorem 25.17 First Theorem
Theorem 25.18 Second Theorem

25.7 a

Right Bracket] 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 25.19 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>.

25.8 Just an Exercise

Checkpoint 25.20 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.

25.9 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.

Checkpoint 25.19 A Right Brace } and a Right Bracket]. Hint. A faux hint to get this exercise to migrate into a <solutions>.
26 Customizations

26.1 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 26.1 (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.

27 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 27.1 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. (HTML Note: The resizer for the knowl is not functioning yet.) Try to evaluate \((-25)^{-1/2}\) 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 27.2 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. (HTML Note: The resizer for the knowl is not functioning yet.) You wish to determine if there is a positive linear correlation between the two variables at a significance level of \(\alpha = 0.001\). You have the following bivariate data set.
\begin{verbatim}
x  y
54  60.5
58.5 137.1
55.1 109.7
57.4 59.1
70.6 122.2
38.3 21.8
69 -79.7
25.1 -56.1
65.3 140.9
63.3 86
26.2 93.3
52.7 85.1
70.2 137.1
55.5 5.9
56.6 117.6
56 44.9
55.7 57.1
38.4 -5.8
40.4 99
56.4 73.1
63.4 -3.5
34.4 99
34.7 -37.7
30.7 51.6
54.5 109.6
34.4 61.4
49.9 26.2
48.5 103
26.2 -66.6
60.9 172.1
25.5 32.3
57.4 87.6
51.2 89.1
58.3 51.2
47.9 107.2
49.2 194.5
23.7 18.7
45.5 100.9
51.8 5.3
71.1 78.7
\end{verbatim}

What is the correlation coefficient for this data set?

\[
r = \]

To find the p-value for a correlation coefficient, you need to convert to a
\[t\]-score: \[t = \sqrt{\frac{r^2(n-2)}{1-r^2}}\] This \(t\)-score is from a \(t\)-distribution with \(n-2\) degrees
of freedom.

What is the p-value for this correlation coefficient?

\[ p = \text{value} \]

Your final conclusion is that...

- There is insufficient sample evidence to support the claim there is a positive correlation between the two variables.

- There is sufficient sample evidence to support the claim that there is a statistically significant positive correlation between the two variables.

Note: In your calculations, round both \( r \) and \( t \) to 3 decimal places in ALL calculations.

**Solution.**

- 0.365
- 0.0103
- There is insufficient sample evidence to support the claim there is a positive correlation between the two variables.

**28 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 a processing switch to enable its appearance:

```
xsltproc -stringparam commentary 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.

Some Commentary.


Some Commentary commentary


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.


29 Worksheets

29.1 About Worksheets

This is a section full of worksheets. Each is a division of its own, via the <worksheet> element. This is a necessary initial <subsection> to the current <section>, since we will have multiple worksheets. 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, with the command-line argument
there will be visual indicators of prescribed whitespace.
29.2 A Geometric Prelude

Objectives

- 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 29.1.

![Figure 29.1: The midpoints of the sides of a quadrilateral are the vertices of a parallelogram.](image1)

In this exercise, we’ll use vectors to show that the medians of any triangle (Figure 29.2) 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 29.4: 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 29.2.1.

3. To show that the point $P$ exists (as the common intersection of the $\vec{M}_i$), show that
   \[
   \vec{A} + \frac{2}{3} \vec{M}_3 = \frac{2}{3} \vec{M}_2 = \ldots
   \]

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.

Wrap-up. 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.
29.3 Networks Worksheet

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 29.5 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 29.6 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 29.7 Kirchoff’s Current Law.** For the circuit below \( I_1 + I_2 = I_3 \).

\[ \square \]

**Theorem 29.8 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 Kirchoffs 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, calculate the amperage in each part of the network by setting up a system of linear equations for the amperages.

\[ \begin{align*}
11 \text{ volt} & \quad 4 \text{ ohm} \\
& \quad 3 \text{ ohm} \\
& \quad 3 \text{ ohm} \\
\end{align*} \]

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.

\[ \begin{align*}
11 \text{ volt} & \quad 4 \text{ ohm} \\
& \quad 3 \text{ ohm} \\
& \quad 3 \text{ ohm} \\
\end{align*} \]

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.

\[ \begin{align*}
11 \text{ volt} & \quad 4 \text{ ohm} \\
& \quad 3 \text{ ohm} \\
& \quad 1 \text{ ohm} \\
& \quad 3 \text{ ohm} \\
& \quad 2 \text{ ohm} \\
& \quad 2 \text{ ohm} \\
\end{align*} \]
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).

![Network Diagram]

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>
<thead>
<tr>
<th></th>
<th>EB Winooski</th>
<th>WB Winooski</th>
<th>Shelburne St</th>
<th>Willow</th>
<th>Jay</th>
</tr>
</thead>
<tbody>
<tr>
<td>into</td>
<td>50</td>
<td>400</td>
<td>0</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>out of</td>
<td>55</td>
<td>390</td>
<td>20</td>
<td>15</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 29.9: Estimated hourly traffic flow for the road network.

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.
This is a mock one-page worksheet for testing purposes. We have specified an overall margin just slightly less than the default.


A two-line paragraph interspersed to check on spacing, breaks and all that.


A two-line paragraph interspersed to check on spacing, breaks and all that.


Activity 29.1 **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.


30 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 30.4.

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 30.2 and 30.2 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 30.2 and Exercise 30.2.

Compare this section with the similar Section 31, next. The following text is mostly nonsense, just for testing purposes.


Theorem 30.3 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. Pellen-

Exercise Collection


![Figure 30.5](image1)

**Figure 30.5:** A figure in an `<exercise>` in an *unstructured* division to test that it is numbered as if the containing `<exercises>` is not present

(a) First Panel, subcaptioned

(b) Second Panel, subcaptioned

cursus ac. Morbi ut auctor lectus.

Figure 30.9: Testing numbering, figure in a `<exercise>` in an un-numbered `<exercises>`


An introduction to an exercise group. This is here to attempt to interrupt the flow of the counting from this division to the next.


**More Reading**

Left intentionally blank, just checking sectioning.


### 31 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 31.3.1.

Compare this section with the similar Section 30, previous. The following text is mostly nonsense, just for testing purposes.

#### 31.1 Faux Subsection


31.2 Drill Exercises


Figure 31.5: A figure in an <exercise> in a structured division to test that it is numbered with consideration of the containing <exercises>

### 31.3 Challenging Exercises


An introduction to an *exercise group*. This is here to attempt to interrupt the flow of the counting from this division to the next.


### 31.4 Impossible Exercises


### 31.5 More Reading

Left intentionally blank, just checking sectioning.


### 32 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 32.4.


   An introduction to an exercise group. This is here to attempt to interrupt the flow of the counting from this division to the next.


8. **Impossible Two.** Aliquam sagittis ex at magna porttitor, quis scelerisque ligula malesuada. Vestibulum vitae mauris id nisi pretium ornare a id lec-


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 on the limit by using the Fundamental Theorem.

    **Solution.** An antiderivative of \( x^2 \) is \( F(x) = \frac{x^3}{3} \), 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.

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.

    (a) Why is this result a Corollary?

    (b) 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).**

    (a) It follows easily.

    (b) 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.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\int_a^b f(x) , dx$</td>
<td>definite integral of $f(x)$</td>
<td>1</td>
</tr>
<tr>
<td>$\int f(x) , dx$</td>
<td>indefinite integral of $f(x)$</td>
<td>2</td>
</tr>
<tr>
<td>$\rho$</td>
<td>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 accommodate really complicated explanations</td>
<td>19</td>
</tr>
<tr>
<td>$\mathbb{Z}/n$ (ring of) integers modulo $n$</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>$\nabla$</td>
<td>gradient operator</td>
<td>20</td>
</tr>
</tbody>
</table>

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 4.4 Essay Question: Compare and Contrast. Hint. Start writing!

4.2.3 · Advice

Checkpoint 4.7 An Inline Exercise. Hint. A good hint.

Answer. 42.

Solution. What was the question?

Activity 4.3 Hints, Answers, Solutions. Answer. The result, but no help in getting there.

Project 4.4 A very structured project.

(c) (i) Answer. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna. Nulla blandit dui metus, male-


(d) **Answer 1.** First answer. In interdum suscipit ullamcorper.


4.2.4 · Exercises

4.2.4.1. **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.

11 · Further Reading

11.3 · More Exercises

<table>
<thead>
<tr>
<th>11.3.6.</th>
<th>11.3.8.</th>
<th>11.3.11.</th>
<th>11.3.13.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hint.</strong> Addition is associative.</td>
<td><strong>Answer.</strong> 15</td>
<td><strong>Hint.</strong> Addition is associative.</td>
<td><strong>Answer.</strong> 15</td>
</tr>
<tr>
<td><strong>Answer.</strong> 12</td>
<td></td>
<td><strong>Answer.</strong> 12</td>
<td></td>
</tr>
<tr>
<td><strong>Solution.</strong> First, add 3 and 4 to get 7, then add 5 to arrive at 12.</td>
<td></td>
<td><strong>Solution.</strong> First, add 3 and 4 to get 7, then add 5 to arrive at 12.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.3.16.</td>
<td>11.3.18.</td>
<td>11.3.21.</td>
<td>11.3.23.</td>
</tr>
<tr>
<td><strong>Hint.</strong> Addition is associative.</td>
<td><strong>Answer.</strong> 15</td>
<td><strong>Hint.</strong> Addition is associative.</td>
<td><strong>Answer.</strong> 15</td>
</tr>
<tr>
<td><strong>Answer.</strong> 12</td>
<td></td>
<td><strong>Answer.</strong> 12</td>
<td></td>
</tr>
<tr>
<td><strong>Solution.</strong> First, add 3 and 4 to get 7, then add 5 to arrive at 12.</td>
<td></td>
<td><strong>Solution.</strong> First, add 3 and 4 to get 7, then add 5 to arrive at 12.</td>
<td></td>
</tr>
</tbody>
</table>

25 · Advanced Numbering

25.7 · A Title with | a Right Bracket
Checkpoint 25.19  A Right Brace } and a Right Bracket]. Hint. A faux hint to get this exercise to migrate into a <solutions>.

25.8  ·  Just an Exercise

Checkpoint 25.20  An Extraneous Exercise. Hint. A hint, so this exercise looks identical in structure to the one in the previous subsection.

29  ·  Worksheets

29.2  ·  A Geometric Prelude

29.2.2.  Hint. Use Worksheet Exercise 29.2.1.

32  ·  Exercises, Top-Level

32.10.  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.

32.42a.  An Exercise with a Hard-Coded Problem Number. Solution. An antiderivative of $x^2$ is $F(x) = x^3/3$, so by the FTC,

$$\int_2^4 x^2 \, dx = F(4) - F(2) = \frac{1}{3} (4^3 - 2^3) = \frac{56}{3}!?!$$

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.

32.12.  Hint 1 (MVT). Consider the definite integral as an area function and employ the Mean Value Theorem.

Hint 2 (Motivator). Think harder!

Answer (Helpful).

(a) It follows easily.

(b) 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>.  

143
D List of Results

We had an automatic list of theorems for just one section, back in Subsection 25.6. 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 20 Program Listings
Theorem 20.3

Section 25 Advanced Numbering

Subsection 25.1 One
Theorem 25.1 First Theorem
Theorem 25.2 Second Theorem

Subsection 25.2 Two
Theorem 25.3 First Theorem!
Theorem 25.4 Second Theorem?
Theorem 25.5 First Theorem?
Theorem 25.6 Second Theorem!

(Continued on next page)
Subsection 25.3 Three
Theorem 25.7 First Theorem
Theorem 25.8 Second Theorem

Subsection 25.4 Four
Theorem 25.9 Good Numbered Theorem One
Theorem 25.10 Good Numbered Theorem Two
Theorem 25.11 First Theorem
Theorem 25.12 Second Theorem
Theorem 25.13 First Theorem
Theorem 25.14 Second Theorem
Theorem 25.15 Bad Numbered Theorem One
Theorem 25.16 Bad Numbered Theorem Two

Subsection 25.5 Five
Theorem 25.17 First Theorem
Theorem 25.18 Second Theorem

Section 29 Worksheets
Theorem 29.5 Ohms Law
Theorem 29.6 Kirchoffs Current Law
Theorem 29.8 Kirchoffs Voltage Law

Section 30 Exercises, One Subsection
Theorem 30.3 Major Result

Section 31 Exercises, Multiple Subsections
Subsection 31.1 Faux Subsection
Theorem 31.3 Major Result

E A Structured Appendix
A deeply-structured appendix for testing purposes.

E.1 A Subsection in an Appendix
Inside a subsection.

E.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.

F 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.

References


G Multiple References

G.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: [G.3.2].
G.2 General References


G.3 Specialized References


Index

\LaTeX{} image, 35
verbatim text, use sortby, 2

A, See F, See G, See H
B, 2
C, See also A, See also D, 2
activity, 7
alert, 15
Archimedian Spiral, 65
assemblage, 14
asymptote graphics language, 38
attributes
doctest, 2
ncbi attribute, 32
autoname, 78

biological names, 32
genus, 32
ncbi attribute, 32
species, 32
taxon, 32

CalcPlot3D, 71
canceling a term, 25
cancelto macro, 25
sorted as if “Cat”, 2
CAYLEY GRAPH, 2
Cayley graph, 2
cayley graph, 2
commentary, 124
contained paragraph, 124
conundrum
repurposed from proposition, 121
cross-reference, 78

D3.js, 62
description list, 51
Desmos graphs, 70
doctest, 2

em
emphasis, 15
embedded interactive elements
authored, 60
server, 69
exploration, 7

fill-in blank, 45
footnotes, 16
Fundamental Theorem of Calculus, 1

Corollary, 5
genus, 32
Geogebra
material, 70
server, 70
Group, 50
group
definition, 50
image, 97
\LaTeX{} image, 35
indefinite integral, 2
integral
indefinite integral, 2
interactive elements
authored, 60
server, 69
internationalization, 83
investigation, 7
JSXGraph, 65
knowl
nested, 80
Led Zeppelin video, 76
link
external, url, 32, 74
list, 53
description, 51
multicolumn, 17–19
named, 52
ordered, 47
unordered, 47
listing, 89
matrix, 87
Maxwell’s equations, 20
mixed-content emphasized, 2
name, 121
named list, 52
ncbi attribute, 32
numbering, 118
numerical, See Sage integration,
See Sage
numerical integration
Sage
cell, See Sage
numerics, See also Sage, 2
ordered list, 47
package
siunitx, 94
panels, 95
paragraph
  initial list, 50
  normal, 53
  opens with list, 53
poem, 108, 117
problem, 7
program listing, 89
project, 7
PSTricks, 38
question, 7
sorted as if “Quorum”, 2
quotations, 28
R, 3
$\rho$ – fibers, 2
reference
  external, 83
  external, url, 32, 74
references
  within a section, 47
rename an environment, 7
comundrum, 121

Sage
  integration, 2
  cell, 2
  numerical, 2
  interacts, 66
Sage cell
  with a title, 4
Sage cells
  linking, 13
Sage plots, 40
scientific names, See biological names

siunitx package, 94
species, 32
strange colors, 17
structured-content emphasized, 2
styling words
  alert, 15
  delete, 15
  em, 15
  emphasis, 15
  insert, 15
  stale, 15
table, 101
task, 7
taxon, 32
test: buried in
  theorem/statement/p, 1
Underdown, Jason, 25
Unicode, 84
units, 94
  Z (sort as A), 2
  A (sort as Z), 2
unordered list, 47
url, 83
URLs, 32
videos
  YouTube, 75
Wolfram CDF, 72
X, See also F, See also G, See also H, 2
  Y, 2
  Z, See A, See D
x as a variable, 2

YouTube videos, 75

Colophon

This article was authored in, and produced with, PreTeXt. It is typeset with the Latin Modern font.