

Derivatives and Integrals

An Annotated Discourse

Robert Beezer

Department of Mathematics and Computer Science

University of Puget Sound

Tacoma, Washington, USA

beezer@pugetsound.edu

November 1, 2017

Abstract

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

1 Introduction

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

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

This is also a demonstration of the capabilities of [PreTeXt](#).

Generated: November 1, 2017, 08:59:38 (-04:00)

2 The Fundamental Theorem

There is a remarkable theorem:¹

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

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

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

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

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, and selectively display them, so you may not see the one here in the output you are looking at now. Or maybe you do see it?

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

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

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

3 Computing Integrals with Sage (\int)

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

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

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

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

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

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

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

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

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

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

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

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 L^AT_EX version (and always be the same).

```
random()
```

```
0.11736021338650582
```

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

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

12.005

Sage, 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, the second most-dangerous special character in XML. You need to escape it by writing `<`; as we have done in the source for this example. (See the discussion and summary in Subsection ??.)

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

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

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

Here is another R cell. Unfortunately, it seems Sage’s doctest facility cannot be used easily with code from other languages. In the source for this example, we have employed a `CDATA` element to protect all the characters from the XML processor.

```
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

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

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

You can place Sage cells inside of a paragraphs if you want to give them a title, but no numbers, etc. Their surrounding box sometimes gets clobbered in L^AT_EX 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 <objectives> element you are reading, and this is its introduction. This early section has really grown and tries to accomplish many things. Not all of them are listed here.

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

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

This is across-reference to one of the objectives above, forced to always using the 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 chapter number: [Objective 1 of Section ??](#). For comparison this is the (forced) type-global cross-reference: [Objective 4.1](#).

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

4.1 Second Version of FTC

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

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

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

$$\frac{d}{dx} \int_a^x f(t) dt = \frac{d}{dx} (F(x) - F(a)) \quad (4.2)$$

$$\begin{aligned} &= \frac{d}{dx} F(x) - \frac{d}{dx} F(a) \\ &= f(x) - 0 = f(x). \end{aligned} \quad (4.3)$$

□

Proof. You can have multiple proofs. Here we just exercise displayed math with no automatic numbering, and an elective number on the middle equation. For L^AT_EX output, with no number on the third line, the tombstone is placed on that line.

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

□

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

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

2+2

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

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

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

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

That's it.

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

2+2

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

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

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

(\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. □

4.2 A Pedagogical Note

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.

Exercise 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 subsubsection 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 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.

¹Which I think sometimes students lose sight of.

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?

Exercise 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 **sectional 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?

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 ??).

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 4.8 (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.

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.

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

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

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

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

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

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

A short paragraph, before an answer.

Answer. 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.

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

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

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

(A) First subsubtask. Short paragraph.

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

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

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

In interdum suscipit ullamcorper

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

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

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

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

- (d) This third top-level task is intermediate in complexity, you are reading the **statement**, which is followed by more items. In interdum suscipit ullamcorper. Morbi sit amet malesuada augue, id vestibulum magna.

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

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

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

Answer 1. First answer. In interdum suscipit ullamcorper.

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

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

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

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

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

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 4.9 (Sample Use of Sage). This would be a good place to talk about Sage, including a cell or two.

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

$4x^3$

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.
---	--

4.3 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 4.10 (The Title Principle). *It is a fundamental principle that many elements can have a title. Try it and see. If you get better formatting, then it is being recognized. If it looks very plain, check the documentation and perhaps make a feature request.*

More precisely, `<theorem>`, `<corollary>`, `<lemma>`, `<algorithm>`, `<proposition>`, `<claim>`, `<fact>`, and `<identity>`, all behave exactly the same, requiring a statement (as a sequence of paragraphs) followed by an optional proof, and may have an optional title. The elements `<axiom>`, `<conjecture>`, `<principle>`, `<heuristic>`, `<hypothesis>`, and `<assumption>` are functionally the same, barring a proof (since they would never have one!). Definitions are an exception, as it is natural to place `<notation>` within—see the source for Definition ?? 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. As a reader you are suggested to “replay” the cell. You have seen the next cell before. More typically such a cell might define a function or set some values of a variable.

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

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

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 (`p`) and side-by-sides. The intent is that these do not have captions, so are not numbered, so cannot be cross-referenced and so do not become knowls (inside of the knowled assemblage). 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 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 ??)
- Sage cells, including with R. (Section ??)
- Lots of document structure, like introductions and conclusions (next). (Section ??)

A sample table, with no caption, follows.

A	B	C
Uno	Dos	Tres

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

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 `` 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.

5 Some Facts and Figures

Because of the Fundamental Theorem, for every derivative we know, there is an antiderivative we might find useful. Because of the Fundamental Theorem of Calculus, 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 gain a greater understanding of derivatives by studying the graphs of functions with their derivatives. Can you discern the derivative–antiderivative relationship in Figure ???

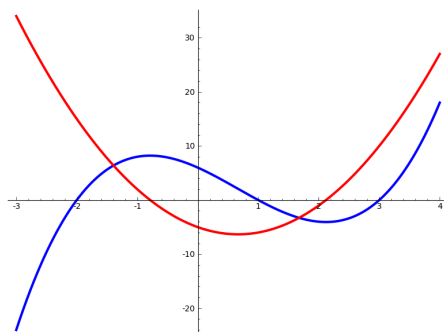


Figure 5.2: A function and its derivative

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

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

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

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

nibh dui vulputate leo, nec tincidunt augue ipsum sed odio. Nunc non erat sollicitudin, iaculis eros consequat, dapibus eros.

- Donec vestibulum auctor nisl. Nullam placerat interdum dui. Quisque lobortis scelerisque augue imperdiet placerat. Maecenas ultricies massa tempor, laoreet urna a, eleifend enim. Integer sed suscipit odio. Pellentesque non dapibus diam, eget tempus dui. Maecenas sollicitudin magna viverra, egestas velit nec, tristique sem. Cras iaculis mattis dui ac cursus. Integer volutpat, urna vel tempus convallis, erat nisi consectetur turpis, id varius dolor lorem vitae mauris. Phasellus erat orci, laoreet commodo gravida quis, congue in lacus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Praesent at bibendum turpis. Pellentesque est nisl, dapibus at sagittis non, ultricies in nunc. Etiam ipsum arcu, porta sed feugiat eget, facilisis nec libero. Mauris tempor convallis felis.

Cras iaculis sapien elit, at convallis ligula convallis nec. Duis ante tortor, euismod a libero vitae, ornare viverra purus. Pellentesque facilisis urna a velit

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

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

Vivamus eu commodo est, pretium fringilla dolor. Curabitur vel sollicitudin libero. Integer sit amet auctor felis. Maecenas sagittis erat at ante feugiat, in tincidunt ligula pretium. Integer eget auctor ipsum, quis volutpat felis. Morbi id dignissim eros. Suspendisse aliquet pulvinar lorem gravida egestas. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Praesent nec massa dui. Suspendisse convallis lacus sit amet adipiscing varius. Suspendisse tem-

pus diam vitae justo ornare, in condimentum metus pharetra. Curabitur sem dolor, auctor vitae sagittis vestibulum, posuere imperdiet metus. Etiam pretium lacus urna, vel auctor diam tincidunt non. Etiam viverra sodales iaculis.

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

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

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,

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

L^AT_EX 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.

$$\mathcal{L}(y')(s) = s\mathcal{L}(y)(s) - y(0) = sY(s) - y(0) \quad (6.1)$$

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

And so it follows that,

$$\mathcal{L}(y')(s)^{++} = s\mathcal{L}(y)(s) - y(0) = sY(s) - y(0) \quad (6.3)$$

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

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

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

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

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

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

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

Table 6.1: Euler's approximation for Duffing's Equation with $h = 0.2$

7 Mathematics

To be able to create both L^AT_EX 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:

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

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

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

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

$$A\overset{\Phi+\Psi+\Theta}{\underset{\text{bijection}}{\rightarrow}}B$$

Look back at the top of the source file of this document to see how to include your T_EX 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.

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

$$\frac{dy}{dt}=x^3-y.\tag{7.2}$$

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 `mrow` tag, rather than using $n-1$ separators (such as `\\`).

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

Example:

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

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

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

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

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

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

$$\begin{array}{lll}\frac{dx}{dt} = x^2 - 4x - y + 4 & \frac{dy}{dt} = x^3 - y & x, y \text{ version} \\ \frac{dw}{dt} = z^3 - w & \frac{dz}{dt} = z^2 - 4z - w + 4 & z, w \text{ version.}\end{array}$$

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

$$\begin{array}{l}\frac{dw}{dt} = z^3 - w \\ \frac{dx}{dt} = x^2 - 4x - y + 4 \frac{dy}{dt} = x^3 - y, y \text{ third column}\end{array}$$

$$\frac{dw}{dt} = z^3 - w.$$

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

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

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

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

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

Beautiful.

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 L^AT_EX output it could be interesting. First, with no extra effort, this page-long display should break naturally, no matter how the preceding material changes.

$$\begin{array}{c} x^2 + y^2 = z^2 \\ a^2 + b^2 = c^2 \\ \alpha^2 + \beta^2 = \gamma^2 \\ m^2 + n^2 = p^2 \\ x^2 + y^2 = z^2 \\ a^2 + b^2 = c^2 \\ \alpha^2 + \beta^2 = \gamma^2 \\ m^2 + n^2 = p^2 \\ x^2 + y^2 = z^2 \\ a^2 + b^2 = c^2 \\ \alpha^2 + \beta^2 = \gamma^2 \\ m^2 + n^2 = p^2 \\ x^2 + y^2 = z^2 \\ a^2 + b^2 = c^2 \\ \alpha^2 + \beta^2 = \gamma^2 \\ m^2 + n^2 = p^2 \\ x^2 + y^2 = z^2 \\ a^2 + b^2 = c^2 \\ \alpha^2 + \beta^2 = \gamma^2 \end{array}$$

$$\begin{aligned}
m^2 + n^2 &= p^2 \\
x^2 + y^2 &= z^2 \\
a^2 + b^2 &= c^2 \\
\alpha^2 + \beta^2 &= \gamma^2 \\
m^2 + n^2 &= p^2 \\
x^2 + y^2 &= z^2 \\
a^2 + b^2 &= c^2 \\
\alpha^2 + \beta^2 &= \gamma^2 \\
m^2 + n^2 &= p^2 \\
x^2 + y^2 &= z^2 \\
a^2 + b^2 &= c^2 \\
\alpha^2 + \beta^2 &= \gamma^2 \\
m^2 + n^2 &= p^2 \\
x^2 + y^2 &= z^2 \\
a^2 + b^2 &= c^2 \\
\alpha^2 + \beta^2 &= \gamma^2 \\
m^2 + n^2 &= p^2 \\
x^2 + y^2 &= z^2 \\
a^2 + b^2 &= c^2 \\
\alpha^2 + \beta^2 &= \gamma^2 \\
m^2 + n^2 &= p^2 \\
x^2 + y^2 &= z^2 \\
a^2 + b^2 &= c^2 \\
\alpha^2 + \beta^2 &= \gamma^2 \\
m^2 + n^2 &= p^2 \\
x^2 + y^2 &= z^2 \\
a^2 + b^2 &= c^2 \\
\alpha^2 + \beta^2 &= \gamma^2 \\
m^2 + n^2 &= p^2.
\end{aligned}$$

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

$$x^2 + y^2 = z^2 \tag{7.3}$$

$$a^2 + b^2 = c^2 \tag{7.4}$$

$$\alpha^2 + \beta^2 = \gamma^2 \tag{7.5}$$

$$m^2 + n^2 = p^2 \tag{7.6}$$

$$x^2 + y^2 = z^2 \tag{7.7}$$

$$a^2 + b^2 = c^2 \tag{7.8}$$

$$\alpha^2 + \beta^2 = \gamma^2 \tag{7.9}$$

$$m^2 + n^2 = p^2 \tag{7.10}$$

$$x^2 + y^2 = z^2 \quad (7.11)$$

$$a^2 + b^2 = c^2 \quad (7.12)$$

$$\alpha^2 + \beta^2 = \gamma^2 \quad (7.13)$$

$$m^2 + n^2 = p^2 \quad (7.14)$$

$$x^2 + y^2 = z^2 \quad (7.15)$$

$$a^2 + b^2 = c^2 \quad (7.16)$$

$$\alpha^2 + \beta^2 = \gamma^2 \quad (7.17)$$

$$m^2 + n^2 = p^2 \quad (7.18)$$

$$x^2 + y^2 = z^2 \quad (7.19)$$

$$a^2 + b^2 = c^2 \quad (7.20)$$

$$\alpha^2 + \beta^2 = \gamma^2 \quad (7.21)$$

$$m^2 + n^2 = p^2 \quad (7.22)$$

$$x^2 + y^2 = z^2 \quad (7.23)$$

$$a^2 + b^2 = c^2 \quad (7.24)$$

$$\alpha^2 + \beta^2 = \gamma^2 \quad (7.25)$$

$$m^2 + n^2 = p^2 \quad (7.26)$$

$$x^2 + y^2 = z^2 \quad (7.27)$$

$$a^2 + b^2 = c^2 \quad (7.28)$$

$$\alpha^2 + \beta^2 = \gamma^2 \quad (7.29)$$

$$m^2 + n^2 = p^2 \quad (7.30)$$

$$x^2 + y^2 = z^2 \quad (7.31)$$

$$a^2 + b^2 = c^2 \quad (7.32)$$

$$\alpha^2 + \beta^2 = \gamma^2 \quad (7.33)$$

$$m^2 + n^2 = p^2 \quad (7.34)$$

$$x^2 + y^2 = z^2 \quad (7.35)$$

$$a^2 + b^2 = c^2 \quad (7.36)$$

$$\alpha^2 + \beta^2 = \gamma^2 \quad (7.37)$$

$$m^2 + n^2 = p^2 \quad (7.38)$$

$$x^2 + y^2 = z^2 \quad (7.39)$$

$$a^2 + b^2 = c^2 \quad (7.40)$$

$$\alpha^2 + \beta^2 = \gamma^2 \quad (7.41)$$

$$m^2 + n^2 = p^2 \quad (7.42)$$

$$x^2 + y^2 = z^2 \quad (7.43)$$

$$a^2 + b^2 = c^2 \quad (7.44)$$

$$\alpha^2 + \beta^2 = \gamma^2 \quad (7.45)$$

$$m^2 + n^2 = p^2 \quad (7.46)$$

$$x^2 + y^2 = z^2 \quad (7.47)$$

$$a^2 + b^2 = c^2 \quad (7.48)$$

$$\alpha^2 + \beta^2 = \gamma^2 \quad (7.49)$$

$$m^2 + n^2 = p^2. \quad (7.50)$$

So. Do not take any extra steps and let L^AT_EX 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.

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 L^AT_EX Packages and MathJax Extensions

If you would like to use macros from a L^AT_EX 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{ }{ }`.

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

Look at the source of this article to see the package name being supplied in a latex-preamble/package element within the docinfo section. That is the only setup required to make the macro usable in L^AT_EX 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 L^AT_EX in web browsers, and improving all the time. You can get fairly fancy with some of its supported commands. In particular, if you need to mix in a few words with your mathematics, the `\text{ }` macro is supported. For example, you might use an “if” or an “otherwise” in the definition of a piecewise function.

Consider that the first line below is text sandwiched in-between two Greek letters, wrapped in a `\text{ }` macro. In HTML output we have taken care that the font for text material within display mathematics should match the font of the surrounding paragraph, as also happens with L^AT_EX output. The second line is nearly identical in the source, but is just naked text being rendered like a slew of variables.

$$\alpha \text{ is not equal to } \beta$$

$$\alpha isnotequalto \beta$$

$$\alpha \neq \beta.$$

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

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

$$y \rightarrow \frac{\sin(0)}{0} \rightarrow \frac{0}{0}.$$

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.

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

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,

$$\begin{array}{ll} A = B + C & \text{Corollary ??} \\ = D + E & \text{The Fundamental Theorem of Calculus.} \end{array}$$

8 Special, Reserved, and Escape Characters

8.1 Reserved 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.

Everybody wants the ampersand, it is the most-dangerous character. It is *the* escape character for XML, and L^AT_EX uses it to organize tables and arrays, and for aligning equations. Consistently use the element `<ampersand />` to make a literal ampersand in normal text, such as in “A&P.” In mathematics, and other places where you are using L^AT_EX syntax, use the pre-defined `\amp` macro. For code listings and other verbatim text, use the escaped XML entity `&`.

The left angle bracket (`<`) is the second most-dangerous character in your source, since it looks to the XML processor like the start of a new XML element. The right angle bracket (`>`) is less dangerous, but for symmetry we treat it the same as the left. Consistently use the elements `<less />` and `<greater />` to make left and right angle brackets in normal text. In mathematics, and other places where you are using L^AT_EX syntax, use the pre-defined `\lt` and `\gt` macros. For code listings and other verbatim text, use the XML entities `<` and `>`.

Sage defines generators of algebraic structures with a syntax that might remind one of common notation for all “combinations” of some generators. It is non-standard Python, but is instead pre-parsed by Sage. No matter, at issue here is the left angle bracket used to specify generators. Here is an example, which can be doctested by Sage to verify the example behaves correctly. Look at the source to see how the generator syntax is created with the XML entities.

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

Univariate Polynomial Ring in t over Integer Ring

There is an alternate Sage syntax, which avoids the angle brackets.

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

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

Ampersands and angle brackets are likely to be necessary in source code, such as Sage code (think generators of field extensions) or TikZ code (think arrowheads), and in matrices (think separating entries). If you have a big matrix, or a huge chunk of TikZ code, you can protect it all at once from the XML processor by wrapping it in “<![CDATA[]]>.” It should be possible to write without ever using the “CDATA” mechanism, but it might get tedious in places to use the supplied macros or XML entities.

The other XML reserved characters are the quotation marks, single and double, ' and ". Their use is only constrained in attributes and so do not present a problem elsewhere. Here are the three XML reserved characters rendered as normal text, see the source to see how they were authored.

& < >

We test the three \LaTeX macros for these characters with a pair of aligned equations:

$$\begin{aligned} a^2 + b^2 &< c^2 \\ c^2 &> a^2 + b^2 \end{aligned}$$

So as a summary of how to avoid conflicts with XML’s reserved characters:

“Normal” Text Use <ampersand />, <less />, <greater />.

Mathematics Within m, me, men, and mrow elements, use \amp, \lt, \gt. Or use CDATA to enclose a large chunk of \LaTeX with many of these characters.

Verbatim, Code Within verbatim text (c and pre elements), Sage code, program listings, and console sessions, use the XML entities &, <, > to get exactly the characters desired.

It might be instructive to see how the paragraphs above about escape characters were written without inadvertently using an escape character improperly.

There are a handful of characters that might render just fine in HTML, but \LaTeX reserves them for special purposes. So if they appear unadorned in your source, they will wreak havoc with the \LaTeX processing. And if you escape them with backslashes to migrate to the \LaTeX output, then you will see those backslashes in your HTML. And the backslash is the escape character for Markdown and JSON. You can’t win. Thus, you need to be aware of these symbols and use the provided PreTeXt elements for each in order to get the right behavior in each type of output. Here are the outputs, look at the source of this document to see the input elements.

\$ % ^ & _ { } ~ \ *

8.2 Pseudo-Characters and Constructions

There are a few other very 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. Again, outputs here, see the source for inputs. 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 .

e.g. i.e. etc. c.

There are a few other characters and marks that get special treatment. Some do not appear on your keyboard, such as the symbol for copyright (and similar business or legal marks in common use). Then there are some characters that do not appear on your keyboard but frequently a keyboard character is used as a substitute. For example, a fraction bar and a forward slash (solidus and slash, respectively) have slightly different slopes. Also, compare a tilde and a swungdash. You can fake a midpoint in \LaTeX by going to math mode, but the midpoint is really a text character. Again, outputs here, see the source for inputs. Using these uniformly will lead to the best quality in all your outputs, though some of these are very infrequent, or the distinctions are not always that important.

© ® ™ ... · ~ ‰ § × / /

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

8.3 URLs

An internet URL can very well contain some of the characters that \LaTeX needs to escape. But the packages we use for embedded links should be smart about this. So we include a long URL for testing \LaTeX output, with one reserved character, though maybe someday it will become stale and we need to change it out: www.pcc.edu/enroll/registration/dropping.html#withdraw. Notice in the source that you *cannot* put a tag inside the href attribute, and do need to use an element within the content (unless you like to wrap the content in a c element). Here is a totally bogus URL, which contains every possible legal character, so if this fails to convert there is some problematic character. Four combinations: with the content as normal text versus with the characters as verbatim text, and as a URL versus not.

```

ABCDEF GHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789%-
._~: /?#[]@!$&'()*+;,=
ABCDEF GHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789%-
._~: /?#[]@!$&'()*+;,=
ABCDEF GHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789%-
._~: /?#[]@!$&'()*+;,=
ABCDEF GHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789%-
._~: /?#[]@!$&'()*+;,=
```

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 />`: two strings of normal text.

- For L^AT_EX output, the verbatim c element will be automatically delimited by a character that is not in the string. The fault is a question mark, which you see here in the string. So we have twice used the latexsep attribute with the value | (the **pipe** character) which cannot ever appear in a URL.

When a url has no content, then its href attribute is displayed as the text, automatically in a verbatim font (so no need to consider the latexsep attribute in any way).

```

ABCDEFHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789%-
._~:/?#[]@!$&'()*+.,;=

```

We are not fans of footnotes, they are totally unstructured¹. A URL in a footnote migrates around, and so care must be taken with special characters, such as the percent and hash². 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.

8.4 Quotations

The q tag will provide beginning and ending double quotations, while the sq tag will behave similarly but provide single quotes.

“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

¹Carleson's Theorem

²ABCDEFHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789%-._~:/?#[]@!\$&'()*+.,;=

double versions. (For example, see Section ?? 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 L^AT_EX 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 L^AT_EX with themselves, or with the grouping tags.

8.5 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], <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.

8.6 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 from the [National Center for Biotechnology Information](http://www.ncbi.nlm.nih.gov/taxonomy). 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](#).

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

You can test the migration of the L^AT_EX special characters in this section title by requesting a 2-deep Table of Contents with --stringparam toc.level 2.

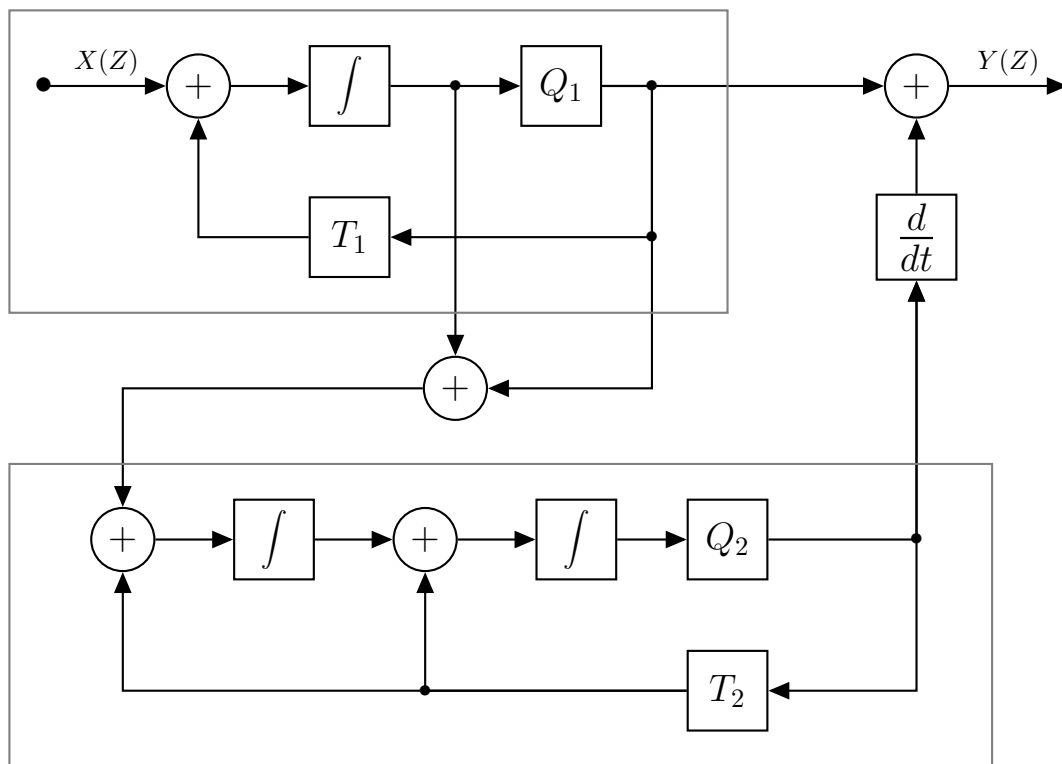
9 Graphics

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 L^AT_EX macros used in the rest of your document may be employed in the L^AT_EX-standalone or Asymptote diagrams (with Sage graphics coming next?).

9.1 L^AT_EX images

There are several graphics engine packages that a \LaTeX document can employ. Code from these packages renders diagrams automatically as part of normal processing of \LaTeX files. For HTML output the `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 `docinfo/latex-image-preamble`, which is also where global package settings should be made. As mentioned in Subsection ??, if any ampersands occur in your \LaTeX code you should use the `\amp` macro. These first examples are from the [TeXample.net](http://www.TeXample.net) site.

FIRST-ORDER NOISE SHAPER



SECOND-ORDER NOISE SHAPER

Figure 9.1: TikZ Electronics Diagram

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

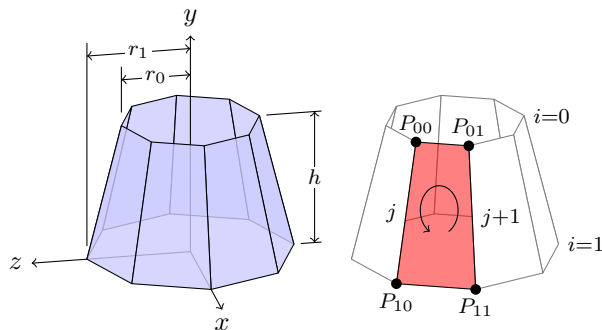


Figure 9.2: TikZ Cone Drawing

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

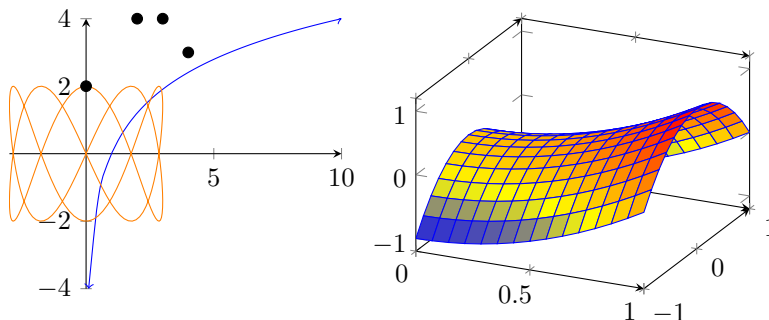


Figure 9.3: Sample pgfplots plot

A plot might use a graphics language to draw the axes and grid, but the data might be from an experiment and live in an external file that you do not wish to place in your source. Place such a file in a subdirectory directly below the directory where your master source file resides. 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 L^AT_EX 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.

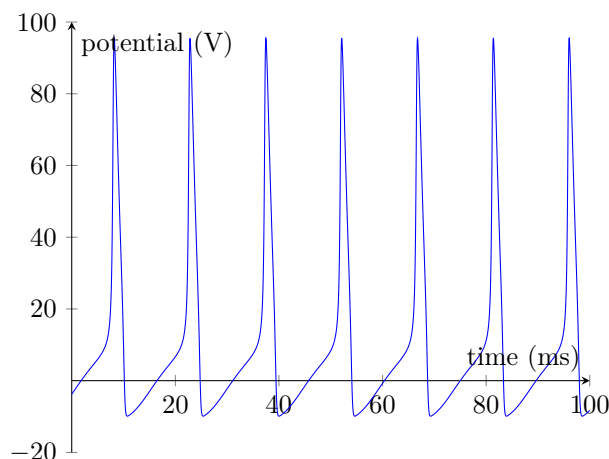
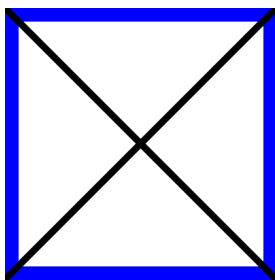


Figure 9.4: External data in a pgfplots plot

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 ?? 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.

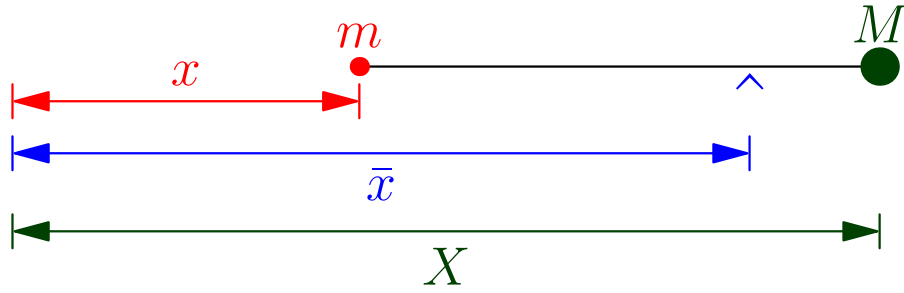


Figure 9.5: Asymptote Lever Demonstration

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

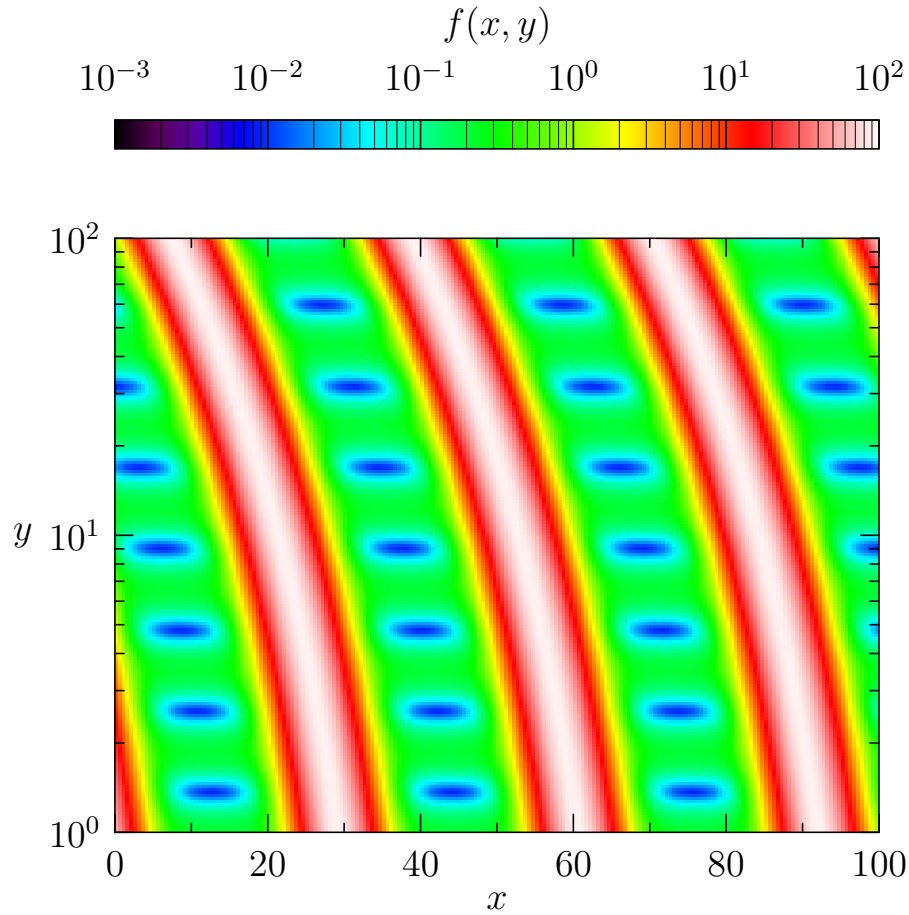


Figure 9.6: Asymptote Contour Plot

Here is the lever diagram again, but now we have added an integral to one of the legends, *using a \LaTeX macro of our own*, which is 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.

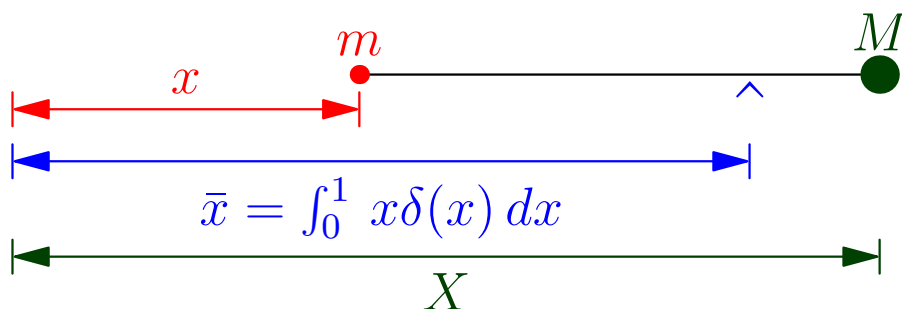


Figure 9.7: Asymptote Lever, plus Integral

And finally, an example of a 3-D graph (from the documentation again).

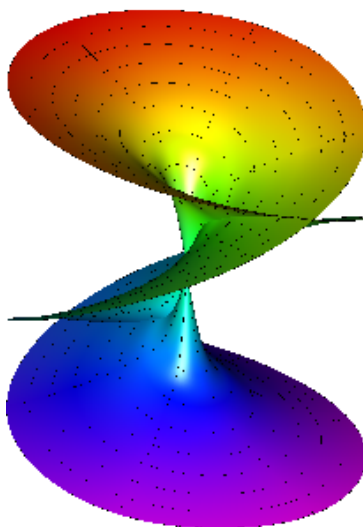


Figure 9.8: Asymptote 3-D Surface

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 *must* return a Sage Graphics object (or 3D plot). The `mbx` script will isolate this last line, use it as the RHS of an assignment statement, and the Sage `.save()` method will be called to generate the image, which is either a Portable Document Format (PDF) file amenable to \LaTeX output, or a Scalable Vector Graphics (SVG) file amenable to HTML output. For visualizations of 3D plots, Sage will only produce Portable Network Graphics (PNG) files, which can be included in HTML pages or \LaTeX output.

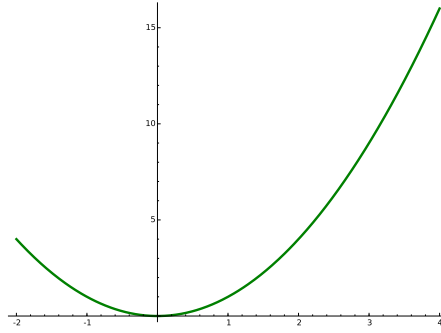


Figure 9.9: A Sage standard parabola, on $[-2, 4]$

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

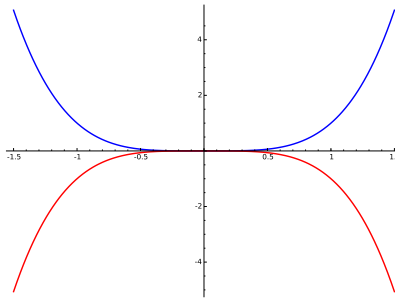


Figure 9.10: Two Sage plots on one set of axes

The following examples are from the [Sage Tour](#). We package them into a `sidebyside` layout element, see Section ??.

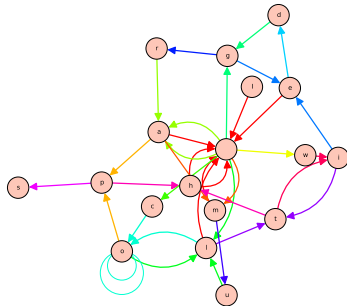


Figure 9.11: A Sage multi-graph of a sentence

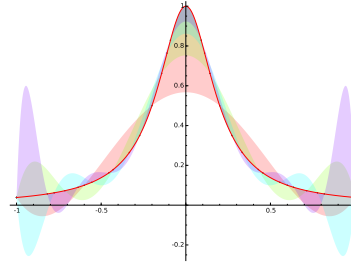


Figure 9.12: Sage polynomial approximations of $f(x) = 1/(1 + 25x^2)$

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

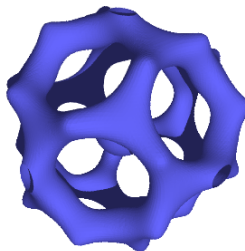


Figure 9.13: A Sage implicitly defined 3D surface

9.5 Images from External Sources

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

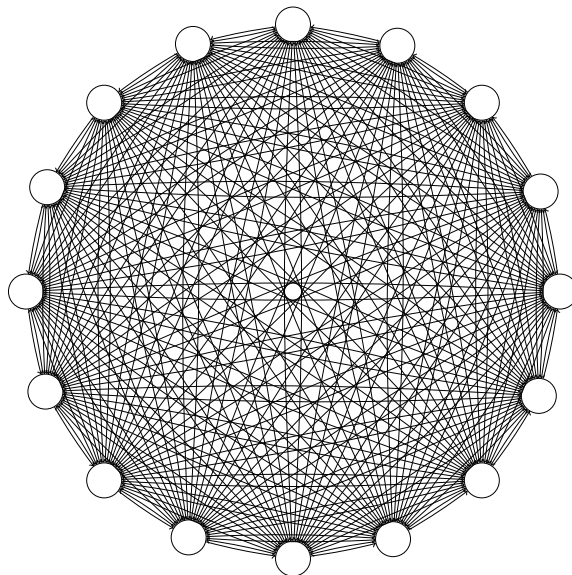


Figure 9.14: Complete graph on 16 vertices, from www.texample.net

9.6 Copies of Images

So you do not have to duplicate the source of an image (and risk them later diverging), or for other reasons of efficiency, you can place an image as a copy

of another one. The copy is an *exact* copy, such as having the identical width. Though, if placed within a figure element, the caption and so on, can be changed.

To use this feature, simply be certain to give the original an `xml:id` and then place an `image` tag where you want the copy. Then use a `@copy` attribute to point to the original. Two test examples, one from TikZ source, the other from a raster image.

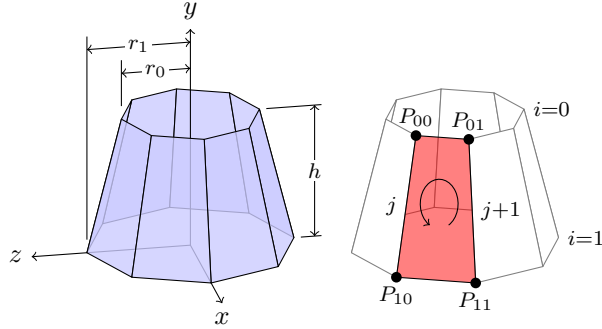


Figure 9.15: TikZ Cone Drawing (Copy)

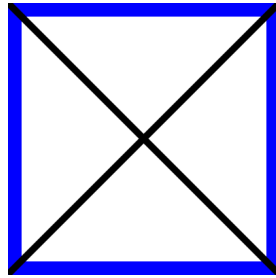


Figure 9.16: Copy of raster image, now numbered, captioned

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 `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 `@source` attribute should include the complete relative path with the subdirectory. Files generated by the `mbx` script will be specified in the output using the relative directory `images`, which can be changed using the `directory.images` stringparam. There is no reason author-provided files cannot also be placed in this same directory (presuming no duplicate names). [This table is presently more readable in HTML, the PDF version will improve.]

MBX Element	Specification	L ^A T _E X/Print	HTML	Notes
image/@source	full relative path, w/ extension	directly included	directly included	author-p
image/@source	full relative path, w/o extension	presumes PDF	presumes SVG	author-p
image/latex-image-code	L ^A T _E X-compatible source	directly included	SVG via mbx	e.g. tikz,
image/sageplot	Sage code	PDF via mbx	SVG via mbx	PNG for
image/asymptote	Asymptote code	PDF via mbx	SVG via mbx	

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

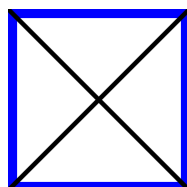


Figure 9.17: A caption can be a whole paragraph with lots of technical details, and maybe a hyperlink to something external, such as mathbook.pugetsound.edu. There could be some inline mathematics, such as $x^2 + y^2 = c^2$. Would a knowl open here? Recursively? Let's see: ?? Display mathematics, side-by-sides, theorems, and lots of other things should be banned. Footnotes sound like a bad idea. Strange characters should be fine: §.

10 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 [11.2.2, Chapter R], included an indication that a specific chapter may be relevant.

11.2 References

These items are here to test basic formatting of references.

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

- So use inside a displayed equation

$$16 \log \text{_____}$$

like this one.

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

$$\begin{aligned} y &= x^7 x^8 \\ &= x \text{---} \end{aligned}$$

11.4 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. $9 + 10$

This feature was designed with short “drill” exercises in mind.

12 List Calisthenics

12.1 Lists, Generally

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

This section contains nested lists, to demonstrate how they get assigned labels (numbering, symbols). But we begin with two simple lists, demonstrating an ordered list and an unordered list. See the end of section for an example of a description list. Since Jupyter notebooks use markdown syntax, their lists are less flexible. So some assertions here may be wrong when viewed as a Jupyter notebook. Note in the source the optional use of a paragraph (`p`) for the list items.

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. The default order of the labels in Markdown/Jupyter (Arabic, Latin, latin, roman) is different than for \LaTeX and HTML (Arabic, latin, roman, Latin), so cross-references are not correct. 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 may be given 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](#).

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 L^AT_EX and HTML (disc, circle, square, disc)

- Level 1, first.
- Level 1, second.
 - Level 2, first.
 - Level 2, second.
 - Level 3, first.
 - Level 3, second.
 - Level 4, first.
 - Level 4, second.
 - Level 4, third.
 - Level 3, third.
 - Level 2, third.
- Level 1, third.

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

1. Level 1, first.
2. Level 1, second.
 - (a) Level 2, first.
 - (b) Level 2, second.
 - Level 3, first.
 - Level 3, second.
 - i. Level 4, first.
 - ii. Level 4, second.
 - A. Level 5, first.
 - B. Level 5, second.
 - Level 6, first.
 - Level 6, second.
 - Level 6, third.
 - C. Level 5, third.
 - iii. Level 4, third.
 - Level 3, third.
 - (c) Level 2, third.
3. Level 1, third.

Now, nested lists with the defaults replaced by custom choices. First, an ordered list, three deep, upper Roman numerals, then upper-case Latin, then more traditional Arabic numerals on the three elements of the third level. Note the adornments of the labels will be rendered in L^AT_EX, but not in HTML, and the label specifications have no effect in a Jupyter notebook.

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 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 L^AT_EX normally begins right after the heading.

Definition 12.1 (Group).

a) There is a binary operation, denoted “.”.

b) The operation is associative.

c) There is an identity element, e .

d) For every element b , there is an element c (the inverse), such that

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

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

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

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

The next two subdivisions are an Exercises subdivision and a References subdivision, which have lists within an exercise and a bibliographic item (respectively).

12.2 List Spacing, I

This is a short list that ends a subsection, so can be used to address the necessary spacing. We also test two XML elements separated by a space (which should not go missing).

1. One item.
2. *Two ducks*.
3. Three items.
4. Four items.

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.

12.4 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.

1111, 2222		3333
aaaa	bbbb,cccc	
AAAA	BBBB	CCCC

Table 12.2: Table Alignment Example

This exercise (a list item really) has a table first. Default L^AT_EX 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: [12.4.1.c](#). Exercise 1, second part, first refinement: [12.4.1.b.i](#).

12.5 Description Lists

Use `d1` 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 L^AT_EX 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. 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.

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.

12.6 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 absurd 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`.

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 12.3: Colors of the Rainbow

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.

12.7 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.

12.8 References (with lists in Annotations)

- [1] Some book would be listed here.
Here is the annotation and an ordered list as part of that annotation.
- (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.

Red	Green	Yellow
Blue	White	Pink

Table 13.1: Some Colors

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”.

				x	$-$	5
x	$+$	2	$\left $	x^2	$-$	$3x$
				x^2	$+$	$2x$
					$-$	8
					$-$	8
					$-$	10
						2

Table 13.2: Polynomial Long Division

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

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

A sequence of `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.

Attributes	Elements				Values
	tabular	col	row	cell	* = default
top	×	×			none*, minor, medium, major
left	×		×		none*, minor, medium, major
bottom	×		×	×	none*, minor, medium, major
right	×	×		×	none*, minor, medium, major
halign	×	×	×	×	left*, center, right, justify
halign		p			decimal, character
header		p	p	p	yes/no*
footer		p	p	p	yes/no*
valign	×		×		top, middle, bottom
colspan				×	1*, integer
rowspan				p	1*, integer
width		×			percentage
colors	p	p	p	p	

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 ??).

In the documentation for his excellent L^AT_EX package, [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.

1234567890	1234567890	1234567890	1234567890
[First	Second	Third	Fourth
A	B	C	D
1	2	3	4

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.

1234567890	1234567890	1234567890	1234567890
First	Second	Third	Fourth
A	B	C	D
1	2	3	4
1	2	3	4

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.

One

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

1234567890	1234567890	1234567890	1234567890
First	Second	Third	Fourth
A	B	C	D
1	2	3	4
1	2	3	4

Table 13.6: Vertical Rules Example

1111	2222	3333
aaaa	bbbb	cccc
AAAA	BBBB	CCCC

Table 13.7: Progressively Thicker Rules Example

1111, 2222		3333
aaaa	bbbb,cccc	
AAAA	BBBB	CCCC

Table 13.8: Column Span Example

Example 13.9 (Example Environment with Leading Table).

1	2+3	4	5+6+7	8+9				
1	2	3	4	5	6	7+8	9	
1	2	3	4	5	6	7	8	9

Table 13.10: Column Spans, No col Elements, Nine Columns

This example tests several things. In L^AT_EX output, figures, tables, listings and side-by-sides are “floats” whose placement can migrate, but we have tries to suppress this behavior. However, a float that is the first item of an “environment” (like a theorem or an example) can still float to a position *before* its title. If that does not happen here, then our additional defenses are working.

This example also checks that the total number of columns is correctly computed from the first row, which features several `colspan` attributes.

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

One

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

Unit	Stands For	Defininion	Roughly
s	second	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 an extraneous paragraph just to demonstrate the inter-paragraph formatting.	the time it takes you to say the phrase “differential calculus”
min	minute	exactly 60 seconds	how long it takes to microwave a full dinner plate from the refrigerator
h	hour	exactly 3600 seconds; exacly 60 minutes	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.

One Fish		Look at me!
Two Fish		Look at me!
Red Fish	I am the Lorax.	Look at me NOW!
Blue Fish	I speak for the trees.	It is fun to have fun.
		But you have
		to know how.

Table 13.13: DR. Seuss lines

This is a table torture test with many combinations of valign, align, colspan, <p> children, and <line> children.

							Cell too wide	
Lf md	Lef mid par cel	Rt md	Rig mid par cel	Cn md	Cen mid par cel	Js md	Jus mid par cel jus mid par cel	
Colspan=2 lef mid with lines		Colspan=3 rig mid			Lines Between Par	Lines Between No Par	Par in row with lines	
L t	Lef top par cel	R t	Rig top par cel	C t	Cen top par cel	J t	Jus top par cel jus top par cel	
L b	Lef bot par cel	R b	Rig bot par cel	C b	Cen bot par cel	J b	Jus bot par cel jus bot par cel	
Colspan=3 lef bot			Colspan=2 rig bot with lines		Lines Under Par	Lines Under No Par	Par in row with lines	

Table 13.14: Table Torture Test

And now a <sidebyside> with a <table> and a <tabular> to check that width is scaled appropriately. See Section ?? to learn about <sidebyside>s.

A1.S1	<p>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.</p> <p>Should be 50% of 45% except perhaps on small screens.</p>	A1.S2.C1	<p>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.</p> <p>Should be 50% of 55% except perhaps on small screens.</p>
-------	---	----------	---

Figure 13.15: Some text from the US Constitution

Tables are formed in L^AT_EX output with copious use of the `\multicolumn` macro to override more global alignment settings, and to spread the content of one cell across several columns. However, we also use them as part of a strategy to accomodate L^AT_EX's special characters in verbatim text. So the table below, two items per row, is just designed for L^AT_EX testing. But of course, it should still render fine in other formats. The five test cases are from ??, but without 50 alphabetic characters and 8 digits, which should not be problems in this context. The first column's entries are *forced* to be wrapped in a `\multicolumn` by specifying their horizontal alignment. The second column's entries *may* be wrapped in a `\multicolumn` depending on their contents (essentially verbatim characters given by escaped versions).

1	<code>09az%-._~:/?#[!\$&'()*+;,=</code>	<code>09az%-._~:/?#[!\$&'()*+;,=</code>
2	<code>09az%-._~:/?#[!\$&'()*+;,=</code>	<code>09az%-._~:/?#[!\$&'()*+;,=</code>
3	<code>09az%-._~:/?#[!\$&'()*+;,=</code>	<code>09az%-._~:/?#[!\$&'()*+;,=</code>
4	<code>09az%-._~:/?#[!\$&'()*+;,=</code>	<code>09az%-._~:/?#[!\$&'()*+;,=</code>
5	<code>09az%-._~:/?#[!\$&'()*+;,=</code>	<code>09az%-._~:/?#[!\$&'()*+;,=</code>

Table 13.16: Problematic Cells for L^AT_EX

14 Embedded Interactive Elements

When outputting Web page versions, it is possible to embed a variety of dynamic interactive elements. In a L^AT_EX/PDF version, these will necessarily need to be replaced by some static substitute, such as a screenshot. See Section ?? for the specifics of embedding instances of the Sage Cell Server.

14.1 GeoGebra

This first example of a GeoGebra demonstration has just the controls for moving the three vertices on the circumference of the circle. This is courtesy of

Danny Parsons at the African Institute of Mathematical Sciences. This demo requires Java, which could be problematic.

GeoGebra will create screenshots of demonstrations in TikZ/L^AT_EX code. For a static version, we use this as a figure.

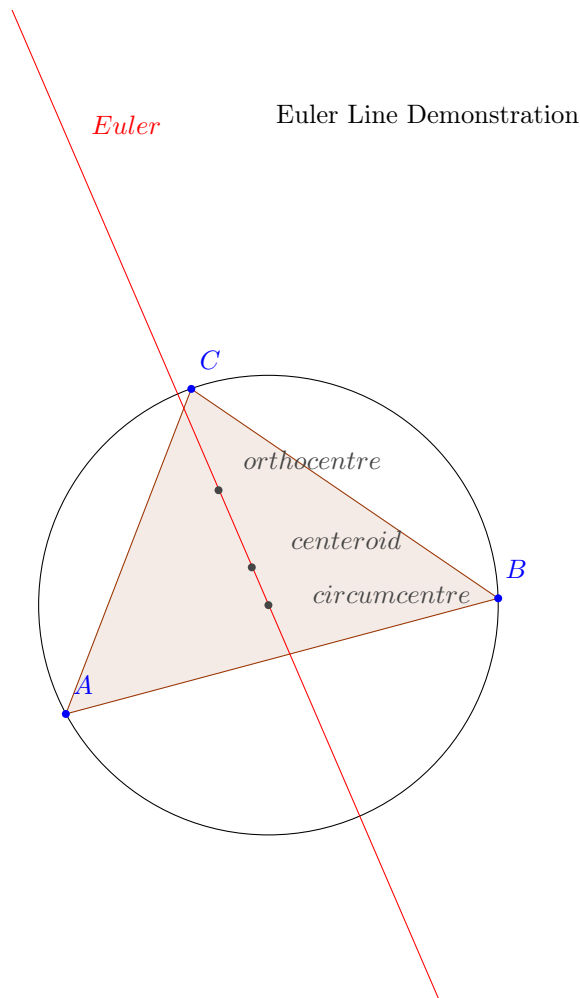


Figure 14.1: GeoGebra demonstration of the Euler Line

With a totally empty “geogebra” element, you will get a blank slate to play around in. This is based on an example of embedding GeoGebra into Sage notebooks by Bruce Cohen. Notice the full suite of menus and tools (in contrast to the previous example).

Again, this example will run with Java. GeoGebra demonstrations can be run via HTML5 without invoking Java, but this seems to only be possible in the Chrome browser. Support will be expanded, especially if requested.

Blank GeoGebra canvas is here in Web version.

14.2 JSXGraph

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 constrained 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](#). Width is 75% and aspect ratio is 4:3.

A JSXGraph interactive demonstration goes here in interactive output.

Figure 14.2: The Archimedian Spiral $r = a + b\theta$, $0 \leq \theta \leq 8\pi$

[JSXGraph](#) is a “cross-browser JavaScript library for interactive geometry, function plotting, charting, and data visualization in the web browser.” Place Javascript inside an input element inside a `jsxgraph` element. You will want an `xml:id` on the `jsxgraph` element, since it is what will be used as the HTML `id` on the `div` that will hold the demonstration, and will typically be used in an early call to the `initBoard()` method.

Here is a more elaborate example, from the [JSXGraph Showcase](#), titled [Infinity](#). Notice that the code here contains a problematic less-than character for a comparison in a loop. We could replace it by the `<` XML escape sequence, but we have instead chosen the expedient of using a `CDATA` construction to wrap the entire hunk of code (see the source for exact syntax). This might be the best solution if the code contains many HTML strings that need to be protected from the XML parser.

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. Width is 60% and aspect ratio is the default, 1:1, i.e. a square.

A JSXGraph interactive demonstration goes here in interactive output.

Figure 14.3: Infinity, from the JSXGraph Showcase

Here are the two new examples, but the first has been given an extreme aspect ratio, and both are smaller overall. They have been included in a `sidebyside` layout element (see Section ??) so they can be placed horizontally across the page. (Note: placement in a `sidebyside` is not yet reflected in the PreTeXt schema.) These are again from the [example wiki](#), the left being Fermat’s Spiral and the right being a demonstration of B-splines.

[jsxgraph]

[jsxgraph]

Now, a piecewise function you can control, with traces of the domain values and range values in two separate JSXGraph boards, laid out using the `sidebyside` layout element.

A JSXGraph interactive demonstration goes here in interactive output.

Figure 14.4: Piecewise Function

[jsxgraph]

[jsxgraph]

Figure 14.5: Domain

Figure 14.6: Range

Be careful about the Javascript code when there are several on the same page, as duplicate variables may affect one another’s performance adversely.

15 Video

First, a gratuitous reference to Exercise [11.3.3](#) about the derivative of a cosine.

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. Support is limited to HTML5-capable browsers. The file format can be MP4, Ogg, or WebM, though this may vary depending upon the browser. The example below is an old promotional video from the University of Puget Sound, in WebM format. The source attribute should *not* include an extension, since the three possibilities above will be searched for preferentially (you need only provide one, but more will increase the chances every browser will find a compatible format).

[video]

Figure 15.1: University of Puget Sound Promotional Video

YouTube videos may be embedded with only knowledge of the “ID”. This 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.

These 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).



Kashmir (Live), Led Zeppelin
www.youtube.com/watch?v=hAzdGU_kpGo

Figure 15.2: Kashmir (Live), Led Zeppelin. O2 Arena, London. December 10, 2007. (8:55)

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



My first day with my drone
www.youtube.com/watch?v=VSHMjW0RFvI

Figure 15.3: First Drone Flight (1:28)

If you are only interested in a piece of the action, you can limit the video with start and end attributes in seconds. You might make those times clear in the caption for readers getting the link out of a PDF. Some videos may not respect these parameters.



My first day with my drone (Splashdown)
www.youtube.com/watch?v=VsHMjWORFvI
 (Start: 54s, End: 72s)

Figure 15.4: First Drone Flight (Splashdown, 0:54 to 1:12)

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 ?? 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.



YouTube:
[5-ptycz7WbY](https://www.youtube.com/watch?v=5-ptycz7WbY)
 (Start: 16s, End: 35s)



YouTube:
[qi7Yh16dA0w](https://www.youtube.com/watch?v=qi7Yh16dA0w)
 (Start: 60s, End: 120s)

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).



YouTube:
[VsHMjWORFvI](https://www.youtube.com/watch?v=VsHMjWORFvI)

[video]

Figure 15.5: Drone Flight

Figure 15.6: UPS Promo

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 is available. (For YouTube, these iamges can be obtained automatically with the mbx script.)



YouTube:
[gFHQ6Y0nxWs](https://www.youtube.com/watch?v=gFHQ6Y0nxWs)



YouTube:
[gFHQ6Y0nxWs](https://www.youtube.com/watch?v=gFHQ6Y0nxWs)
 (Start: 2s, End: 4s)



YouTube:
[gFHQ6Y0nxWs](https://www.youtube.com/watch?v=gFHQ6Y0nxWs)

(a) Embedded Only

(b) Pop-Out Only, w/
Start/End

(c) User Selection

Figure 15.7: 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.



YouTube:
[gFHQ6Y0nxWs](https://www.youtube.com/watch?v=gFHQ6Y0nxWs)



YouTube:
[gFHQ6Y0nxWs](https://www.youtube.com/watch?v=gFHQ6Y0nxWs)
 (Start: 2s, End: 4s)



YouTube:
[gFHQ6Y0nxWs](https://www.youtube.com/watch?v=gFHQ6Y0nxWs)

(a) Embedded Only

(b) Pop-Out Only, w/
Start/End

(c) User Selection

Figure 15.8: YouTube with Generic Preview

Now, the six combinations above with an author-hosted video.

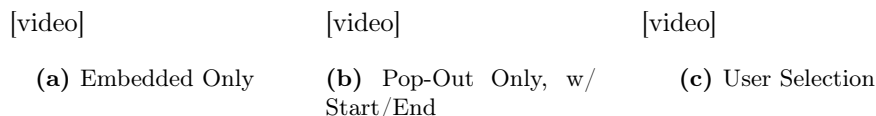


Figure 15.9: Author-Hosted Play Location Options (play-at attribute)

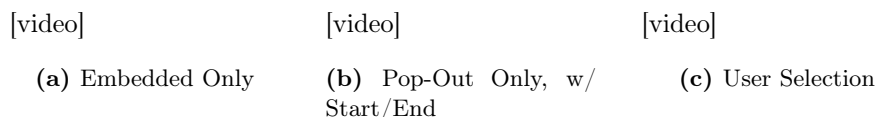


Figure 15.10: Author-Hosted, Generic Preview, Experimental Aspect Ratios

16 Exercises

Exercises in an “exercises” section are numbered automatically. However, what if you have a busted problem and remove it? Then a bunch of problem numbers change and your list of homework for your students changes as well. What a mess. Use auto-numbering while writing and refining. Once stable, go ahead and hard-code problem numbers as you delete/add. Notice the oddly numbered problem below. Once you go down this road, you can’t stop. But instructors will thank you for it.

More precisely, once you hard-code a number for a problem, you will likely need to hard-code every subsequent problem number in that section of exercises. This is because the automatic numbering is unlikely to be what you really want, or what you had before.

1. (An Exercise in a Section) Exercises can appear in a “section” of their own. You need to give the section a title, even if it seems obvious what to call it. Individual exercises may have titles, as you choose. Problem: How should we hide solutions?

Solution. Maybe a global switch should be used to suppress solutions, while a separate processing regime could use them as part of a solutions manual.

42a. (An Exercise with a Hard-Coded Problem Number) Compute the definite integral $\int_2^4 x^2 dx$, not as an approximate value from a Riemann sum, but as an exact value based of the limit by using the Fundamental Theorem.

Solution. An antiderivative of x^2 is $F(x) = x^3/3$, so by the FTC,

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

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.

3. Can you prove Corollary ?? 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.

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: `??`. 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: `??`
- Always named: Corollary `??`

You might also wish to provide a prefix to a cross-reference and have it incorporated into the text of what you would click on in an electronic version. So if you make an `xref` with some content, then that content will prefix the cross-reference within the clickable/pokeable text and be attached with a non-breaking space. This `xref` content totally overrides any prefix that might happen otherwise. So the name of an item (e.g. “corollary”) could be replaced, and if you make a cross-reference with a title as the clickable, then that text can be replaced also. An example:

- A grand result: Major Corollary `??`
- 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 Theorem by providing the content Theorems on the first xref and `text="global"` on the second 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 ?? and ??. (But notice that it is up to you to be certain the types of these targets do not change without you changing the content of the first xref. The “author-tools” mode and careful choices of `xml:id` strings can help avoid this trap.)

One final twist. If you say `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 [16.42a](#) 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 [\[1\]](#)
- Note: just the annotation of previous citation at [1.1](#)
- Examples: Mystery derivative at ??, or a question at ??.
- Definition-like: A mathematical statement with no proof ??.
- A numbered Note: ??
- A link to a proposition element, while this document has globally re-named propositions as “Conundrum”s, so this link should use the new name: Conundrum ??
- Theorems: Fundamental Theorem of Calculus, with proof at ??
- Proof: of second version of FTC at [4.1.1](#)
- Figures: A plot with a derivative at ??.
- A Figure within a side-by-side panel, with its own number: ??
- A Table within a side-by-side panel, with a subnumber: ??
- A Figure, containing a side-by-side with two sub-captioned images: ??
- Display Mathematics: single, first with no name: (??). Then with an autoname: (??).
- Display Mathematics: multi-row, first with no name: (??). Then with an autoname: (??). And two, with a plural form: Equations (??) and (??).
- Exercises (sectional), a range, with plural form provided to override autonaming: [Exercises 11.3.1–11.3.3](#).

- Exercise (inline): with enclosed hint at ??
- A group of two exercises, with introduction, conclusion: [Exercise Group 11.3.2–3](#)
- Solution: An autonamed portion of an exercise: [Solution 16.42a.1](#)
- Parts of a complicated exercise: [Hint 16.3.2](#) [Answer 16.3.1](#)
- An item buried in nested ordered lists (local reference): [Item 2.b.ii.C](#)
- A subsidiary part of an exercise (fully-qualified reference): [12.4.1.b.i](#)
- List item as knows in HTML, including nested lists: [2](#), [Item 2.b.ii](#)
- A titled list: ??
- List item inside a named list, second color in rainbow list: [Item 12.3:2](#)
- 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: [Mathematics](#). Note that you need to include the attribute autoname="number" even if that is obvious from the situation. This requirement may be relaxed in a future refactoring of the cross-reference system.
- 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 ??: [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 ??](#)
- 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 ??](#)”
- An activity with full details following: ??
- An interactive program inside a program listing, to test if the Javascript will execute properly within a knowl: ??

- 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 [??](#)??, the encompassing project: [??](#), and a local reference [c.iii](#).
- A subcaptioned named list: [??](#)
- 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 ??](#)
- 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](#)

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 ??](#).

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).

$x^2 + y^2 = z^2$	Theorem ??
$a^2 + b^2 = c^2$	Section ??

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 ??](#), [Theorem ??](#), [Theorem ??](#)
- Two equations, no spaces, autonamed: [\(??\)](#), [\(??\)](#)

- Two bibliographic items, no autoname: [1, 2]

If you have a long list of items (such as homework exercises, not in an `exercisegroup`, or perhaps several chapters, you can get a cross-reference that prints as a range by using `xref` with two attributes `first` and `last`, which may contain a single `xml:id` each. As with multiple references, `first` will control 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): Exercise ??–4.2.4.1
- A range of equations: (??)–(??)
- A system of equations, given as range from first to last: (??)–(??)
- A range of sections, hand-named to be plural: Sections ??–??
- 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 somewhere on the Internet, then just use the complete address. Here is one from Microsoft: [Sample Excel Spreadsheet](#).

For a link like the previous one, you might want to provide advice appropriate for your audience about using a context menu to download a file, or how to configure helper/viewer applications.

You can also provide a file yourself, but now it is your obligation to distribute the file with your document (HTML, PDF, etc.) and provide a relative link. This creates some complications, such as making sure an electronic PDF has the associated file in the same place relative to the PDF file. Of course, if you make a print PDF, this becomes impossible. Here is a test example anyway, which is highly likely to be broken in a PDF (including at the PreTeXt project site) unless you build this example on your own computer, locally. Here is a template from the Apache OpenOffice project, provided via the Public Documentation License (PDL): [Running Statistics Template](#).

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 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 special characters, even if targeting HTML output, before resorting to a Unicode character. For example, use `<times />` for a small “multiplication sign” in text before resorting to the Unicode character U+00D7.

- 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/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/paste this text 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 L^AT_EX, PDF, print The situation for L^AT_EX is much more complicated, since T_EX 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 L^AT_EX 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 should all render using `pdflatex` or `xelatex` with no special setup (and 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 also replaced reserved L^AT_EX characters by their PreTeXt equivalent empty elements.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
002_		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
003_	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
004_	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
005_	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
006_	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
007_	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

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 `pdflatex` or `xelatex` with no special setup (and HTML).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00A_		ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
00B_	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
00C_	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
00D_	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
00E_	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
00F_	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

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 `xelatex`, however our focus with this sample article for PDF output is the capabilities when processed with `pdflatex`. First, characters from U+0000–U+007F.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
002_		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
003_	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
004_	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
005_	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
006_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
007_	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

Table 18.3: Basic Latin, Monospace

Note that the single and double quotes are upright and dumb, not curly and smart: " ' " ' " '. The zero is distinguished from the capital “oh”: 0 0 0 0 0. And the numeral one is slightly different from the lower-case “ell”: 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. Inline code (the `c` tag) and the `program` tag are implemented in `LATEX` via the `listing` package and these characters require ad-hoc replacements for processing by `pdflatex`. (You can see the replacements in the preamble of the `LATEX` source for this document.) The replacement mechanism provided by the `listing` package will cause the characters below to produce a `LATEX` compilation error if processed by `pdflatex` and in a table cell in certain situations (which we have avoided in the table below). The only workaround in this case is to switch to `xelatex`.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00A_		ı	¢	£	¤	¥	¦	§		©	ª	«	¬		®	
00B_	°	±	²	³		µ	¶	·	¸	¹	º	»	¼	½		¿
00C_	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
00D_	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
00E_	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
00F_	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Table 18.4: Latin-1 Supplement, Monospace

The `pre` tag is implemented in L^AT_EX with the `fancyvrb` package. You can compare results here with the table above, lines here are rows above.

```
ı ¢ £ ¤ ¥ ¦ § © ª « ¬ ®
° ± ² ³ µ ¶ · ¸ ¹ º » ¼ ½ ¿
À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï
Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
à á â ã ä å æ ç è é ê ë ì í î ï
ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ
```

The `console` tag is also implemented with `fancyvrb`, with adjustments for the input lines. It will not look like it, but these are 8 such inputs, with similar results to above, but now bolded.

```
ı ¢ £ ¤ ¥ ¦ § © ª « ¬ ®
° ± ² ³ µ ¶ · ¸ ¹ º » ¼ ½ ¿
À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï
Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
à á â ã ä å æ ç è é ê ë ì í î ï
ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ
```

Again, more examples and more thorough explanations can be found in the sample: `examples/fonts/fonts-and-characters.xml`. Be aware that the nature of the more advanced sample is that it will likely produce many errors when processed with `pdflatex`. Adding `-interaction batchmode` or `-interaction nonstopmode` to the `pdflatex` command-line will sometimes be less painless than acknowledging each error. The more advanced sample will perform well when processed with `xelatex`.

19 Pre-Formatted Text

In Sage, if you wanted to build a matrix, then you would use the `matrix()` constructor. Here is the matrix of second partials of $f(x, y) = x^3 + 8x^2y^3 + y^4$, as you would enter it in Sage. Notice that `SR` is the ring of symbolic expressions, Symbolic Ring.

```
var('x', 'y')
J = matrix(SR, [
    [6*x + 16*y^3, 48*x*y^2],
    [48*x*y^2, 48*x^2*y + 12*y^2]
])
```

That accomplished, Sage will easily and naturally provide a L^AT_EX representation of the matrix with the command `latex(J)`.

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

```
48 \, x y^{2} & 48 \, x^{2} y + 12 \, y^{2} \\
\end{array}\right)
```

The realization of preformatted text should be robust enough that it can be cut from documents and pasted without any substitutions of “fancier” Unicode characters for generic ASCII characters. Try the “minus” sign on the 48 above to see if it does not become a dash, or the single quotes on the Sage variables.

For Sage input code, the first non-whitespace character sets the left margin, since legitimate Python code has no subsequent lines outdented. For pre-formatted code, the line with the *least* whitespace leading the line will determine the left margin. If preserving indentation is important, do not mix spaces and tabs. For syntax highlighting of text representing computer programs, or parts of them, see Section ???. Examine the source of the following example to help understand this paragraph.

```
A normal line
      An indented line
An outdented line
```

The `<c>` element, for inline code snippets (or anything else in a monospace font) uses the question-mark character to tell L^AT_EX where the text begins and ends. This will be a problem if the text has a question mark in it! So there is the attribute `latexsep` that allows you to specify another character that does not appear in your text. For example, XML directives use question-marks now and then, so writing about them in PreTeXt requires specifying a different separation character, as in: `<?xml version="1.0" encoding="UTF-8" ?>`. The `<pre>` element does not suffer from this quirk.

Snippets should also be robust for cut/paste operations. For example, you should not get “curly” “smart” quote marks in verbatim text: this should have “dumb” quote marks. Here are a few characters that should migrate through L^AT_EX to a PDF unmolested: `'"-----'`

If you write a very long snippet of inline code it can impinge on the right margin, especially if it were to begin close to that margin. For output in L^AT_EX we allow line-breaking, but we do not get hyphenation and the font is fixed-width. So not always perfect. Consider a display (program listing as above with the `pre` element) or maybe reorganizing a sentence. And it looks like a space after that left parenthesis just got ignored in the PDF created from the latex output.

An intermediate type of verbatim text can be accomplished with the `<cd>` tag, short for “code display.” It allows for larger chunks of verbatim text to show up in the middle of a paragraph, but with some vertical space above and below. It can be

```
authored as a single line
```

```
or if you wish to have multiple lines
```

```
there is the cline tag
meant to model the line tag
and short for "code line"
```

```
and you may even
```

```
use a single cline
```

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

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

If you write a very long snippet of inline code it can impinge on the right margin, especially if it were to begin close to that margin. For output in LaTeX we allow line-breaking, but we do not get hyphenation and the font is fixed-width. So not always perfect. Consider a display (program listing as above with the `pre` element) or maybe reorganizing a sentence.

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 ??.

In the meantime, program listings, especially with syntax highlighting, is useful all by itself. The “R” language might not be a bad stand-in for pseudo-code, as it supports assignment with a left arrow and has fairly generic procedural syntax for control structures and data structures. Or maybe Pascal would be a good choice? Here is an example of R. Note in the source that the entire block of code is wrapped in a CDATA section due to the four left angle brackets. We do not recommend this technique for isolated problem characters, but it is a life-saver for situations like the XSLT code just following.

```
n_loops <- 10
x.means <- numeric(n_loops) # create a vector of zeros for
  results
for (i in 1:n_loops){
  x <- as.integer(runif(100, 1, 7)) # 1 to 6, uniformly
  x.means[i] <- mean(x)
}
x.means
```

And some self-referential XSL:

```
<xsl:template match="biblio" mode="number">
  <xsl:apply-templates select="."
    mode="structural-number" />
  <xsl:text>.</xsl:text>
  <xsl:number from="references"
    level="any" count="biblio" />
</xsl:template>
```

You can write made-up pseudo-code and get reasonable syntax highlighting, 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 in the source the use of escaped characters for the four angle brackets.

```

input m, n and A
r := 0
for j := 1 to n
  i := r+1
  while i <= m and A[i,j] == 0
    i := i+1
  if i < m+1
    r := r+1
    swap rows i and r of A (row op 1)
    scale A[r,j] to a leading 1 (row op 2)
    for k := 1 to m, k <> r
      make A[k,j] zero (row op 3, employing row r)
output r and A

```

Look in the `mathbook-common.xml` file to see the strings to use to identify languages. Always all-lowercase, no symbols, no punctuation.

Note that the above examples all have slightly different widths (theser are very evident in print with the frames). As 2-D atomic objects, to place them in the narrative requires the layout features of a `sidebyside` element. Then width and/or margin attributes will influence the width of the panel.

A program may also be nested inside a listing, which behaves similar to a figure. You can provide a 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 ?? . 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.

```

/* 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 (c. 276 BC–c. 195/194 BC) by Nicomachus (c. 60–c. 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 $2p, 3p, \dots$
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. \square

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 L^AT_EX macros such as `\texttt` for a monospace text font, and `\;` for spacing/grouping the bits of the binary number.

$$6C2A_{16} = 0110\ 1100\ 0010\ 1010_2 \quad (20.1)$$

If you use these constructions repeatedly, then some L^AT_EX 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. \square

A specialized version of a program listing is an interactive command/response session at a command-line, where differing fonts are used to differentiate the system prompt, the user’s commands, and the system’s reaction. A console session may be used by itself inside a `sidebyside`, or it can be wrapped in a listing to get a number and a caption. As elsewhere, you will need to escape ampersands and angle brackets (such as if you have a command using redirection), using `&`, `<`, and `>` in your source.

```
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 `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. (But be careful, a backslash, “\,” will require you to define the `latex.console.macro-char` xsltproc stringparam to something else, as explained below.)

```
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&paste that the prompts are not included.

There is one subtlety with a console session rendered as \LaTeX output. The user input is made bold by a \LaTeX macro, which means that your code cannot contain the special \LaTeX characters “\”, “{”, and “}”, which are used to begin a macro, begin a group, and end a group (respectively). You will get an error message if this condition exists, and there are parameters `latex.console.macro-char`, `latex.console.begin-char`, and `latex.console.end-char` that will allow you to specify alternatives (which need to be characters that do not appear in *any* of your console sessions, document-wide). The characters `&` `%` `$` `#` `_` `{` `}` `~` `^` `\` have special meaning in \LaTeX but should be available for duty as these alternative characters (though not all have been tested). The backslash used in pathnames for Windows is a highly likely case where this needs adjustment.

There is no good way to provide an example of this situation, without making a document with an error in it, out-of-the-box. So experiment by using `--stringparam` on the xsltproc command-line with alternative characters that will behave with the example above, and with characters that will cause the example above to raise errors. In practice, you may want to specify alternative characters in a thin XSL extension file specific to your project.

We conclude with a longer example, 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    locals,28          @ space for the structs

@ Constant program data
    .section .rodata
    .align  2
displayX:
    .asciz      "x fields:\n"
displayY:
    .asciz      "y fields:\n"
dispAChar:
```

```

        .asciz      "          aChar = "
dispAnInt:
        .asciz      "          anInt = "
dispOtherChar:
        .asciz      "      anotherChar = "

@ The program
        .text
        .align 2
        .global main
        .type  main, %function
main:
        stmfd      sp!, {r4, fp, lr}    @ save caller's info
        add        fp, sp, #8           @ our frame pointer
        sub        sp, sp, #locals @ for the structs

@ fill the x struct
        add        r0, fp, #x           @ address of x struct
        mov        r1, #'1
        mov        r2, #456
        mov        r3, #'2
        bl         loadStruct

@ fill the y struct
        add        r0, fp, #y           @ address of y struct
        mov        r1, #'a
        mov        r2, #123
        mov        r3, #'b
        bl         loadStruct

@ display x struct
        add        r4, fp, #x           @ address of x struct
        ldr        r0, displayXaddr
        bl         writeStr
        ldr        r0, dispACharAddr @ display aChar
        bl         writeStr
        ldrb       r0, [r4, #aChar]
        bl         putChar
        bl         newLine
        ldr        r0, dispAnIntAddr @ display anInt
        bl         writeStr
        ldr        r0, [r4, #anInt]
        bl         putDecInt
        bl         newLine
        ldr        r0, dispOtherCharAddr @ display
        anotherChar
        bl         writeStr
        ldrb       r0, [r4, #anotherChar]
        bl         putChar
        bl         newLine

@ display y struct
        add        r4, fp, #y           @ address of y struct
        ldr        r0, displayXaddr
        bl         writeStr
        ldr        r0, dispACharAddr @ display aChar
        bl         writeStr
        ldrb       r0, [r4, #aChar]
        bl         putChar

```

```

        bl        newLine
        ldr       r0, dispAnIntAddr @ display anInt
        bl        writeStr
        ldr       r0, [r4, #anInt]
        bl        putDecInt
        bl        newLine
        ldr       r0, dispOtherCharAddr @ display
        anotherChar
        bl        writeStr
        ldrb      r0, [r4, #anotherChar]
        bl        putChar
        bl        newLine

        mov       r0, #0           @ return 0;
        sub       sp, fp, #8       @ restore sp
        ldmfd     sp!, {r4, fp, pc} @ restore and return

        .align    2
@ addresses of messages
displayXAddr:
        .word     displayX
displayYAddr:
        .word     displayY
dispACharAddr:
        .word     dispAChar
dispAnIntAddr:
        .word     dispAnInt
dispOtherCharAddr:
        .word     dispOtherChar

```

Listing 20.6: A longer program listing

In HTML output, a program can be interactive. This is an example program provided by [Python Tutor](#).

```

# 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 20.7: 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 achive 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 $9.8 \frac{\text{m}}{\text{s}^2}$. Force is measured in $\frac{\text{kg m}}{\text{s}^2}$, also known as one N. A quantity with rather ridiculous units is $23 \frac{\mu\text{ha}^{23}}{\text{C s}^2}$. One Hz is the same as $\frac{1}{\text{s}}$. You can have a unitless quantity, like 42, which may help with consistency between such numbers and units in the \LaTeX output. Some non-SI units are available, such as the absurd $\frac{^\circ\text{F ft lb}}{\text{gal}}$. The \LaTeX command `\pi` is recognized within `mag` in conversions to HTML, which is consistent with the behavior with a conversion to \LaTeX , for example there are 2π rad in a full circle.

For a full list of the allowed units and prefixes, see `mathbook-units.xml`. If you have a need for more units, they need to be added to `mathbook-units.xml` 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`, `table`, `listing`, `paragraphs`, `image`, `tabular`, `p`, `ol`, `ul`, `dl`, `pre`, `poem`, and more. The individual components of a `<sidebyside>` are generically called **panels**.

As a layout device, the `<sidebyside>` does not allow a `<caption>`, is never numbered, and therefore cannot be cross-referenced. You may cross-reference whatever element holds the `<sidebyside>`, and many of the panels may be cross-referenced individually.

As a first example, we have two single paragraphs, laid out with different widths, and slight margins on each side. The widths have been chosen experimentally to get roughly identical heights for the two paragraphs of varying length.

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

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

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 ?? and Figure ??

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 ?. You can also cross-reference the subfigures individually, for example: Figure ?.

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.

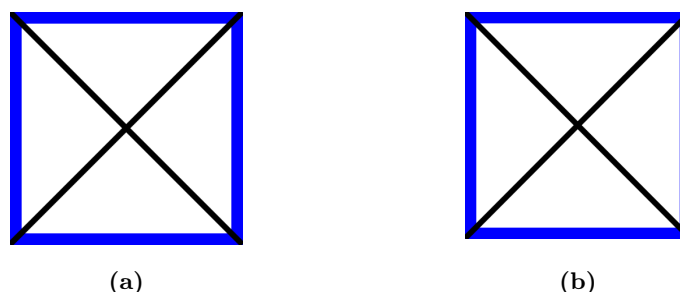


Figure 22.1: Side-by-Side, with figures as children, automatic margin

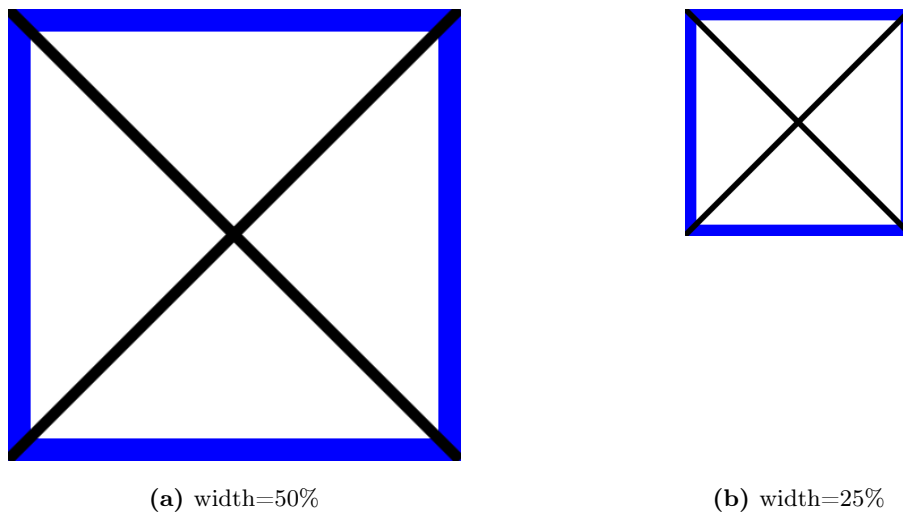


Figure 22.2: Side-by-Side, with figures as children, margin set to zero

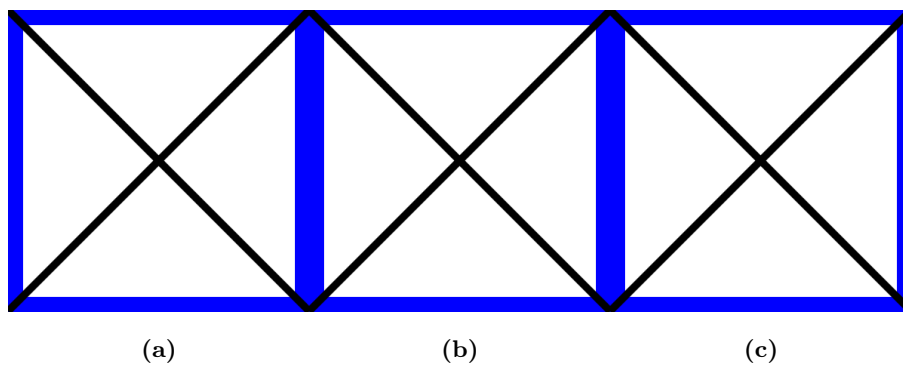


Figure 22.3: Widths calculated automatically, all defaults

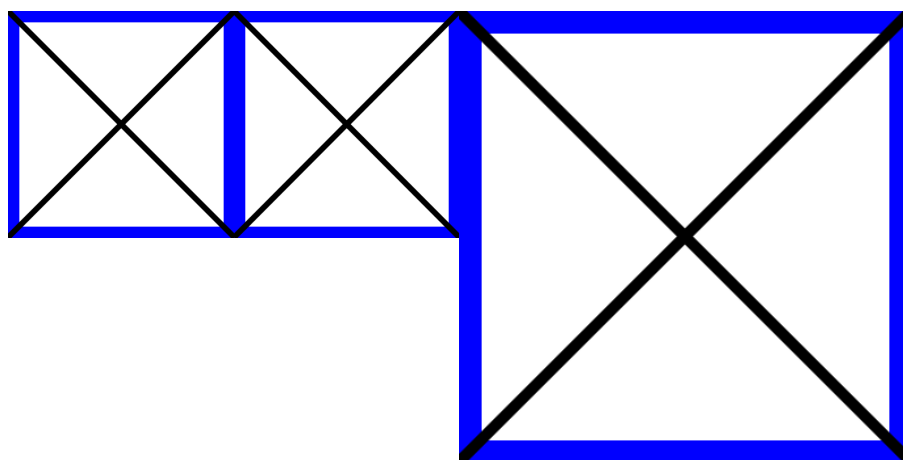


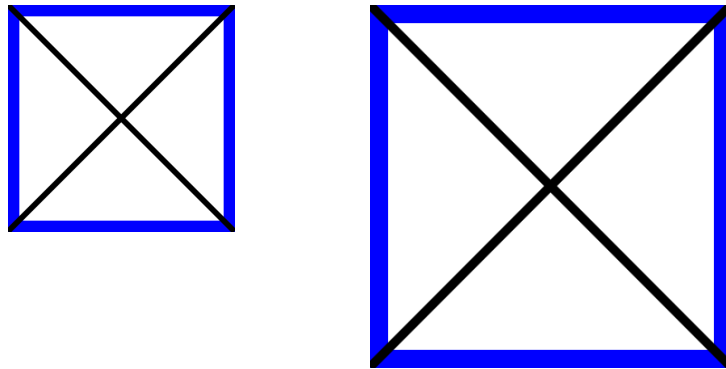
Figure 22.4: Interior figure

Figure 22.5: Regular numbering

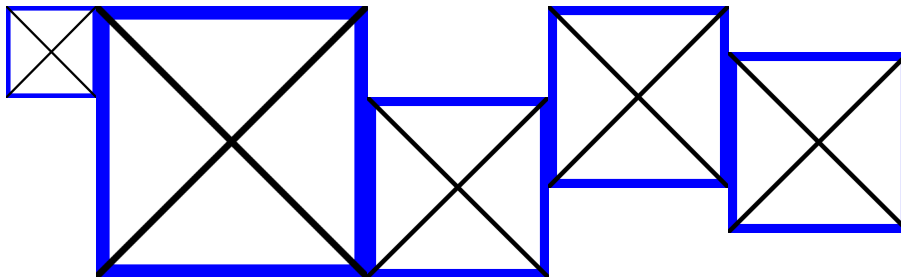
Figure 22.6: Regular numbering

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 ??

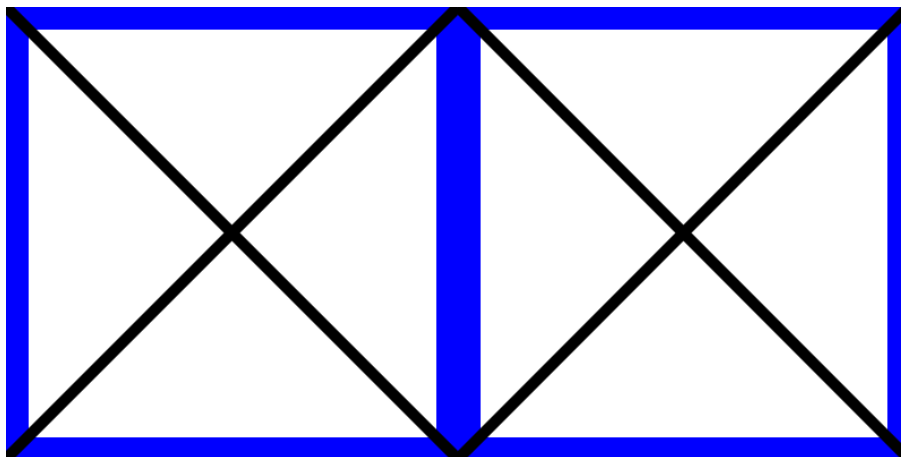


Figure 22.7: Two equally-spaced (identical) images

22.3 Common Side-By-Side Constructions

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

Outer Figure	Inner Figure	Effect
Absent	Absent	Layout only, no numbers nor captions
Absent	Present	Numbers and captions on figure(s)
Present	Absent	Number and overall caption
Present	Present	Number and overall caption, sub-numbers and captions on figure(s)

Table 22.8: `sidebyside` and figure interactions

22.4 A Single Captionless Image

Sometimes you may want to include a single image or table in an example or exercise without a caption. You can also achieve this using `sidebyside`. The first has width 10%, and the second has width 10% and margins 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`.

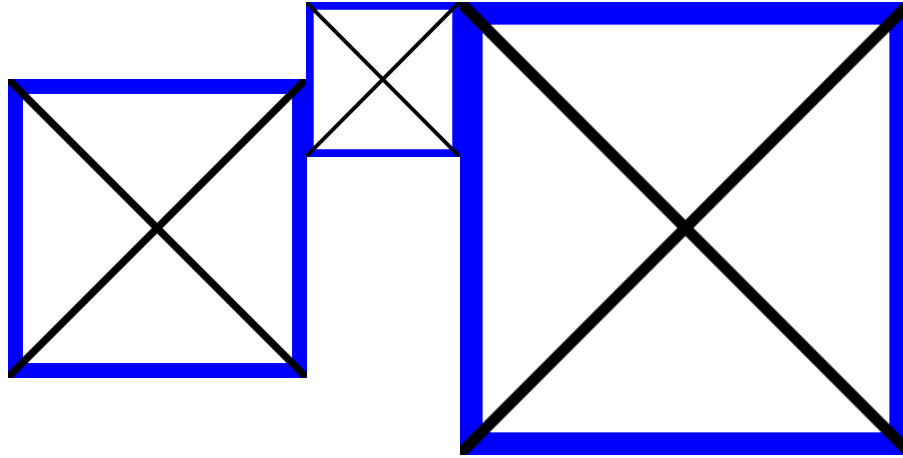
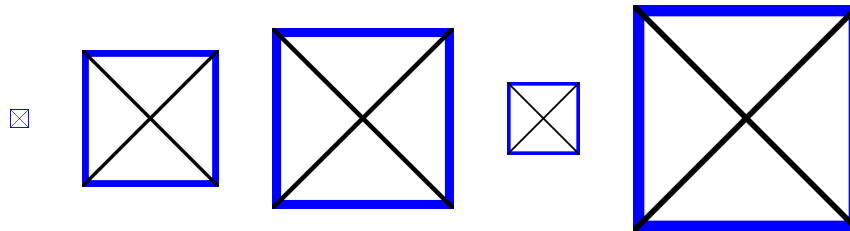


Figure 22.9: Middle

Figure 22.10:
Top

Figure 22.11: Middle

The singular version of the attribute, `valign`, can provide the same alignment to each panel, here we use five different widths, but all with vertical alignment of `middle`.



22.6 Text Next to Text and Images

Text can be put next to other blocks of text using either the `paragraphs` element, which can contain multiple paragraphs using the `p` element.

If only one paragraph is required, simply use the `p` element on its own. In addition to captions, elements which allow titles will have those displayed above the panel. So `paragraphs` can carry a title, but a single paragraph, `p`, will not.

Note: The `paragraphs` element, used as a container for several `p` in a panel, and without a title, will be replaced by some sort of stacking mechanism. A single `p` may be used now as the contents of a panel, without a title.

Sample Title

A Really Long Title on a Fairly Skinny Paragraphs Element

here is some
text here is
some text here
is some text
here is some
text here is
some text here
is some text
here is some
text here is
some text here
is some text
here is some
text here is
some text here
is some text
here is some
text here is
some text here
is some text
here is some
text here is
some text here
is some text

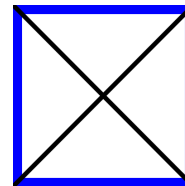
here is some
text here is
some text here
is some text
here is some
text here
here is some
text here is
some text here
is some text
here is some
text here
here is some
text here is
some text here
is some text
here is some
text here

here is some
text here is
some text here
is some text
here is some
text here

here is some
text here is
some text here
is some text
here is some
text here

Similarly, text can be put next to images.

here is some text here is some text here
is some text here is some text here is
some text here is some text here is some
text here is some text here is some text
here is some text here is some text here
is some text here is some text here is
some text here is some text here is some
text here is some text here is some text
here is some text here is some text here
is some text cross reference: Figure ??
and math: x^2



You can place text next to numbered figures, as shown below in Figure ??.

here is some text here is some text here
 is some text here is some text here is
 some text here is some text here is some
 text here is some text here is some text
 here is some text here is some text here
 is some text here is some text here is
 some text here is some text here is some
 text here is some text here is some text
 here is some text here is some text here
 is some text; cross reference: Figure ??
 and math: x^2

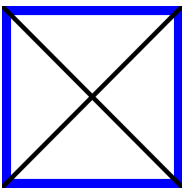
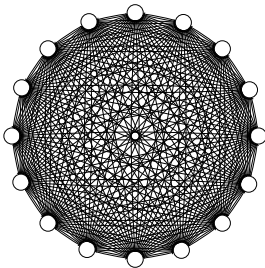


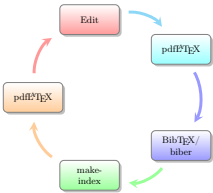
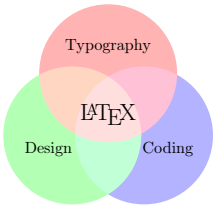
Figure 22.12:
 Text next to a
 figure

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 ?? . 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 L^AT_EX 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](#).



Images by Stefan Kottwitz

- [Venn Diagram](#)
- [Work Flow](#)

Figure 22.13:
 T_EX Work Flow

22.8 Tables Side-By-Side

Tables can also be put side-by-side, as demonstrated below in Figure ??; naturally, subtables can be referenced as in Table ??.

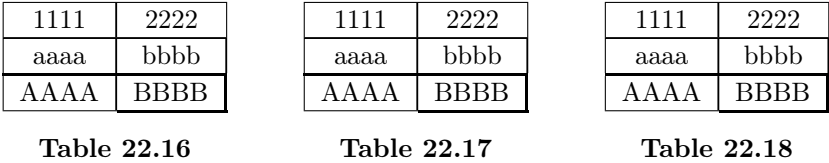


Figure 22.14: Side-by-Side, with tables as children



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 ??-??.



22.9 Tables Next to Figures

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

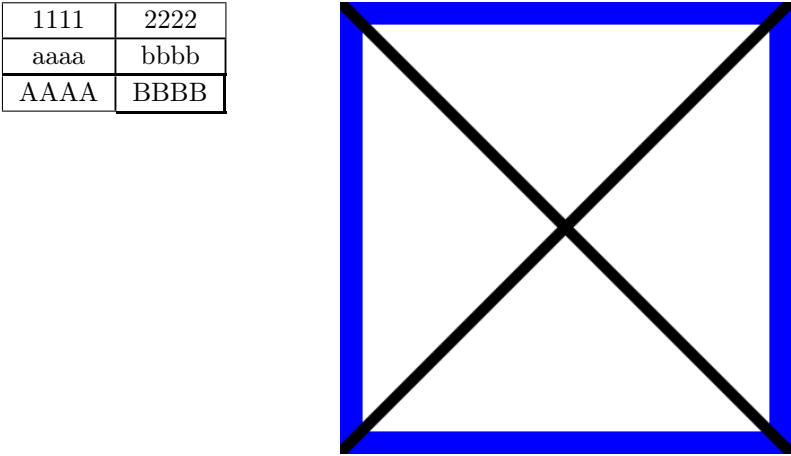
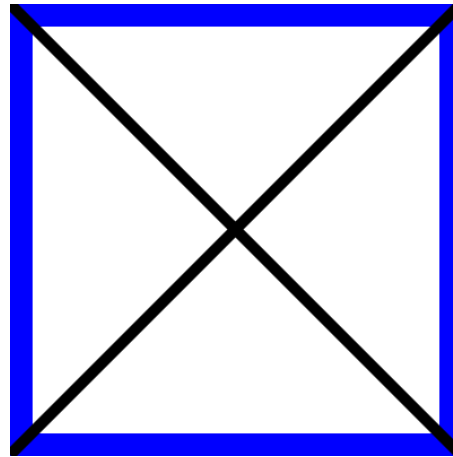


Table 22.19: Table next to a Figure Figure 22.20: Figure next to a Table

1111	2222
aaaa	bbbb
AAAA	BBBB



(a) Table next to a Figure

(b) Figure next to a Table

Figure 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 <paragraphs> element.

1111	2222
aaaa	bbbb
AAAA	BBBB

here is some
text here is
some text here
is some text here is some
here is some text here is
text here some text here
here is some is some text
text here is here is some
some text here text here
is some text here is some
here is some text here is
text here some text here
here is some is some text
text here is here is some
some text here text here
is some text
here is some
text here

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.

1111	2222	1111	2222	1111	2222	1111	2222
aaaa	bbbb	aaaa	bbbb	aaaa	bbbb	aaaa	bbbb
AAAA	BBBB	AAAA	BBBB	AAAA	BBBB	AAAA	BBBB
CCCC	DDDD						

22.12 Lists in Side-by-Sides

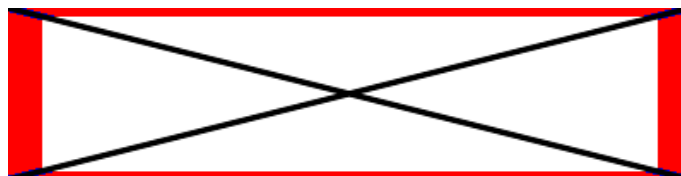
A “regular” list normally belongs in a p but it can be placed unadorned into a panel of a side-by-side, as demonstrated below in Subsection ???. You can also put “named” lists into a panel, and then the title, introduction, conclusion, and caption will behave as expected, along with a number that might be used in a cross-reference (??), or perhaps we might cross-reference by title, A List of Colors.

		A List of Colors
Dr. Seuss again.		
<ul style="list-style-type: none"> • One fish • Two fish • Red fish • Blue fish 	<ol style="list-style-type: none"> 1. Blue <ol style="list-style-type: none"> (a) Light (b) Navy (c) Royal 2. Red <ol style="list-style-type: none"> (a) Maroon (b) Pink (c) Shocking 	
This ends our example.		
(a) Sea Life		(b) Color Shades

Figure 22.23: Two named lists

We also need to test a `sidebyside` in a list. The widths are now relative to the space given over to an indented item. Here we nest and nest and nest and nest to get a big, obvious indentation, and then include an image at 100% width and no margin. In your mind’s eye, or with a ruler, check that the image spans all the way over to the right margin.

1. This is
 - (a) a very
 - i. wide
 - A. rectangle



22.13 Other Panels

Other elements may be placed within a `sidebyside` element. Pure lists first.

- | | |
|--|---|
| 1. Footnotes: Fermat allusion at 2.1. | • Footnotes: Fermat allusion at 2.1. |
| 2. Examples: Mystery derivative at ??. | • Examples: Mystery derivative at ??. |
| 3. Definition-like: A mathematical statement with no proof ??. | • Definition-like: A mathematical statement with no proof ??. |
| 4. Figures: An early plot, Figure ??. | • Figures: An early plot, Figure ??. |

You can place *aligned* equations in paragraphs within a *sidebyside* element.

	here is some text, and here is an equation that contains alignment.
here is some text, and here is an equation that contains alignment.	$f(x) = x^2 + 3x + 2$ $= (x + 2)(x + 1)$
$f(x) = x^2 + 3x + 2$ $= (x + 2)(x + 1)$	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.

program HelloWorld;	#include
begin	
WriteLn('Hello, world!');	int main()
end.	{
	std::cout << "Hello, world!";
	return 0;
	}

Figure 22.24: “Hello, World!” in Pascal and C++

```
graph1.txt
9
6 2
1 5
1 7
6 8
9 1
4 3
5 7
1 3
5 9
7 9
```

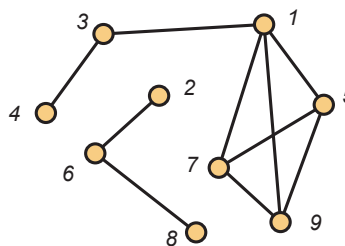


Figure 22.25: A graph defined by data (from Keller and Trotter’s *Applied Combinatorics*)

22.14 Poems as Side-By-Side Panels

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

Commentary

You might have several things to say about a poem and you could use a sequence of paragraphs immediately adjacent.
This is a second paragraph of commentary.

Poems are not horizontally-rigid, but they are not perfectly horizontally-flexible either. The left copy of this next poem is in a panel roughly 2/3 the width of the page and fits there. The right copy has the first five lines and is in space about half the previous width, and you can see the lines being wrapped with obvious indentation. So you *can* constrain the width of a poem if you do not mind the additional indentation. (Recognize that this example is a bit extreme.)

Sonnet to Liberty

Not that I love thy children, whose dull eyes
See nothing save their own unlovely woe,
Whose minds know nothing, nothing care to know,
But that the roar of thy Democracies,
Thy reigns of Terror, thy great Anarchies,
Mirror my wildest passions like the sea,
And give my rage a brother! Liberty!
For this sake only do thy dissonant cries
Delight my discreet soul, else might all kings
By bloody knout or treacherous cannonades
Rob nations of their rights inviolate
And I remain unmoved-and yet, and yet,
These Christs that die upon the barricades,
God knows it I am with them, in some things.

Oscar Wilde

Sonnet to Liberty

Not that I love thy
children,
whose dull
eyes
See nothing save their
own unlovely
woe,
Whose minds know
nothing,
nothing care
to know,
But that the roar of
thy
Democracies,
Thy reigns of Terror,
thy great
Anarchies,

...

Oscar Wilde

22.15 Side-By-Side Groups

A “side-by-side group,” `<sbsgroup>`, is still in development. (Notably, subcaptions do not behave as expected.) It is a sequence of `sidebyside`, which may conceivably use the same margins, widths and vertical alignments for each horizontal run of panels. Attributes on the `sbsgroup` are global to the group’s enclosed `sidebyside`, and will be used by each contained `sidebyside`. If attributes are present on an individual `sidebyside`, they override the global values. The next two examples demonstrate some of this behavior, in a limited way.

One.	Two.	Three.
Four.	Five.	Six.

Figure 22.26: 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.

Stanza One

Some commentary

Stanza Two

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

Stanza Three

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

Stanza Four

Some commentary

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 `sbsgroup` is to layout a grid of items, and by placing the layout parameters on the `sbsgroup` element, the items can line up across sidebyside and subcaptioning can run across the whole group. So, for example, if you have images to compare by placing in a grid, then making them all the same size, or of the same aspect ratio, can help with the overall consistency.

This example has three sidebyside, each with four figure containing an identical image. Since the images are identical and the width is set to 20% they should all line up nicely with little effort. Since the default for margins is automatic, the remaining 20% of the overall width will be used for three inter-panel spaces of 5% and two margins of 2.5% each. Note the numbering of these as independent figures. We have left the captions empty for reasons of space, but you could add more information. Note that in print, a page break is allowed between any two of the sidebyside and cannot be suppressed.

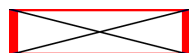


Figure 22.27

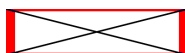


Figure 22.28

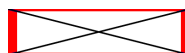


Figure 22.29

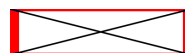


Figure 22.30

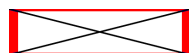


Figure 22.31

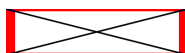


Figure 22.32

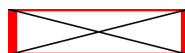


Figure 22.33

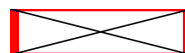


Figure 22.34



Figure 22.35



Figure 22.36



Figure 22.37



Figure 22.38

We recycle the prior `sbsgroup` but now put it in its own overall figure. That will allow a caption for the whole group, and will cause the twelve figures to be subcaptioned. Except the subcaptioning is not implemented. Soon.

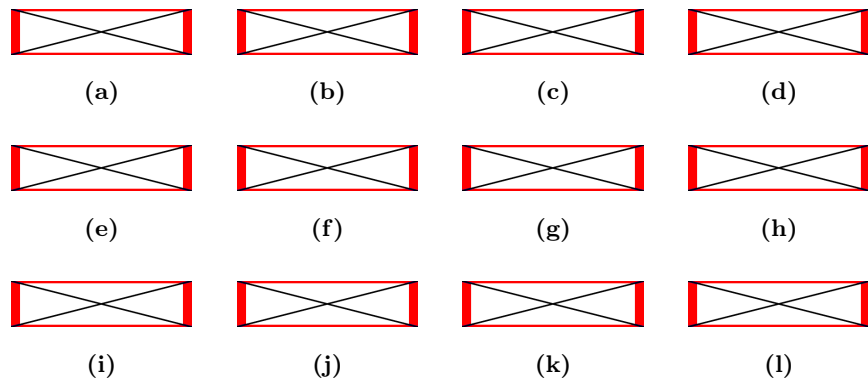
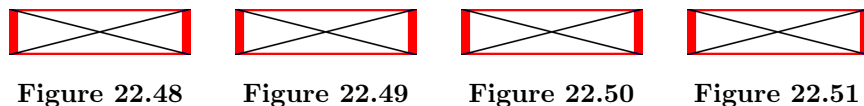
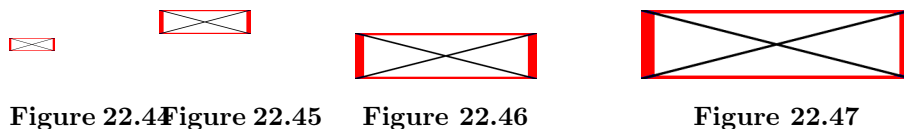
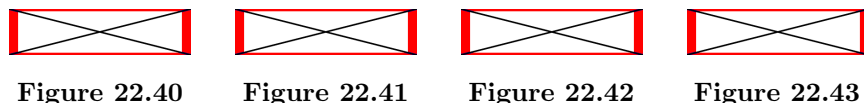


Figure 22.39: Twelve images, arranged in a grid

One more test. We override the spacing and vertical alignments of the middle sidebyside. Note that it is easy to make a panel so skinny that even the smallest possible caption does not fit in the width.



22.16 Testing Styling of Related Elements

This subsection has non-side-by-side structures, to aid with the effects of styling decisions across the range of possibilities. First a figure with a caption holding a scaled image and a cross-reference for knowl testing: Figure ??.

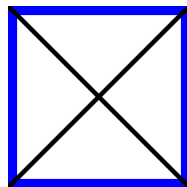


Figure 22.52: A traditional figure

23 Side-by-Side Gallery

This subsection attempts to survey all the possible items that can be placed into a sidebyside element, in various combinations. While intended to be

exhaustive across contents, it does not test all possibilities, and is not meant to be instructive (see Section ?? 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.

Single p

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.

Paragraphs

Lorem Ipsum

Aliquam dui nisi, pharetra id enim vel, imperdiet laoreet risus. Nunc convallis elit eu erat imperdiet tincidunt. Sed eget augue et nunc mollis tempor. Suspendisse luctus elit non lorem scelerisque, nec lacinia lectus dictum.

Vivamus ut orci nisl. Donec eleifend ultricies tortor, a pellentesque neque dignissim in. Praesent maximus, augue eu pretium auctor, dolor quam feugiat augue, ut vulputate nunc eros vitae massa. Phasellus quis ante quis est venenatis dapibus eget luctus ipsum.

Figure 23.1: Single p (left), paragraphs (right)

ol with simple items

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

ul with items with paragraphs

- 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

<pre>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</pre>	<pre>pi@rpi ~\$ gcc -o intAndFloat intAndFloat.c pi@rpi ~\$./intAndFloat 19088743 (integer) and 19088.742188 (float) pi@rpi ~\$</pre>
--	--

Figure 23.3: A program and a console

Note that these two chunks of verbatim text will very likely exceed the right side of a too-skinny panel. We have 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

Organism	Classification
Trout	Fish
Monkey	Mammal
Crow	Bird
Crimini	Fungus
Bee	Insect

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.

<pre>Vestibulum sit amet est non lacus accumsan iaculis aliquam nec leo. Maecenas placemat consequat quam, a lobortis odio convallis vitae.</pre>	<pre>Vestibulum sit amet est non lacus accumsan iaculis aliquam nec leo. Maecenas placemat consequat quam, a lobortis odio convallis vitae.</pre>
---	---

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.

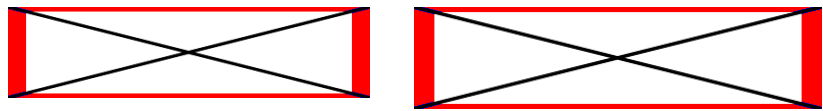


Figure 23.6: An identical image, twice

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).

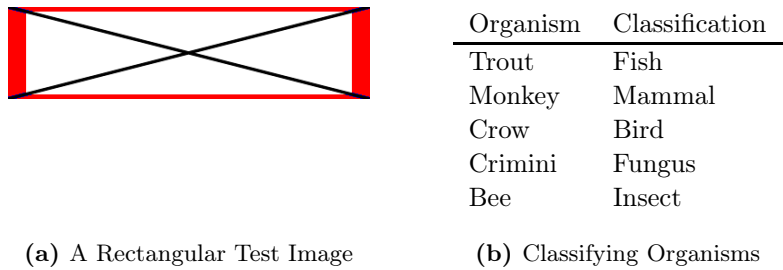


Figure 23.7: A figure and a table

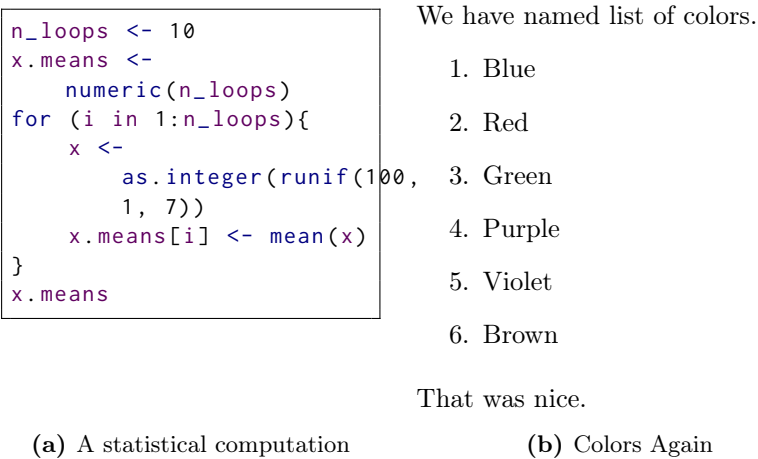


Figure 23.8: A listing and a list

Now we have some more interactive elements

[video]



YouTube:UYPoMjR6-Ao

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 is displayed in print. We could relax our common spacing to make it a bit better. For other examples see Section ??.

Sometime Around Midnight



Sometime Around Midnight

www.youtube.com/watch?v=UYPoMjR6-Ao

(Start: 48s, End: 100s)

(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 ??).

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:
Theirs not to make reply,
Theirs not to reason why,
Theirs but to do and die:

Into the valley of Death
Rode the six hundred.

Alfred Lord Tennyson

Ken Levasseur, who teaches at UMass-Lowell, has limericks in his Applied Discrete Structures textbook. When he reported that they were unable to be the target of a cross-reference, Karl-Dieter Crisman penned the following limerick.

CS students studying in Lowell
Required their books to have soul.
Along came their teacher
Who asked for this feature:
A poem that lives in a knowl.

Karl-Dieter Crisman

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). *No statement.*

Theorem 25.2 (Second Theorem). *No statement.*

25.2 Two

Further subdivided.

25.2.1 Uno

A document leaf.

Theorem 25.3 (First Theorem). *No statement.*

Theorem 25.4 (Second Theorem). *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.*

Theorem 25.14 (Second Theorem). *No statement.*

Conclusion now. We include two theorems as numbered items in the conclusion to test their numbers, which are sometimes totally illogical and are inconsistent across output formats. To see the effect, use `--stringparam numbering.theorems.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 `??`. See a slightly different example in Appendix `??`.

Theorem 25.1	First Theorem
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 Title with] a Right Bracket

L^AT_EX has trouble with brackets that end up inside optional arguments, so this is only a check on the defense against that.

26 Customizations

26.1 Renaming Document Parts

“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.

Conundrum 26.1. *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.

A Notation

This is some notation introduced in the article.

Symbol	Description	Page
$\int_a^b f(x) dx$	definite integral of $f(x)$??
$\int f(x) dx$	indefinite integral of $f(x)$??
ρ	this symbol could be used for lots of things, but we are just trying to make a super-long description to get it to wrap within the column where it belongs, which is sometimes set to a fixed width to accomodate really complicated explanations	??
∇	gradient operator	??

B Solutions to Selected Exercises

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.

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.

11.4 More Exercises

6. $3 + 4 + 5$

Addition is associative.

12

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

16 Exercises

1. 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?

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. Compute the definite integral $\int_2^4 x^2 dx$, not as an approximate value from a Riemann sum, but as an exact value based of the limit by using the Fundamental Theorem.

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.

3. Can you prove Corollary ?? 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?

Consider the definite integral as an area function and employ the Mean Value Theorem.

Think harder!

- (a) It follows easily.
- (b) Yes.

We could prove either result first, then obtain the other as an easy consequence.

C List of Results

We had an automatic list of theorems for just one section, back in Subsection ???. 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

(Continued on next page)

Subsection 25.2 Two

Theorem 25.3	First Theorem
Theorem 25.4	Second Theorem
Theorem 25.5	First Theorem
Theorem 25.6	Second Theorem

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

D 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 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

- [1] Tom Judson, *Abstract Algebra: Theory and Applications*.
Another online, open-source offering.
- [2] David C. Lay, *Subspaces and Echelon Forms*. The College Mathematics Journal, January 1993, **24** no. 1, 57–62.

Index

L^AT_EX image, 28

verbatim text, use sortby, 2

A, *see also* H, *see* F, *see* G, 2

B, *see also* C

C, *see* D

activity, 7

Archimedian Spiral, 52

assemblage, 12

asymptote graphics language, 30

attributes

doctest, 2

ncbi attribute, 27

autoname, 56

biological names, 27

genus, 27

ncbi attribute, 27

species, 27

taxon, 27

canceling a term, 22

cancelto macro, 22

sorted as if “Cat”, 2

colors

shades, 81

conundrum

repurposed from proposition,
94

cross-reference, 56

description list, 42

doctest, 2

exercise, 55

exploration, 7

fill-in blank, 37

Fundamental Theorem of

Calculus, 1

Corollary, 5

genus, 27

GeoGebra, 50

Group, 41

image, 74

L^AT_EX image, 28

indefinite integral, 2

integral

indefinite integral, 2

internationalization, 60

investigation, 7

JSXGraph, 51

Led Zeppelin video, 53

list, 43

description, 42

multicolumn, 14–16

named, 43

ordered, 39

unordered, 39

listing, 65

matrix, 63

Maxwell’s equations, 17

mixed-content *emphasized*, 2

name, 94

named list, 43

ncbi attribute, 27

numbering, 92

numerical, *see* Sage integration,
see Sage

numerical integration

Sage

cell, *see* Sage, 2

numerics, *see also* Sage

ordered list, 39

package

siunitx, 70

panels, 71

paragraph

normal, 44

opens with list, 44

poem, 83, 91

problem, 7

program listing, 65

project, 7

question, 7

sorted as if “Quorum”, 2

quotations, 26

R, 3

ρ -fibers, 2

references

within a section, 36

rename an environment, 7

Sage

integration, 2

cell, 2

numerical, 2

Sage cell

- with a title, [4](#)
- Sage cells
 - linking, [11](#)
- Sage plots, [32](#)
- scientific names, *see* biological names
- siunitx package, [70](#)
- species, [27](#)
- strange colors, [14](#)
- structured-content *emphasized*, [2](#)
- table, [79](#)
- task, [7](#)
- taxon, [27](#)
- test: buried in
- theorem/statement/p, [1](#)
- Underdown, Jason, [22](#)
- Unicode, [60](#)
- units, [70](#)
 - Z (*sort as* A), [2](#)
 - A (*sort as* Z), [2](#)
- unordered list, [39](#)
- url, [60](#)
- URLs, [25](#)
- videos
 - YouTube, [53](#)
- YouTube videos, [53](#)

Colophon

This article was authored in PreTeXt.