Demonstration of features

Learning goals

- Learn about the capabilities of this lesson building framework
- Take a look around. Hey, math! $\frac{\pi^2}{6} = \sum_{n=1}^{\infty} \frac{1}{n^2}$.
- You get the idea.

With Pollen, we can define custom HTML and LaTeX environments for common setups. Here are a few examples:

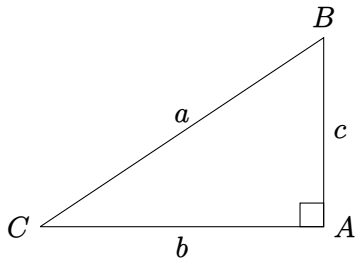
Warmup!

Find the line perpendicular to the line $y = \frac{2}{3}x - 7$ which passes through the point $(6, 0)$.

This is an optional sidenote—a great place to put non-essential information. On mobile devices where screen space is limited, sidenotes like these collapse out of view.

PYTHAGOREAN THEOREM

In a right triangle, the square of the length of the hypotenuse of a right triangle equals the sum of the squares of the lengths of the other two sides.



For this triangle, this means that $a^2 + b^2 = c^2$.

Example: Some boring problem

The bottom of a ladder must be placed 3 feet from a wall. The ladder is 10 feet long. How far above the ground does the ladder touch the wall?

Your turn!

A soccer field is a rectangle 100 meters wide and 130 meters long. The coach asks players to run from one corner to the other corner diagonally across. What is that distance?

A HOME FOR MANIMATIONS

Manim is a python library for creating mathematical animations. Originally developed by Grant Sanderson (of 3Blue1Brown), the software is

now maintained and actively developed by the Manim Community[1].

[1] *https://www.manim.community/*

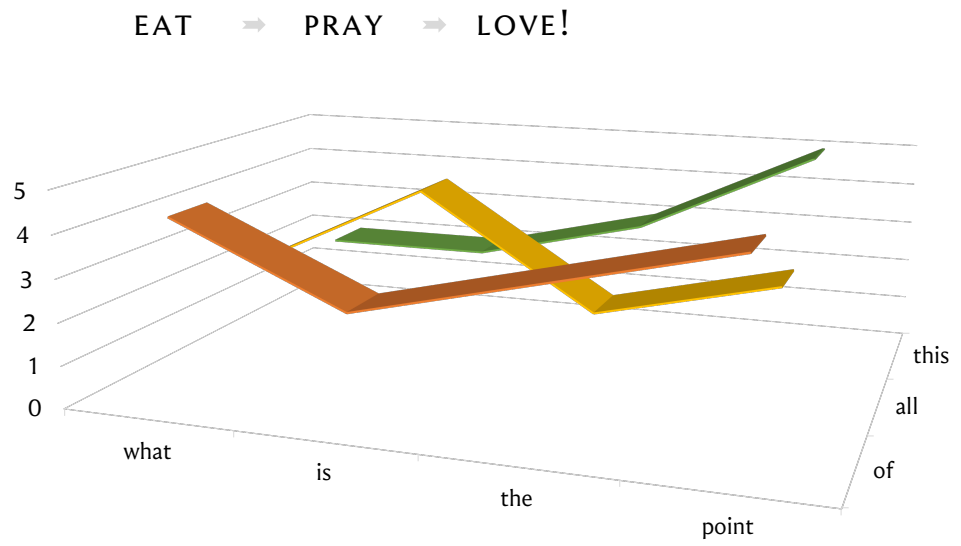
I've incorporated Manim animations in my lessons before, but I've noticed that pausing the animations to narrate can be a little tricky to time correctly. Wouldn't it be nice if the video somehow *knew* where to pause?

It turns out that in Manim, you can set splitting points within a scene and render that scene into multiple separate animations. If we were making a PowerPoint, we would then manually drag every individual animation over to our presentation. But with Pollen, we can define our own custom markup that does the heavy lifting for us. The command `◊manim["name-of-scene"]` grabs all of individual animations for a given scene and lays them out in a carousel for us.

Widescreen animations work also. And in the slides version of this lesson, the animation flows perfectly with the rest of the content.

POWERPOINT FIGURES: YOU CAN HAVE IT ALL

I appreciate how quick and easy figure-making can be with Powerpoint, so I want to have some way of incorporating Powerpoint-generated figures into these lessons. But because I am allergic to rasterization, the vector graphics need to be preserved (as SVG in HTML, and as PDF in LaTeX). The command `◊pptfig["fig-name"]` grabs `fig-name.pdf`, converts it to SVG, and inserts the appropriately-scaled image into the page—for LaTeX, it inserts the PDF.



LaTeX: NO MORE MATHJAX

MathJax is the defacto standard for typesetting mathematics on the web. But there are some downsides:

1. Like anything rendered on the client's side with JavaScript, there's a start-up time tax. As of MathJax 3.0, that wait is brief (and with KaTeX, even briefer) but the more math on the page, the more noticeable the wait.
2. There are only a subset of LaTeX commands and environments to choose from. That's not a failing of MathJax—it was only ever meant to handle inline and display math. But I want it all.
3. Limited fonts (with MathJax 3.0, only one). I don't have major problems with the default font, but like most things in life, I like to have options. More importantly, I incorporate text in my math equations all the time and prefer it to match the body text:

$$z\text{-score} = \frac{\text{“data” } - \text{ “mean”}}{\text{“standard deviation”}} = \frac{y - \mu}{\sigma}$$

With Pollen, we don't have to compromise—we can write markup for whatever the heck we want. In this case, I've written some custom markup that takes in LaTeX code, spits out a PDF, converts that PDF to SVG, and throws that image into our document.

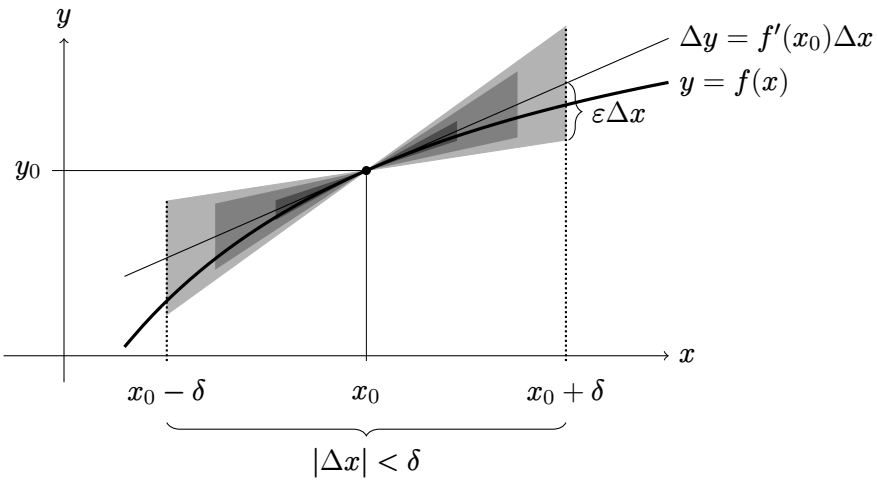
Let's try it out. For display LaTeX, wrap your equation around ◊$$ ... :

```
◊\$\${\mathcal{L}_{\mathcal{T}}(\vec{\lambda}) = \sum_{\mathbf{x},\mathbf{s}\in\mat
```

$$\mathcal{L}_{\mathcal{T}}(\vec{\lambda}) = \sum_{\mathbf{x},\mathbf{s}\in\mathcal{T}} \log P(\mathbf{x}|\mathbf{S}) - \sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}$$

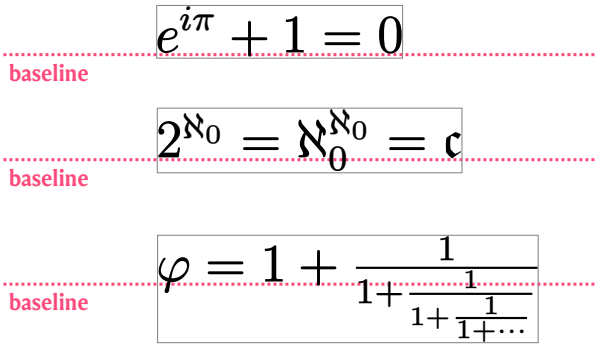
Or do whatever you want with the more general ◊latex command. Here's a TikZ figure just because we can:

Figure courtesy of Charles Staats (*https://tex.stackexchange.com/a/158732/148973*)



To be clear, this figure wasn't generated externally and dropped into the document—the source code is right inline with everything else. Is this as cool as I think it is?

Surprisingly, inline math was the most involved part. Since our equations get rendered as SVG, there's nothing telling the browser how to position the image vertically relative to the surrounding text.



My solution: slip a microscopic hot-pink-colored period into the beginning of every inline equation, have Pollen search the SVG for that pink dot, grab its height, and throw in some CSS that shifts the equation into place. Is it hacky? Yes. But it works perfectly.

Wanna know the coolest part? On most browsers, text-selecting an image actually selects the alt-text. By setting the alt-text as the LaTeX code, selecting and copying text on this page will also copy the LaTeX code (yes, even the TiKZ figure). Go ahead, try it out on this next paragraph.

Let $A \in M_n$. Then $x^*Ax = 0$ for all $x \in \mathbb{C}^n$ implies that $A = 0$, and if $\int_A^B x^*Ax = 0$ for all $x \in \mathbb{C}^n$, then $\sigma(A) = \{0\}$. Furthermore, if A is nondefective then $A = 0$.

If you are concerned that these figures will be compiling every time the document is updated, then you will be pleased to know that's not a problem. A `◊latex` command is only sent to get rendered if the content has changed (or if you set the `POLLEN` environment variable (*https://docs.racket-lang.org/pollen/raco-pollen.html*) to TEX).

Until I encounter an inline equation in which I must use the exact color `#FF0FF0`, I think this solution is safe.

One last cool LaTeX thing: tables. Tables in HTML suck, and tables in LaTeX suck a little less. With Pollen, we can make them awesome. The markup is similar to that of markdown, with hyphens (—) and pipes (|) to create rows and columns, but with an extension to allow for multicolumn headers and custom spacing.

I still need to figure out how to align decimals. That's coming.

		ALPHA			BRAVO	
		a_1	a_2	a_3	b_2	b_3
CHARLIE	c_1	1902	1290	293	12	234
	c_2	213	345	23	234	456
DAVID	d_1	34	4356	98	23	435
	d_2	892	283	108	890	523

GEOGEBRA EMBEDDING

MISCELLANEOUS FEATURES

- The ◊book command hides content in the brackets from the slideshow.
- Likewise, the ◊slide command is for slide-only content.
-
- Light and dark modes. The site automatically detects if you are using light or dark mode on your system, but you can override this by clicking the button on the top menu bar

COMING FEATURES

- multiple choice quizzes for live polls
- homework question environments
- collapsible solutions environment