

Create title page using `%%title` magic or
`self.title()` context manager.

Author: Abdul Saboor

Create Slides using

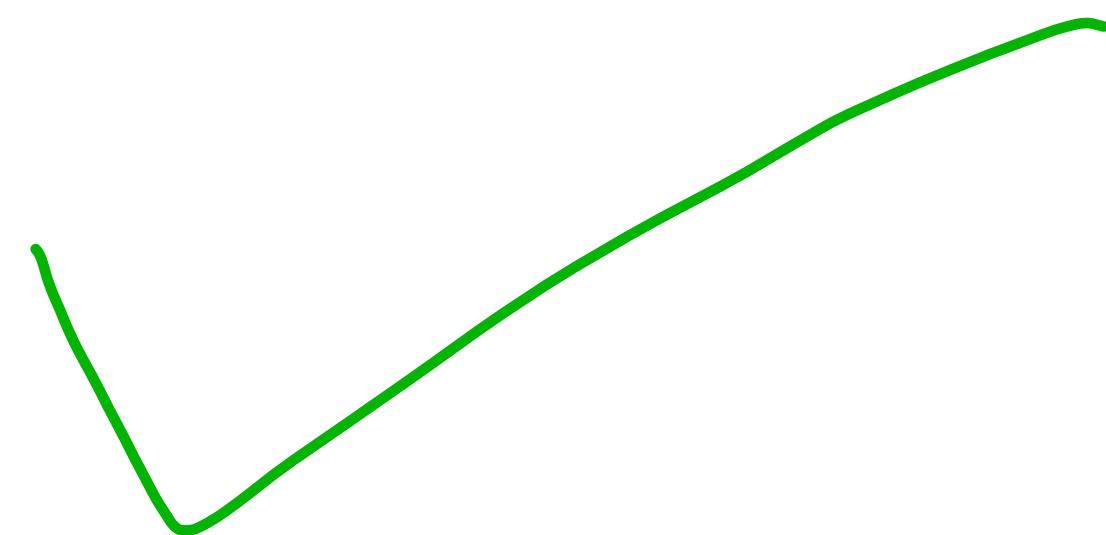
`%%slide`

or with

`self.slide()`

context manager.

Read instructions by clicking on left-bottom button



I am created using `with slides.slide(1) context manager!`

I am **Alerted** and I am *colored and italic text*

Code

```
1 | write(f'I am {slides.alert("Alerted")}' and I am *{slides.colored("Colored and italic text","magi
```

I am created using magic %%slide 2

I am created using @slides.slides

----- Above text generated by this!-----

```
## I am created using `@slides.slides`
```

IPySlides Online Running Sources

Launch as voila slides (may not work as expected ¹)  [launch](#) [binder](#)

[Edit on Kaggle](#)

Launch example Notebook  [launch](#) [binder](#)

-
1. Add references like this per slide. Use slides.cite() to add citations generally. [←](#)

----- Above text generated by this! -----

```
# IPySlides Online Running Sources Launch as voila slides (may not work as expected [^1])[![Binder]
(https://mybinder.org/badge\_logo.svg)](https://mybinder.org/v2/gh/massgh/ipyslides-voila/HEAD?urlpath=voila%2Frender%2Fnotebooks%2Fipyslides.ipynb) [Edit on Kaggle]
(https://www.kaggle.com/massgh/ipyslides) Launch example Notebook [![Binder]
(https://mybinder.org/badge\_logo.svg)](https://mybinder.org/v2/gh/massgh/ipyslides-voila/HEAD?urlpath=lab%2Ftree%2Fnotebooks%2Fipyslides.ipynb)
[^1]: Add references like this per slide. Use slides.cite() to add citations generally.
```

IPython Display Objects

Any object with following methods could be in `write` command:

`_repr_pretty_`, `_repr_html_`, `_repr_markdown_`, `_repr_svg_`, `_repr_png_`, `_repr_jpeg_`, `_repr_latex_`,
`_repr_json_`, `_repr_javascript_`, `_repr_pdf_`

Such as `IPython.display.<HTML,SVG,Markdown,Code>` etc. or third party such as `plotly.graph_objects.Figure`.

Plots and Other Data Types

These objects are implemented to be writable in `write` command:

`matplotlib.pyplot.Figure`, `altair.Chart`, `pygal.Graph`, `pydeck.Deck`, `pandas.DataFrame`, `bokeh.plotting.Figure`

Many will be extentended in future. If an object is not implemented, use `display(obj)` to show inline or use library's specific

command to show in Notebook outside `write`.

Interactive Widgets

Any object in `ipywidgets`

Link to ipywidgtes right here using `textbox` command

or libraries based on ipywidgtes such as `bqplot`, `ipyvolume`, plotly's `FigureWidget`¹(reference at end) can be included in `iwrite` command. Text/Markdown/HTML inside `iwrite` is made available through `ihtml` command.

Commands which do all Magic!

slides.write / ipyslide.utils.write

```
1 def write(*columns, width_percents=None):
2     '''Writes markdown strings or IPython objects to a string.
3     Each column should be a valid object that can be converted to
4     a string via str(). See notes below for details.
5
6     - Pass int, float, dict, function etc. Pandas DataFrames
7     - Give a code object from `ipyslides`.
8     - Give a matplotlib `figure/Axes` to `FigureConverter`.
9     - Give an interactive plotly figure.
10    - Give a pandas dataframe `df` or `df.to_html()`.
11    - Give any object which has `to_html` method.
12    - Give an IPython object which has `__html__` method.
13    - Give a function/class/module (without parentheses) which
14      has a `__html__` method.
```

slides.iwrite/ipyslide.utils.iwrite

```
1 def iwrite(*columns, width_percents=None):
2     """Each obj in columns should be an IPy
3     Text and other rich IPython content 1:
4     display(_fmt_iwrite(*columns, width_perce
```

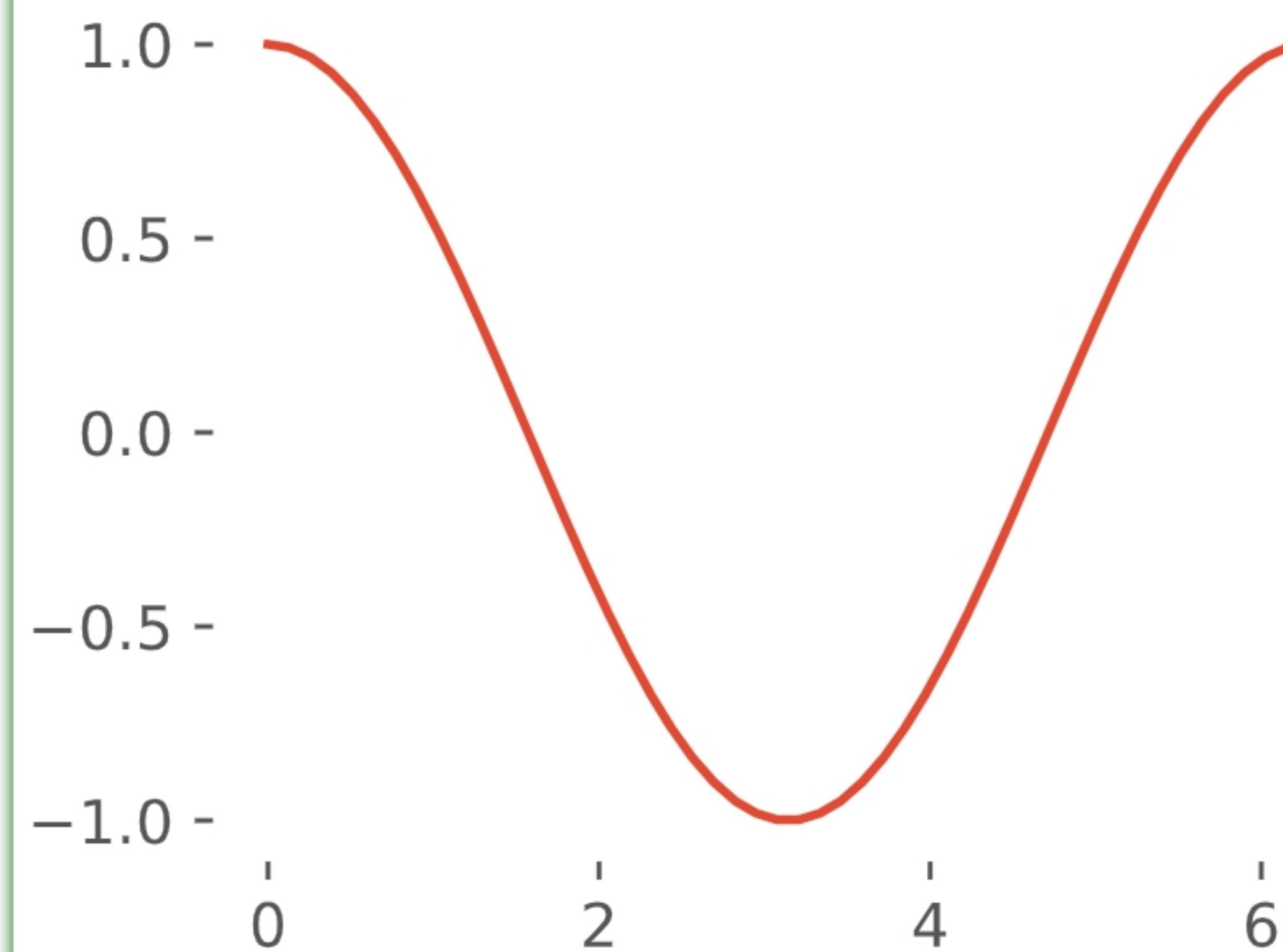
slides.html/ipyslide.utils.html

```
1 def ihtml(*columns, width_percents=None):
2     "Returns an ipywidgets.HTML widget. Accepts
3     return ipw.HTML(_fmt_write(*columns, width_percents))
```

Plotting with Matplotlib

Matplotlib inside block!

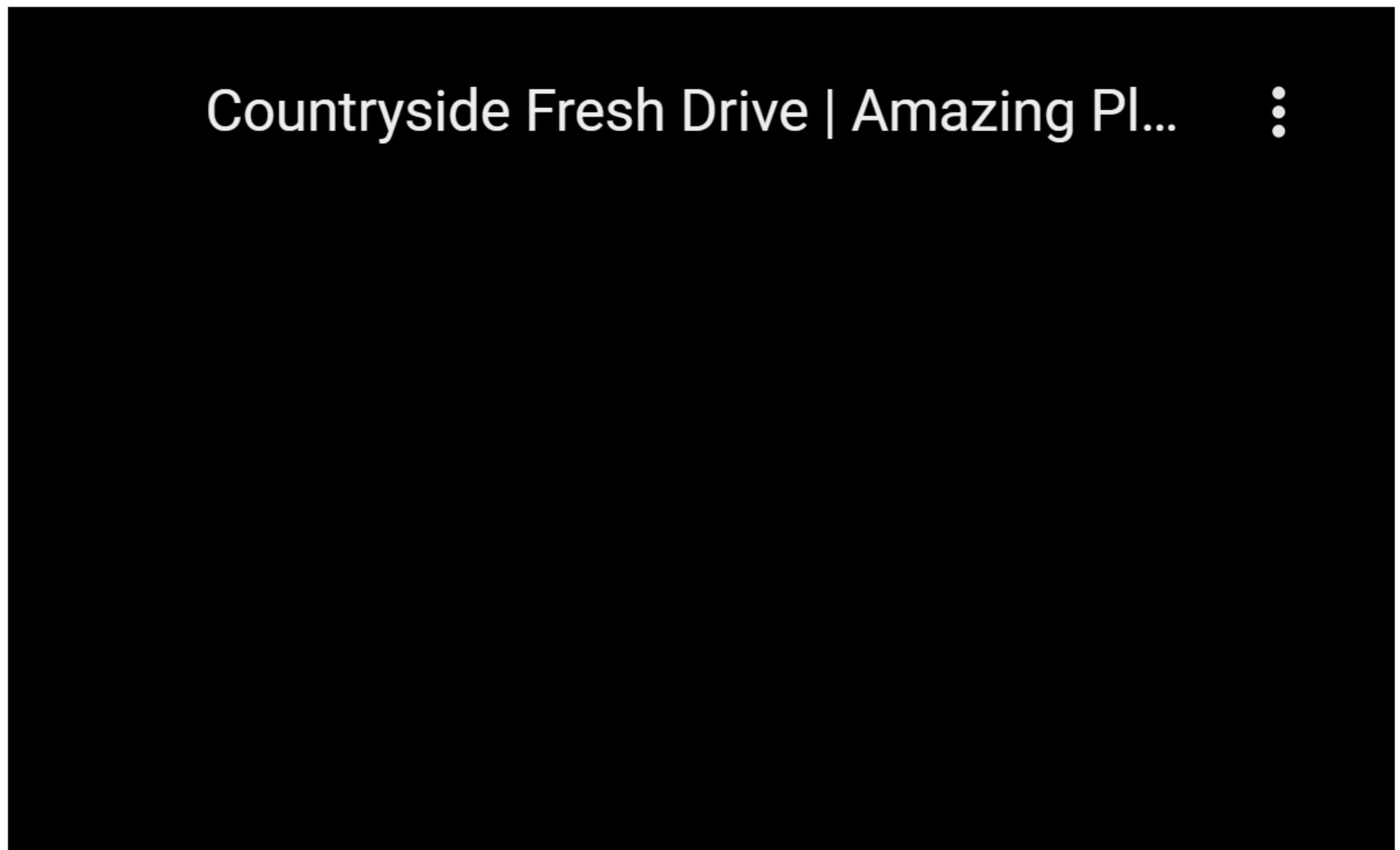
Alerting inside block!



No need to save me in file, I directly show up here!

Watching Youtube Video?

Countryside Fresh Drive | Amazing Pl... ::



Data Tables

Here is Table

h1	h2	h3
d1	d2	d3
r1	r2	r3

Here is Code

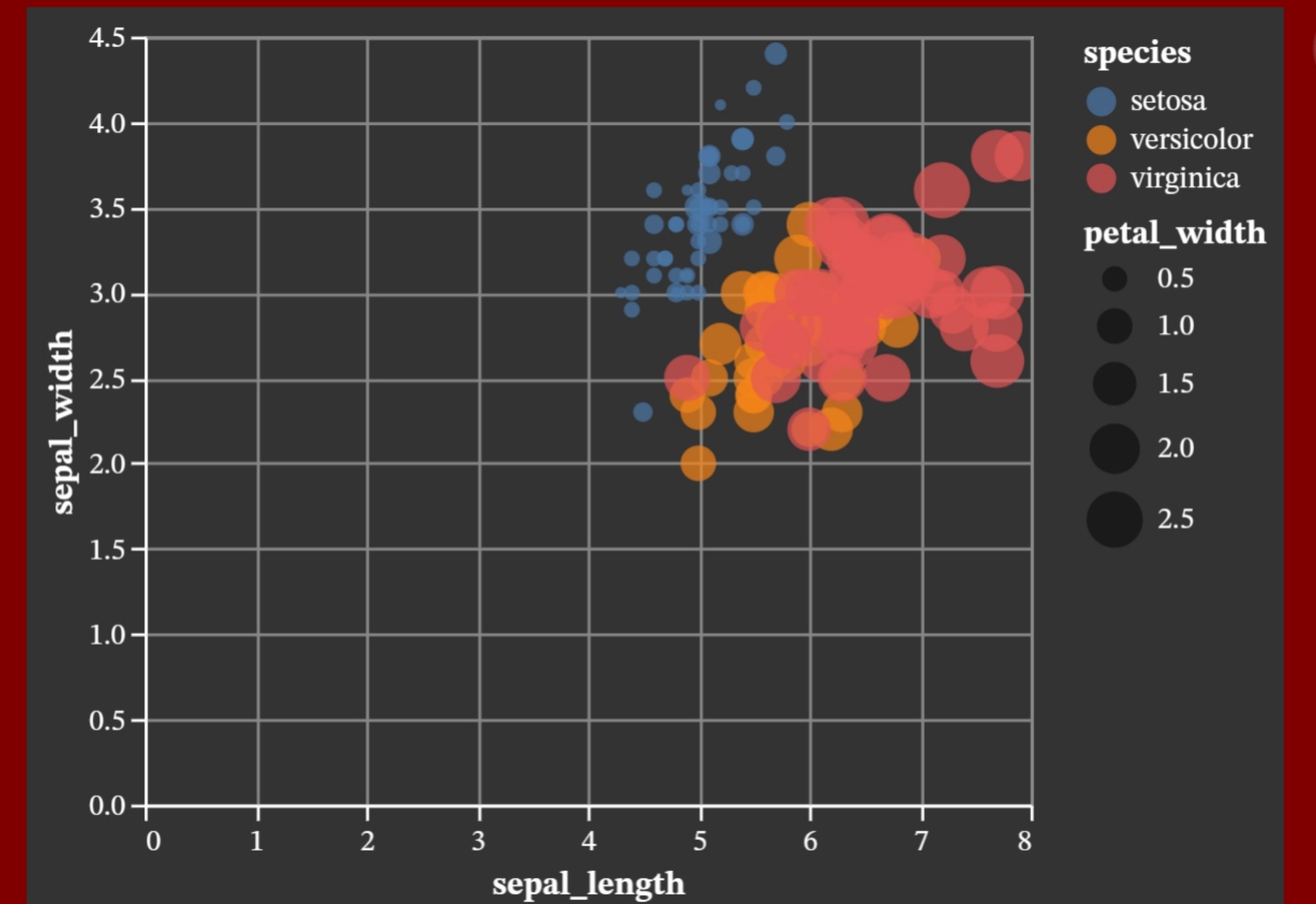
```
1 slides.block_o("""
2 |h1|h2|h3|
3 |---|---|---|
4 |d1|d2|d3|
5 |r1|r2|r3|
6 """)
```

Writing Pandas DataFrame

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Writing Altair Chart

May not work everywhere, needs javascript



Writing Plotly Figure

Install `plotly` to view output

Interactive Apps on Slide

Use ipywidgets , bqplot , ipyvolume , plotly Figurewidget etc. to show live apps like this!



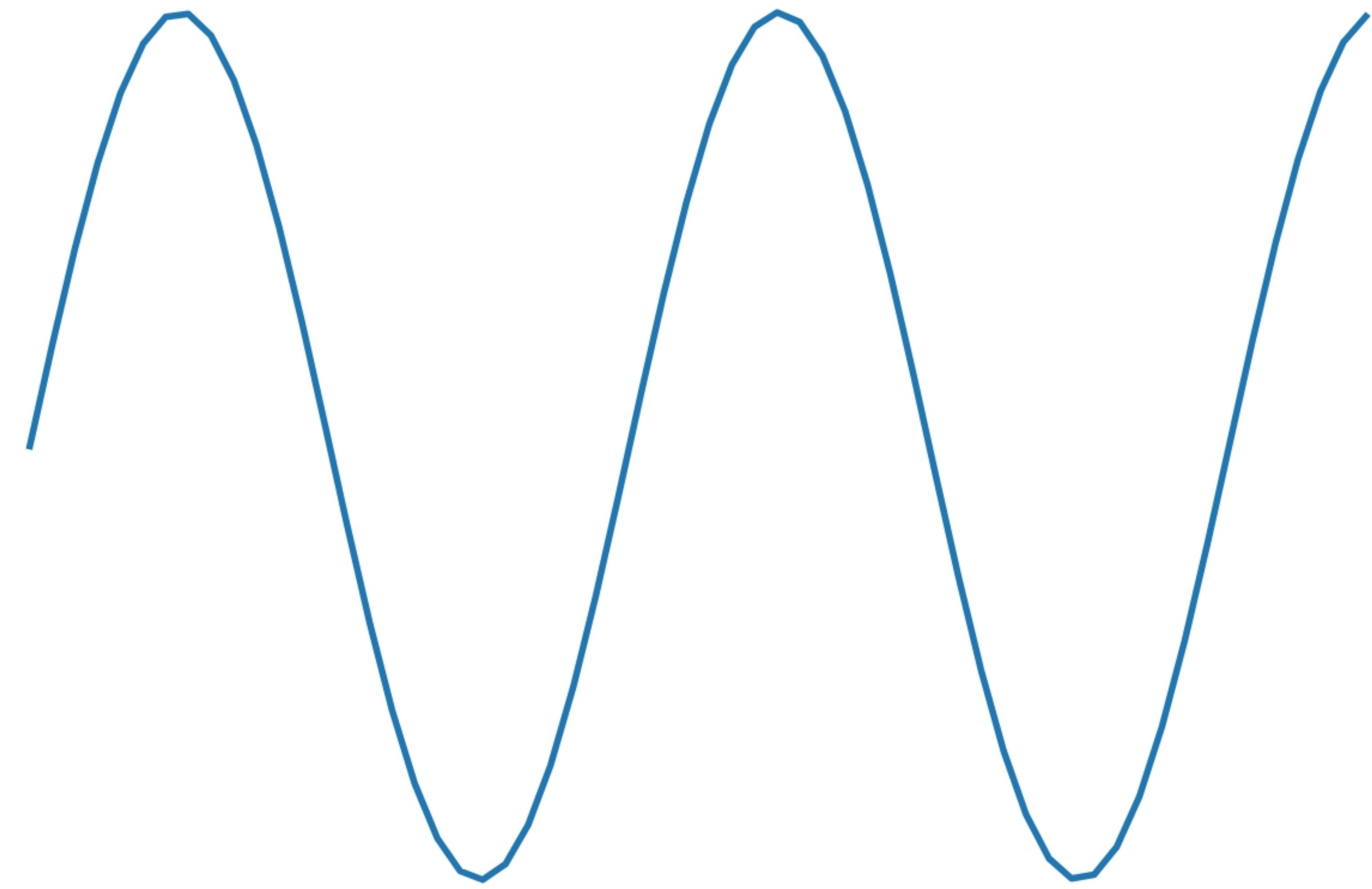
Current Value is 10

[Check out this app](#)

This is Slide 15

and we are animating
matplotlib

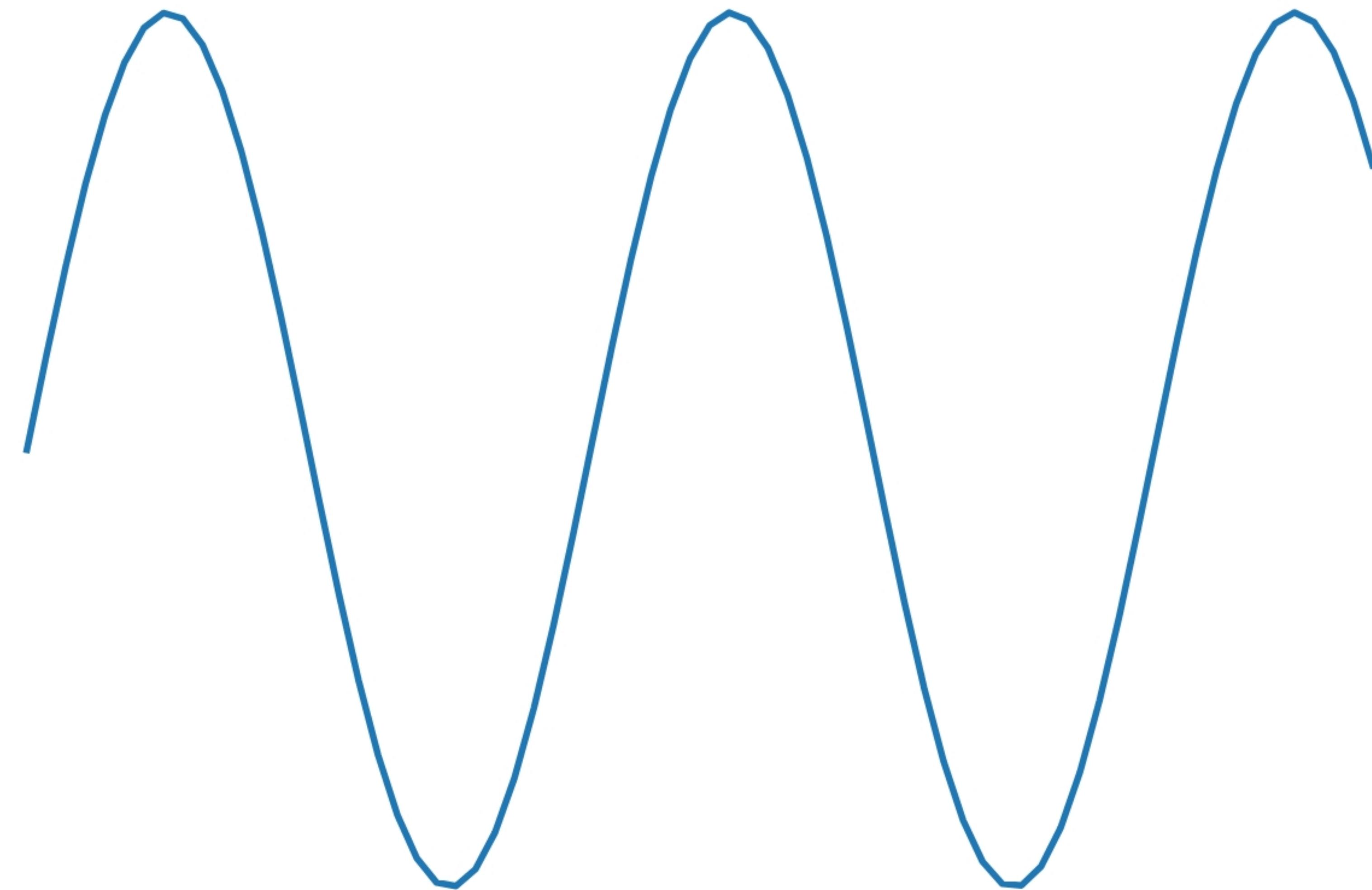
$$f(x) = \sin(x), 0 < x < 1$$



This is Slide 16

and we are animating
matplotlib

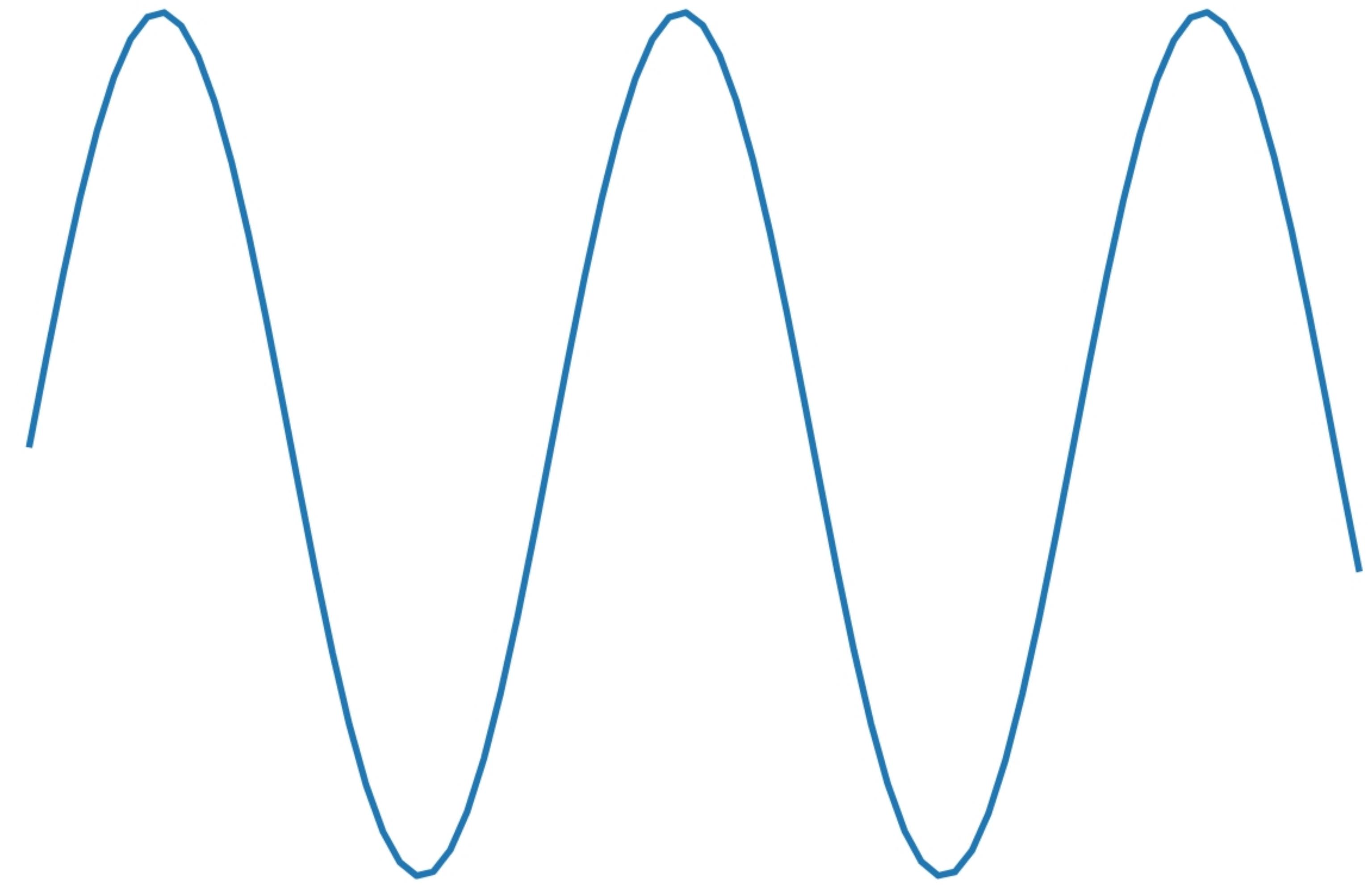
$$f(x) = \sin(x), 0 < x < 2$$



This is Slide 17

and we are animating
matplotlib

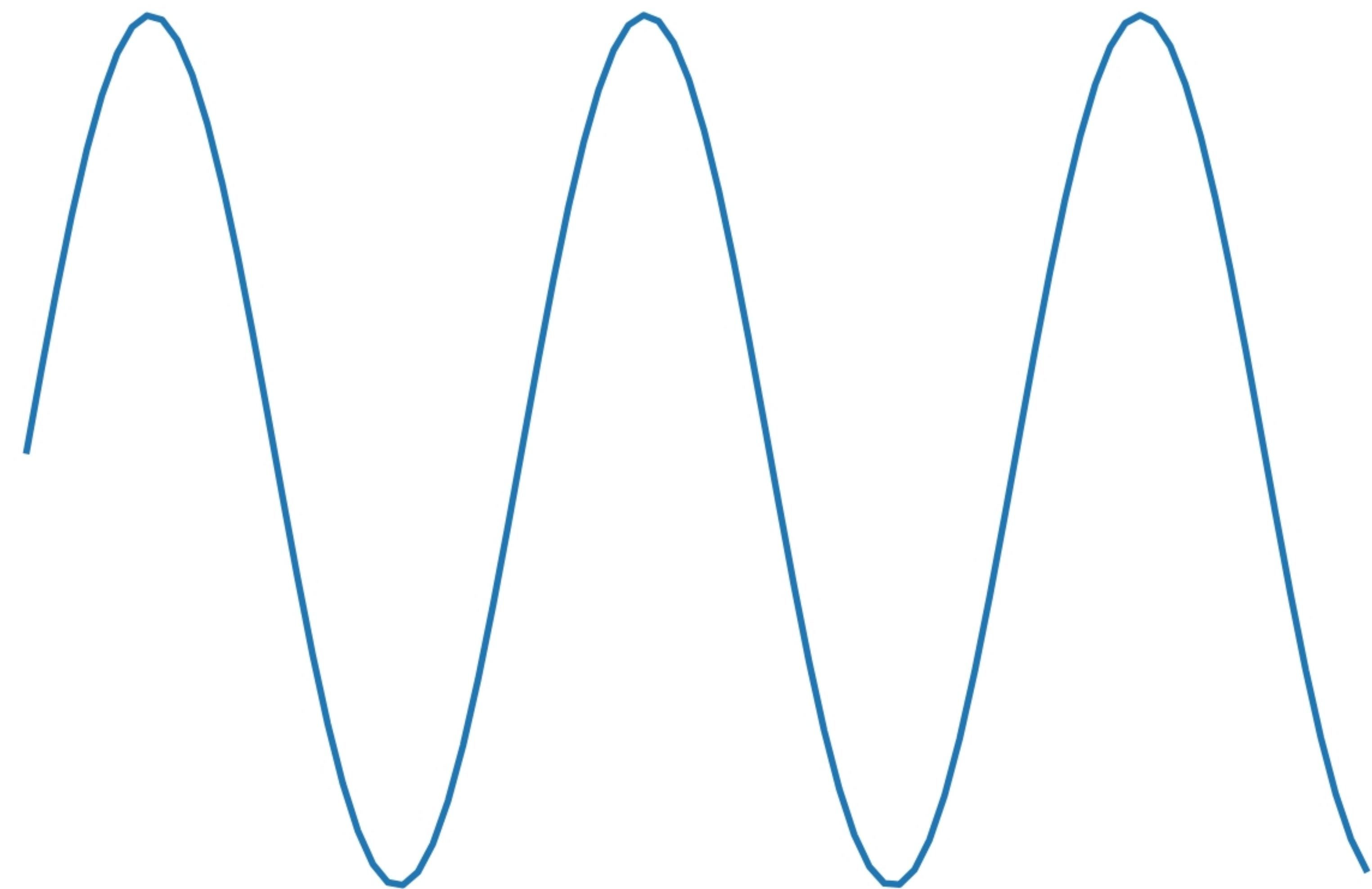
$$f(x) = \sin(x), 0 < x < 3$$



This is Slide 18

and we are animating
matplotlib

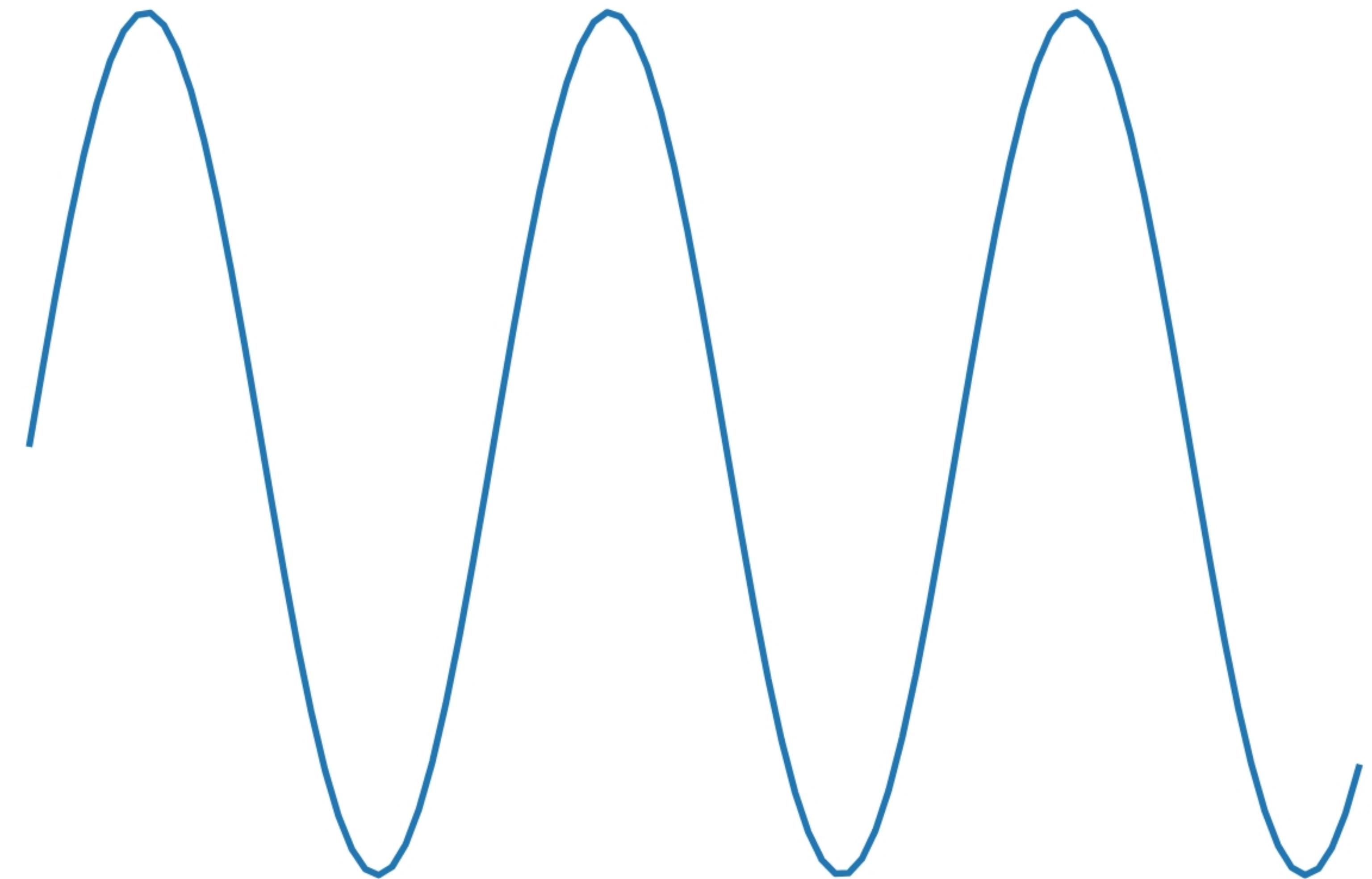
$$f(x) = \sin(x), 0 < x < 4$$



This is Slide 19

and we are animating
matplotlib

$$f(x) = \sin(x), 0 < x < 5$$



This is Slide 20 added with enum_slides

This is Slide 21 added with enum_slides

This is all code to generate slides

```
1 #Author: Abdul Saboor
2 # This demonstrates that you can generate slides from a .py file too, which you can import in no
3 from .core import LiveSlides
4 from .utils import write, ihtml, plt2html, iwrite, __reprs__, textbox
5 from .objs_formatter import libraries
6 slides = LiveSlides()
7 slides.convert2slides(True)
8 slides.set_footer('Author: Abdul Saboor')
9 slides.set_logo('''<svg viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg">
10             <circle cx="50" cy="50" r="50" fill="green"/>
11             <text x="35" y="50" fill="white">Logo</text></svg>''', width=50)
12
13 #title is skipped to show instructions
14 ...
```

```
1 def demo():
2     import os
3     from . import _demo, utils
4
5     slides = _demo.slides
6     with slides.slide(100):
7         write('## This is all code to generate slides')
8         write(_demo)
9         write(demo)
10    with slides.slide(101, background='#9ACD32'):
```

References

¹This is reference to FigureWidget using `slides.cite` command