

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

| Author: Abdul Saboor

Create Slides using

`%%slide`

or with

`self.slide(slide_number)`

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*

```
1 write('## I am created using `with slides.slide(1)` context manager')
2 write(f'I am {slides.alert("Alerted")} and I am *{slides.colored('
```

I am created using magic

%%slide 2

I am created using @slides.slides

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

I am created using `@slides.slides`

```
1 slides.write('----- Above text generated by this!-----')
2 slides.write(slides.keep_format(item))
```

IPySlides Online Running Sources

Launch as voila slides (may not work as expected [1](#)) [!\[\]\(2e897e890e69d81eae4503a8342c36b0_img.jpg\) launch](#) [!\[\]\(ce4e2504c7100a62a9a9496b2e01b6e4_img.jpg\) binder](#)

[Edit on Kaggle](#)

Launch example Notebook [!\[\]\(e2376d476d06eb31946dc01a69a4403a_img.jpg\) launch](#) [!\[\]\(bbb3388d591ef640dd8a8c4262f2866a_img.jpg\) 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])![Binderv](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 [Binderv]
(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.
```

```
1 slides.write('----- Above text generated by this!-----')
2 slides.write(slides.keep_format(item))
```

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 object with method `__re
3     Each column should be a valid object (text/markdown/html/ h
4
5     - Pass int,float,dict,function etc. Pass list/tuple in a wr
6     - Give a code object from `ipyslides.get_cell_code()` to it
7     - Give a matplotlib `figure/Axes` to it or use `ipyslides.u
8     - Give an interactive plotly figure.
9     - Give a pandas dataframe `df` or `df.to_html()` .
10    - Give any object which has `to_html` method like Altair ch
11    - Give an IPython object which has `__repr__<repr>_` method w
12    - Give a function/class/module (without calling) and it wil
13
14    If an object is not in above listed things, `obj.__repr__()` m
15    methods specific to that library to show in jupyter notebook
16
17    Note: Use `keep_format` method to keep format of object for
18    Note: You can give your own type of data provided that it i
19    Note: `__repr__<format>_` takes precedence to `to<format>` m
20
21    ...
22
23    return display(HTML(_fmt_write(*columns, width_percents=width_p
```

slides.iwrite/ipyslide.utils.iwrite

```
1 def iwrite(*columns, width_percents=None):
2     """Each obj in columns should be an IPython widget like `ipyw
3         Text and other rich IPython content like charts can be adde
4     display(_fmt_iwrite(*columns, width_percents=width_percents))
```

slides.ihtml/ipyslide.utils.ihtml

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

If an object does not render as you want, use

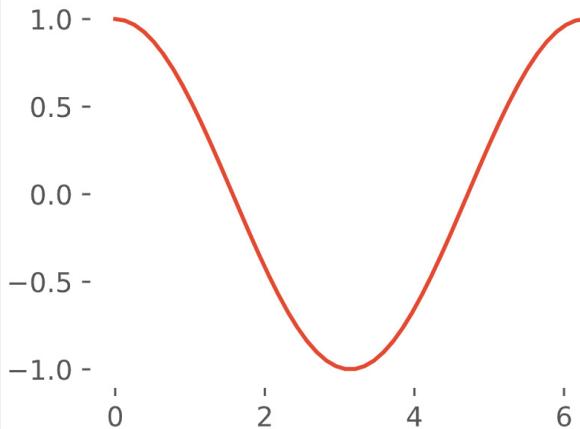
`display(object)` or it's own library's method to display
inside Notebook.

```
1 write(slides.block_r('slides.write/ipyslide.utils.write', write),
2       slides.rows(slides.block_b('slides.iwrite/ipyslide.utils.iwrit
3       slides.block_b('slides.ihtml/ipyslide.utils.ihtml', ihtml)
4       )
5   )
6 write("#### If an object does not render as you want, use `displa
```

Plotting with Matplotlib

Matplotlib inside block!

Alerting inside block!



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

```
1 import numpy as np, matplotlib.pyplot as plt
2 x = np.linspace(0,2*np.pi)
3 with plt.style.context('ggplot'):
4     fig, ax = plt.subplots(figsize=(3.4,2.6))
5     _ = ax.plot(x,np.cos(x))
6
7 write('## Plotting with Matplotlib')
8 write(slides.block_g('Matplotlib inside block!',slides.alert('Alert')))
```

Watching Youtube Video?



```
1 write(f"### Watching Youtube Video?")
2 write(YouTubeVideo('Z3iR551KgpI',width='100%',height='266px'))
```

Data Tables

Here is Table

h1	h2	h3
d1	d2	d3
r1	r2	r3

```
1 write('## Data Tables')
2 write(slides.block_r('Here is Table',
3   textwrap.dedent(''
4     |h1|h2|h3|
5     |---|---|---|
6     |d1|d2|d3|
7     |r1|r2|r3|
8     ''')))
```

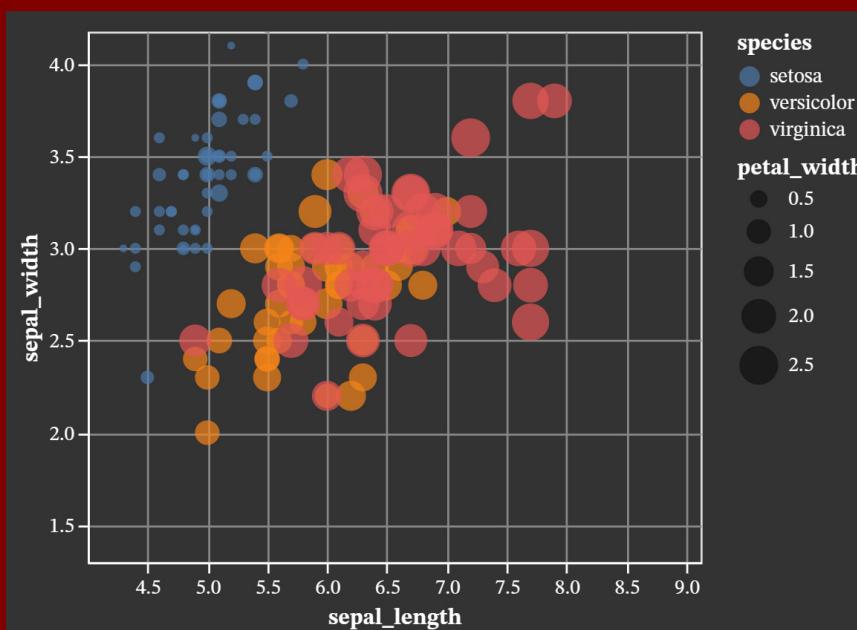
Writing Pandas DataFrame

	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



```
1 try:
2     import pandas as pd
3     import altair as alt
4     alt.themes.enable('dark')
5     df = pd.read_csv('https://raw.githubusercontent.com/mwaskom/seaborn-data/master/iris.csv')
6     chart = alt.Chart(df, width=300, height=260).mark_circle(size=60).encode(
7         x='sepal_length',
8         y='sepal_width',
9         color='species',
10        size = 'petal_width',
11        tooltip=['species', 'sepal_length', 'sepal_width','petal_width'])
12    ).interactive()
13    df = df.describe() #Small for display
14 except:
15     df = '### Install `pandas` to view output'
16     chart = '### Install Altair to see chart'
17     write((## Writing Pandas DataFrame',df),
18           ('## Writing Altair Chart\nMay not work everywhere, needs java
```

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!

Click Me To see Progress

Current Value is 20

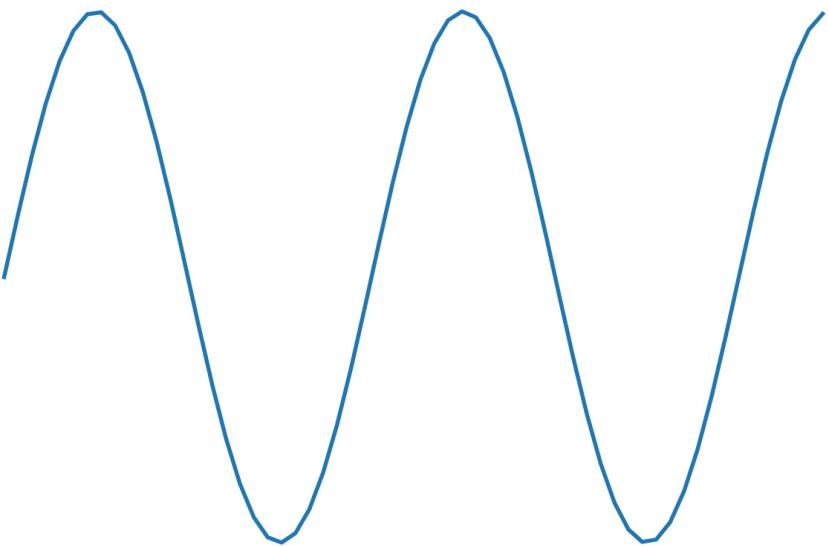
[Check out this app](#)

```
1 import ipywidgets as ipw
2 btn = ipw.Button(description='Click Me To see Progress', layout=ipw.
3 prog = ipw.IntProgress(value=10)
4 html = ihtml(f"Current Value is {prog.value}")
5 def onclick(btn):
6     prog.value = prog.value + 10
7     if prog.value > 90:
8         prog.value = 0
9     html.value = f"Current Value is {prog.value}"
10
11 btn.on_click(onclick)
12
13 write('## Interactive Apps on Slide\n Use `ipywidgets` , `bqplot` , `ipyvolume` , `plotly Figurewidget`')
14 iwrite(prog,[btn,html])
15 write("[Check out this app](https://massgh.github.io/pivotpy/Widge")
```

This is Slide 15

and we are animating matplotlib

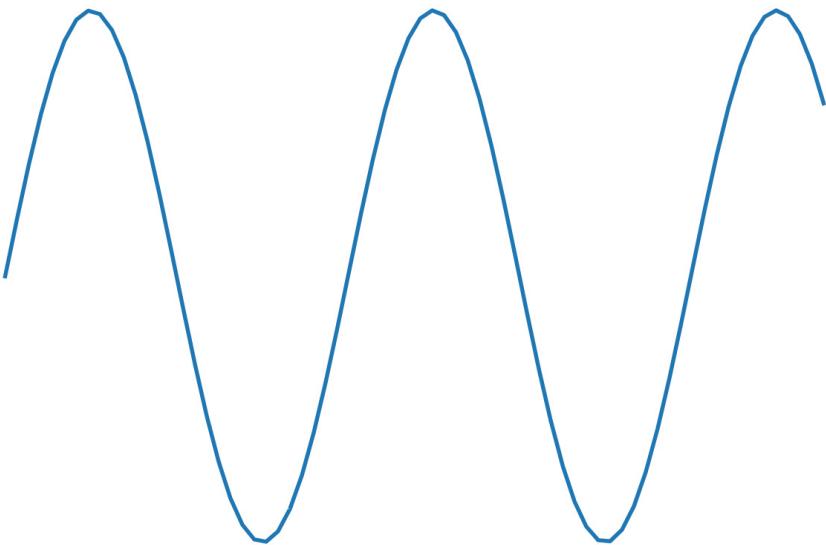
$$f(x) = \sin(x), 0 < x < 1$$



This is Slide 16

and we are animating matplotlib

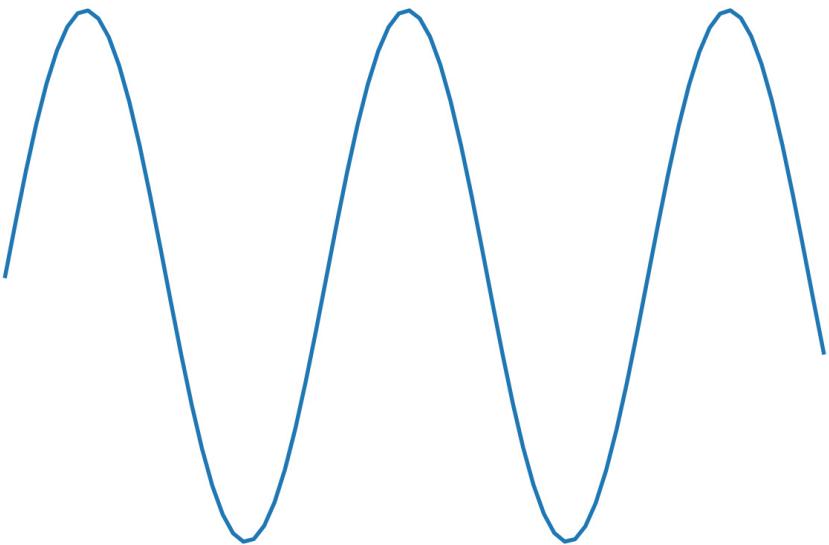
$$f(x) = \sin(x), 0 < x < 2$$



This is Slide 17

and we are animating matplotlib

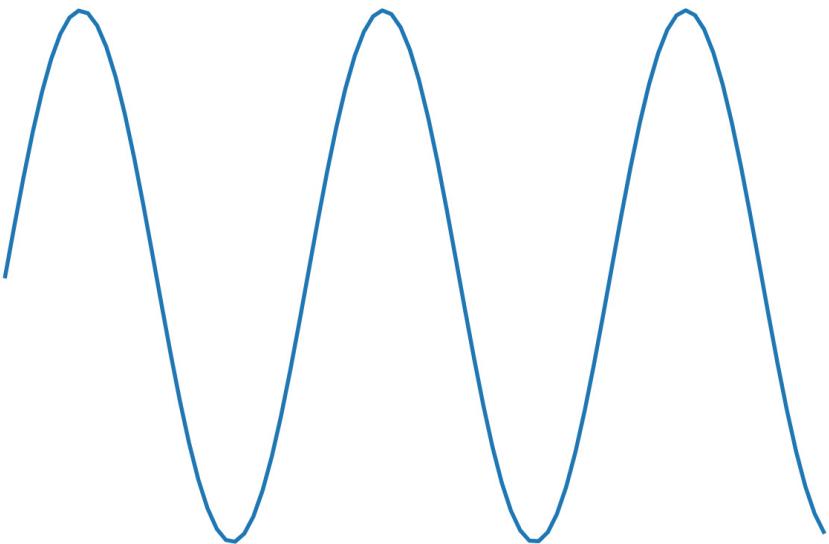
$$f(x) = \sin(x), 0 < x < 3$$



This is Slide 18

and we are animating matplotlib

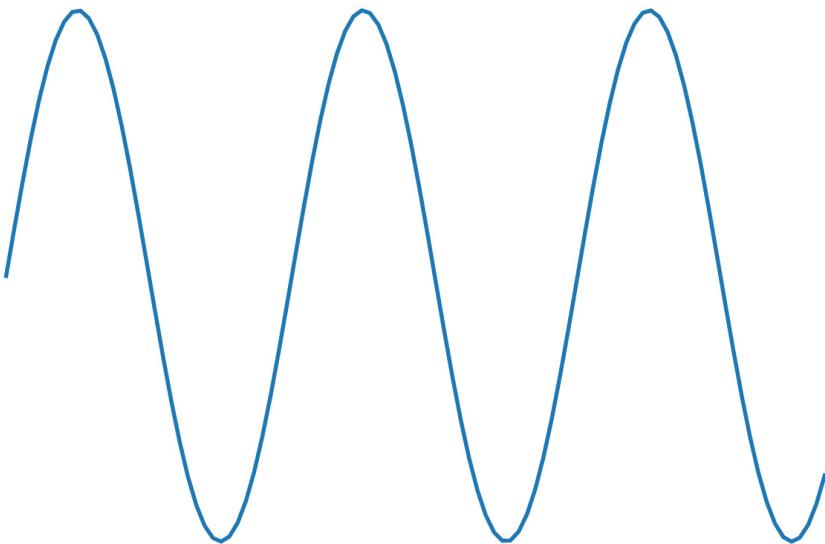
$$f(x) = \sin(x), 0 < x < 4$$



This is Slide 19

and we are animating matplotlib

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



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
3 import textwrap
4 from .core import LiveSlides
5 from .utils import write, ihtml, plt2html, iwrite, __reprs__, textbox
6 from .objs_formatter import libraries
7 slides = LiveSlides()
8 slides.convert2slides(True)
9 slides.set_footer('Author: Abdul Saboor')
10 slides.set_logo('''<svg viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg">
11 <circle cx="50" cy="50" r="50" fill="green"/>
12 <text x="35" y="50" fill="white">Logo</text></svg>''')
13
14 #title is skipped to show instructions
15 with slides.slide(1): #slide 1
16     with slides.source():
17         write('## I am created using `with slides.slide(1)` context manager')
18         write(f'I am {slides.alert("Alerted")}) and I am *{slides.colour}*')
19 slides.shell.user_ns['write'] = write #Inject variable in IPython shell
20
21 #slide 2
22 slides.shell.run_cell_magic('slide', '2', 'write("## I am created using `with slides.slide(2)` context manager")')
23 #slide 3
24 online_sources = '''# IPySlides Online Running Sources
25 Launch as voila slides (may not work as expected [^1])![!Binder]({{ ipyslides.launch_url }})[Edit on Kaggle]({{ ipyslides.kaggle_url }}){{ ipyslides.kaggle_url }})
26 Launch example Notebook [!{{ ipyslides.binder_url }}]({{ ipyslides.binder_url }}){{ ipyslides.binder_url }})
27 <br>
28 [^1]: Add references like this per slide. Use slides.cite() to add them to the presentation
29
```

```
30 ...
31 @slides.slides(2,'## I am created using `@slides.slides`',online_
32 def func(item):
33     slides.write(item)
34     with slides.source():
35         slides.write('----- Above text generated by this!-----')
36         slides.write(slides.keep_format(item))
37
38 #Now generate many slides in a loop
39 __contents = [f"""## IPython Display Objects
40 ##### Any object with following methods could be in`write` command
41 {', '.join([f'{repr_}` for repr_ in __reprs_])}
42 Such as `IPython.display.<HTML,SVG,Markdown,Code>` etc. or third
43 """,
44 f"""## Plots and Other Data Types
45 ##### These objects are implemented to be writable in `write` comm
46 {', '.join([f'{lib['name']}.{lib['obj']}' for lib in libraries])}
47 Many will be extended in future. If an object is not implement
48 command to show in Notebook outside `write`.
49 """,
50 f"""## Interactive Widgets
51 ### Any object in `ipywidgets`{textbox('<a href="https://ipywidge
52 or libraries based on ipywidgtes such as `bqplot`, `ipyvolume`, plo
53 can be included in `iwrite` command. Text/Markdown/HTML inside `i
54 """,
55 '## Commands which do all Magic!']
56 for i in range(3,7):
57     with slides.slide(i, background=f'linear-gradient(to right, oliv
58     write(__contents[i-3])
59     if i == 6:
60         with slides.source():
61             write(slides.block_r('slides.write/ipyslide.utils.write',\n
62                 slides.rows(slides.block_b('slides.iwrite/ipyslide.util
```

```
63         slides.block_b('slides.ihtml/ipyslide.utils.ihtml',iht
64             )
65         )
66         write("#### If an object does not render as you want, use
67
68 # Matplotlib
69 with slides.slide(7):
70     with slides.source():
71         import numpy as np, matplotlib.pyplot as plt
72         x = np.linspace(0,2*np.pi)
73         with plt.style.context('ggplot'):
74             fig, ax = plt.subplots(figsize=(3.4,2.6))
75             _ = ax.plot(x,np.cos(x))
76
77         write('## Plotting with Matplotlib')
78         write(slides.block_g('Matplotlib inside block!',slides.alert(
79
80 # Youtube
81 from IPython.display import YouTubeVideo
82 with slides.slide(8):
83     with slides.source():
84         write(f"### Watching Youtube Video?")
85         write(YouTubeVideo('Z3iR551KgpI',width='100%',height='266px')
86
87 # Data Table
88 with slides.slide(9):
89     with slides.source():
90         write('## Data Tables')
91         write(slides.block_r('Here is Table',
92             textwrap.dedent('''
93                 |h1|h2|h3|
94                 |---|---|---|
95                 |d1|d2|d3|
96                 '''))
```

```
96         |r1|r2|r3|
97         '')))
98
99 # Plotly and Pandas DataFrame only show if you have installed
100 with slides.slide(10,background="#800000"):
101     with slides.source():
102         try:
103             import pandas as pd
104             import altair as alt
105             alt.themes.enable('dark')
106             df = pd.read_csv('https://raw.githubusercontent.com/mwaskom/:
107             chart = alt.Chart(df, width=300, height=260).mark_circle(size=100)
108             chart.encode(x='sepal_length',
109                         y='sepal_width',
110                         color='species',
111                         size='petal_width',
112                         tooltip=['species', 'sepal_length', 'sepal_width', 'petal_width'])
113             chart.interactive()
114             df = df.describe() #Small for display
115         except:
116             df = '### Install `pandas` to view output'
117             chart = '### Install Altair to see chart'
118             write((## Writing Pandas DataFrame',df),
119                   ('## Writing Altair Chart\nMay not work everywhere, needs jupyter')
120                   )
121
122     try:
123         import plotly.graph_objects as go
124         fig = go.Figure()
125         fig.add_trace(go.Bar([1,5,8,9]))
126     except:
127         fig = '### Install `plotly` to view output'
128     with slides.slide(11):
```

```
129     write('## Writing Plotly Figure',fig)
130
131 # Interactive widgets can't be used in write command, but still t
132
133 with slides.slide(12):
134     with slides.source():
135         import ipywidgets as ipw
136         btn = ipw.Button(description='Click Me To see Progress',layout
137         prog = ipw.IntProgress(value=10)
138         html = ihtml(f"Current Value is {prog.value}")
139         def onclick(btn):
140             prog.value = prog.value + 10
141             if prog.value > 90:
142                 prog.value = 0
143             html.value = f"Current Value is {prog.value}"
144
145         btn.on_click(onclick)
146
147         write('## Interactive Apps on Slide\n Use `ipywidgets` , `bqpl
148         iwrite(prog,[btn,html])
149         write("[Check out this app](https://massgh.github.io/pivotpy/
150
151 # Animat plot in slides
152 @slides.slides(12,*range(13,18))
153 def func(item):
154     fig, ax = plt.subplots()
155     x = np.linspace(0,item+1,50+10*(item - 12))
156     ax.plot(x,np.sin(x));
157     ax.set_title(f'$f(x)=\sin(x)$, $0 < x < {item - 12}$')
158     ax.set_axis_off()
159     slides.write(f'### This is Slide {item+2}\n and we are animatin
160
161 # Use enumerate to iterate over slides
```

```
161 # Use enumerate to iterate over slides
162 for i,s in slides.enum_slides(13,15,background='var(--secondary-bg)')
163     with s:
164         write(f'### This is Slide {i+7} added with `enum_slides`')
```

```
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"):
11        slides.write_citations()
12
13    slides.prog_slider.value = 0 # back to title
14    return slides
```

References

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