How to Use

Jupyter Lab Only

Having your cursor over slides:

- Press `Ctrl + Shift + C` to change the theme, create console/terminal etc.
- Press `Ctrl + Shift + [` , `Ctrl + Shift + ]` to switch to other tabs like console/terminal/notebooks and do coding without leaving slides!
- Press `F` to toggle fullscreen mode.

Jupyter Lab + Others (Notebook, VSCode, Voila etc.)

Author: Abdul Saboor عبدالصبور | Oct-29-2021

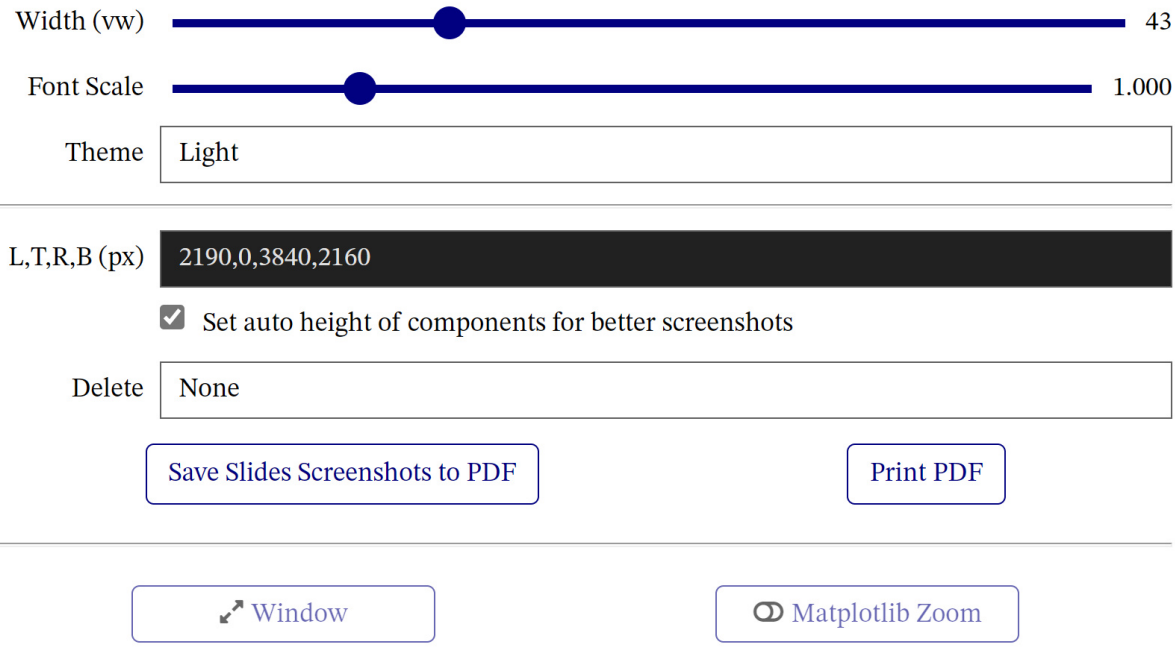
How to Use

Jupyter Lab Only

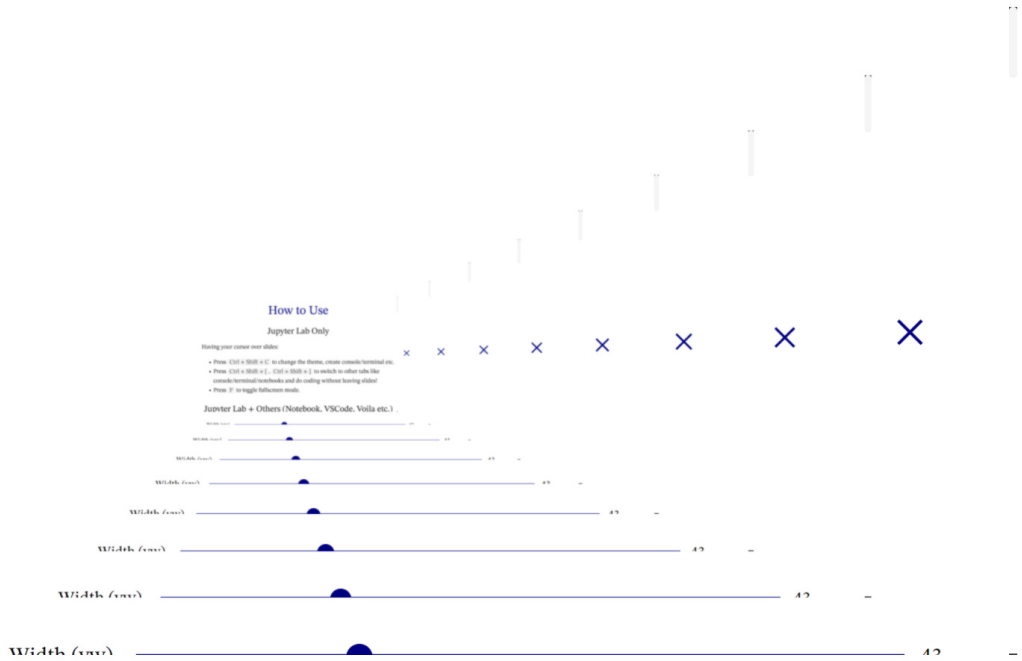
Having your cursor over slides:

- Press `Ctrl + Shift + C` to change the theme, create console/terminal etc.
- Press `Ctrl + Shift + [` , `Ctrl + Shift + ]` to switch to other tabs like console/terminal/notebooks and do coding without leaving slides!
- Press `F` to toggle fullscreen mode.

Jupyter Lab + Others (Notebook, VSCode, Voila etc.)



Author: Abdul Saboor عبدالصبور | Oct-29-2021



Width (vw) 43

Font Scale 1.000

Theme Light

L,T,R,B (px) 2190,0,3840,2160

Set auto height of components for better screenshots

Delete None

Save Slides Screenshots to PDF

Print PDF

Window

Matplotlib Zoom

Author: Abdul Saboor عبدالصبور | Oct-29-2021

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

Author: Abdul Saboor عبدالصبور | Oct-29-2021

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 m
2 write(f'I am {slides.alert("Alerted")} and I am *{slides.colo
```

I am created using magic

%%slide 2

I am created using

@slides.frames

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

I am created using `@slides.frames`

```
slides.write('------ Above text generated by this!-------')
slides.write(slides.keep_format(obj))
```

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.

```
slides.write('------ Above text generated by this!-------')
slides.write(slides.keep_format(obj))
```

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

```
def write(*columns,width_percents=None):
  '''Writes markdown strings or IPython object with method
    Each column should be a valid object (text/markdown/ht

    - Pass int,float,dict,function etc. Pass list/tuple in
    - Give a code object from `ipyslides.get_cell_code()`
    - Give a matplotlib `figure/Axes` to it or use `ipysli
    - Give an interactive plotly figure.
    - Give a pandas dataframe `df` or `df.to_html()`.
    - Give any object which has `to_html` method like Alta
    - Give an IPython object which has `_repr_<repr>_` met
    - Give a function/class/module (without calling) and i

    If an object is not in above listed things, `obj.__rep
    methods specific to that library to show in jupyter no

    Note: Use `keep_format` method to keep format of objec
    Note: You can give your own type of data provided that
    Note: `_repr_<format>_` takes precedence to `to_<forma

    '''
  return display(HTML(_fmt_write(*columns,width_percents=wi
```

slides.iwrite/ipyslide.utils.iwrite

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

slides.ihtml/ipyslide.utils.ihtml

```
def ihtml(*columns,width_percents=None):
  "Returns an ipywidgets.HTML widget. Accepts content type
  return ipw.HTML(_fmt_write(*columns,width_percents=width_
```

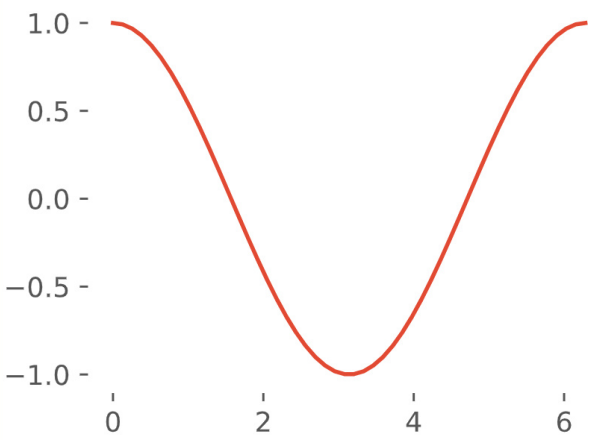
If an object does not render as you want, use `display(object)` or it's own library's mehod to display inside Notebook.

```
write(slides.block_r('slides.write/ipyslide.utils.write',wri
   slides.rows(slides.block_b('slides.iwrite/ipyslide.utils.
    slides.block_b('slides.ihtml/ipyslide.utils.ihtml',ihtml
    )
  )
write("#### If an object does not render as you want, use `d
```

Plotting with Matplotlib

Matplotlib inside block!

Alerting inside block!



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

```
import numpy as np, matplotlib.pyplot as plt
x = np.linspace(0,2*np.pi)
with plt.style.context('ggplot'):
  fig, ax = plt.subplots(figsize=(3.4,2.6))
  _ = ax.plot(x,np.cos(x))

write('## Plotting with Matplotlib')
write(slides.block_g('Matplotlib inside block!',slides.alert
```

Watching Youtube Video?


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

Data Tables

Here is Table

h1	h2	h3
d1	d2	d3
r1	r2	r3

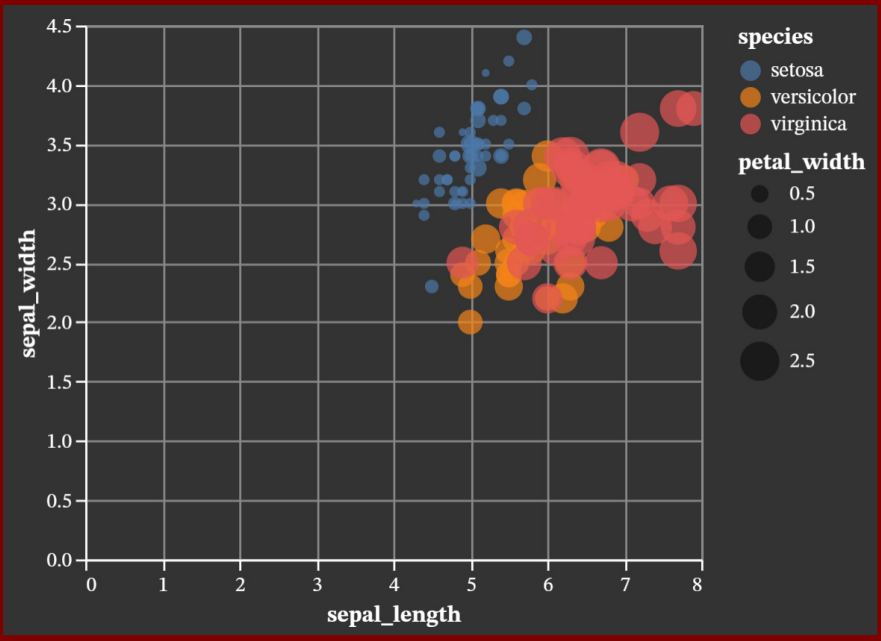
```
write('## Data Tables')
write(slides.block_r('Here is Table',
  textwrap.dedent('''
    |h1|h2|h3|
    |---|---|---|
    |d1|d2|d3|
    |r1|r2|r3|
    ''')))
```

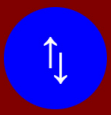
Writing Pandas DataFrame

	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





```
try:
  import pandas as pd
  import altair as alt
  alt.themes.enable('dark')
  df = pd.read_csv('https://raw.githubusercontent.com/mwaskom
  chart = alt.Chart(df,width=300,height=260).mark_circle(size
    x='sepal_length',
    y='sepal_width',
    color='species',
    size = 'petal_width',
    tooltip=['species', 'sepal_length', 'sepal_width','petal_
    ).interactive()
  df = df.describe() #Small for display
except:
  df = '### Install `pandas` to view output'
  chart = '### Install Altair to see chart'
write(('## Writing Pandas DataFrame',df),
  ('## Writing Altair Chart\nMay not work everywhere, needs
```

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 10

Check out this app

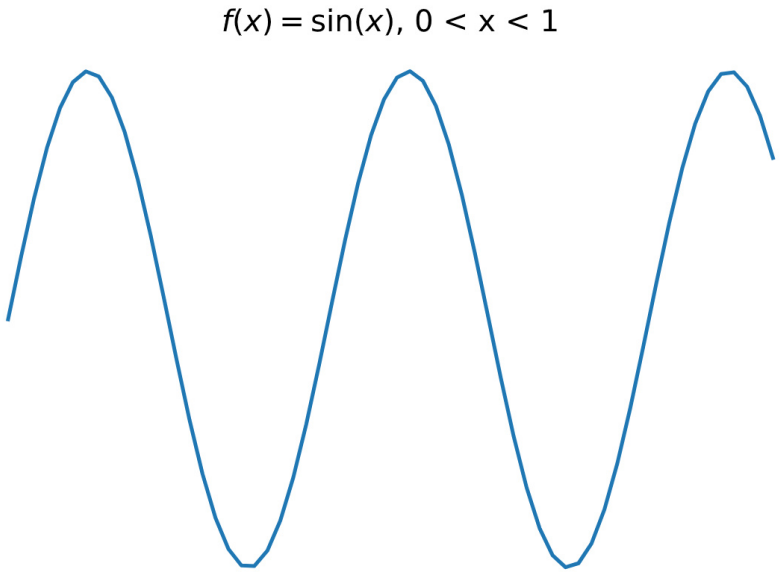
```
import ipywidgets as ipw
btn = ipw.Button(description='Click Me To see Progress',layou
prog = ipw.IntProgress(value=10)
html = ihtml(f"Current Value is {prog.value}")
def onclick(btn):
  prog.value = prog.value + 10
  if prog.value > 90:
    prog.value = 0
  html.value = f"Current Value is {prog.value}"

btn.on_click(onclick)

write('## Interactive Apps on Slide\n Use `ipywidgets`, `bqp
iwrite(prog,[btn,html])
write("[Check out this app](https://massgh.github.io/pivotpy
```

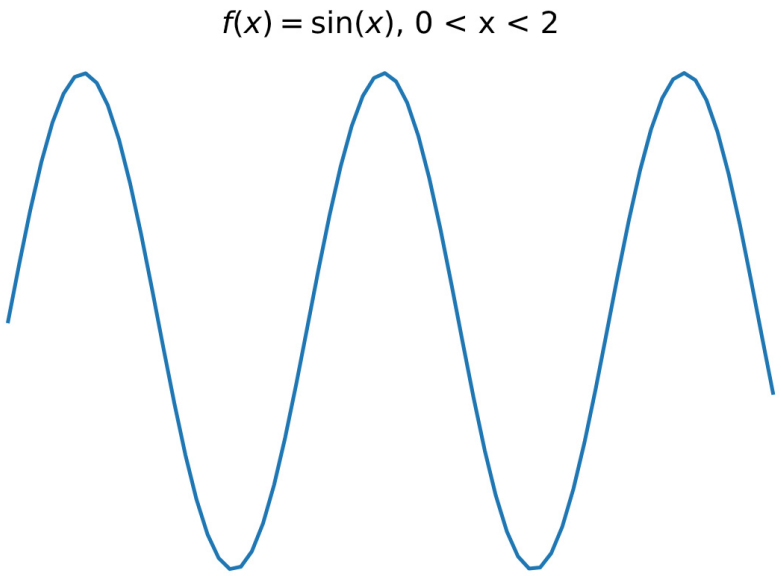
This is Slide 14.1

and we are animating matplotlib



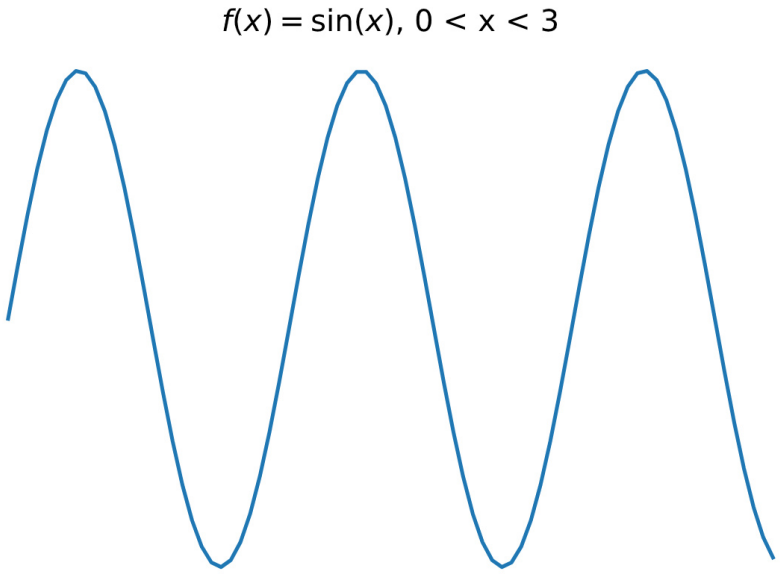
This is Slide 14.2

and we are animating matplotlib



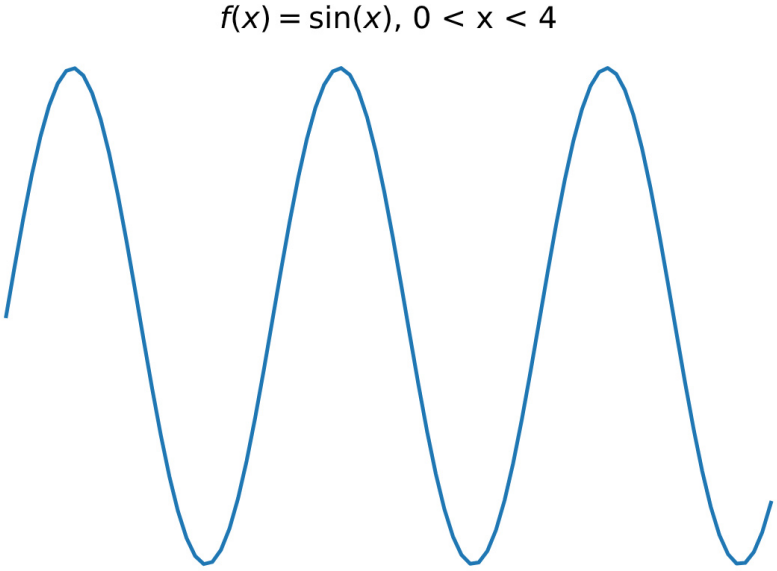
This is Slide 14.3

and we are animating matplotlib



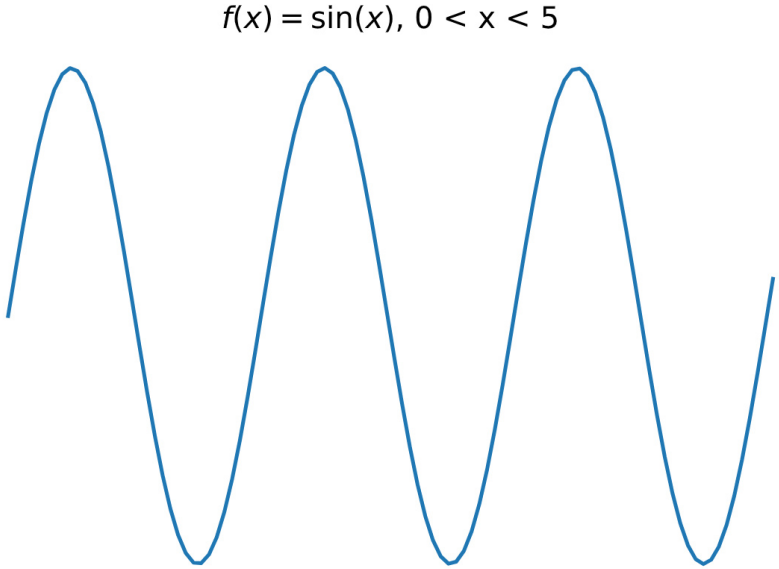
This is Slide 14.4

and we are animating matplotlib



This is Slide 14.5

and we are animating matplotlib



This is Slide 15 added with

enum_slides

This is Slide 16 added with

enum_slides

This is all code to generate slides

```
#Author: Abdul Saboor
# This demonstrates that you can generate slides from a .py
import textwrap
from .core import LiveSlides
from .utils import write, ihtml,plt2html, iwrite, __reprs__, tex
from .objs_formatter import libraries
slides = LiveSlides()
slides.convert2slides(True)
slides.set_footer('Author: Abdul Saboor عبدالصبور')
slides.set_logo('''<svg viewBox="0 0 100 100" xmlns="http://
        <circle cx="50" cy="50" r="50" fill="blue"/>
        <text x="45%" y="45%" fill="white" font-size="4em" d
        <text x="55%" y="60%" fill="white" font-size="4em" d

#title is skipped to show instructions
with slides.slide(1): #slide 1
  with slides.source():
    write('## I am created using `with slides.slide(1)` cont
    write(f'I am {slides.alert("Alerted")} and I am *{slides
slides.shell.user_ns['write'] = write #Inject variable in IPy

#slide 2
slides.shell.run_cell_magic('slide','2','write("## I am crea
```

```
#slide 3
online_sources = '''# IPySlides Online Running Sources
Launch as voila slides (may not work as expected [^1])[![Bin
[Edit on Kaggle](https://www.kaggle.com/massgh/ipyslides)
Launch example Notebook [![Binder](https://mybinder.org/badg
<br>
[^1]: Add references like this per slide. Use slides.cite()
'''
@slides.frames(3,'## I am created using `@slides.frames`',on
def func(obj):
  slides.write(obj)
  with slides.source():
    slides.write('------ Above text generated by this!------
    slides.write(slides.keep_format(obj))

#Now generate many slides in a loop
__contents = [f"""## IPython Display Objects
#### Any object with following methods could be in`write` co
{', '.join([f'`_repr_{rep}_`' for rep in __reprs__])}
Such as `IPython.display.<HTML,SVG,Markdown,Code>` etc. or t
""",
f"""## Plots and Other Data Types
#### These objects are implemented to be writable in `write`
{', '.join([f"`{lib['name']}.{lib['obj']}`" for lib in librari
Many will be extentended in future. If an object is not impl
command to show in Notebook outside `write`.
""",
f"""## Interactive Widgets
### Any object in `ipywidgets`{textbox('<a href="https://ipy
or libraries based on ipywidgtes such as `bqplot`,`ipyvolume
```

```
can be included in `iwrite` command. Text/Markdown/HTML insi
""",
'## Commands which do all Magic!']
for i in range(4,8):
  with slides.slide(i, background=f'linear-gradient(to right,
    write(__contents[i-4])
    if i == 7:
      with slides.source():
        write(slides.block_r('slides.write/ipyslide.utils.wr
           slides.rows(slides.block_b('slides.iwrite/ipyslide
            slides.block_b('slides.ihtml/ipyslide.utils.ihtml
            )
          )
        write("#### If an object does not render as you want

# Matplotlib
with slides.slide(8):
  with slides.source():
    import numpy as np, matplotlib.pyplot as plt
    x = np.linspace(0,2*np.pi)
    with plt.style.context('ggplot'):
      fig, ax = plt.subplots(figsize=(3.4,2.6))
      _ = ax.plot(x,np.cos(x))

    write('## Plotting with Matplotlib')
    write(slides.block_g('Matplotlib inside block!',slides.a

# Youtube
from IPython.display import YouTubeVideo
with slides.slide(9):
```

```
  with slides.source():
    write(f"### Watching Youtube Video?")
    write(YouTubeVideo('Z3iR551KgpI',width='100%',height='26

# Data Table
with slides.slide(10):
  with slides.source():
    write('## Data Tables')
    write(slides.block_r('Here is Table',
      textwrap.dedent('''
            |h1|h2|h3|
            |---|---|---|
            |d1|d2|d3|
            |r1|r2|r3|
            ''')))

# Plotly and Pandas DataFrame only show if you have installe
with slides.slide(11,background='#800000'):
  with slides.source():
    try:
      import pandas as pd
      import altair as alt
      alt.themes.enable('dark')
      df = pd.read_csv('https://raw.githubusercontent.com/mwa
      chart = alt.Chart(df,width=300,height=260).mark_circle(
        x='sepal_length',
        y='sepal_width',
        color='species',
        size = 'petal_width',
        tooltip=['species', 'sepal_length', 'sepal_width','pe
```

```
114         ).interactive()
115       df = df.describe() #Small for display
116     except:
117       df = '### Install `pandas` to view output'
118       chart = '### Install Altair to see chart'
119     write(('## Writing Pandas DataFrame',df),
120       ('## Writing Altair Chart\nMay not work everywhere, ne
121       )
122
123 try:
124   import plotly.graph_objects as go
125   fig = go.Figure()
126   fig.add_trace(go.Bar([1,5,8,9]))
127 except:
128   fig = '### Install `plotly` to view output'
129 with slides.slide(12):
130   write(('## Writing Plotly Figure',fig))
131
132 # Interactive widgets can't be used in write command, but st
133
134 with slides.slide(13):
135   with slides.source():
136     import ipywidgets as ipw
137     btn = ipw.Button(description='Click Me To see Progress',l
138     prog = ipw.IntProgress(value=10)
139     html = ihtml(f"Current Value is {prog.value}")
140     def onclick(btn):
141       prog.value = prog.value + 10
142       if prog.value > 90:
143         prog.value = 0
```

```
144         html.value = f"Current Value is {prog.value}"
145
146     btn.on_click(onclick)
147
148     write('## Interactive Apps on Slide\n Use `ipywidgets`,
149     iwrite(prog,[btn,html])
150     write("[Check out this app](https://massgh.github.io/piv
151
152 # Animat plot in slides
153 @slides.frames(14,*range(14,19))
154 def func(obj):
155   fig, ax = plt.subplots()
156   x = np.linspace(0,obj+1,50+10*(obj - 13))
157   ax.plot(x,np.sin(x));
158   ax.set_title(f'$f(x)=\sin(x)$, 0 < x < {obj - 13}')
159   ax.set_axis_off()
160   slides.write(f'### This is Slide {14}.{obj-13}\n and we ar
161
162 # Use enumerate to iterate over slides
163 for i,s in slides.enum_slides(15,17,background='var(--secondar
164   with s:
165     write(f'### This is Slide {i} added with `enum_slides`')
```

```python
def demo():
  import os
  from . import _demo, utils

  slides = _demo.slides
  with slides.slide(100):
    with slides.source():
      write('## This is all code to generate slides')
      write(_demo)
      write(demo)
  with slides.slide(101,background='#9ACD32'):
    with slides.source():
      slides.write_citations()

  slides.prog_slider.value = 0 # back to title
  return slides
```

```python
write('## This is all code to generate slides')
write(_demo)
write(demo)
```

References

[1]This is refernce to FigureWidget using `slides.cite` command

```
slides.write_citations()
```