IPySlides

▶ Show Code

Creating Slides

Abdul Saboor[1], Unknown Author[2]

Dec 04, 2022

[1]My University is somewhere in the middle of nowhere
[2]Their University is somewhere in the middle of nowhere

👈🏻 Read instructions in left panel

IPySlides

Home End

Table of Contents

s

r^2

here
where

anel

IPySlides

Table of Contents

IPySlides

▶ Show Code

Introduction

To see how commands work, use `Slides.docs()` to see the documentation. Here we will focus on using all that functionality to create slides.

ⓘ This is inline markdown parsed by magic

Version: 2.1.7 as executed from below code in markdown.

Python

```python
# get the slides instance under a python block in Markdown file, we will use it la
myslides = get_slides_instance()
import ipyslides as isd
version = isd.__version__
%xmd #### This is inline markdown parsed by magic {.Note .Warning}
```

I was added at end using `s2.insert_markdown`

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() or in markdown cite`key` to add citations generally. ↩

IPySlides

Table of Contents

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`, `IPython.display.Image` 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 as well as other objects that can be passed to `write` with caveat of Javascript.

IPySlides

Commands which do all Magic!

Slides.**write**(*columns, width_percents=None, className=None)

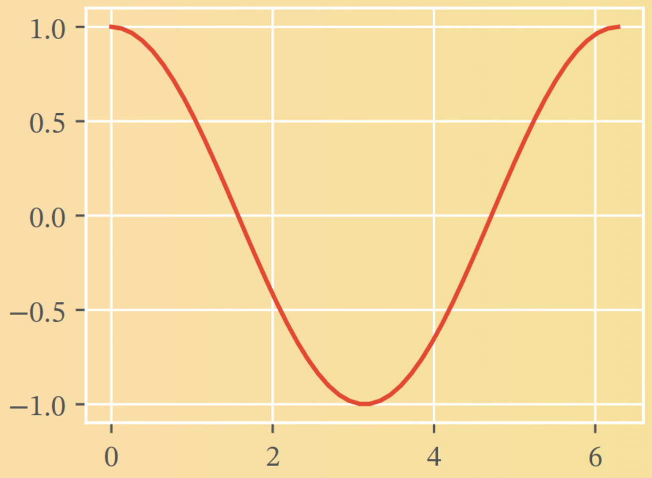
Writes markdown strings or IPython object with method `_repr_<html,svg,png, ... >_` in each column of same with. If width_percents is given, column width is adjusted. Each column should be a valid object (text/markdown/html/ have *repr or to* method) or list/tuple of objects to form rows or explictly call `rows`.

- Pass int,float,dict,function etc. Pass list/tuple in a wrapped list for correct print as they used for rows writing too.
- Give a code object from `Slides.source.context[from_ ... ]` to it, syntax highlight is enabled.
- Give a matplotlib `figure/Axes` to it or use `ipyslides.objs_formatter.plt2html()`.
- Give an interactive plotly figure.
- Give a pandas dataframe `df` or `df.to_html()`.
- Give any object which has `to_html` method like Altair chart. (Note that chart will not remain interactive, use display(chart) if need interactivity like brushing etc.)

IPySlides

Table of Contents

Plotting with Matplotlib



Python

```python
import numpy as np, matplotlib.pyplot as plt
plt.rcParams['svg.fonttype'] = 'none' # Global setting, enforce same fonts as pre
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([ax, s.focus_lines([1,3,4])])
```

IPySlides

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

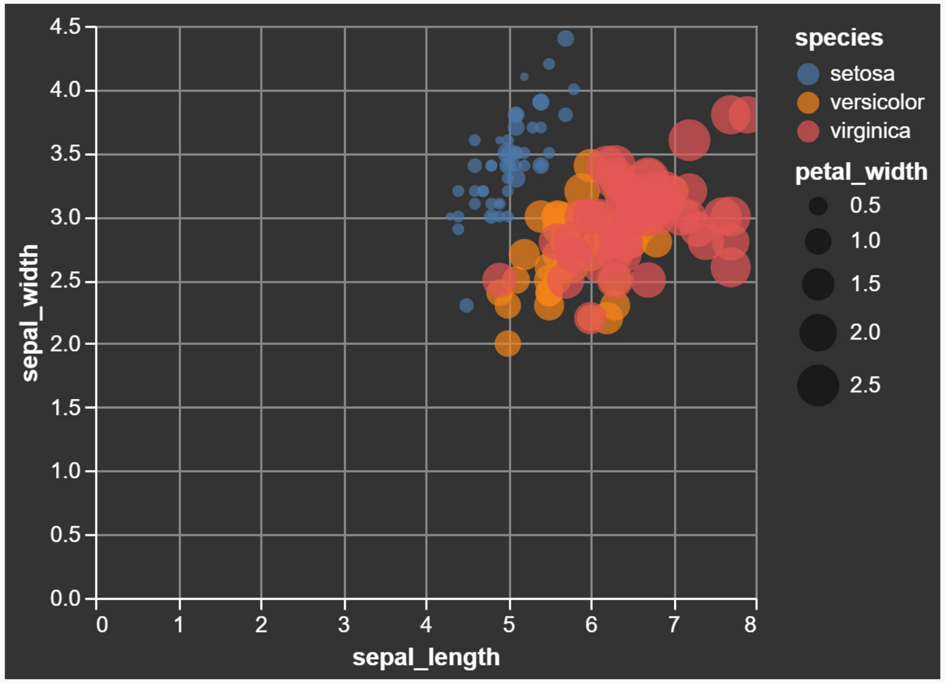
Python

```python
import pandas as pd
 + 2 more lines ...
df = pd.read_csv('https://raw.githubusercontent.com/mwaskom/seaborn-data/master/i
 + 7 more lines
```

IPySlides

Writing Altair Chart

ⓘ May not work everywhere, needs javascript

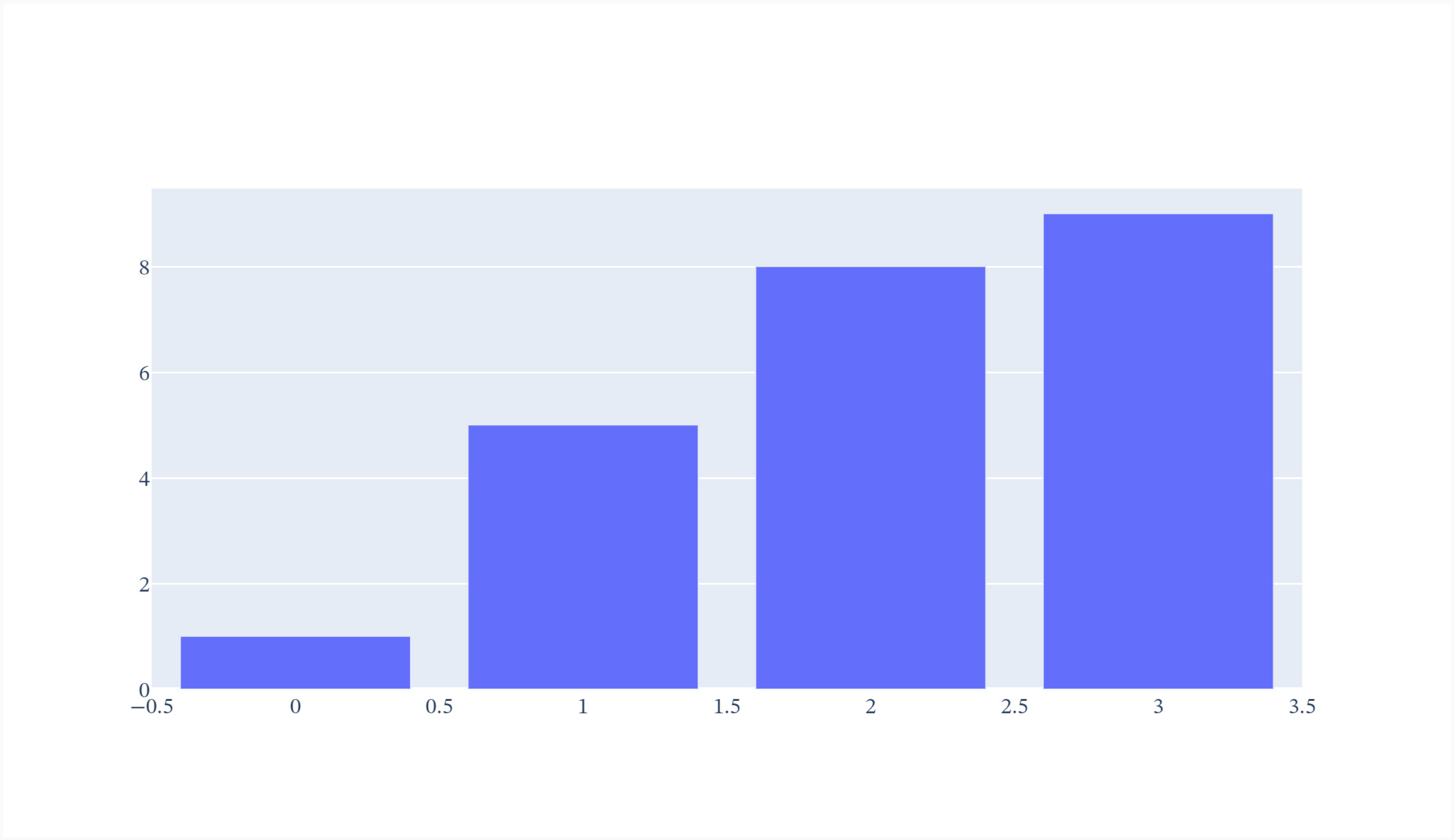


Python

```
 + 1 more lines ...
import altair as alt
alt.themes.enable('dark')
df = pd.read_csv('https://raw.githubusercontent.com/mwaskom/seaborn-data/master/i
chart = alt.Chart(df,width=300,height=260).mark_circle(size=60).encode(
```

IPySlides

Writing Plotly Figure



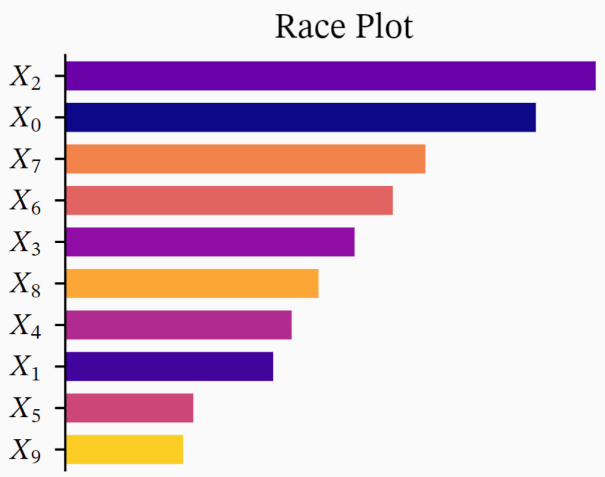
Python

```
import plotly.graph_objects as go
fig = go.Figure()
fig.add_trace(go.Bar(y=[1,5,8,9]))
```

IPySlides

Interactive Apps on Slide

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



Click me to update race plot

Check out this app

Python

```
import ipywidgets as ipw
import numpy as np, matplotlib.py

write('## Interactive Apps on color
writer, (plot,button, _), code = s
        '## Plot will be here! Click b
        ipw.Button(description='Click
        "[Check out this app](https://

def update_plot():
        x = np.linspace(0,0.9,10)
        y = np.random.random((10,))
        _sort = np.argsort(y)

        fig,ax = plt.subplots(figsize
```

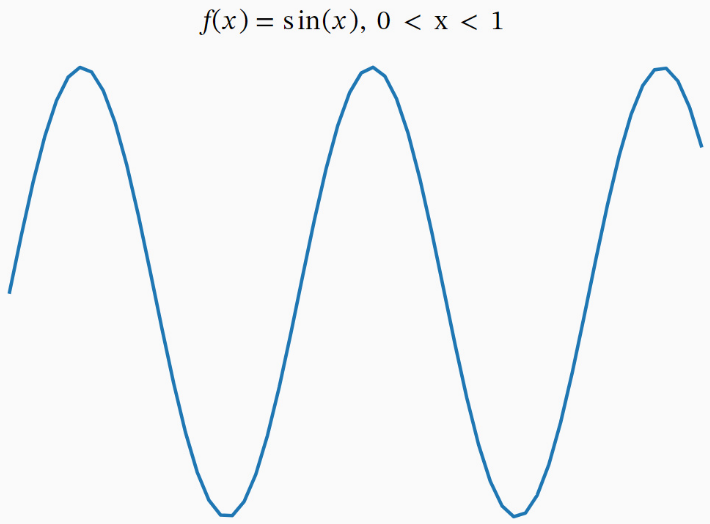
IPySlides

This is Slide 15.0

and we are animating matplotlib

Python

```python
fig, ax = plt.subplots()
 + 6 more lines ...
```



Python

```python
 + 5 more lines ...
slides.notes.insert(f'## This is under @frames decorator!')
slides.notify_later()(lambda: f'This is under @frames decorator!')
```

2

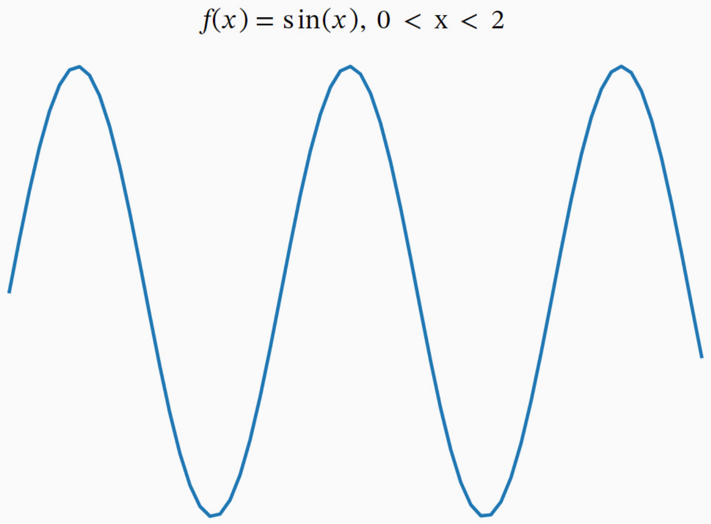
IPySlides

This is Slide 15.1

and we are animating matplotlib

Python

```
  + 1 more lines ...
x = np.linspace(0,obj+1,50+
  + 5 more lines ...
```



2

IPySlides

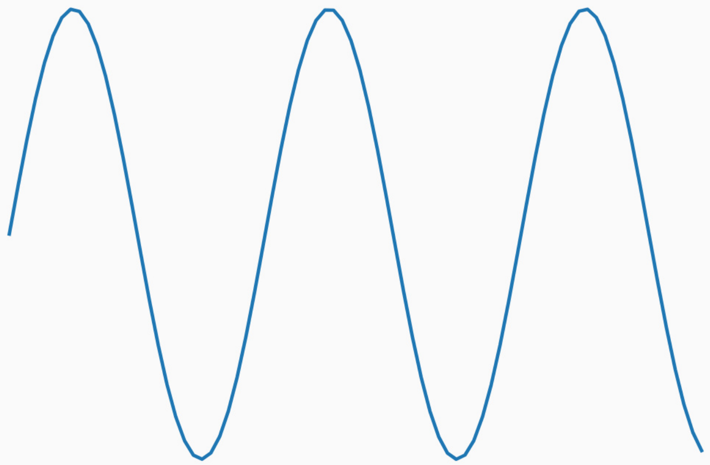
This is Slide 15.2

and we are animating matplotlib

Python

```
  + 2 more lines ...
ax.plot(x, np.sin(x));
  + 4 more lines ...
```

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



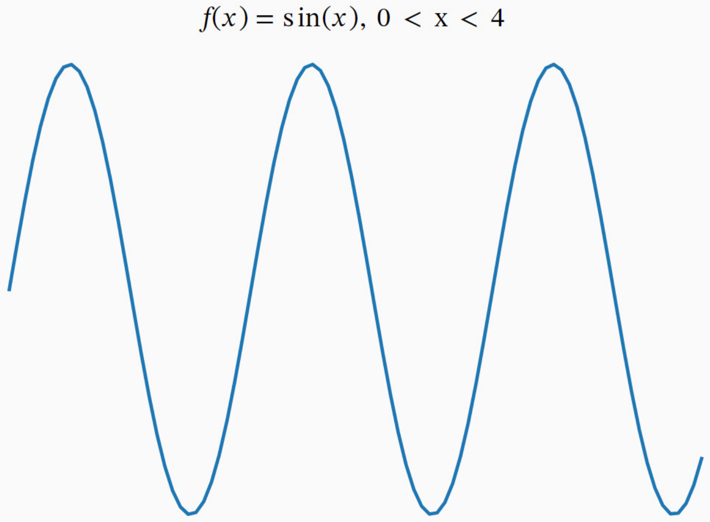
2

This is Slide 15.3

and we are animating matplotlib

Python

```
 + 3 more lines ...
ax.set_title(f'$f(x)=\sin(x)
 + 3 more lines ...
```



2

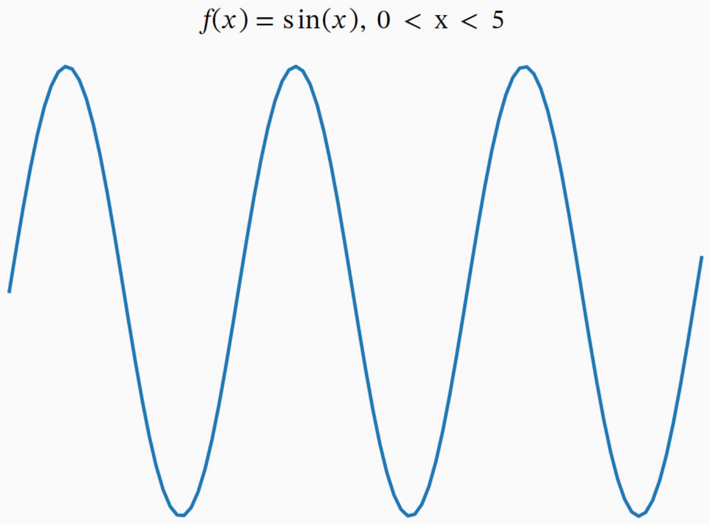
IPySlides

This is Slide 15.4

and we are animating matplotlib

Python

```
 + 4 more lines ...
ax.set_axis_off()
 + 2 more lines ...
```



2

IPySlides

Table of Contents

IPySlides

Frames with

`repeat = False`

1

IPySlides

Frames with

`repeat = False`

2

IPySlides

Frames with

`repeat = False`

3

IPySlides

Frames with `repeat = False`

4

IPySlides

Frames with

`repeat = True` and Fancy Bullet List

💘 1

IPySlides

Frames with
`repeat = True` and Fancy Bullet List

- 💘 **1**
- 💘 **2**

Frames with

`repeat = True` and Fancy Bullet List

- 💘 **1**
- 💘 **2**
- 💘 **3**

IPySlides

Frames with

`repeat = True` and Fancy Bullet List

- 💘 1
- 💘 2
- 💘 3
- 💘 4

IPySlides

Frames with

`repeat = [(0,1),(2,3)]`

1	2

Python

```python
slides.write('# Frames with \n#### `repeat = [(0,1),(2,3)]`')
slides.write(*obj)
```

IPySlides

Frames with

`repeat = [(0,1),(2,3)]`

3

4

Python

```python
slides.write('# Frames with \n#### `repeat = [(0,1),(2,3)]`')
slides.write(*obj)
```

IPySlides

Displaying image from url from somewhere in Kashmir (کشمیر)

Python

```
slides.goto_button(slides.running.number - 5, 'Skip All Previous Frames')
slides.write('## Displaying image from url from somewhere in Kashmir color[crimsor
```

IPySlides

Watching Youtube Video?

Python

```python
write(f"### Watching Youtube Video?")
write(YouTubeVideo('Z3iR551KgpI',width='100%',height='266px'))
@slides.notify_later()
def push():
        t = time.localtime()
        return f'You are watching Youtube at Time-{t.tm_hour:02}:{t.tm_min:02}'

s.display()  # s = source.context(style='vs', className="Youtube")
```

IPySlides

Data Tables

Here is Table

h1	h2	h3
d1	d2	d3
r1	r2	r3

Python

```python
write('## Data Tables')
# Remember myslides variable was assigned in a python block
# in markdown just in start. Magic!
write(myslides.block_r('Here is Table','<hr/>','''
        |h1|h2|h3|
        |---|---|---|
        |d1|d2|d3|
        |r1|r2|r3|
        '''))
```

$\LaTeX$ in Slides

ⓘ Use `$ $` or `$$ $$` to display latex in Markdown, or embed images of equations $\LaTeX$ needs time to load, so keeping it in view until it loads would help.

$$\int_0^1\frac{1}{1-x^2}dx$$

$$\int_0^1\frac{1}{1-x^2}dx$$

Built-in CSS styles

Python

```python
slides.css_styles.display()
slides.write('Info',className='Info')
slides.write('Warning',className='Warning')
slides.write('سارے جہاں میں دھوم ہماری زباں کی ہے۔',className='Right RTL')
```

```
Use any or combinations of these styles in className argument of writing functions:
------------------------------------------------------------------------------------
 className          | Formatting Style
====================================================================================
 'Center'           | ------Text------
 'Left'             | Text------------
 'Right'            | ------------Text
 'RTL'              | ------ اردو عربی
 'Info'             | Blue Text
 'Warning'          | Orange Text
 'Success'          | Green Text
 'Error'            | Red Text
```

IPySlides

Python

```python
slides.rows(
        '## Can skip `write` commnad sometimes',
        slides.cols('### Column A','### Column B',className='Info'),
        '||### Column C {.Warning}||### Column D {.Success}||',
).display()
slides.write('----')  # In Python < 3.8, context manager does not properly handle e
```

Can skip `write` commnad sometimes

Column A

Column B

Column C

Column D

IPySlides

Table of Contents

Serialize Custom Objects to HTML

This is useful for displaying user defined/third party objects in slides

0 1 2 3 4 5 6 7 8 9

Python

```python
@slides.serializer.register(int)
def colorize(obj):
    color = 'red' if obj % 2 == 0 else 'green'
    return f'<span style="color:{color};">{obj}</span>'

slides.write(*range(10))
```

This is all code to generate slides

e:\research\ipyslides\ipyslides_demo.py

```python
# Author: Abdul Saboor
# This demonstrates that you can generate slides from a .py file too, which you ca
import time, textwrap

from ipyslides.writers import write, iwrite
from ipyslides.formatters import libraries, __reprs__
from ipyslides._base.intro import logo_svg

markdown_str = """# Creating Slides
class`Center`
alert`Abdul Saboor`sup`1`, Unknown Authorsup`2`

today``

class`TextBox`
```

ⓘ Slides made by using `from_markdown` or `%%slide` magic preserve their full code

IPySlides

Source Code

Markdown: Slide 0

```
# Creating Slides
class`Center`
alert`Abdul Saboor`sup`1`, Unknown Authorsup`2`

today``

class`TextBox`
sup`1`My University is somewhere in the middle of nowhere
sup`2`Their University is somewhere in the middle of nowhere
^^^
^^^
<h4 style=""color:green;"> 👈🏻 Read instructions in left pane
```

Markdown: Slide 1

```
section`Introduction`
```

Markdown: Slide 2

Python

```
slides.write('citations`## Reference via Markdown\n----`',
                   ['## Reference via Python API\n----',
                    *slides.citations])
slides.write('Markdown is easier to write and read, but Python API is more powerfu
```

Reference via Markdown

1. This is refernce to FigureWidget using `slides.cite` command
2. Set citation for key 'This'.

Reference via Python API

1. This is refernce to FigureWidget using `slides.cite` command
2. Set citation for key 'This'.

Markdown is easier to write and read, but Python API is more powerful.