Creating Slides

Abdul Saboor[1], Unknown Author[2]

Feb 21, 2024

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

Read instructions by clicking **i** button in quick menu.

Contents

1. Introduction
2. Variety of Content Types to Display
3. Plotting and DataFrame
4. Interactive Widgets
5. Simple Animations with Frames
6. Controlling Content on Frames
7. Miscellaneous Content
8. Custom Objects Serilaization
9. Code to Generate Slides

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.

> 📝 Note
>
> This is inline markdown parsed by magic

Version: 3.6.9 as executed from below code in markdown.

Python

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

IPySlides Online Running Sources

Note

- Edit on Kaggle
- Launch example Notebook launch binder
- Watch a Youtube Video

1. Add references like this per slide or citekey to add citations generally. ↩

Contents

1. Introduction
2. Variety of Content Types to Display
3. Plotting and DataFrame
4. Interactive Widgets
5. Simple Animations with Frames
6. Controlling Content on Frames
7. Miscellaneous Content
8. Custom Objects Serilaization
9. Code to Generate Slides

This is summary for current section created using block syntax of toc. See `Slides.xmd_syntax` for details.

- Item 1
- Item 2

$$E = mc^2$$

Markdown

```
```toc Table of contents
Extra content for current section which is on right
```
```

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 as well.

1. This is refernce to FigureWidget using citep`f`syntax

Commands which do all Magic!

Slides.**write**(*objs, widths=None)

Write `objs` to slides in columns. To create rows in a column, wrap objects in a list or tuple. You can optionally specify `widths` as a list of percentages for each column.

Write any object that can be displayed in a cell with some additional features:

- Strings will be parsed as as extended markdown that can have citations/python code blocks/Javascript etc.
- Display another function in order by passing it to a lambda function like `lambda: func()`. Only body of the function will be displayed/printed. Return value will be ignored.
- Dispaly IPython widgets such as `ipywidgets` or `ipyvolume` by passing them directly.
- Display Axes/Figure form libraries such as `matplotlib`, `plotly` `altair`, `bokeh`, `ipyvolume` ect. by passing them directly.
- Display source code of functions/classes/modules or other languages by passing them directly or using `Slides.code` API.
- Use `Slides.alt(widget, func)` function to display widget on slides and alternative content

Contents

1. Introduction
2. Variety of Content Types to Display
3. **Plotting and DataFrame**
4. Interactive Widgets
5. Simple Animations with Frames
6. Controlling Content on Frames
7. Miscellaneous Content
8. Custom Objects Serilaization
9. Code to Generate Slides

Plotting with Matplotlib

Python

```python
sl.set_css({'background':'linear-gradie
import numpy as np, matplotlib.pyplot a
plt.rcParams['svg.fonttype'] = 'none' #
x = np.linspace(0,2*np.pi)
with plt.style.context('ggplot'):
    fig, ax = plt.subplots(figsize=(3.4
    _ = ax.plot(x,np.cos(x))
write(ax, s.focus_lines([0,3,4]))
```

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
try:
    import pandas as pd
    df = pd.read_csv('https://raw.githubusercontent.com/mwaskom/seaborn-data/master/
    df = df.describe() #Small for display
except:
    df = '### Install `pandas` to view output'
```

Writing Plotly Figure

Install `plotly` to view output

Python

```python
try:
    import plotly.graph_objects as go
    fig = go.Figure()
    fig.add_trace(go.Bar(y=[1,5,8,9]))
except:
    fig = '### Install `plotly` to view output'
```

Interactive Apps with Widgets

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

> 💡 Tip
>
> Export to Slides to see what happens to this slide and next slide!

A Silly Plot

Python

```python
import ipywidgets as ipw

write('''
    ## Interactive Apps with Widgets se
    Use `ipywidgets`, `bqplot`,`ipyvolu
     ::: note-tip
         Export to Slides to see what ha
    ''')
plot_html = ipw.HTML('Plot will be here
```

Dynamic Content without Widgets

Use refresh button below to update plot! Compare with previous slide!

A Silly Plot

Python

```python
write('''
    ## Dynamic Content without Widgets
    Use refresh button below to update p
    ''')

def display_plot(): return race_plot().

write(lambda: slides.on_refresh(display_
slides.code.cast(race_plot).display()
```

Python

```python
def race_plot():
    import numpy as np
```

Contents

1. Introduction
2. Variety of Content Types to Display
3. Plotting and DataFrame
4. Interactive Widgets
5. **Simple Animations with Frames**
6. Controlling Content on Frames
7. Miscellaneous Content
8. Custom Objects Serilaization
9. Code to Generate Slides

Skip All Next Frames

This is Slide 16.0

and we are animating matplotlib

Python

```
fig, ax = plt.subplots()
 + 5 more lines ...
```

1

1. I was cited for no reason

This is Slide 16.1

and we are animating matplotlib

Python

```
 + 1 more lines ...
x = np.linspace(0,obj+1,50+10*(
 + 4 more lines ...
```

[1]

1. I was cited for no reason

This is Slide 16.2

and we are animating matplotlib

Python

```
 + 2 more lines ...
ax.plot(x,np.sin(x));
 + 3 more lines ...
```

[1]

1. I was cited for no reason

This is Slide 16.3

and we are animating matplotlib

Python

```python
 + 3 more lines ...
ax.set_title(f'$f(x)=\sin(x)$,
 + 2 more lines ...
```

1

1. I was cited for no reason

This is Slide 16.4

and we are animating matplotlib

Python

```python
 + 4 more lines ...
ax.set_axis_off()
 + 1 more lines ...
```

[1]

1. I was cited for no reason

Contents

1. Introduction
2. Variety of Content Types to Display
3. Plotting and DataFrame
4. Interactive Widgets
5. Simple Animations with Frames
6. **Controlling Content on Frames**
7. Miscellaneous Content
8. Custom Objects Serilaization
9. Code to Generate Slides

Frames with

```
repeat = False
```

```
1
```

Frames with

```
repeat = False
```

```
2
```

Frames with

```
repeat = False
```

```
3
```

Frames with

```
repeat = False
```

```
4
```

Frames with

`repeat = True` and Fancy Bullet List

1

Frames with

`repeat = True` and Fancy Bullet List

- 1
- 2

Frames with

`repeat = True` and Fancy Bullet List

- 1
- 2
- 3

Frames with

`repeat = True` and Fancy Bullet List

- 1
- 2
- 3
- 4

Frames with

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

1

2

Python

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

Frames with

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

3	4

Python

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

Skip Previous Frames

Displaying image from url from somewhere in Kashmir

(کشمیر)

Watching Youtube Video?

Want to do some drawing instead? Click on pencil icon and draw something on tldraw!

Python

```
write(f"### Watching Youtube Video?")
write('**Want to do some drawing instead?**\nClick on pencil icon and draw something
```

Block API

New `block` API is as robust as `write` command. On top of it, it makes single unit of related content.

Table

h1	h2	h3
d1	d2	d3
r1	r2	r3

Widgets

Click to do nothing

Python

```python
write('## Block API\nNew `block` API is as robust as `write` command. On top of it,
slides.block_red(
    [
        '### Table',
        '''
        |h1 |h2 |h3 |
        |   |   |   |
```

$\LaTeX$ 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.

> 💡 Tip
>
> Varibale formatting alongwith $\LaTeX$ {var}`→ I was a variable is seamless.

$\LaTeX$ 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.

> 💡 Tip
>
> Varibale formatting alongwith $\LaTeX$ {var}`→ I was a variable is seamless.

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

$\LaTeX$ 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.

> 💡 Tip
>
> Varibale formatting alongwith $\LaTeX$ {var}`→ I was a variable is seamless.

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

$$ax^2 + bx + c = 0$$

Python

```python
slides.write(fmt('## Built-in CSS styles\n`{s.css_styles}`',s=slides))
```

Built-in CSS styles

Use any or combinations of these styles in className argument of writing functions:

className	Formatting Style
'text-[value]'	[value] should be one of tiny, small, big, large, huge.
'align-[value]'	[value] should be one of center, left, right.
'rtl'	——— اردو عربی
'info'	Blue text. Icon ℹ️ for note-info class.
'tip'	Blue Text. Icon 💡 for note-tip class.
'warning'	Orange Text. Icon ⚠️ for note-warning class.
'success'	Green text. Icon ✅ for note-success class.
'error'	Red Text. Icon ⚡ for note-error class.
'note'	📝 Text with note icon.
'export-only'	Hidden on main slides, but will appear in exported slides.
'jupyter-only'	Hidden on exported slides, but will appear on main slides.
'block'	Block of text/objects

Contents

1. Introduction
2. Variety of Content Types to Display
3. Plotting and DataFrame
4. Interactive Widgets
5. Simple Animations with Frames
6. Controlling Content on Frames
7. Miscellaneous Content
8. **Custom Objects Serilaization**
9. Code to Generate Slides

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.write('## Serialize Custom Objects to HTML\nThis is useful for displaying use
with slides.suppress_stdout(): # suppress stdout from register fuction below
    @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))

some_slide.get_source().display()
```

This is all code to generate slides

Python

```python
def demo(self):
    "Demo slides with a variety of content."
    from .._demo import demo_slides
    return demo_slides(self)
```

e:\development\ipyslides\ipyslides_demo.py

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

from ipyslides.formatters import libraries, supported_reprs
from ipyslides.xmd import fmt

def demo_slides(slides):
    slides.close_view() # Close any previous view to speed up loading 10x faster on
    slides.clear() # Clear previous content
```

> 📝 Note
> Slides keep their full code if they are not made by @frames decorator!

Source Code

Markdown: Slide 0

```
# Creating Slides
::: align-center
    alert`Abdul Saboor`sup`1`, Unknown Authorsup`2`
    center`today```
    ::: text-box
        sup`1`My University is somewhere in the middle of nowhere
        sup`2`Their University is somewhere in the middle of nowhere

Read instructions by clicking ℹ button in quick menu.
```

Markdown: Slide 1

```
section`Introduction` toc`### Contents`
```