Creating Slides

Abdul Saboor[1], Unknown Author[2]

Nov 19, 2025

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

Read instructions by clicking on

ⓘ

or same button in quick menu

⌄ Show Code

Contents

Introduction

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

> ✴ **Info**
>
> This slide was built purely from markdown, so you can create a variable `test` to overwite this →
>
> ```
> Exception: Could not resolve '%{test}':
> NameError: name 'test' is not defined
> You can update this variable by `Slides[int,|list|slice].vars.update` or by defining it
> in notebook if `Auto Rebuild` is enabled.
> ```

Version: `6.5.6`

Author: Abdul Saboor عبدالصبور

Contents

This is summary for current section created using block syntax of toc. See `Slides.xmd.syntax` for details.
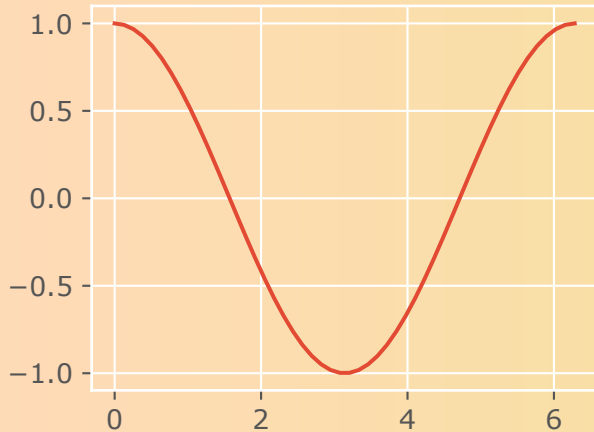
- Item 1
- Item 2

$$E = mc^2$$

✏️

> 💡 Tip
>
> Above `btn` variable can be updated later via `Slides[number,].vars.update` method.

Markdown

```
1   section`Adding informative TOC`
```

Author: Abdul Saboor عبدالصبور

Plotting with Matplotlib

```python
1  sl.set_css(bg1 = 'linear-gradient(to right, #FFDA
2
3  import numpy as np, matplotlib.pyplot as plt
4  plt.rcParams['svg.fonttype'] = 'none' # Global se
5  x = np.linspace(0,2*np.pi)
6  with plt.style.context('ggplot'):
7      fig, ax = plt.subplots(figsize=(3.4,2.6))
8      _ = ax.plot(x,np.cos(x))
9  slides.write(ax, s.focus([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/iris.csv')
```

Writing Plotly Figure

Install `plotly` to view output

```python
1   try:
2       import ipywidgets as ipw
3       import plotly.graph_objects as go
4
5       fig = slides.patched_plotly(go.FigureWidget()) # prefere Widget for interactivity and correct display
6       fig.add_trace(go.Bar(y=[1,5,8,9], customdata=["A","B"]))
7
8       # We have clicked and selected traits on patched plotly
9       html = ipw.HTML()
10
11      def observe_click(change):
12          html.value = "<br/>".join(f"  {k} = {v}" for k, v in change['new'].items())
13
14      fig.observe(observe_click, names='clicked')
```

Refreshable Content

Use refresh button below to update plot! See race_plot function at end of slides.

Race Plot
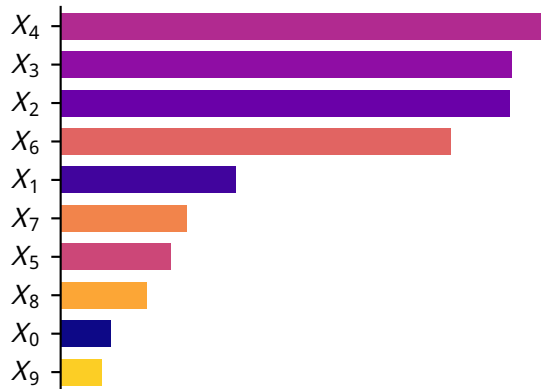
A Silly Plot

```python
1   slides.write('''
2       ## Refreshable Content
3       Use refresh button below to update plot!
4       See alert`race_plot` function at end of slide:
5   ''')
6
7   def display_plot(btn): return race_plot().display
8
9   slides.write(
10      slides.dl.interactive(display_plot, btn = sl:
11      rslide.get_source()
12  ) # Only first columns will update
```
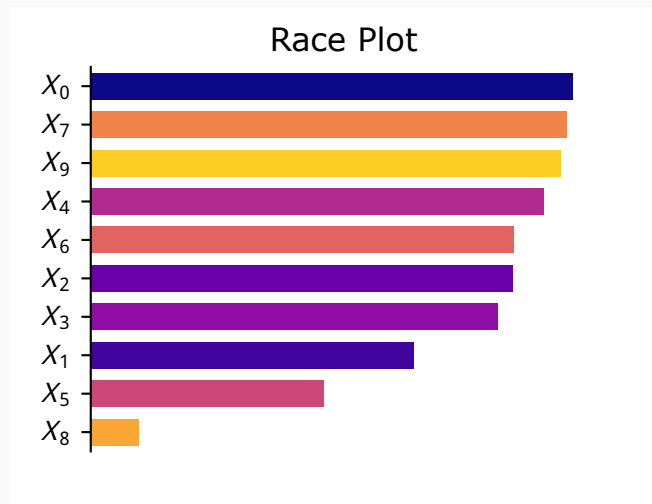
Python

⟳ Refresh Plot

🍥 Author: Abdul Saboor عبدالصبور

Animations with Widgets

anim ▶ ↺ ━━━━●━━━━ 7

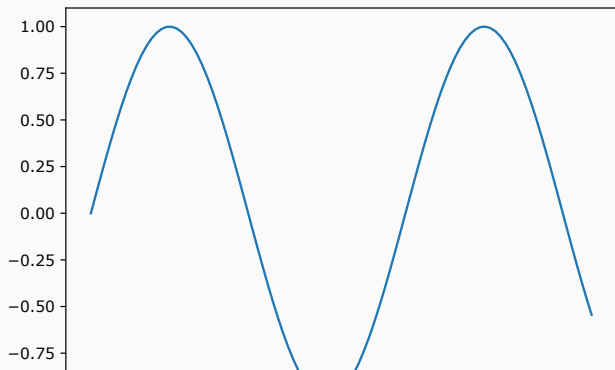`Animation Frame: 7`

A Silly Plot

Author: Abdul Saboor عبدالصبور

Rich Content `ListWidget`

Execute a code block

```python
lambda: print(np.random.random((10,2))),

lambda: plt.plot(np.random.random((10,2))),

def plot_sine():
    plt.plot(np.sin(np.linspace(0,10,100)))
```


Author: Abdul Saboor عبدالصبور

Python

```python
1   import numpy as np
2   import matplotlib.pyplot as plt
3
4   def plot_sine():
5       plt.plot(np.sin(np.linspace(0,10,100)))
6
7   lw = slides.ListWidget(description='Execute a cod
8       options = [
9           lambda: print(np.random.random((10,2))),
10          lambda: plt.plot(np.random.random((10,2)
11          plot_sine,
12      ], transform = lambda value: slides.code(val
13  )
14
15  def run(c):
16      if callable(c): c() # avoid None value when
17      plt.show()
18
19  css = {'.out-main': {'height':'300px'}, 'grid':'a
```

Contents

Animating Matplotlib!

$f(x) = \sin(x),\ 0 < x < 11$

Python

```python
for idx in slides.PAGE.iter(range(10,19)):
    fig, ax = plt.subplots(figsize=(3.4,2.6))
    x = np.linspace(0,idx,50)
    ax.plot(x,np.sin(x))
    ax.set_title(rf'$f(x)=\sin(x)$, 0 < x < {idx+1}')
    ax.set_xlim([0,18])
    ax.set_axis_off()
    slides.write(s.focus([idx - 10]),ax,widths=[60,40])

    if idx == 10:
        slides.write('Unlike `interact/interactive`, this a
```


💡 **Tip** Unlike `interact/interactive`, this animation is based on slide frames, all of which are exported to HTML.

🔷 Author: Abdul Saboor عبدالصبور

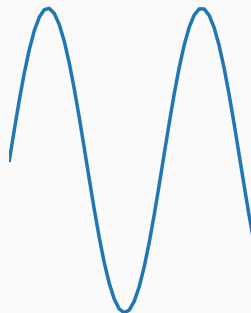# Animating Matplotlib!

Python

```python
for idx in slides.PAGE.iter(range(10,19)):
    fig, ax = plt.subplots(figsize=(3.4,2.6))
    x = np.linspace(0,idx,50)
    ax.plot(x,np.sin(x))
    ax.set_title(rf'$f(x)=\sin(x)$, 0 < x < {idx+1}')
    ax.set_xlim([0,18])
    ax.set_axis_off()
    slides.write(s.focus([idx - 10]),ax,widths=[60,40])

    if idx == 10:
        slides.write('Unlike `interact/interactive`, this a
```
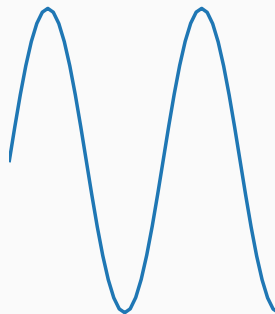
$f(x) = \sin(x), 0 < x < 12$

Animating Matplotlib!

Python

```python
1   for idx in slides.PAGE.iter(range(10,19)):
2       fig, ax = plt.subplots(figsize=(3.4,2.6))
3       x = np.linspace(0,idx,50)
4       ax.plot(x,np.sin(x))
5       ax.set_title(rf'$f(x)=\sin(x)$, 0 < x < {idx+1}')
6       ax.set_xlim([0,18])
7       ax.set_axis_off()
8       slides.write(s.focus([idx - 10]),ax,widths=[60,40])
9
10      if idx == 10:
11          slides.write('Unlike `interact/interactive`, this a
```
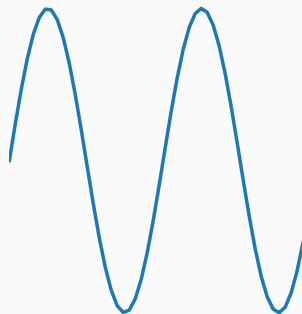
$f(x) = \sin(x),\ 0 < x < 13$

Author: Abdul Saboor عبدالصبور

11

Animating Matplotlib!

Python

```python
for idx in slides.PAGE.iter(range(10,19)):
    fig, ax = plt.subplots(figsize=(3.4,2.6))
    x = np.linspace(0,idx,50)
    ax.plot(x,np.sin(x))
    ax.set_title(rf'$f(x)=\sin(x)$, 0 < x < {idx+1}')
    ax.set_xlim([0,18])
    ax.set_axis_off()
    slides.write(s.focus([idx - 10]),ax,widths=[60,40])

    if idx == 10:
        slides.write('Unlike `interact/interactive`, this a
```

$f(x) = \sin(x), 0 < x < 14$

Author: Abdul Saboor عبدالصبور

11

Animating Matplotlib!

Python

```python
for idx in slides.PAGE.iter(range(10,19)):
    fig, ax = plt.subplots(figsize=(3.4,2.6))
    x = np.linspace(0,idx,50)
    ax.plot(x,np.sin(x))
    ax.set_title(rf'$f(x)=\sin(x)$, 0 < x < {idx+1}')
    ax.set_xlim([0,18])
    ax.set_axis_off()
    slides.write(s.focus([idx - 10]),ax,widths=[60,40])

    if idx == 10:
        slides.write('Unlike `interact/interactive`, this a
```
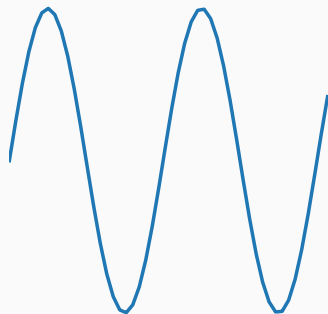
$f(x) = \sin(x),\ 0 < x < 15$

Author: Abdul Saboor عبدالصبور

11

Animating Matplotlib!

Python

```python
for idx in slides.PAGE.iter(range(10,19)):
    fig, ax = plt.subplots(figsize=(3.4,2.6))
    x = np.linspace(0,idx,50)
    ax.plot(x,np.sin(x))
    ax.set_title(rf'$f(x)=\sin(x)$, 0 < x < {idx+1}')
    ax.set_xlim([0,18])
    ax.set_axis_off()
    slides.write(s.focus([idx - 10]),ax,widths=[60,40])

    if idx == 10:
        slides.write('Unlike `interact/interactive`, this a
```
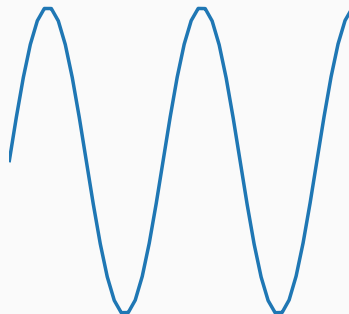
$f(x) = \sin(x),\ 0 < x < 16$

Author: Abdul Saboor عبدالصبور

11

Animating Matplotlib!

Python

```python
for idx in slides.PAGE.iter(range(10,19)):
    fig, ax = plt.subplots(figsize=(3.4,2.6))
    x = np.linspace(0,idx,50)
    ax.plot(x,np.sin(x))
    ax.set_title(rf'$f(x)=\sin(x)$, 0 < x < {idx+1}')
    ax.set_xlim([0,18])
    ax.set_axis_off()
    slides.write(s.focus([idx - 10]),ax,widths=[60,40])

    if idx == 10:
        slides.write('Unlike `interact/interactive`, this a
```
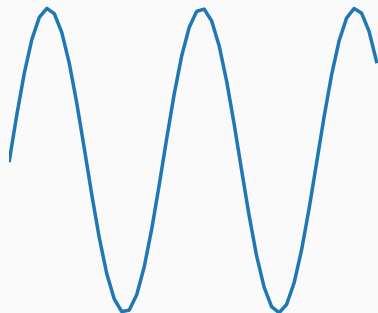
$f(x) = \sin(x), \; 0 < x < 17$

Author: Abdul Saboor عبدالصبور

11

Animating Matplotlib!

Python

```python
1  for idx in slides.PAGE.iter(range(10,19)):
2      fig, ax = plt.subplots(figsize=(3.4,2.6))
3      x = np.linspace(0,idx,50)
4      ax.plot(x,np.sin(x))
5      ax.set_title(rf'$f(x)=\sin(x)$, 0 < x < {idx+1}')
6      ax.set_xlim([0,18])
7      ax.set_axis_off()
8      slides.write(s.focus([idx - 10]),ax,widths=[60,40])
9
10     if idx == 10:
11         slides.write('Unlike `interact/interactive`, this a
```
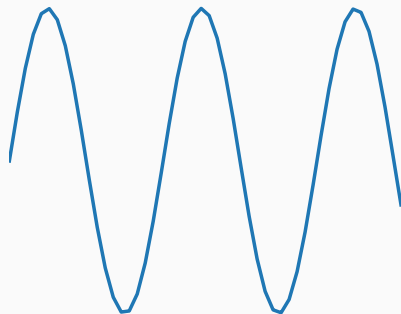
$f(x) = \sin(x),\ 0 < x < 18$

🔶Author: Abdul Saboor عبدالصبور

11

Animating Matplotlib!

Python

```python
 1  for idx in slides.PAGE.iter(range(10,19)):
 2      fig, ax = plt.subplots(figsize=(3.4,2.6))
 3      x = np.linspace(0,idx,50)
 4      ax.plot(x,np.sin(x))
 5      ax.set_title(rf'$f(x)=\sin(x)$, 0 < x < {idx+1}')
 6      ax.set_xlim([0,18])
 7      ax.set_axis_off()
 8      slides.write(s.focus([idx - 10]),ax,widths=[60,40])
 9
10      if idx == 10:
11          slides.write('Unlike `interact/interactive`, this a
```
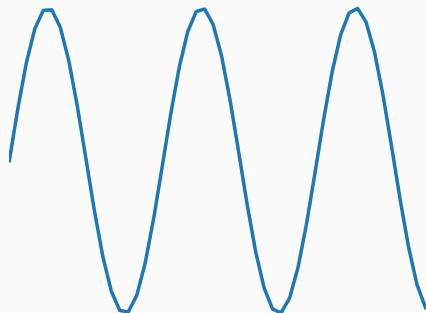
$f(x) = \sin(x), 0 < x < 19$

Author: Abdul Saboor عبدالصبور

11

Contents

Default Frames

```python
1  slides.write('# Default Frames')
2  s.get_source().focus([2,3]).display()
3  slides.PAGE() # want to show source alone first
4  for item in slides.PAGE.iter(boxes):
5      slides.write(item)
```

Default Frames

```python
1  slides.write('# Default Frames')
2  s.get_source().focus([2,3]).display()
3  slides.PAGE() # want to show source alone first
4  for item in slides.PAGE.iter(boxes):
5      slides.write(item)
```

1

Default Frames

```python
1  slides.write('# Default Frames')
2  s.get_source().focus([2,3]).display()
3  slides.PAGE() # want to show source alone first
4  for item in slides.PAGE.iter(boxes):
5      slides.write(item)
```

2

Default Frames

```python
1  slides.write('# Default Frames')
2  s.get_source().focus([2,3]).display()
3  slides.PAGE() # want to show source alone first
4  for item in slides.PAGE.iter(boxes):
5      slides.write(item)
```

3

Default Frames

```python
1  slides.write('# Default Frames')
2  s.get_source().focus([2,3]).display()
3  slides.PAGE() # want to show source alone first
4  for item in slides.PAGE.iter(boxes):
5      slides.write(item)
```

4

Frames with

PAGE`.iter()` and Fancy Bullet List

```python
slides.write('# Frames with \n#### code`PAGE.iter()` and Fancy Bullet List yoffset`0`')
s.get_source().focus([2,3,4]).display()
slides.PAGE() # want to show source alone first
for item in slides.PAGE.iter(boxes):
    slides.bullets([item], marker='🐦').display()
```

Author: Abdul Saboor عبدالصبور

Frames with

PAGE.iter() and Fancy Bullet List

```python
slides.write('# Frames with \n#### code`PAGE.iter()` and Fancy Bullet List yoffset`0`')
s.get_source().focus([2,3,4]).display()
slides.PAGE() # want to show source alone first
for item in slides.PAGE.iter(boxes):
    slides.bullets([item], marker='💘').display()
```

💘

1

💘 Author: Abdul Saboor عبدالصبور

Frames with

PAGE.iter() and Fancy Bullet List

```python
1  slides.write('# Frames with \n#### code`PAGE.iter()` and Fancy Bullet List yoffset`0`')
2  s.get_source().focus([2,3,4]).display()
3  slides.PAGE() # want to show source alone first
4  for item in slides.PAGE.iter(boxes):
5      slides.bullets([item], marker='💘').display()
```

💘

2

Frames with

PAGE`.iter()` and Fancy Bullet List

```python
slides.write('# Frames with \n#### code`PAGE.iter()` and Fancy Bullet List yoffset`0`')
s.get_source().focus([2,3,4]).display()
slides.PAGE() # want to show source alone first
for item in slides.PAGE.iter(boxes):
    slides.bullets([item], marker='💘').display()
```

💘   3

💘Author: Abdul Saboor عبدالصبور   14

Frames with

PAGE.iter() and Fancy Bullet List

```python
slides.write('# Frames with \n#### code`PAGE.iter()` and Fancy Bullet List yoffset`0`')
s.get_source().focus([2,3,4]).display()
slides.PAGE() # want to show source alone first
for item in slides.PAGE.iter(boxes):
    slides.bullets([item], marker='💘').display()
```

💘

4

Frames with

PART`.iter()` and 2x2 grid of boxes

```python
1  slides.write('# Frames with \n#### code`PART.iter()` and 2x2 grid of boxes yoffset`0`')
2  s.get_source().focus(range(2,7)).display()
3  objs = [boxes[:2],boxes[2:]]
4  widths = [(1,3),(3,2)]
5  for ws, cols in slides.PART.iter(zip(widths,objs)):
6      slides.write(*cols, widths=ws)
```

⬥Author: Abdul Saboor عبدالصبور

Frames with

PART`.iter()` and 2x2 grid of boxes

Python

```python
1  slides.write('# Frames with \n#### code`PART.iter()` and 2x2 grid of boxes yoffset`0`')
2  s.get_source().focus(range(2,7)).display()
3  objs = [boxes[:2],boxes[2:]]
4  widths = [(1,3),(3,2)]
5  for ws, cols in slides.PART.iter(zip(widths,objs)):
6      slides.write(*cols, widths=ws)
```

1

Frames with

PART`.iter()` and 2x2 grid of boxes

Python

```
1   slides.write('# Frames with \n#### code`PART.iter()` and 2x2 grid of boxes yoffset`0`')
2   s.get_source().focus(range(2,7)).display()
3   objs = [boxes[:2],boxes[2:]]
4   widths = [(1,3),(3,2)]
5   for ws, cols in slides.PART.iter(zip(widths,objs)):
6       slides.write(*cols, widths=ws)
```

1	2

Author: Abdul Saboor عبدالصبور

Frames with

Python

```python
1  slides.write('# Frames with \n#### code`PART.iter()` and 2x2 grid of boxes yoffset`0`')
2  s.get_source().focus(range(2,7)).display()
3  objs = [boxes[:2],boxes[2:]]
4  widths = [(1,3),(3,2)]
5  for ws, cols in slides.PART.iter(zip(widths,objs)):
6      slides.write(*cols, widths=ws)
```

| 1 | 2 |

| 3 |

Frames with

PART`.iter()` and 2x2 grid of boxes

Python

```python
slides.write('# Frames with \n#### code`PART.iter()` and 2x2 grid of boxes yoffset`0`')
s.get_source().focus(range(2,7)).display()
objs = [boxes[:2],boxes[2:]]
widths = [(1,3),(3,2)]
for ws, cols in slides.PART.iter(zip(widths,objs)):
    slides.write(*cols, widths=ws)
```

1	2
3	4

◈Author: Abdul Saboor عبدالصبور

Watching Youtube Video?

Want to do some drawing instead? Click on button on the right!

IPySlides-Demo

Blocks with CSS classes

Table

h1	h2	h3
d1	d2	d3
r1	r2	r3

A rich content table

h1	h2	h3
⌒	2	3
3	`import numpy as np`	5

Widgets

44
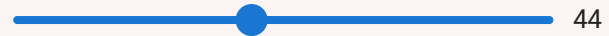
Click to do nothing

☑ Select to do nothing

⌄ Show Code

Author: Abdul Saboor عبدالصبور

● ● ● ● ●   17

$\mathit{L\!\!^{A}\!T_{E}X}$ in Slides

Use `$ $` or `$$ $$` to display latex in Markdown, or embed images of equations $\mathit{L\!\!^{A}\!T_{E}X}$ needs time to load, so keeping it in view until it loads would help.

> 💡 Tip
>
> Varibale formatting alongwith $\mathit{L\!\!^{A}\!T_{E}X}$ %{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.

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

$\LaTeX$ 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.

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

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

$\mathit{L\!^{\!A}\!T_{\!E}X}$ in Slides

Use `$ $` or `$$ $$` to display latex in Markdown, or embed images of equations $\mathit{L\!^{\!A}\!T_{\!E}X}$ needs time to load, so keeping it in view until it loads would help.

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

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

⬆ Show Code

```
1   ++
2   ```multicol 50 50
3   $$ \int_0^1\\frac{1}{1-x^2}dx $$
4   {.align-left .text-big .info}
5   +++
6   ··· success
```

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 user defined/third party
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 import in notebook.

def demo_slides(slides):
    slides.close_view() # Close any previous view to speed up loading 10x faster on average
    slides.clear() # Clear previous content
    raw_source = slides.code.cast(__file__).raw
    N = raw_source.count('.build') + raw_source.count('\n---')
    slides.create(range(N)) # Create slides first, this is faster
```

Source Code

```md-src.collapsed
# Creating Slides
 ::: align-center width=50%
    alert`Abdul Saboor`^`1`, Unknown Author^`2`
    center`//today``//`
     ::: align-left text-box
        ^`1`My University is somewhere in the middle of nowhere
        ^`2`Their University is somewhere in the middle of nowhere

 ::: display align-center
    vspace`2`Read instructions by clicking on %{btn} or same button in quick menu
```
```

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

Author: Abdul Saboor عبدالصبور