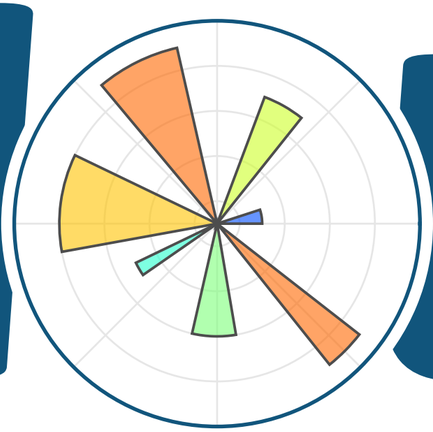
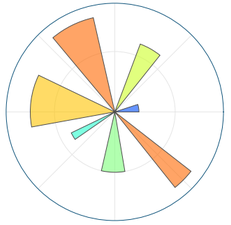
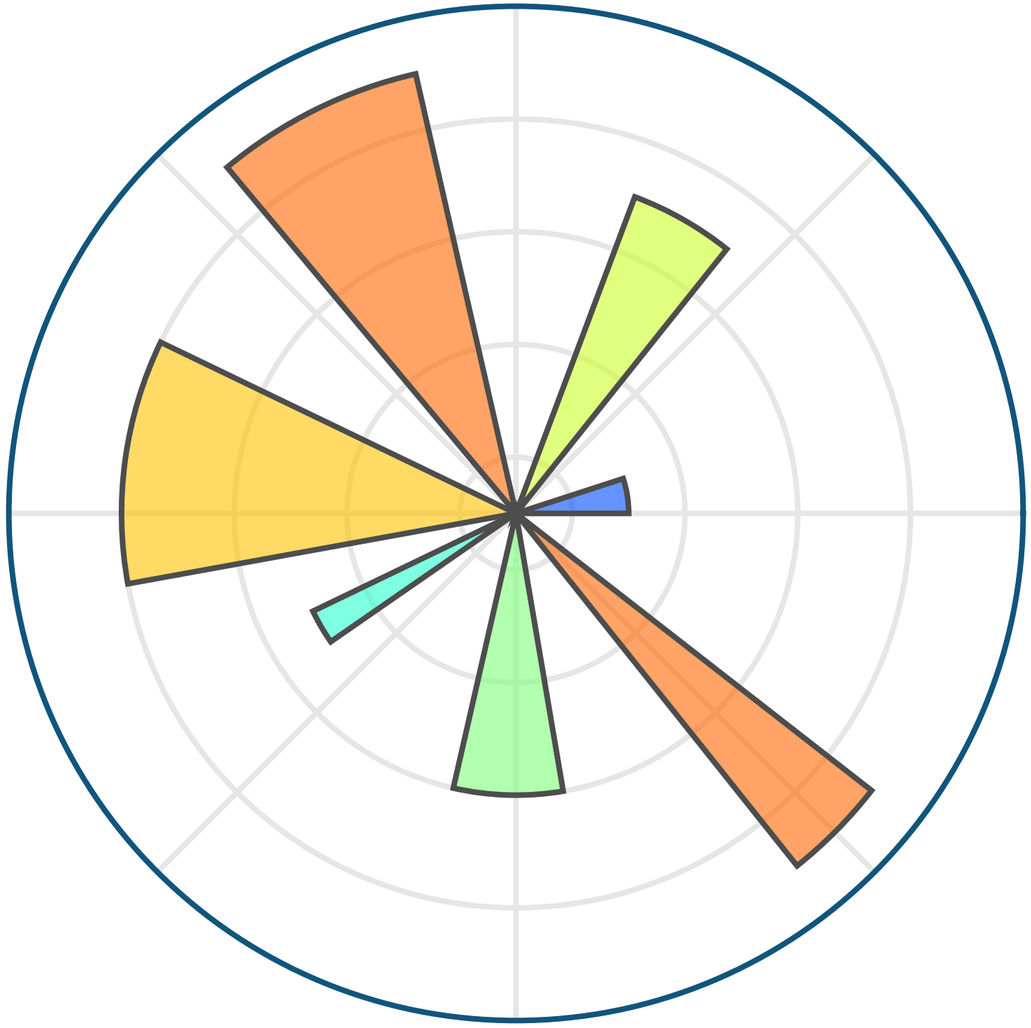
matplotlib



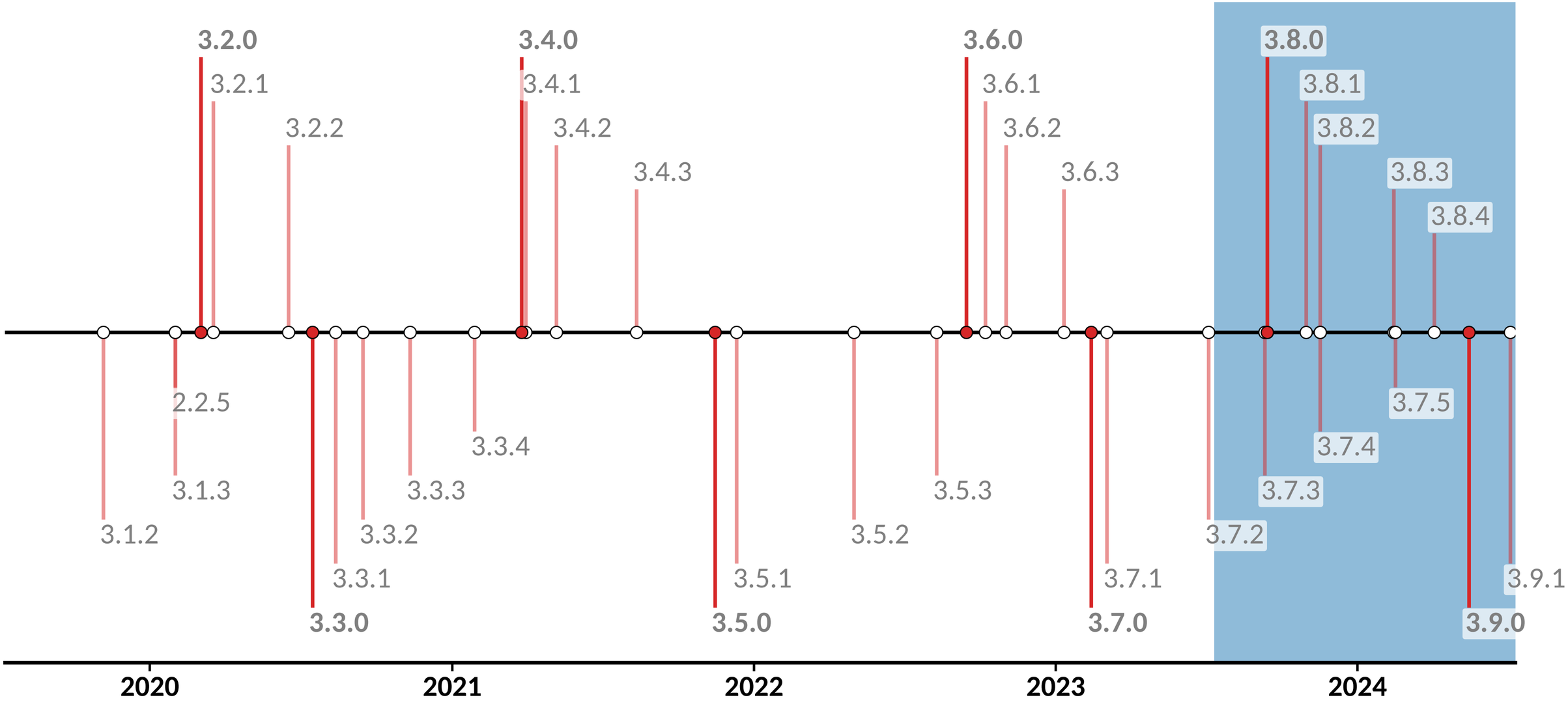
Plotting Slides in @matplotlib

Example: Logo

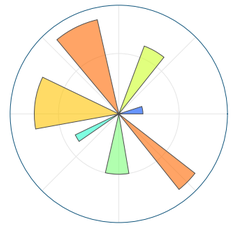




Example: Release History



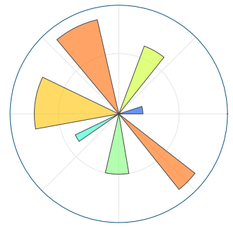
Examples



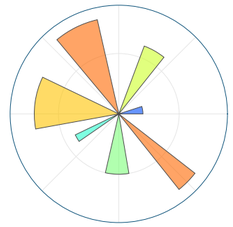
Both are examples from Matplotlib gallery

https://matplotlib.org/stable/gallery/misc/logos2.html

https://matplotlib.org/stable/gallery/lines_bars_and_markers/timeline.html

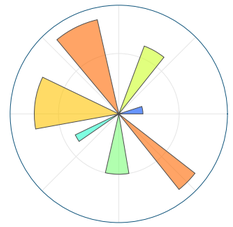


🤔 WHY??? 🤔



Why Not?

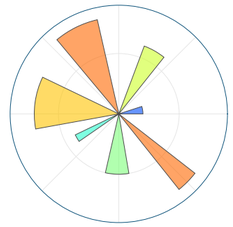
Slide setup



Set a big figure (1080p):

```
plt.rcParams["figure.figsize"] = (
    19.2, 10.8)
plt.rcParams["figure.dpi"] = 100
```

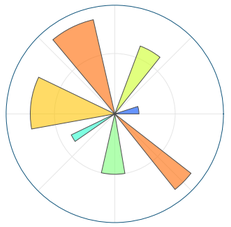
Slide setup



Set a nice font:

```
plt.rcParams['font.family'] = [
    'Carlito',
    'OpenMoji Black',
]
plt.rcParams['font.weight'] = 'bold'
plt.rcParams['font.size'] = 64
plt.rcParams['text.color'] = 'tab:grey'
```

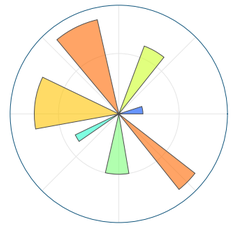
Slide setup



Set better Axes sizes:

```
plt.rcParams['axes.linewidth'] = 3
plt.rcParams['axes.labelsize'] = 40
plt.rcParams['xtick.labelsize'] = 32
plt.rcParams['xtick.major.width'] = 2
plt.rcParams['xtick.major.size'] = 7
plt.rcParams['ytick.labelsize'] = 32
plt.rcParams['ytick.major.width'] = 2
plt.rcParams['ytick.major.size'] = 7
plt.rcParams['lines.linewidth'] = 3
```

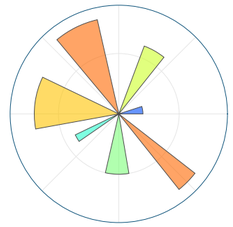
Save the slides



```
from matplotlib.backends.backend_pdf import (
    PdfPages)

figures = [...]
with PdfPages('name.pdf') as pdf:
    for fig in figures:
        add_logo(fig)
        pdf.savefig(fig)
```

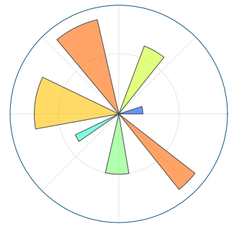
Add a slide title



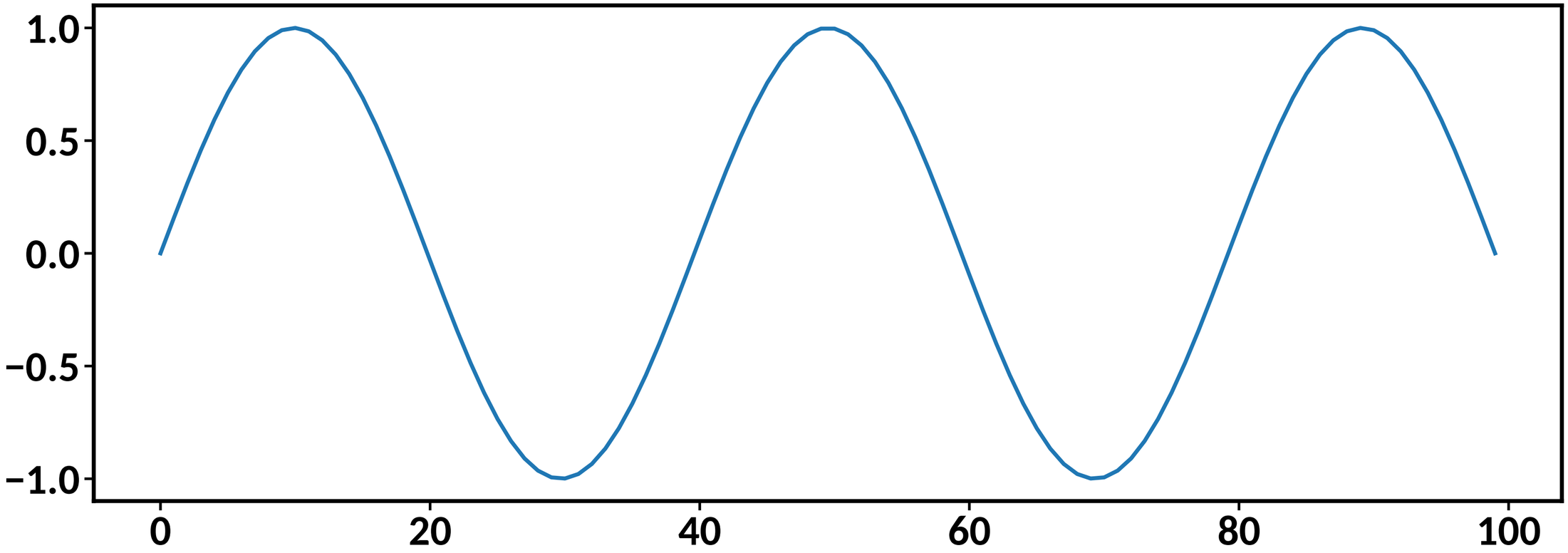
```
def slide_heading(fig, text):
    """
    Add a heading to a slide,
    using a common style.
    """


    fig.text(0.05, 0.85, text,
                color='tab:blue', fontsize=72)
```

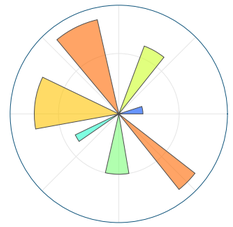
Add a plot



```
fig, ax = plt.subplots()
slide_heading(fig, 'Add a plot')
ax.plot(np.sin(np.linspace(0, 5*np.pi, 100)))
```



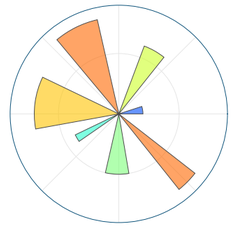
Write some text



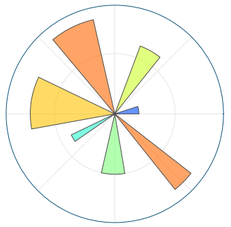
Here is some explanatory text

```
fig.text(0.05, 0.75,
         'Here is some explanatory text')
```

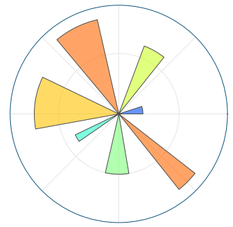
Add an image



```
img = plt.imread('grace_hopper.jpg')
fig.figimage(img, xo=..., yo=...)
```

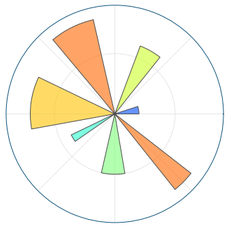


And that's all we need!

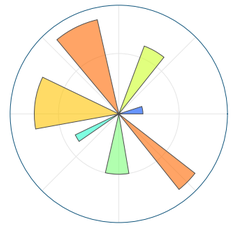


Was that a good idea?

Bonus: QR codes



```
import segno
qrcode = segno.make(url)
out = io.BytesIO()
qrcode.save(out, kind='png', dark=MPL_BLUE)
out.seek(0)
img = Image.open(out).convert('RGB')
ax = fig.add_axes(
    location, frameon=False, xticks=[], yticks=[])
ax.imshow(img)
```



Demo

https://github.com/QuLogic/scipy2024-lightning-mpl-slides