
basics - pillole - blog

basics

28 nov 2024

Indice

1	Animazioni in un Jupyter book	3
2	Test carta	5
2.1	Libreria <code>ipyleaflet</code>	5

Questo bbook è pensato per raccogliere alcuni approfondimenti e spunti.

Percezione umana

- Udito
- Vista
- ...

Animazioni in un Jupyter book

Primo approccio alle animazioni¹ in un Jupyter book.

```
%matplotlib inline

import numpy as np
import matplotlib.pyplot as plt

# Create a figure and axes.
fig = plt.figure(figsize=(12,5))
ax1 = plt.subplot(1,2,1)
ax2 = plt.subplot(1,2,2)

# Set up the subplots
ax1.set_xlim((0,2))
ax1.set_ylim((-2,2))
ax1.set_xlabel('Time')
ax1.set_ylabel('Magnitude')

ax2.set_xlim((-2,2))
ax2.set_ylim((-2,2))
ax2.set_xlabel('X')
ax2.set_ylabel('Y')
ax2.set_title('Phase plane')

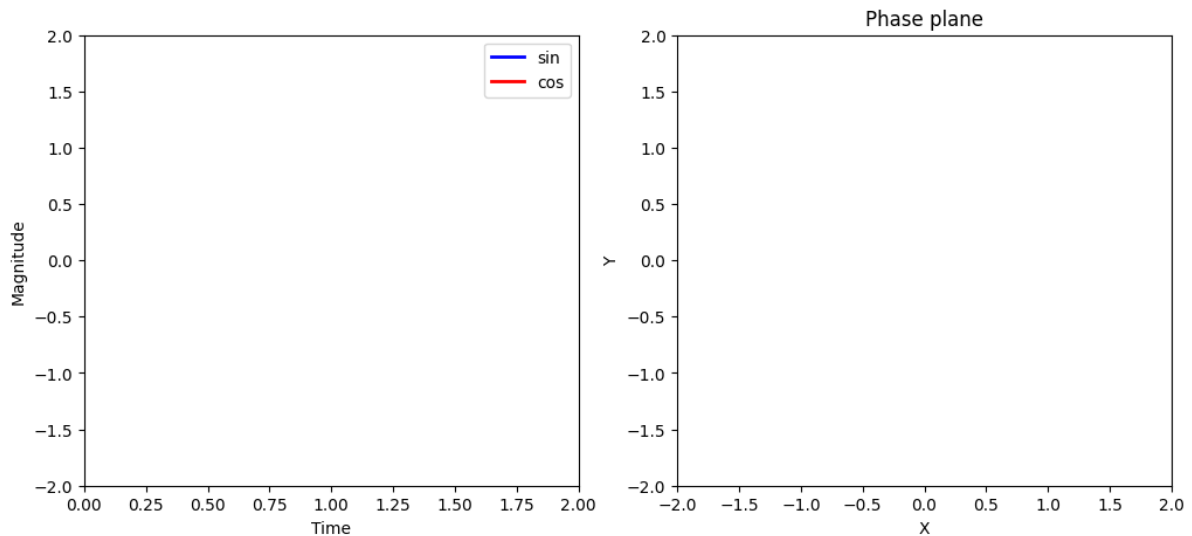
# Create objects that will change in the animation.
# These objects are initially empty, and will be given new values for each frame in
# the animation.
txt_title = ax1.set_title('')
line1, = ax1.plot([], [], 'b', lw=2) # ax.plot returns a list of 2D line objects.
line2, = ax1.plot([], [], 'r', lw=2)
pt1, = ax2.plot([], [], 'g.', ms=20)
line3, = ax2.plot([], [], 'y', lw=2)
```

(continues on next page)

¹ https://interactivetextbooks.tudelft.nl/open-textbooks-demonstration/content/Basic_animation_demo.html

(continua dalla pagina precedente)

```
ax1.legend(['sin', 'cos']);
```



```
# Animation function. This function is called sequentially.
def drawframe(n):
    x = np.linspace(0, 2, 1000)
    y1 = np.sin(2 * np.pi * (x - 0.01 * n))
    y2 = np.cos(2 * np.pi * (x - 0.01 * n))
    line1.set_data(x, y1)
    line2.set_data(x, y2)
    line3.set_data(y1[0:50], y2[0:50])
    pt1.set_data([y1[0]], [y2[0]]) # Note that matplotlib will throw an error if we
    ↪supply only numbers (i.e., pt1.set_data(y1[0], y2[0]))
    txt_title.set_text('Frame = {0:4d}'.format(n))
    return (line1, line2)
```

```
# Initialization function.
def init():
    line1.set_data([], [])
    line2.set_data([], [])
    return (line1, line2)
```

```
from matplotlib import animation

#anim = animation.FuncAnimation(fig, drawframe, init_func=init, frames=100,
    ↪interval=20, blit=True)
anim = animation.FuncAnimation(fig, drawframe, frames=100, interval=20, blit=True)
# blit = True re-draws only the parts that have changed.
```

```
from IPython.display import HTML
HTML(anim.to_jshtml())
```

```
<IPython.core.display.HTML object>
```


2.1 Libreria ipyleaflet

2.1.1 Import librerie

```
%pip install ipyleaflet
%pip install ipywidgets
%pip install jupyterlab_widgets

# %jupyter nbextension enable --py widgetsnbextension
# %jupyter nbextension enable --py ipyleaflet
```

2.1.2 Carta semplice

```
VBox(children=(Map(center=[45.0, 10.0], controls=(ZoomControl(options=['position',
↵'zoom_in_text', 'zoom_in_ti...
```

```
import os
print(f"cwd: {os.getcwd()}")
```

```
cwd: /home/davide/Documents/basics-books/books/bbooks-insights/ch
```

2.1.3 Carta con qualche dettaglio e interattività

```
Map(center=[45, 10], controls=(ZoomControl(options=['position', 'zoom_in_text',  
↪ 'zoom_in_title', 'zoom_out_tex...
```