

# ITS INFORMATION AND COMMUNICATIONS TECHNOLOGY Academy

---

**NOME MODULO: IA**  
**UNITÀ DIDATTICA: IA.1**  
**Lezione 2**  
**Stefano Puglia**

Biennio 2024-2026



## INDICE DEGLI ARGOMENTI

- Introduzione a Matplotlib
  - Prime visualizzazioni (plot, chart)

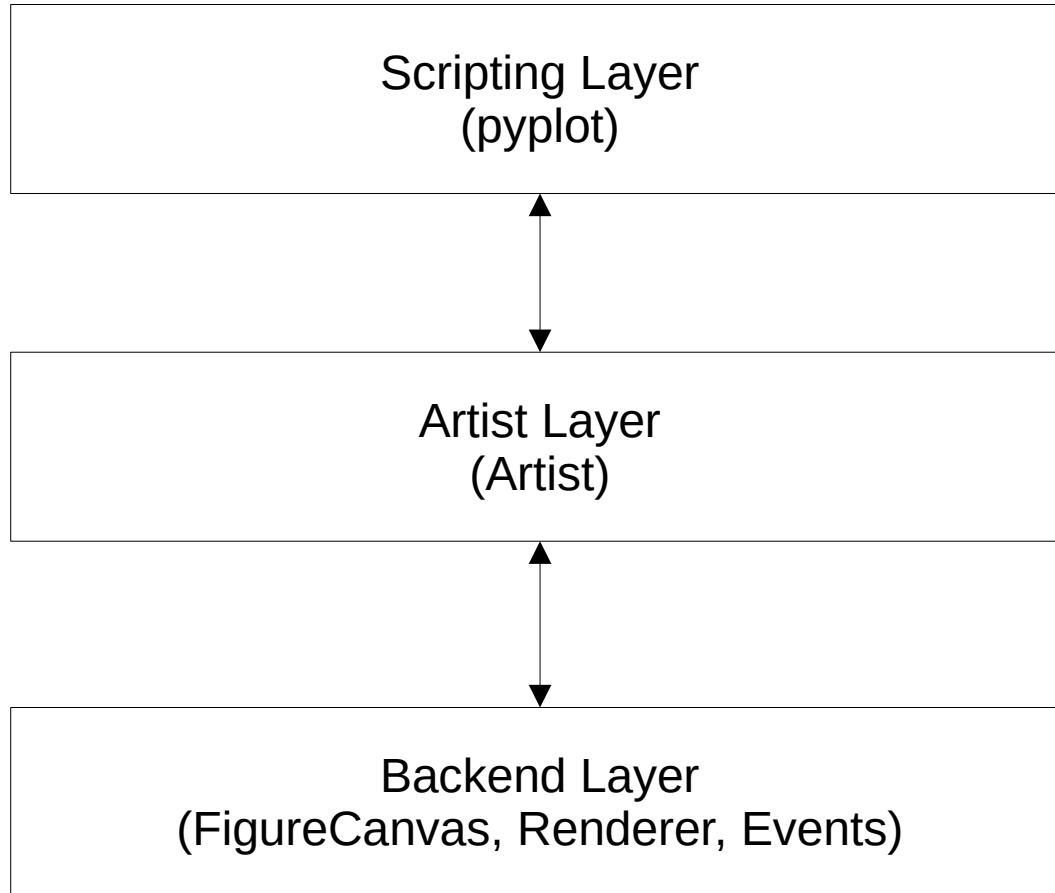


# matplotlib

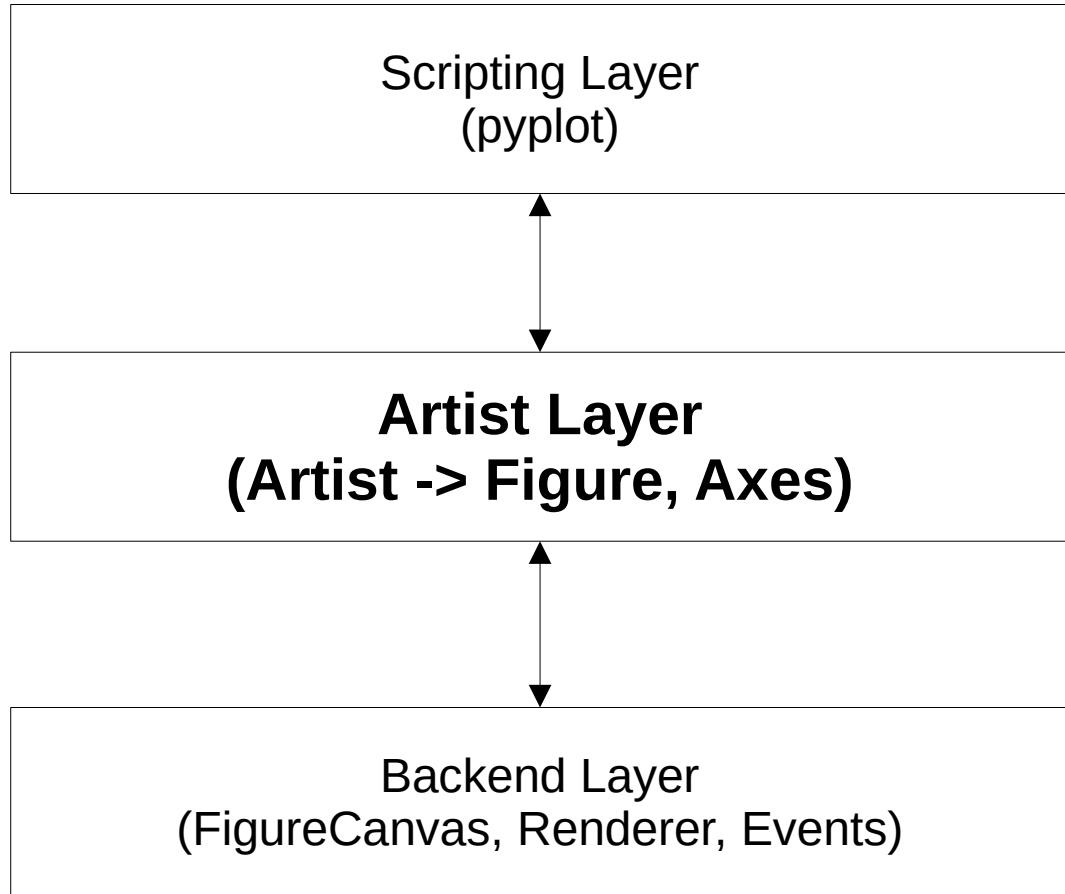
```
import matplotlib.pyplot as plt
```



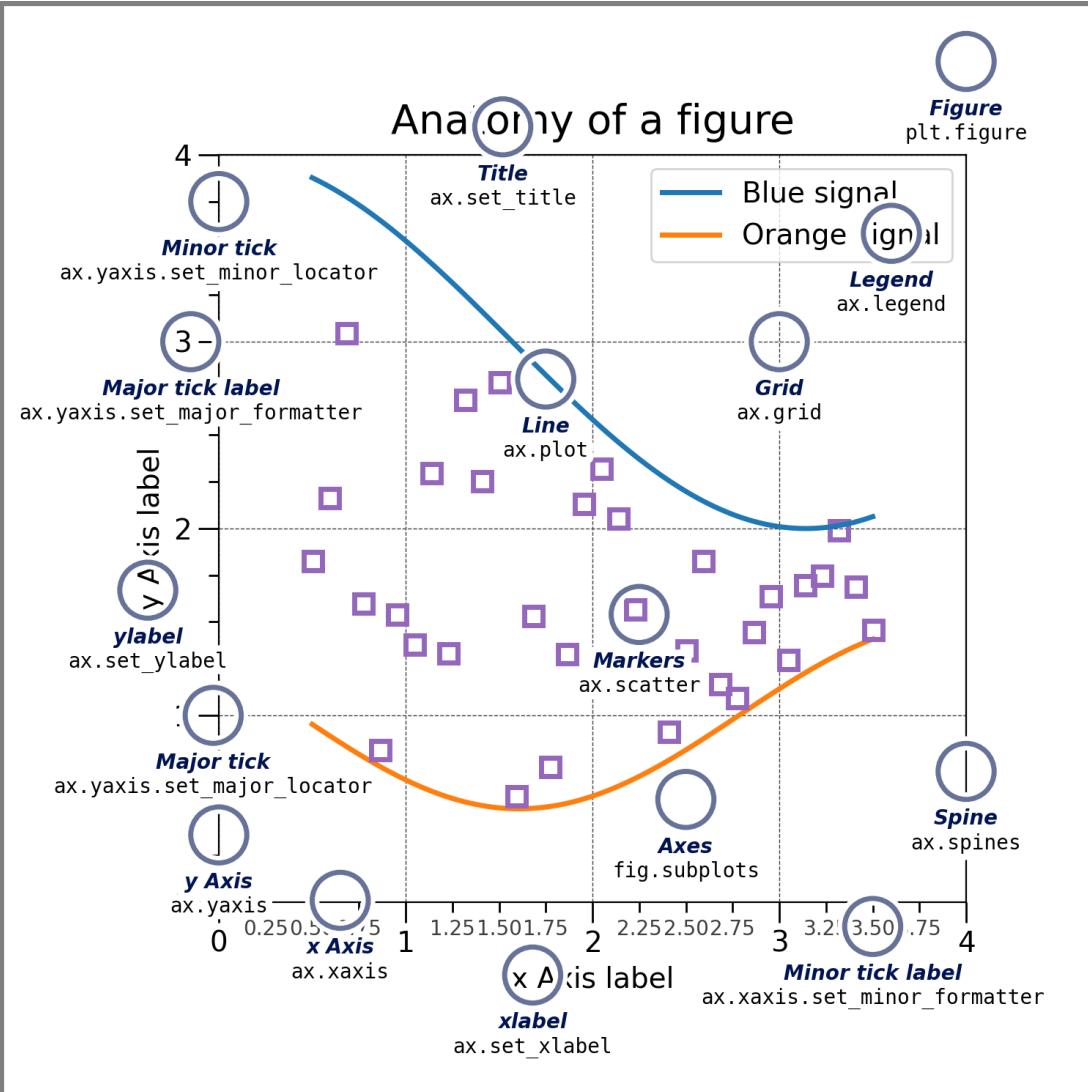
# Matplotlib - Architettura



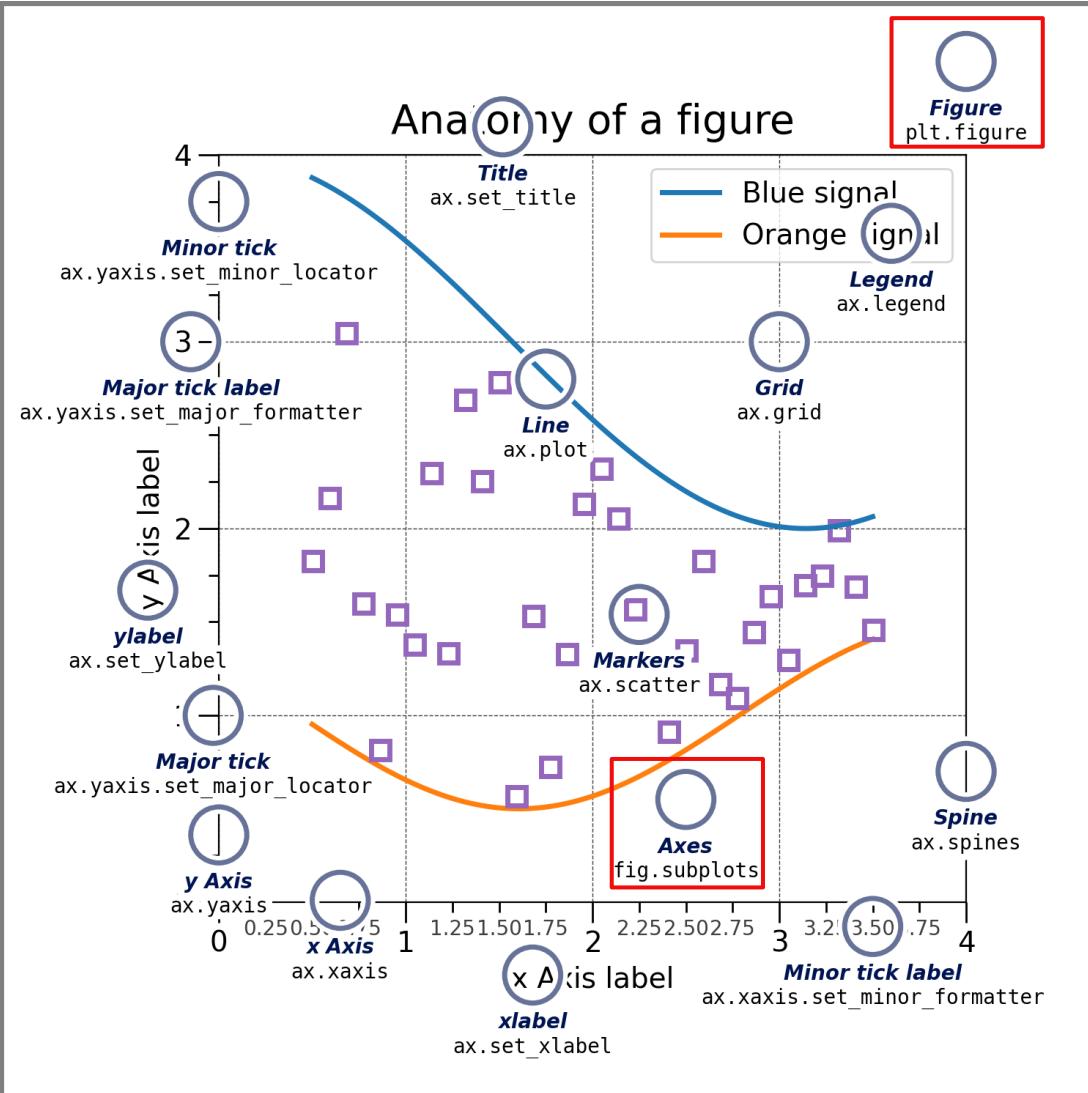
# Matplotlib - Artist



# Matplotlib - Artist



# Matplotlib - Artist



# Matplotlib for beginners

Matplotlib is a library for making 2D plots in Python. It is designed with the philosophy that you should be able to create simple plots with just a few commands:

## 1 Initialize

```
import numpy as np  
import matplotlib.pyplot as plt
```

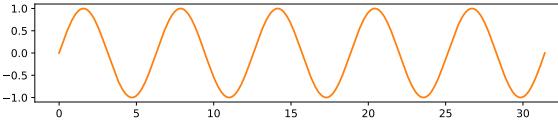
## 2 Prepare

```
X = np.linspace(0, 10*np.pi, 1000)  
Y = np.sin(X)
```

## 3 Render

```
fig, ax = plt.subplots()  
ax.plot(X, Y)  
plt.show()
```

## 4 Observe



## Choose

Matplotlib offers several kind of plots (see Gallery):

```
X = np.random.uniform(0, 1, 100)  
Y = np.random.uniform(0, 1, 100)  
ax.scatter(X, Y)
```



```
X = np.arange(10)  
Y = np.random.uniform(1, 10, 10)  
ax.bar(X, Y)
```



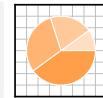
```
Z = np.random.uniform(0, 1, (8, 8))  
ax.imshow(Z)
```



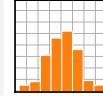
```
Z = np.random.uniform(0, 1, (8, 8))  
ax.contourf(Z)
```



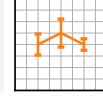
```
Z = np.random.uniform(0, 1, 4)  
ax.pie(Z)
```



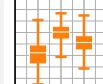
```
Z = np.random.normal(0, 1, 100)  
ax.hist(Z)
```



```
X = np.arange(5)  
Y = np.random.uniform(0, 1, 5)  
ax.errorbar(X, Y, Y/4)
```



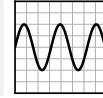
```
Z = np.random.normal(0, 1, (100, 3))  
ax.boxplot(Z)
```



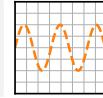
## Tweak

You can modify pretty much anything in a plot, including limits, colors, markers, line width and styles, ticks and ticks labels, titles, etc.

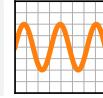
```
X = np.linspace(0, 10, 100)  
Y = np.sin(X)  
ax.plot(X, Y, color="black")
```



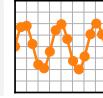
```
X = np.linspace(0, 10, 100)  
Y = np.sin(X)  
ax.plot(X, Y, linestyle="--")
```



```
X = np.linspace(0, 10, 100)  
Y = np.sin(X)  
ax.plot(X, Y, linewidth=5)
```



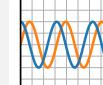
```
X = np.linspace(0, 10, 100)  
Y = np.sin(X)  
ax.plot(X, Y, marker="o")
```



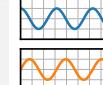
## Organize

You can plot several data on the same figure, but you can also split a figure in several subplots (named Axes):

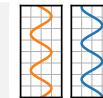
```
X = np.linspace(0, 10, 100)  
Y1, Y2 = np.sin(X), np.cos(X)  
ax.plot(X, Y1, X, Y2)
```



```
fig, (ax1, ax2) = plt.subplots(2, 1)  
ax1.plot(X, Y1, color="C1")  
ax2.plot(X, Y2, color="C0")
```

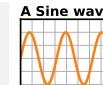


```
fig, (ax1, ax2) = plt.subplots(1, 2)  
ax1.plot(Y1, X, color="C1")  
ax2.plot(Y2, X, color="C0")
```

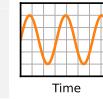


## Label (everything)

```
ax.plot(X, Y)  
fig.suptitle(None)  
ax.set_title("A Sine wave")
```



```
ax.plot(X, Y)  
ax.set_ylabel(None)  
ax.set_xlabel("Time")
```



## Explore

Figures are shown with a graphical user interface that allows to zoom and pan the figure, to navigate between the different views and to show the value under the mouse.

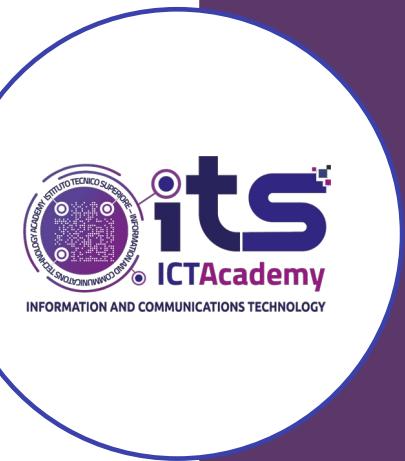
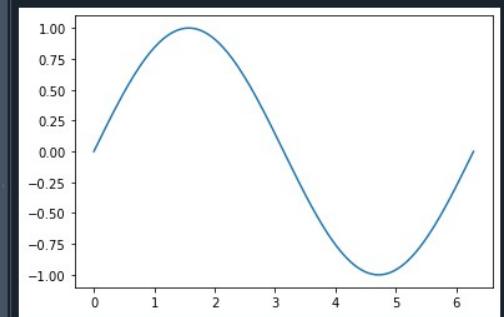
## Save (bitmap or vector format)

```
fig.savefig("my-first-figure.png", dpi=300)  
fig.savefig("my-first-figure.pdf")
```

# Matplotlib - Artist

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 2 * np.pi, 200)
y = np.sin(x)
fig, ax = plt.subplots() # Oggetti di tipo figura e asse per il grafico
ax.plot(x, y)
plt.savefig('matplotlib_sine_line.png')
plt.show()
```

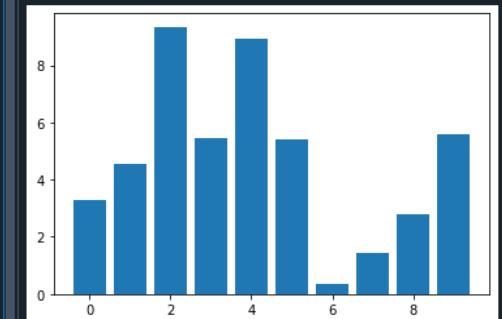
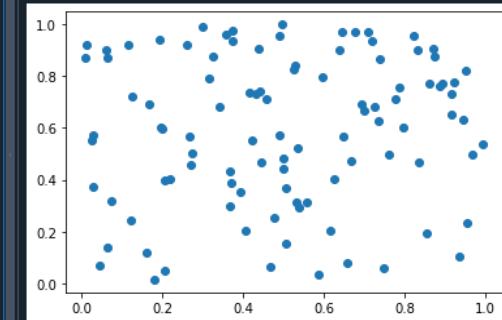


# Matplotlib - Artist

```
import matplotlib.pyplot as plt
import numpy as np

X = np.random.uniform(0, 1, 100)
Y = np.random.uniform(0, 1, 100)
fig, ax = plt.subplots() # Oggetti di tipo figura e asse per il grafico
ax.scatter(X, Y)
plt.savefig('matplotlib_scatter.png')
plt.show()
```

```
X = np.arange(10)
Y = np.random.uniform(0, 10, 10)
fig, ax = plt.subplots() # Oggetti di tipo figura e asse per il grafico
ax.bar(X, Y)
plt.savefig('matplotlib_bar.png')
plt.show()
```

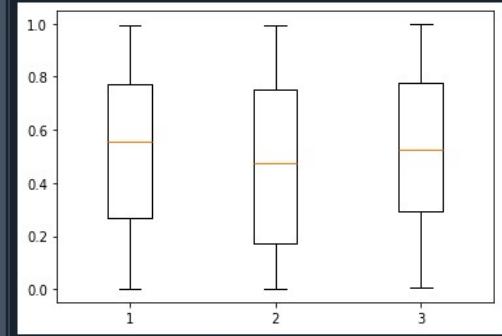


# Matplotlib - Artist

```
import matplotlib.pyplot as plt
import numpy as np

Z = np.random.uniform(0, 1, 4)
fig, ax = plt.subplots() # Oggetti di tipo figura e asse per il grafico
ax.pie(Z)
plt.savefig('matplotlib_pie.png')
plt.show()

Z = np.random.uniform(0, 1, (100, 3))
fig, ax = plt.subplots() # Oggetti di tipo figura e asse per il grafico
ax.boxplot(Z)
plt.savefig('matplotlib_boxplot.png')
plt.show()
```

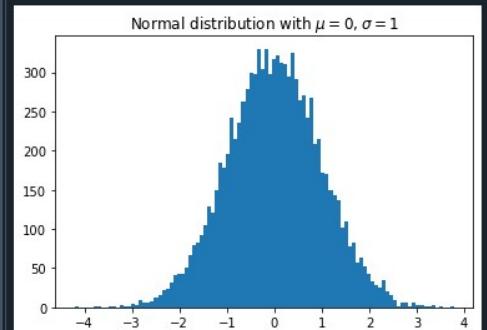
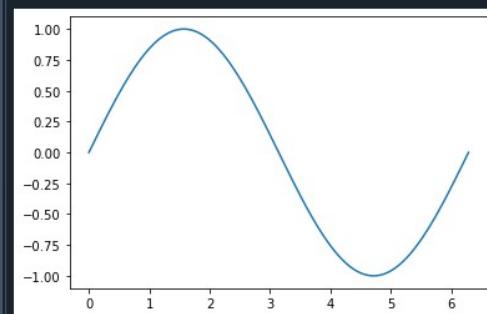


# Matplotlib - Artist

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 2 * np.pi, 200)
y = np.sin(x)
fig, ax = plt.subplots() # Oggetti di tipo figura e asse per il grafico
ax.plot(x, y)
plt.savefig('matplotlib_sine_line.png')
plt.show()
```

```
x = np.random.randn(10000) # 10000 punti distribuiti secondo una gaussiana (standard normal distribution)
# centrata sullo zero e con varianza uguale a 1.
# Es. -0.455, 0.947, -1.123, ..., 1.675
plt.hist(x, 100) # numero di punti raggruppati in 100 intervalli (bins)
plt.title(r'Normal distribution with $\mu=0, \sigma=1$')
plt.savefig('matplotlib_histogram.png')
plt.show()
```

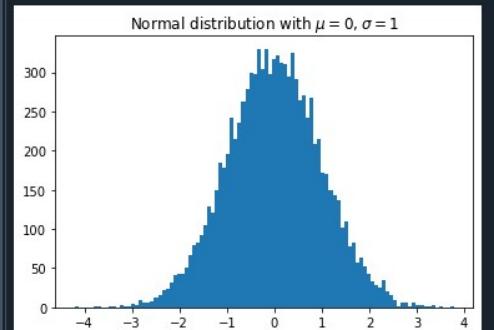
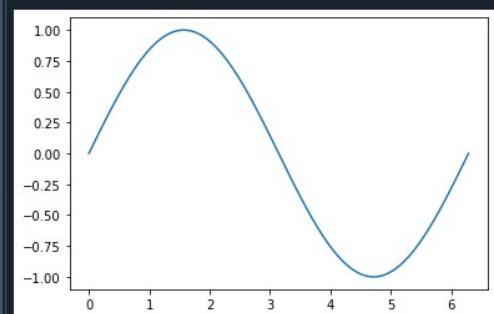


# Matplotlib - Artist e pyplot

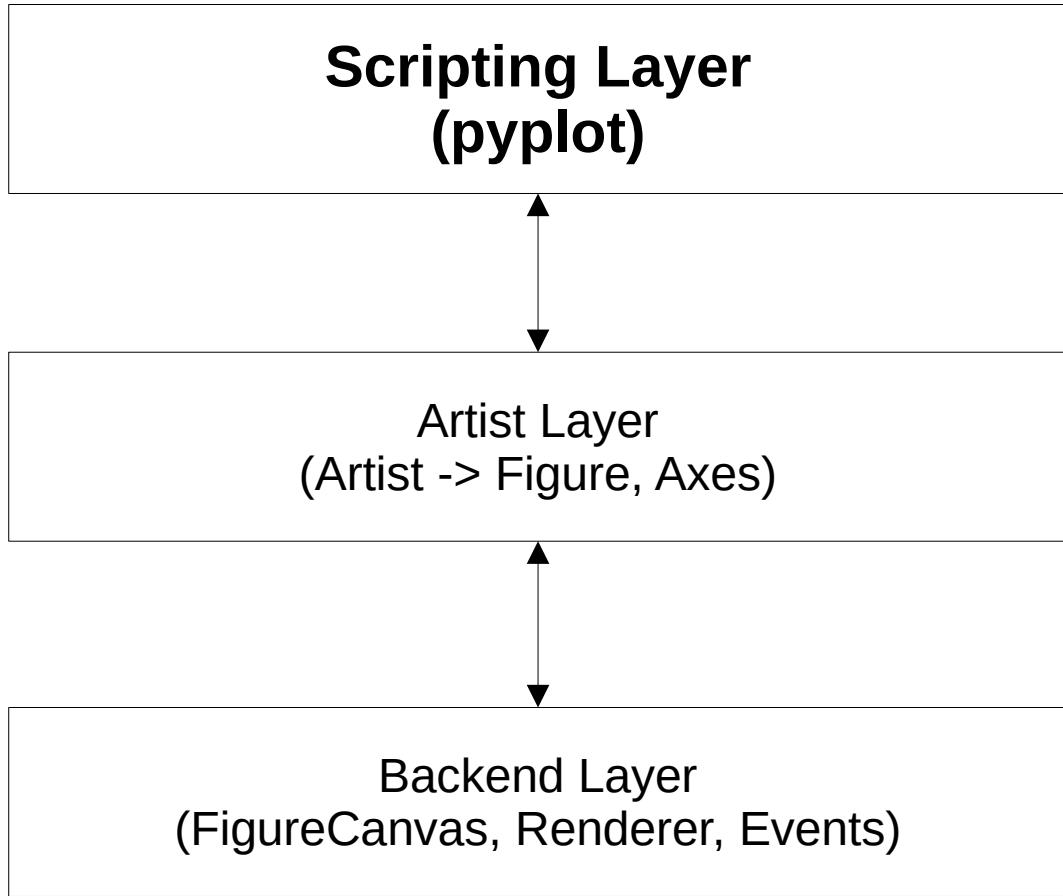
```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 2 * np.pi, 200)
y = np.sin(x)
fig, ax = plt.subplots() # Oggetti di tipo figura e asse per il grafico
ax.plot(x, y)
plt.savefig('matplotlib_sine_line.png')
plt.show()
```

```
x = np.random.randn(10000) # 10000 punti distribuiti secondo una gaussiana (standard normal distribution)
# centrata sullo zero e con varianza uguale a 1.
# Es. -0.455, 0.947, -1.123, ..., 1.675
plt.hist(x, 100) # numero di punti raggruppati in 100 intervalli (bins)
plt.title(r'Normal distribution with $\mu=0, \sigma=1$')
plt.savefig('matplotlib_histogram.png')
plt.show()
```



# Matplotlib - pyplot



# Matplotlib - Artist e pyplot

```
df_agg_1 = df.groupby('preferred_deck').agg(deck_count=('PassengerId', 'count'))
```

```
X = df_agg_1.index.astype(str)
Y = df_agg_1['deck_count']
fig, ax = plt.subplots()
ax.bar(X, Y)
```

OPPURE

```
plt.bar(x=df_agg_1.index.astype(str), height=df_agg_1['deck_count'])
```



# Matplotlib - Artist e pyplot

```
df_agg_1 = df.groupby('preferred_deck').agg(deck_count=('PassengerId', 'count'))
```

```
X = df_agg_1.index.astype(str)
Y = df_agg_1['deck_count']
fig, ax = plt.subplots()
ax.bar(X, Y)
```



OPPURE

```
plt.bar(x=df_agg_1.index.astype(str), height=df_agg_1['deck_count'])
```

OPPURE

# Matplotlib(Artist e pyplot) e pandas

```
df_agg_1 = df.groupby('preferred_deck').agg(deck_count=('PassengerId', 'count'))
```

```
X = df_agg_1.index.astype(str)
Y = df_agg_1['deck_count']
fig, ax = plt.subplots()
ax.bar(X, Y)
```



OPPURE

```
plt.bar(x=df_agg_1.index.astype(str), height=df_agg_1['deck_count'])
```

OPPURE

```
df_agg_1.plot(kind = 'bar')
```

# Matplotlib (pyplot) e pandas

```
df_agg_1 = df.groupby('preferred_deck').agg(deck_count=('PassengerId', 'count'))
```

```
X = df_agg_1.index.astype(str)
Y = df_agg_1['deck_count']
fig, ax = plt.subplots()
ax.bar(X, Y)
```



OPPURE

```
plt.bar(x=df_agg_1.index.astype(str), height=df_agg_1['deck_count'])
```

OPPURE

```
df_agg_1.plot(kind = 'bar')
```

## RIFERIMENTI BIBLIOGRAFICI

- click to pandas documentation
- click to matplotlib documentation

