

July 27, 2024

Riprendiamo dal ciclo **for**

```
[ ]: for età in range(30):
    print("La tua età è di anni " + str(età) + "?", " Ma sei un giovincello!!!")
for età in range(30,41):
    print("La tua età è di anni " + str(età) + "?", " Dai, sei un giovincello!!!")
for età in range(41,99):
    print("La tua età è di anni " + str(età) + "?", " Non sei più un ragazzino!!!")
```

La funzione range() può essere personalizzata

range(start, stop, step)

Esercizio: scrivere un codice (usando il ciclo for) che identifichi i numeri pari presenti in una lista di numeri naturali.

Esercizio: scrivere un codice (usando il ciclo for) che conta il numero di lettere di un certo tipo (quante a?) ci sono in una parola

Ciclo **While**

Parliamo di **DEFINIZIONI**

```
[ ]: def moltiplicazione(a,b):
    print('Questa è la funzione moltiplicazione.')
    print('Fornisce il prodotto di due numeri passati come parametri.')
    risultato = a * b
    print('Il risultato della moltiplicazione_
    ↳di',str(a),"per",str(b),"è",str(risultato))

moltiplicazione(3,7)
```

```
[ ]: # Area del cerchio
def area_cerchio(r):
    print("L'area del cerchio è direttamente proporzionale al quadrato del raggio.
    ↳")
    area = 3.14159*(r**2)
    print("Dato il raggio di lunghezza", str(r)+",")
    print("l'area del cerchio", "è " + str(area))

area_cerchio(4)
```

L'area del cerchio è direttamente proporzionale al quadrato del raggio.
Dato il raggio di lunghezza 4,
l'area del cerchio è 50.26544

```
[ ]: # Costruiamo una definizione che trovi il minimo tra tre numeri assunti come
      ↪parametri

def massimo(m1,m2,m3):
    """Questa funzione trova il massimo tra tre numeri"""
    if m1>m2 and m1>m3:
        return m1
    if m2>m3:
        return m2
    return m3

massimo(7,5,3)
```

```
[ ]: 7
```

Esercizio: costruire una funzione che permetta di passare da gradi Celsius a gradi Kelvin e viceversa.

Esercizio: costruire una funzione che conti le vocali di una parola

Esercizio: costruire una funzione che inverta una stringa (con il ciclo while!)

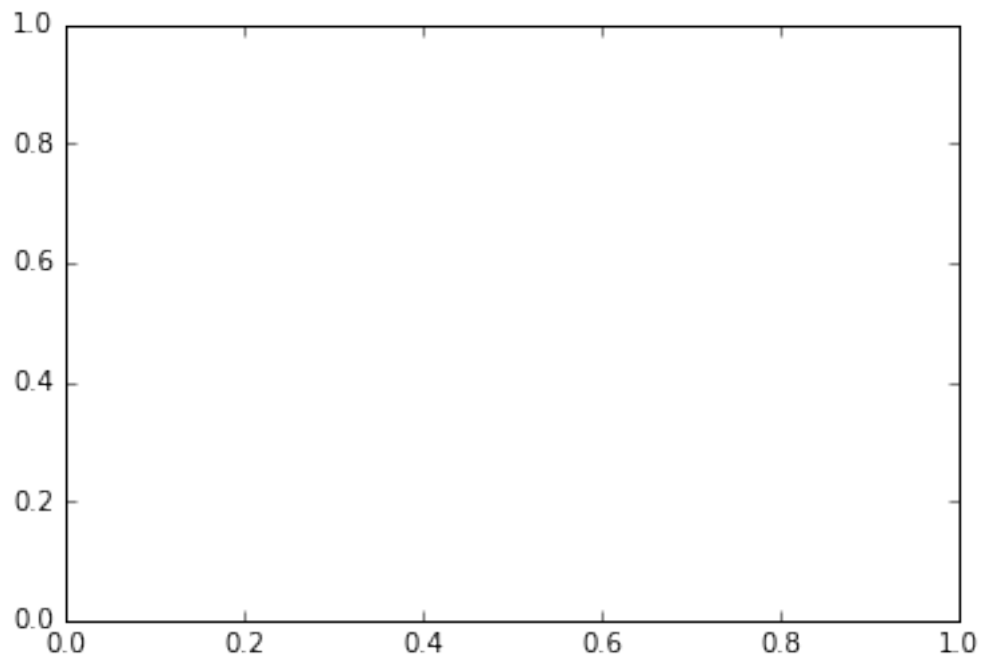
1 Matplotlib

<https://jakevdp.github.io/PythonDataScienceHandbook/index.html>

<https://matplotlib.org/tutorials/index.html>

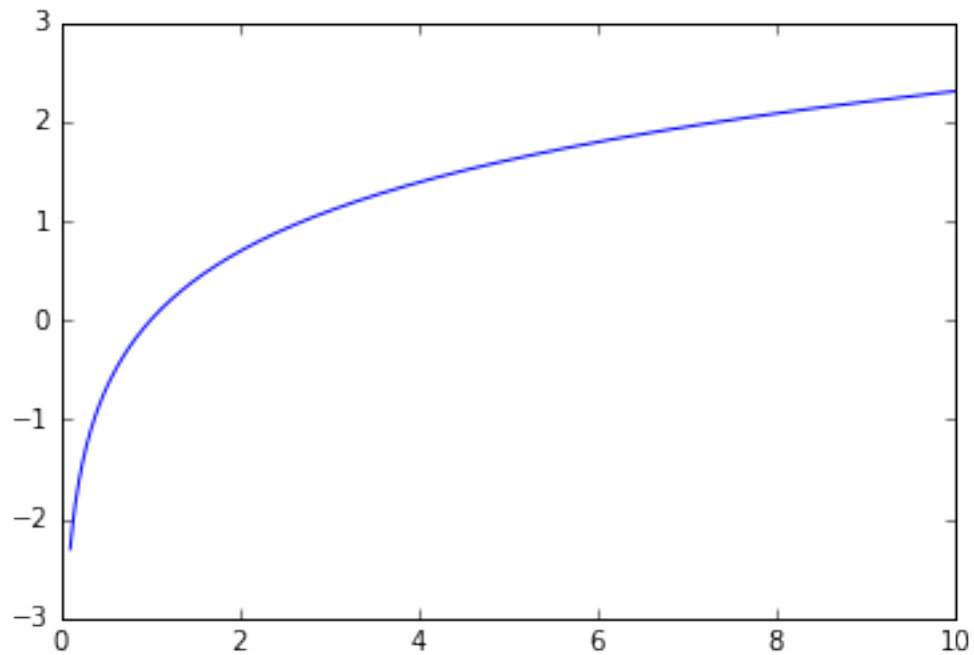
```
[ ]: %matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('classic') # https://tonysyu.github.io/raw_content/
      ↪matplotlib-style-gallery/gallery.html
import numpy as np
```

```
[ ]: fig = plt.figure() # costruiamo un ambiente figura
ax = plt.axes() # inseriamo gli assi
```



```
[ ]: fig = plt.figure()
      ax = plt.axes()

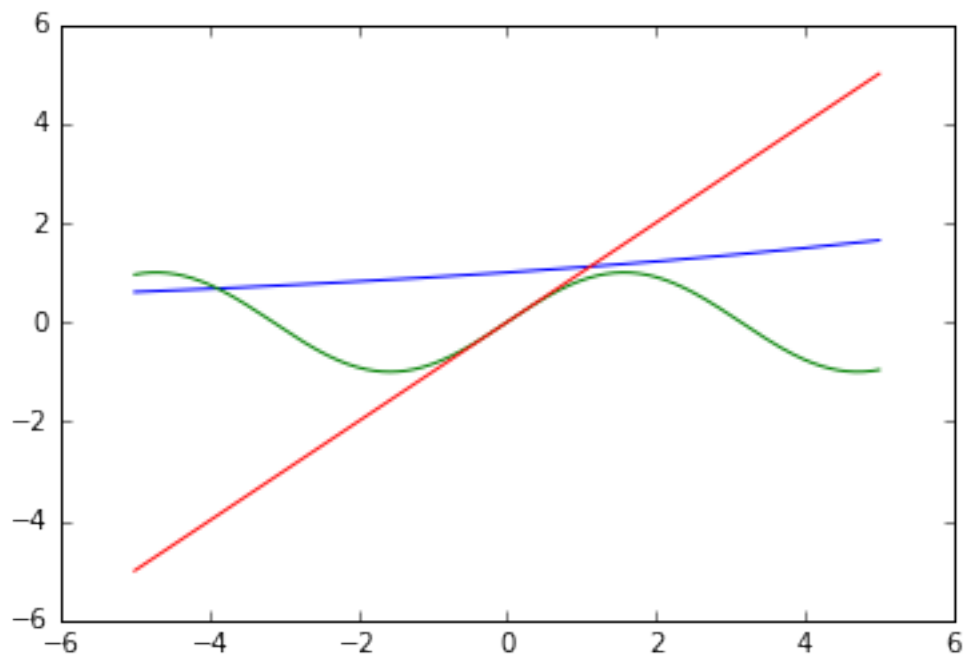
      x = np.linspace(0.1, 10, 1000) # estremi asse x e numero di punti per costruire
      ↪ la funzione
      ax.plot(x, np.log(x));
```



Possiamo anche inserire più grafici nella stessa figura

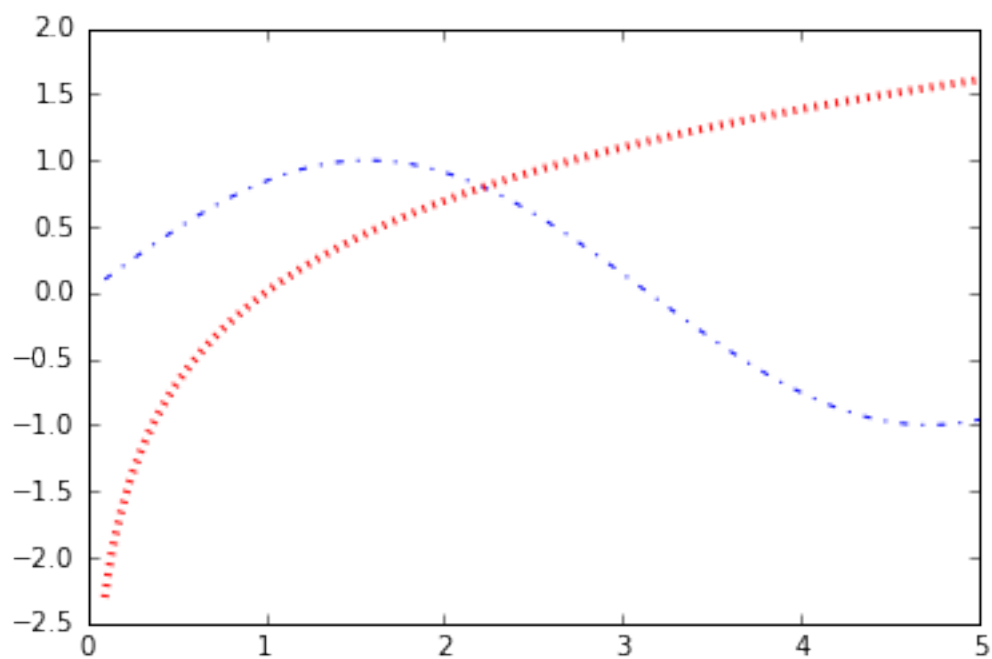
```
[ ]: fig = plt.figure()
      ax = plt.axes()

      x = np.linspace(-5,5, 1000) #https://docs.scipy.org/doc/numpy/reference/
      ↪generated/numpy.linspace.html
      ax.plot(x, np.exp(x/10))
      ax.plot(x, np.sin(x))
      ax.plot(x,x);
```



2 Stile linee e colori

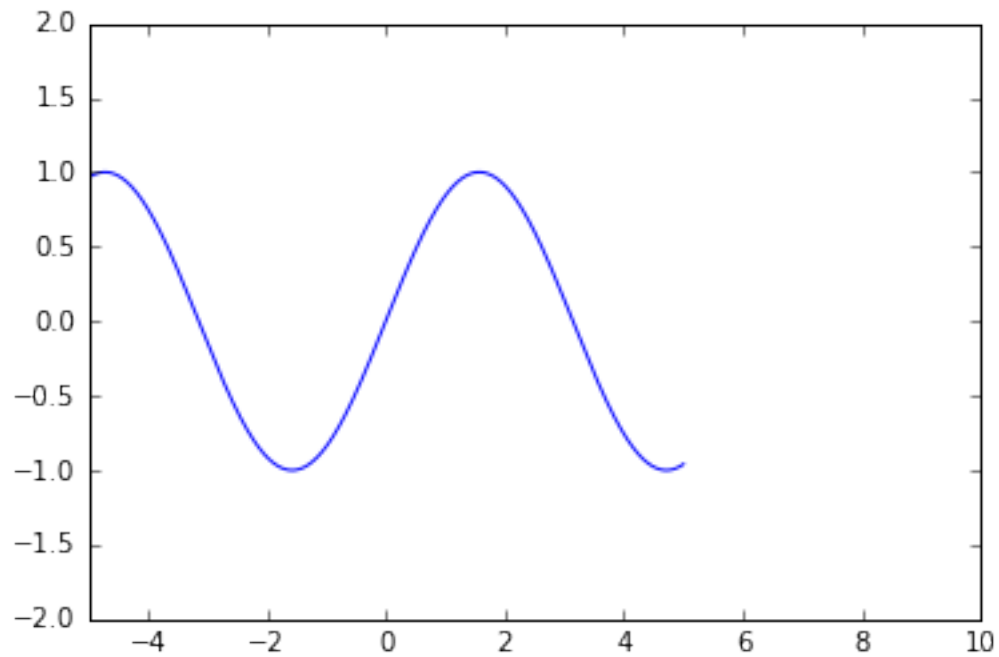
```
[ ]: x = np.linspace(0.1,5, 1000)
plt.plot(x, np.sin(x), color='blue', linestyle='dashdot')
plt.plot(x, np.log(x), color='red', linestyle='dotted', linewidth=3);
```



Possiamo modificare le impostazioni degli assi

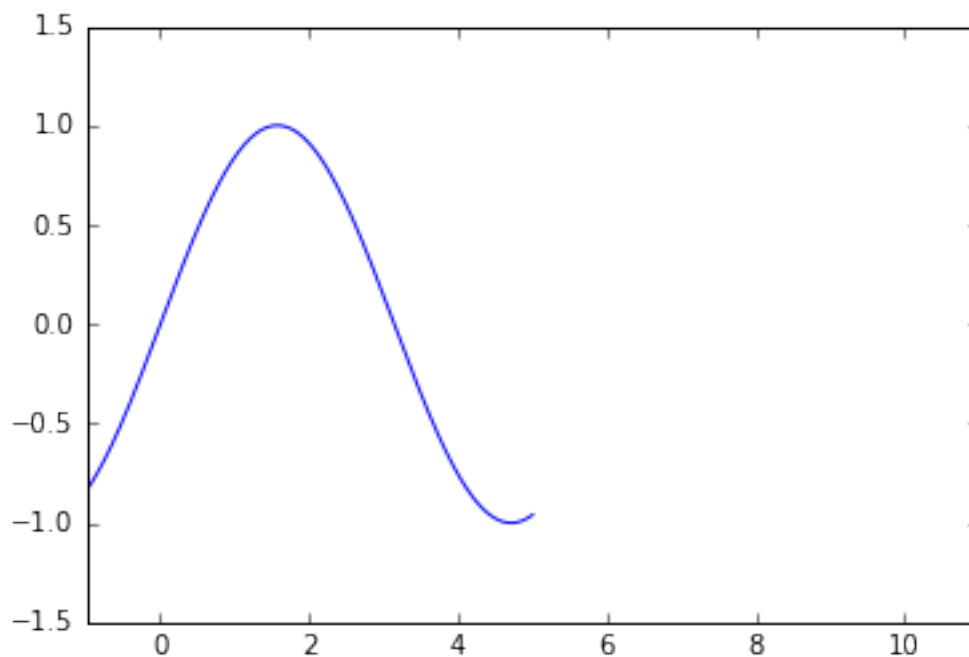
```
[ ]: x = np.linspace(-5,5, 1000)
plt.plot(x, np.sin(x))

plt.xlim(-5, 10)
plt.ylim(-2, 2);
```



Possiamo fare la stessa cosa con un singolo comando

```
[ ]: plt.plot(x, np.sin(x))
plt.axis([-1, 11, -1.5, 1.5]);
```



Esercizio:

Rappresentare su uno stesso grafico la funzione

$$y = \frac{x}{x^2 - 1}$$

e la funzione

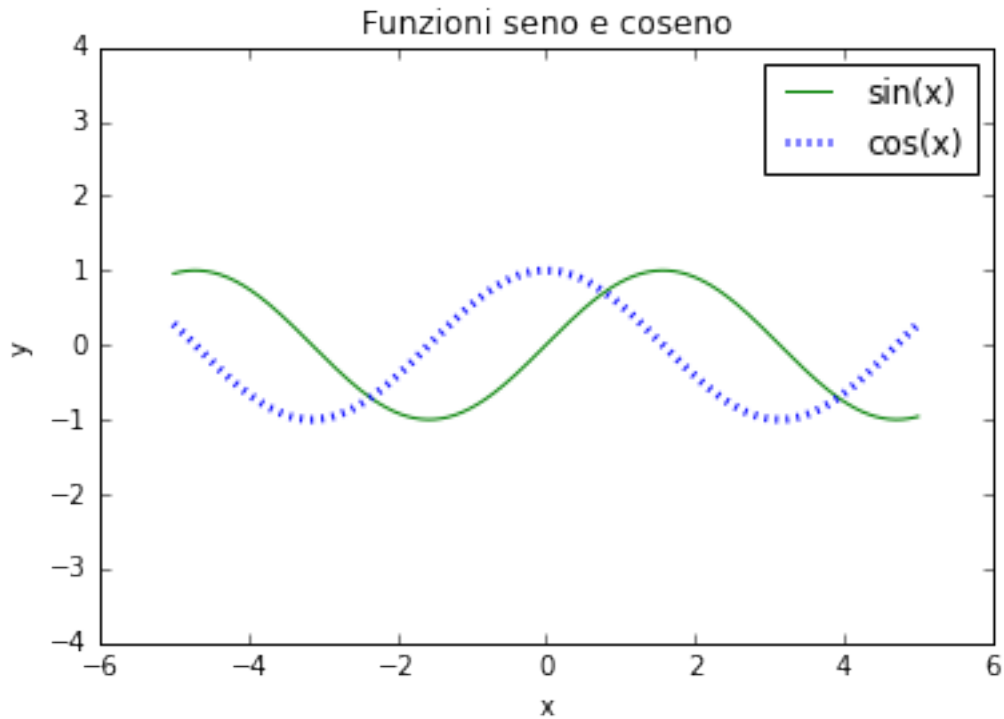
$$y = x^3$$

in modo che abbiano due stili diversi, due colori diversi e due spessori di linea diversi

3 Titolo assi legenda

```
[ ]: plt.plot(x, np.sin(x), '-g', label='sin(x)') # abbreviazione per tratto e colore
plt.plot(x, np.cos(x), ':b', label='cos(x)', linewidth=3)
plt.title("Funzioni seno e coseno")
plt.xlabel("x")
plt.ylabel("y")
plt.axis('equal')

plt.legend();
```



Tutto in uno!

```
[ ]: ax = plt.axes()
      ax.plot(x, np.arctan(x))
      ax.set(xlim=(-3, 10), ylim=(-2, 2),
              xlabel='x', ylabel='sin(x)',
              title='A Simple Plot');
```

Esercizio:

Rappresentare la funzione

$$y = \sqrt{x}$$

dando delle opportune limitazioni sugli assi, inserendo il nome della funzione sull'asse y e mettendo nel titolo la scritta "Funzioni irrazionali".