

In [1]:

```
#Importo numpy

import numpy as np
from numpy import mean, std
```

In [2]:

```
#Dati

v = np.array([1,2,4,3,5])
a = np.array([[1,2],[3,4]])
b = np.array([[5,6],[7,8]])
```

In [3]:

```
#1.1 - Visualizzare gli attributi size e shape di v e a

print("L'array v è composto da " + str(np.size(v)) + " elementi" + "\n")
print("L'array a è composto da " + str(np.size(a)) + " elementi" + "\n")
print("L'array v è composto da " + str(np.shape(v)) + " elementi" + "\n")
print("L'array v è composto da " + str(np.shape(a)) + " elementi" + "\n")
```

L'array v è composto da 5 elementi

L'array a è composto da 4 elementi

L'array v è composto da (5,) elementi

L'array v è composto da (2, 2) elementi

In [4]:

```
#1.2 - Calcolare massimo, minimo, media e deviazione standard degli elementi di v

print("L'elemento massimo dell'array v è " + str(np.max(v)) + "\n")
print("L'elemento minimo dell'array v è " + str(np.min(v)) + "\n")
print("La media degli elementi dell'array v è " + str(np.mean(v)) + "\n")
print("La deviazione standard degli elementi dell'array v è " + str(np.std(v)) + "\n")
```

L'elemento massimo dell'array v è 5

L'elemento minimo dell'array v è 1

La media degli elementi dell'array v è 3.0

La deviazione standard degli elementi dell'array v è 1.4142135623730951

In [5]:

```
#1.3 - Ordinare gli elementi di v

print("L'array v è stato ordinato " + str(np.sort(v)))
```

L'array v è stato ordinato [1 2 3 4 5]

In [6]:

```
#1.4 - Calcolare il risultato delle operazioni +, -, *, / e ** tra le matrici a e b

print("La somma delle matrici è " + str(a + b) + "\n")
print("La differenza delle matrici è " + str(a - b) + "\n")
print("Il prodotto delle matrici è " + str(a * b) + "\n")
print("La divisione delle matrici è " + str(a / b) + "\n")
print("La potenza delle matrici è " + str(a ** b))
```

La somma delle matrici è [[ 6 8]  
[10 12]]

La differenza delle matrici è [[-4 -4]  
[-4 -4]]

Il prodotto delle matrici è [[ 5 12]  
[21 32]]

La divisione delle matrici è [[0.2 0.33333333]  
[0.42857143 0.5 ]]

La potenza delle matrici è [[ 1 64]  
[ 2187 65536]]

In [7]:

```
#1.6 - Creare un vettore con la sequenza 0-9. Creare un vettore con solo i multipli di 3 tra 3 e 27

#La funzione arange() restituisce valori uniformemente distanziati all'interno di un determinato intervallo.

n1 = 10
#Valori generati con un mezzo intervallo [0, stop]. Include l'inizio ma non la fine
c = np.arange(n1)
print("Il vettore c ha i seguenti elementi " + str(c) + "\n")

d = np.array([])
#Valori generati all'interno dell'intervallo semiaperto [start, stop, step], con una distanza tra i valori data dal step.
d = np.arange(3, 27, 3)

print(d)
```

Il vettore c ha i seguenti elementi [0 1 2 3 4 5 6 7 8 9]

[ 3 6 9 12 15 18 21 24]

In [8]:

```
#1.8 - Creare una matrice di tutti 0, una matrice di tutti 1 e una matrice diagonale (tutte di 2x2)

m1 = [[0 for column in range(2)]for row in range(2)]
m2 = [[1 for column in range(2)]for row in range(2)]
m3 = np.zeros((2, 2), int) #Altro modo per creare una matrice con tutti 0
np.fill_diagonal(m3, 1)

print(str(m1) + "\n")
print(str(m2) + "\n")
print(m3)
```

[[0, 0], [0, 0]]

[[1, 1], [1, 1]]

[[1 0]
 [0 1]]

In [9]:

```
#1.9 - Trovare ed estrarre solo gli elementi maggiori di 2 di v

mask = v > 2
result = v[mask]
print(result)
```

[4 3 5]

In [10]:

```
#1.10 - Trovare ed estrarre solo gli elementi sulla diagonale di a

result = np.diag(a)
print(result)
```

[1 4]

In [11]:

```
#Importo matplotlib

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

In [12]:

```
#2.1 Disegna grafico
```

```
#Genera grafico
```

```
plt.plot([1, 2, 3], label="esempio", color="red")
```

```
#Imposta Legenda
```

```
plt.legend(loc='best')
```

```
#Minimo e massimo assi X e Y
```

```
plt.axis([-1, 7, -5, 4])
```

```
# Assegna nome agli assi
```

```
plt.ylabel('asse y')
```

```
plt.xlabel('asse x')
```

```
plt.show()
```

