**CHEAT\_SHEET NUMPY**

|  |  |  |
| --- | --- | --- |
| np.array | Crea un array NumPy. | np.array([1, 2]) |
| np.zeros(shape) | Genera un array di zeri con la forma specificata. | np.zeros((3, 3)) |
| np.ones(shape) | Genera un array di uni con la forma specificata. | np.ones((2, 2)) |
| len(array) | Ritorna la lunghezza dell’array. |  |
| np.arange(start, stop, step) | Genera una sequenza di numeri con un intervallo specificato. | np.arange(0, 10, 2) |
| np.linspace(start, stop, num) | Genera num valori equidistanti tra start e stop. | np.linspace(0, 1, 5) |
| np.random.randint(low, high, size) | Genera un array di interi casuali. | np.random.randint(0, 10, 3) |
| np.random.normal(loc, scale, size) | Genera valori casuali da una distribuzione normale. | np.random.normal(0, 1, 100) |
| np.random.uniform(low, high, size) | Genera valori casuali da una distribuzione uniforme. | np.random.uniform(0, 1, 10) |
| np.random.choice(a, size) | Sceglie elementi casuali da un array | np.random.choice([1, 2, 3], 2) |
| np.sum(array) | Calcola la somma degli elementi dell'array. | np.sum(array) |
| np.mean(array) | Calcola la media degli elementi in un array |  |
| np.min(array) | Trova il valore minimo in un array |  |
| np.max(array) | Trova il valore massimo in un array |  |
| np.std(array) | Calcola la deviazione standard degli elementi dell'array | np.std(array) |
| .dtype | Ritorna il tipo di dati dell'array |  |
| .shape | Ritorna la forma (dimensioni) dell’array |  |
| .reshape(shape) | Cambia la forma dell’array | array.reshape((2, 3)) |
| array.copy() | Crea una copia dell'array | array.copy() |
| np.round(array, decimals) | Arrotonda i valori in un array a un numero specifico di decimali. | np.round([1.23, 2.56], 1) |
| np.count\_nonzero(array) | Conta gli elementi non zero |  |
| np.exp(x) | Calcola l’esponenziale di ogni elemento in x | np.exp([0, 1, 2]) |
| np.sin(x) | Calcola il seno di ogni elemento in x | np.sin(np.pi / 2) |
| np.pi | Rappresenta il valore di pi greco |  |
| np.logical\_not(array) | Applica l'operazione logica 'not' a ogni elemento | np.logical\_not(array) |
| np.logical\_and(array1, array2) | Applica 'and' logico elemento per elemento tra due array | np.logical\_and(array1, array2) |
| np.logical\_or(array1, array2) | Applica 'or' logico elemento per elemento tra due array | np.logical\_or(array1, array2) |
| np.unique | Trova i valori unici in un array | np.unique(array) |
| np.isclose(array1, array2) | Confronta elementi tra due array e restituisce True se sono vicini. | np.isclose(array1, array2) |
| np.roll(array, shift) | Ruota l’array di un numero specificato di posizioni | np.roll([1, 2, 3], 1) |
| np.where(condition) | Restituisce indici degli elementi che soddisfano una condizione. | np.where(array > 0) |
| plt.plot(x, y) | Genera un grafico a linee | plt.plot(x, y) |
| plt.imshow(image) | Mostra un'immagine | plt.imshow(image, cmap='viridis') |
| plt.imread(filepath) | Legge un’immagine dal percorso specificato. | plt.imread('image.png') |
| plt.show() | Mostra la grafica o l'immagine |  |
| fig, ax = plt.subplots(figsize=(8, 6)) | Crea una figura e una serie di assi per il grafico | fig, ax = plt.subplots(figsize=(8, 6)) |
| ax.plot(x, y) | Genera un grafico su assi specifici | ax.plot(x, y) |
| ax.set\_xlabel('X Label') | Imposta l'etichetta dell'asse x | ax.set\_xlabel('X Label') |
| ax.set\_ylabel('Y Label') | Imposta l'etichetta dell'asse y | ax.set\_ylabel('Y Label') |
| ax.set\_title(r'$y = \sin(x)$') | Imposta il titolo del grafico (supporta formule matematiche) | ax.set\_title(r'$y = \sin(x)$') |
| ax.grid(True) | Aggiunge una griglia al grafico. | ax.grid(True) |
| ax.set\_ylim(-1, 1) | Imposta i limiti dell'asse y | ax.set\_ylim(-1, 1) |
| ax.legend(['Serie 1', 'Serie 2']) | Mostra la legenda del grafico. | ax.legend(['Serie 1', 'Serie 2']) |