

NumPy Methods in Python

1. Creating Arrays

```
import numpy as np
```

```
# Creating an array from a list
```

```
arr = np.array([1, 2, 3, 4])
```

```
# Creating arrays with specific values
```

```
zeros = np.zeros((2, 3)) # 2x3 array of zeros
```

```
ones = np.ones((3, 3)) # 3x3 array of ones
```

```
full = np.full((2, 2), 7) # 2x2 array filled with 7
```

```
eye = np.eye(3) # Identity matrix of size 3x3
```

```
# Creating arrays using ranges
```

```
arange = np.arange(0, 10, 2) # [0, 2, 4, 6, 8]
```

```
linspace = np.linspace(0, 1, 5) # 5 evenly spaced numbers between 0 and 1
```

```
# Random arrays
```

```
rand = np.random.rand(3, 3) # 3x3 array with random values between 0 and 1
```

```
randint = np.random.randint(1, 10, (2, 2)) # Random integers between 1 and 10
```

2. Array Attributes

```
arr.shape # Returns the shape of the array (rows, cols)
```

```
arr.ndim # Returns the number of dimensions
```

```
arr.size # Returns the total number of elements
```

```
arr.dtype # Returns the data type of the array elements
```

```
arr.itemsize # Returns the size of each element in byte
```

3. Reshaping and Manipulating Arrays

```
arr.reshape(2, 2) # Reshape array to 2x2
arr.flatten()     # Flatten the array to a 1D array
arr.T             # Transpose of the array
np.concatenate([arr, arr]) # Concatenating arrays
np.hstack([arr, arr]) # Horizontal stacking
np.vstack([arr, arr]) # Vertical stacking
```

4. Mathematical Operations

```
arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])

arr1 + arr2 # Element-wise addition
arr1 - arr2 # Element-wise subtraction
arr1 * arr2 # Element-wise multiplication
arr1 / arr2 # Element-wise division
np.sqrt(arr1) # Square root of elements
np.exp(arr1) # Exponential
np.log(arr1) # Natural logarithm
np.sin(arr1) # Sine function
np.cos(arr1) # Cosine function
np.dot(arr1, arr2) # Dot product
```

5. Statistical Methods

```
arr = np.array([1, 2, 3, 4, 5])

np.min(arr) # Minimum value
np.max(arr) # Maximum value
np.sum(arr) # Sum of all elements
```

```
np.mean(arr)    # Mean (average)
np.median(arr)  # Median
np.std(arr)     # Standard deviation
np.var(arr)     # Variance
np.percentile(arr, 50) # 50th percentile
np.corrcoef(arr) # Correlation coefficient
```

6. Indexing and Slicing

```
arr = np.array([[1, 2, 3], [4, 5, 6]])

arr[0, 1] # Access element at row 0, column 1 (output: 2)
arr[:, 1] # Access all rows in column 1 (output: [2, 5])
arr[1, :] # Access all columns in row 1 (output: [4, 5, 6])
arr[0:2, 1:3] # Slice array
```

7. Boolean Indexing

```
arr = np.array([1, 2, 3, 4, 5])

arr > 2    # Returns boolean array: [False False True True True]
arr[arr > 2] # Returns elements where condition is True: [3 4 5]
```

8. Copying and Sorting

```
arr.copy()    # Creates a copy of the array
np.sort(arr)   # Sorts the array in ascending order
np.argsort(arr) # Returns indices that would sort the array
```

9. Unique Values and Counting

```
arr = np.array([1, 2, 2, 3, 3, 3])

np.unique(arr)    # Returns unique values: [1 2 3]
np.bincount(arr)  # Counts occurrences of value
```