

```
import numpy as np
```

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

```
↵ Array: [1 2 3 4 5]
```

```
arr = np.arange(0, 11)  
print("Arange array:", arr)
```

```
↵ Arange array: [ 0  1  2  3  4  5  6  7  8  9 10]
```

```
rand_arr = np.random.randint(1, 101, (3, 3))  
print("Random 3x3 array:\n", rand_arr)
```

```
↵ Random 3x3 array:  
[[76 84 65]  
 [25 83 74]  
 [ 4 43 70]]
```

```
print("Zeros:\n", np.zeros((2, 2)))  
print("Ones:\n", np.ones((2, 2)))  
print("Empty:\n", np.empty((2, 2))) # values are uninitialized
```

```
↵ Zeros:  
[[0. 0.]  
 [0. 0.]  
 Ones:  
 [[1. 1.]  
 [1. 1.]  
 Empty:  
 [[1. 1.]  
 [1. 1.]
```

```
floats = np.random.rand(5)  
print("Random floats:", floats)
```

```
↵ Random floats: [0.1759916  0.57439999 0.6027019  0.01488555 0.1760676 ]
```

```
a = np.arange(6)  
reshaped = a.reshape((2, 3))  
print("Reshaped:\n", reshaped)
```

```
↵ Reshaped:  
[[0 1 2]  
 [3 4 5]]
```

```
a = np.array([1, 3, 5, 2])  
print("Max value:", np.max(a))
```

```
↵ Max value: 5
```

```
a = np.array([10, 2, 30, 4])  
print("Index of min:", np.argmin(a))
```

```
↵ Index of min: 1
```

```
a = np.array([[1, 2], [3, 4]])  
print("Shape:", a.shape)
```

```
↵ Shape: (2, 2)
```

```
a = np.array([[1, 2], [3, 4], [5, 6]])
print("Total elements:", a.size)
```

➦ Total elements: 6

```
a = np.array([[1, 2], [3, 4]])
print("Dimensions:", a.ndim)
```

➦ Dimensions: 2

```
a = np.array([1, 2, 3])
print("Data type:", a.dtype)
```

➦ Data type: int64

```
a = np.array([1, 2, 3])
b = a.copy()
print("Original:", a, "\nCopy:", b)
```

➦ Original: [1 2 3]
Copy: [1 2 3]

```
a = np.array([1, 2, 3])
a = np.append(a, 4)
print("Appended array:", a)
```

➦ Appended array: [1 2 3 4]

```
a = np.array([1, 2, 4])
a = np.insert(a, 2, 3)
print("After insert:", a)
```

➦ After insert: [1 2 3 4]

```
a = np.array([5, 1, 3])
a_sorted = np.sort(a)
print("Sorted:", a_sorted)
```

➦ Sorted: [1 3 5]

```
a = np.array([5, 1, 3])
a_desc = np.sort(a)[::-1]
print("Descending sort:", a_desc)
```

➦ Descending sort: [5 3 1]

```
a = np.array([10, 20, 30, 40, 50])
a = np.delete(a, 3)
print("After deletion:", a)
```

➦ After deletion: [10 20 30 50]

```
a = np.array([1, 2])
b = np.array([3, 4])
c = np.concatenate((a, b))
print("Concatenated:", c)
```

➦ Concatenated: [1 2 3 4]

```
a = np.array([1, 2, 3, 4, 5])
a = np.delete(a, [1, 3])
print("After multiple deletions:", a)
```

↻ After multiple deletions: [1 3 5]

```
a = np.array([1, 2])
b = np.array([3, 4])
v = np.vstack((a, b))
print("Vertical stack:\n", v)
```

↻ Vertical stack:

```
[[1 2]
 [3 4]]
```

```
a = np.array([1, 2])
b = np.array([3, 4])
h = np.hstack((a, b))
print("Horizontal stack:", h)
```

↻ Horizontal stack: [1 2 3 4]

```
a = np.arange(10)
split = np.split(a, 2)
print("Split array:", split)
```

↻ Split array: [array([0, 1, 2, 3, 4]), array([5, 6, 7, 8, 9])]

```
a = np.arange(16).reshape(4, 4)
parts = np.vsplit(a, 2)
print("2D Split:\n", parts[0], "\n", parts[1])
```

↻ 2D Split:

```
[[0 1 2 3]
 [4 5 6 7]
 [ 8  9 10 11]
 [12 13 14 15]]
```

```
a = np.array([1, 2, 3])
b = np.array([1, 2, 3])
print("Arrays equal:", np.array_equal(a, b))
```

↻ Arrays equal: True

```
a = np.array([1, 2, 3])
b = a + 10
print("Broadcasting result:", b)
```

↻ Broadcasting result: [11 12 13]

```
a = np.array([1, 2])
b = np.array([3, 4])
print("Element-wise product:", a * b)
```

↻ Element-wise product: [3 8]

```
a = np.array([1, 2, 3])
try:
    a.reshape((2, 2))
except Exception as e:
    print("Error:", e)
```

↻ Error: cannot reshape array of size 3 into shape (2,2)

```
a = np.array([[10, 20, 30], [40, 50, 60]])  
print("Element at row 1, col 2:", a[1, 2])
```

↗ Element at row 1, col 2: 60

```
a = np.array([10, 5, 3, 15])  
print("Max:", np.max(a), "Min:", np.min(a))
```

↗ Max: 15 Min: 3