

```
import numpy as np
```

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

```
↗ [1 2 3 4 5]
```

```
arr = np.arange(0, 11)
print(arr)
```

```
↗ [ 0  1  2  3  4  5  6  7  8  9 10]
```

```
arr = np.random.randint(1, 101, size=(3, 3))
print(arr)
```

```
↗ [[64 75  4]
   [83 43 89]
   [70 38 52]]
```

```
print("Zeros:", np.zeros((2, 2)))
print("Ones:", np.ones((2, 2)))
print("Empty (random values):", np.empty((2, 2)))
```

```
↗ Zeros: [[0. 0.]
         [0. 0.]]
Ones: [[1. 1.]
       [1. 1.]]
Empty (random values): [[1. 1.]
                        [1. 1.]]
```

```
arr = np.random.rand(5)
print(arr)
```

```
↗ [0.48712538 0.00091961 0.14799843 0.48569993 0.82106591]
```

```
arr = np.arange(9)
reshaped = arr.reshape((3, 3))
print(reshaped)
```

```
↗ [[0 1 2]
   [3 4 5]
   [6 7 8]]
```

 **Generate**



Close

```
arr = np.array([4, 7, 2, 9, 5])
print("Max value:", np.max(arr))
```

```
↗ Max value: 9
```

```
arr = np.array([4, 7, 2, 9, 5])
print("Index of min value:", np.argmin(arr))
```

```
↗ Index of min value: 2
```

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
print("Shape:", arr.shape)
```

```
↗ Shape: (2, 3)
```

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

Size: 5

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

Dimensions: 2

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

Data type: int64

```
original = np.array([1, 2, 3])
copy = original.copy()
copy[0] = 100
print("Original:", original)
print("Copy:", copy)
```

Original: [1 2 3]
Copy: [100 2 3]

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

[1 2 3 4]

```
arr = np.array([1, 2, 3, 4])
arr = np.insert(arr, 2, 99)
print(arr)
```

[1 2 99 3 4]

```
arr = np.array([4, 2, 7, 1])
arr_sorted = np.sort(arr)
print(arr_sorted)
```

[1 2 4 7]

```
arr = np.array([4, 2, 7, 1])
arr_sorted_desc = np.sort(arr)[::-1]
print(arr_sorted_desc)
```

[7 4 2 1]

```
arr = np.array([10, 20, 30, 40, 50])
arr = np.delete(arr, 3)
print(arr)
```

[10 20 30 50]

```
arr = np.array([10, 20, 30, 40, 50, 60])
arr = np.delete(arr, [1, 3, 4])
print(arr)
```

[10 30 60]

```
a = np.array([1, 2])
b = np.array([3, 4])
combined = np.concatenate((a, b))
```

```
print(combined)
```

```
↵ [1 2 3 4]
```

```
a = np.array([1, 2])
b = np.array([3, 4])
v_stack = np.vstack((a, b))
print(v_stack)
```

```
↵ [[1 2]
   [3 4]]
```

```
a = np.array([1, 2])
b = np.array([3, 4])
h_stack = np.hstack((a, b))
print(h_stack)
```

```
↵ [1 2 3 4]
```

```
arr = np.arange(10)
split = np.split(arr, 2)
print(split)
```

```
↵ [array([0, 1, 2, 3, 4]), array([5, 6, 7, 8, 9])]
```

```
arr = np.arange(16).reshape(4, 4)
split = np.vsplit(arr, 2)
print(split)
```

```
↵ [array([[0, 1, 2, 3],
         [4, 5, 6, 7]]), array([[ 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
```

```
arr = np.array([1, 2, 3])
broadcasted = arr + 5
print(broadcasted)
```

```
↵ [6 7 8]
```


```
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
result = a * b
print(result)
```

```
↵ [ 4 10 18]
```


```
arr = np.arange(10)
try:
    arr.reshape((3, 4))
except ValueError as e:
    print("Error:", e)
```

```
↵ Error: cannot reshape array of size 10 into shape (3,4)
```

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

 Element at [1,2]: 60

```
arr = np.array([4, 8, 1, 9, 3])  
print("Max:", np.max(arr))  
print("Min:", np.min(arr))
```

 Max: 9
Min: 1