In [1]:
```
import numpy as np
```

In [2]:
```

array_1d = np.array([1, 2, 3, 4, 5])
print("1D Array:", array_1d)
array_2d = np.array([[1, 2, 3], [4, 5, 6]])
print("2D Array:\n", array_2d)
```

```
1D Array: [1 2 3 4 5]
2D Array:
 [[1 2 3]
 [4 5 6]]
```

In [4]:
```
zero_array = np.zeros((3, 4))
print(zero_array)
```

```
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
```

In [6]:
```
scalar_array = np.full((3, 3), 4)
print(scalar_array)
```

```
[[4 4 4]
 [4 4 4]
 [4 4 4]]
```

In [7]:
```
random_array = np.random.rand(3, 2)
print(random_array)
```

```
[[0.3674191  0.52785559]
 [0.57623038 0.27704469]
 [0.91477727 0.02147448]]
```

In [8]:
```
reshaped_array = np.reshape(array_2d, (3, 2))
print(reshaped_array)

flattened_array = array_2d.flatten()
print(flattened_array)
```

```
[[1 2]
 [3 4]
 [5 6]]
[1 2 3 4 5 6]
```

In [9]:
```
float_array = array_1d.astype(float)
print( float_array)
```

```
[1. 2. 3. 4. 5.]
```

In [10]:
```
sliced_array = array_2d[0, 1:]
print(sliced_array)


sliced_2d = array_2d[1:, :2]
print(sliced_2d)
```

```
[2 3]
[[4 5]]
```

In [15]:
```
array_a = np.array([[1, 4], [3, 9]])
array_b = np.array([[7, 6], [3, 8]])

joined_array = np.concatenate((array_a, array_b), axis=0)
print(joined_array)
```

```
[[1 4]
 [3 9]
 [7 6]
 [3 8]]
```

In [12]:
```
h_joined_array = np.hstack((array_a, array_b))
print(h_joined_array)
```

```
[[1 4 7 6]
 [3 9 3 8]]
```

In [13]:
```
v_joined_array = np.vstack((array_a, array_b))
print(v_joined_array)
```

```
[[1 4]
 [3 9]
 [7 6]
 [3 8]]
```

In [14]:
```
d_joined_array = np.dstack((array_a, array_b))
print(d_joined_array)
```

```
[[[1 7]
  [4 6]]

 [[3 3]
  [9 8]]]
```

In [17]:
```
unsorted_array = np.array([3, 1, 2, 5, 4])
sorted_array = np.sort(unsorted_array)
print(sorted_array)
```

```
[1 2 3 4 5]
```

In [18]:
```
filter_array = np.array([10, 20, 30, 40, 50])
filtered_elements = filter_array[filter_array < 30]
print(filtered_elements)
```

```
[10 20]
```

In [19]:
```
vector_a = np.array([1, 2, 3])
vector_b = np.array([4, 5, 6])

vector_add = vector_a + vector_b
print(vector_add)


vector_sub = vector_a - vector_b
print(vector_sub)


vector_mul = vector_a * vector_b
print(vector_mul)

vector_div = vector_a / vector_b
print(vector_div)
```

```
[5 7 9]
[-3 -3 -3]
[ 4 10 18]
[0.25 0.4  0.5 ]
```

In [20]:
```
scalar = 5

scalar_add = vector_a + scalar
print(scalar_add)

vectorized_mul = vector_a * 2
print(vectorized_mul)
```

```
[6 7 8]
[2 4 6]
```

In [ ]:
```

```