In [1]:

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

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

In [2]:

```
zero_array = np.zeros((3, 3))
print(zero_array)
```

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

In [3]:

```
scalar_array = np.full((3, 3), 7)
print(scalar_array)
```

```
[[7 7 7]
 [7 7 7]
 [7 7 7]]
```

In [4]:

```
random_array = np.random.random((3, 3))
print(random_array)
```

```
[[0.42311982 0.76131557 0.85355245]
 [0.54142233 0.02371802 0.62786925]
 [0.36197407 0.47523331 0.46876061]]
```

In [5]:

```
reshaped_array = random_array.reshape(1, 9)
flattened_array = reshaped_array.flatten()
print(reshaped_array)
print(flattened_array)
```

```
[[0.42311982 0.76131557 0.85355245 0.54142233 0.02371802 0.62786925
  0.36197407 0.47523331 0.46876061]]
[0.42311982 0.76131557 0.85355245 0.54142233 0.02371802 0.62786925
 0.36197407 0.47523331 0.46876061]
```

In [6]:

```
int_array = random_array.astype(int)
print(int_array)
```

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

In [7]:

```
sliced_array = arr[1:4]
print(sliced_array)
```

```
[2 3 4]
```

In [8]:

```
joined_array = np.concatenate((arr, arr))
print(joined_array)
```

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

In [9]:

```
horizontally_joined_array = np.hstack((arr, arr))
print(horizontally_joined_array)
```

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

In [10]:

```
vertically_joined_array = np.vstack((arr, arr))
print(vertically_joined_array)
```

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

In [11]:

```
depth_joined_array = np.dstack((arr, arr))
print(depth_joined_array)
```

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

In [12]:

```
index_retrieved = arr[2]
print("Index 2:", index_retrieved)

arr[2] = 10
print("Updated Array:", arr)
```

```
Index 2: 3
Updated Array: [ 1  2 10  4  5]
```

In [13]:

```
sorted_array = np.sort(arr)
print(sorted_array)
```

```
[ 1  2  4  5 10]
```

In [14]:

```
filtered_array = arr[arr > 2]
print(filtered_array)
```

```
[10  4  5]
```

In [15]:

```
vector_1 = np.array([1, 2, 3])
vector_2 = np.array([4, 5, 6])

addition = vector_1 + vector_2
subtraction = vector_1 - vector_2
multiplication = vector_1 * vector_2
division = vector_1 / vector_2

print("Addition:", addition)
print("Subtraction:", subtraction)
print("Multiplication:", multiplication)
print("Division:", division)
```

```
Addition: [5 7 9]
Subtraction: [-3 -3 -3]
Multiplication: [ 4 10 18]
Division: [0.25 0.4  0.5 ]
```

In [16]:

```
scalar_multiplication = vector_1 * 3
print("Scalar Multiplication:", scalar_multiplication)
```

```
Scalar Multiplication: [3 6 9]
```

In [17]:

```
vectorized_operation = np.add(vector_1, vector_2)
print("Vectorized Addition:", vectorized_operation)
```

```
Vectorized Addition: [5 7 9]
```

In [ ]: