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

In [3]:
```
#1 Declare the numpy array
arr_1 = np.array([1,2,3,4,5])
arr_2 = np.array([[1,2,3],[4,5,6]])
print(arr_1)
print(arr_2)
```

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

In [5]:
```
#2 create an array with full of zero values
b = np.zeros((4,4))
print(b)
```

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

In [34]:
```
#3 create an array with full of scalar values filled
arr = np.full((4,5),2)
print(arr)
```

```
[[2 2 2 2 2]
 [2 2 2 2 2]
 [2 2 2 2 2]
 [2 2 2 2 2]]
```

In [8]:
```
#4 create an array with scalar values filled
a = np.random.random((2,3))
print(a)
```

```
[[0.18916302 0.03875692 0.64422785]
 [0.34632101 0.13898375 0.07254455]]
```

In [9]:
```
#5 reshape and flattening the array
newarr = arr_2.reshape(3,2)
print(newarr)
```

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

In [11]:
```
flarr = arr_2.flatten()
print(flarr)
```

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

In [13]:
```
#6 convert an array from one type to another
e = np.arange(0,20,2)
anarr=e.astype('f')
print(anarr)
```

```
[ 0.  2.  4.  6.  8. 10. 12. 14. 16. 18.]
```

In [17]:

```
#7 slicing operation in an array
arr = np.array([[1,2,3,4],[5,2,4,2],[5,9,2,0],[1,2,0,1]])
print(arr[0:4:2])
print(arr[:2,:3])
print(arr[2:,2:])
print(arr[:2])
```

```
[[1 2 3 4]
 [5 9 2 0]]
[[1 2 3]
 [5 2 4]]
[[2 0]
 [0 1]]
[[1 2 3 4]
 [5 2 4 2]]
```

In [18]:

```
#8 join functions
x=np.array([1,2,3])
y = np.array([4,5,6])
z = np.concatenate((x,y))
print(x,y)
print(z)
```

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

In [19]:

```
hori = np.hstack((x,y))
print(hori)
```

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

In [20]:

```
vert = np.vstack((x,y))
print(vert)
```

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

In [22]:

```
#9 index retrivel and basic operation
arr = np.array([1,2,3,4,5,4,4])
print(arr)
x = np.where(arr==4)
print(x)
```

```
[1 2 3 4 5 4 4]
(array([3, 5, 6], dtype=int64),)
```

In [27]:

```
#10 sorting of array
arr = np.array([[30,20,40],[50,0,11]])
print(arr)
print(np.sort(arr))
```

```
[[30 20 40]
 [50  0 11]]
[[20 30 40]
 [ 0 11 50]]
```

In [28]:

```
#11 filtering operation based on array value
arr = np.array([44,43,45,67,89])
filter_arr = arr>43
newarr = arr[filter_arr]
print("\n original array:",arr)
print("\n filter array:condition - > 43")
print("\nnew array:",newarr)
```

```

 original array: [44 43 45 67 89]

 filter array:condition - > 43

new array: [44 45 67 89]
```

In [30]:

```
#12 vector operation
ar1 = np.array([1,2,3,4,5])
ar2 = np.array([2,4,5,3,9])
print("\n vector addition")
print(ar1 + ar2)
print("\n vector subtraction")
print(ar1 - ar2)
print("\n vector multiplication")
print(ar1 * ar2)
print("\n vector division")
print(ar1/ar2)
```

```

 vector addition
[ 3  6  8  7 14]

 vector subtraction
[-1 -2 -2  1 -4]

 vector multiplication
[ 2  8 15 12 45]

 vector division
[0.5        0.5        0.6        1.33333333 0.55555556]
```

In [32]:

```
#13 scalar operation and vectorize operation
a = np.array([1,3,4,5,6])
b = np.array([2,4,6,7,9])
c = np.array([1,24,5,6,7,8])
print(c*3)

def my_fun(x,y):
    if(x>y):
        retrunx-y
    else:
        return x+y
vec_function = np.vectorize(my_fun)
print(vec_function(a,b))
```

```
[ 3 72 15 18 21 24]
[ 3  7 10 12 15]
```

In [ ]: