In [1]:

```
# Load libraries
import numpy as np
```

In [2]:

```
# Create a random array with given dimension
array = np.random.rand(6,4)
# Print it for checking
print(np.shape(array))
print(array)
```

```
(6, 4)
[[0.51064131 0.41821853 0.11066622 0.80089847]
 [0.51714368 0.03258209 0.83950015 0.12597796]
 [0.39489994 0.04145781 0.96385097 0.06945781]
 [0.47433235 0.77882002 0.5348074  0.98744227]
 [0.56463454 0.0060091  0.90595607 0.05451798]
 [0.49693528 0.54350653 0.08538033 0.78033091]]
```

In [3]:

```
# Let us use operations like mean, sum,max and create new matrices
mean_rows=np.mean(array, axis = 0).reshape(1,4)
mean_col=np.mean(array, axis = 1).reshape(6,1)
max_rows=np.max(array, axis = 0).reshape(1,4)
max_col=np.max(array, axis = 1).reshape(6,1)
min_rows=np.min(array, axis = 0).reshape(1,4)
min_col=np.min(array, axis = 1).reshape(6,1)
sum_rows=np.sum(array, axis = 0).reshape(1,4)
sum_col=np.sum(array, axis = 1).reshape(6,1)
# A 2x4 array made by combinig mean and max of rows
print("Combination of mean and max of rows:")
new_array1=np.concatenate((mean_rows,max_rows), axis=0)
print('Shape:',np.shape(new_array1))
print(new_array1)
# A 4x4 array made by combining mean,max,min,sum of all rows
print("Combination of mean,max,min,sum of all rows:")
new_array2=np.concatenate((mean_rows,max_rows,min_rows,sum_rows), axis=0)
print('Shape:',np.shape(new_array2))
print(new_array2)
# A 6x4 made by combining all operations on columns
print("Combination of mean,max,min,sum of all columns:")
new_array3=np.concatenate((mean_col,max_col,min_col,sum_col), axis=1)
print('Shape:',np.shape(new_array3))
print(new_array3)
# Until now we only made matrices whose size was smaller than the original matrix
# Now let us define a 10x4 matrix by combining array3 and array4
new_array4=np.concatenate((new_array2,new_array3),axis=0)
print('Shape:',np.shape(new_array4))
print(new_array4)
```

```
Combination of mean and max of rows:
Shape: (2, 4)
[[0.49309785 0.30343235 0.57336019 0.4697709 ]
 [0.56463454 0.77882002 0.96385097 0.98744227]]
Combination of mean,max,min,sum of all rows:
Shape: (4, 4)
[[0.49309785 0.30343235 0.57336019 0.4697709 ]
 [0.56463454 0.77882002 0.96385097 0.98744227]
 [0.39489994 0.0060091  0.08538033 0.05451798]
 [2.9585871  1.82059409 3.44016115 2.8186254 ]]
Combination of mean,max,min,sum of all columns:
Shape: (6, 4)
[[0.46010613 0.80089847 0.11066622 1.84042454]
 [0.37880097 0.83950015 0.03258209 1.51520388]
 [0.36741663 0.96385097 0.04145781 1.46966654]
 [0.69385051 0.98744227 0.47433235 2.77540205]
 [0.38277942 0.90595607 0.0060091  1.5311177 ]
 [0.47653826 0.78033091 0.08538033 1.90615305]]
```

```
Shape: (10, 4)
[[0.49309785 0.30343235 0.57336019 0.4697709 ]
 [0.56463454 0.77882002 0.96385097 0.98744227]
 [0.39489994 0.0060091  0.08538033 0.05451798]
 [2.9585871  1.82059409 3.44016115 2.8186254 ]
 [0.46010613 0.80089847 0.11066622 1.84042454]
 [0.37880097 0.83950015 0.03258209 1.51520388]
 [0.36741663 0.96385097 0.04145781 1.46966654]
 [0.69385051 0.98744227 0.47433235 2.77540205]
 [0.38277942 0.90595607 0.0060091  1.5311177 ]
 [0.47653826 0.78033091 0.08538033 1.90615305]]
```

In [4]:

```
# Test case 1 ---  array(0,1)
test=array[0,1]
print(np.shape(test))
print(np.ndim(test))
```

```
()
0
```

In [5]:

```
# Test case 2 ---  array(4,3)
test=array[4,3]
print(np.shape(test))
print(np.ndim(test))
```

```
()
0
```

In [6]:

```
# Test case 3 ---  array(:,1)
test=array[:,1]
print(np.shape(test))
print(np.ndim(test))
```

```
(6,)
1
```

In [7]:

```
# Test case 4 ---  array(1,:)
test=array[1,:]
print(np.shape(test))
print(np.ndim(test))
```

```
(4,)
1
```

In [8]:

```
# Test case 5 ---  array(3,:)
test=array[:,1]
print(np.shape(test))
print(np.ndim(test))
```

```
(6,)
1
```

In [9]:

```
# Test case 6 ---  array(1:4,:)
test=array[1:4,:]
print(test)
print(np.shape(test))
```

```
print(np.shape(test))
print(np.ndim(test))
```

```
[[0.51714368 0.03258209 0.83950015 0.12597796]
 [0.39489994 0.04145781 0.96385097 0.06945781]
 [0.47433235 0.77882002 0.5348074  0.98744227]]
(3, 4)
2
```

In [10]:

```
# Test case 7 ---  array(:,0:3)
test=array[:,0:3]
print(test)
print(np.shape(test))
print(np.ndim(test))
```

```
[[0.51064131 0.41821853 0.11066622]
 [0.51714368 0.03258209 0.83950015]
 [0.39489994 0.04145781 0.96385097]
 [0.47433235 0.77882002 0.5348074 ]
 [0.56463454 0.0060091  0.90595607]
 [0.49693528 0.54350653 0.08538033]]
(6, 3)
2
```

In [ ]:

```

```