Numpy Library

- Processing N-dimensional arrays

In [2]:
```
import numpy as np
li=[1,2,3,'z']
a=np.array(li)

b=np.arange(15)  #array range
b
```

Out[2]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14])
```

In [3]:
```
rn=np.random.randint(0,100,size=10)
rn
```

Out[3]:
```
array([77, 70, 19, 34, 25, 39, 53, 88, 22,  7])
```

In [4]:
```
m2=np.random.randint(0,2,size=(3,3))
m2
```

Out[4]:
```
array([[1, 1, 0],
       [1, 0, 0],
       [0, 1, 0]])
```

In [5]:
```
m3=np.random.randint(0,2,size=(3,3,3))
m3
```

Out[5]:
```
array([[[1, 1, 0],
        [0, 1, 0],
        [1, 1, 1]],

       [[0, 1, 1],
        [0, 0, 1],
        [1, 1, 1]],

       [[0, 0, 0],
        [1, 1, 1],
        [0, 1, 0]]])
```

In [10]:
```
m4=np.random.randint(0,2,size=(3,3,3,3))
m4
m4[2][2][2][2]
```

Out[10]:
```
array([[[[0, 0, 1],
         [1, 1, 0],
         [1, 0, 0]],

        [[0, 0, 0],
         [1, 1, 0],
         [1, 0, 1]],

        [[0, 1, 0],
         [1, 0, 1],
         [0, 1, 1]]],


       [[[1, 1, 1],
         [1, 0, 0],
         [0, 0, 1]],

        [[1, 1, 1],
         [0, 1, 0],
         [1, 0, 1]],

        [[1, 1, 0],
         [1, 1, 1],
         [0, 1, 1]]],


       [[[0, 0, 1],
         [1, 0, 1],
         [0, 0, 0]],

        [[1, 1, 1],
         [0, 0, 0],
         [1, 0, 0]],

        [[1, 1, 1],
         [1, 0, 0],
         [1, 0, 0]]]])
```

In [11]:
```
m4.ndim
m4.size
m4.shape
m4.dtype
m4.itemsize
m4.nbytes
```

Out[11]:
```
324
```

In [12]:
```
print(b)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14]
```

In [16]:
```
1 b.reshape(5,3)
```

Out[16]:
```
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11],
       [12, 13, 14]])
```

In [17]:
```
1 c=b.reshape(5,3)
```

In [18]:
```
1 d=c+1
2 d
```

Out[18]:
```
array([[ 1,  2,  3],
       [ 4,  5,  6],
       [ 7,  8,  9],
       [10, 11, 12],
       [13, 14, 15]])
```

In [19]:
```
1 m=np.ones((3,3))
2 print(m)
```

```
[[1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]]
```

Pandas

Use cases

- Data cleaning
- Data Transformation
- Data Analysis

Notations

- series
- Data frames

In [4]:
```
1 import pandas as pd
2 internal1={'s1':21,'s2':18,'s3':24}
3 internal1=pd.Series(internal1)
4 internal2={'s1':15,'s2':25,'s3':12}
5 internal2=pd.Series(internal2)
6 internal2
```

Out[4]:
```
s1    15
s2    25
s3    12
dtype: int64
```

In [5]:
```
final={'Internal1':internal1,'Internal2':internal2}
final=pd.DataFrame(final)
final
```

Out[5]:

	Internal1	Internal2
s1	21	15
s2	18	25
s3	24	12

In [6]:
```
final.columns  #names of all colmns
```

Out[6]:
```
Index(['Internal1', 'Internal2'], dtype='object')
```

In [7]:
```
final.values
```

Out[7]:
```
array([[21, 15],
       [18, 25],
       [24, 12]], dtype=int64)
```

In [28]:
```
final.values[2]
```

Out[28]:
```
array([12, 25], dtype=int64)
```

In [9]:
```
final.values[2,0]=25
final.values[2][0]
```

Out[9]:
```
25
```

In [10]:
```
for row in final.values:
    print('Internal1-',row[0],',Internal2-',row[1])
```

```
Internal1- 21 ,Internal2- 15
Internal1- 18 ,Internal2- 25
Internal1- 25 ,Internal2- 12
```

In [14]:
```
final.loc['s4']=[15,23]
final
```

Out[14]:

	Internal1	Internal2
s1	21	15
s2	18	25
s3	25	12
s4	15	23

In [22]:
```
final.values[2]=[12,25]
```

In [ ]:
```

```

In [16]:
```
# Reading CSV file data
import pandas as pd
filepath='DataFiles/income.csv'
incomedf=pd.read_csv(filepath)
incomedf
```

Out[16]:

	GEOID	State	2005	2006	2007	2008	2009	2010	2011	2012	2013
0	04000US01	Alabama	37150	37952	42212	44476	39980	40933	42590	43464	41381
1	04000US02	Alaska	55891	56418	62993	63989	61604	57848	57431	63648	61137
2	04000US04	Arizona	45245	46657	62993	46914	45739	46896	48621	47044	50602
3	04000US05	Arkansas	36658	37057	40795	39586	36538	38587	41302	39018	39919
4	04000US06	California	51755	55319	55734	57014	56134	54283	53367	57020	57528

In [17]:
```
# Extract  income of all states in 2013
#calfornia :57528
for row in incomedf.values:
    print(row[1],':',row[-1])
```

```
Alabama : 41381
Alaska : 61137
Arizona : 50602
Arkansas : 39919
California : 57528
```

In [20]:
```
# average income of  Arizona
sum=0
for i in range(2,11):
    sum+=incomedf.values[2][i]
print(sum/len(incomedf.values[2][2:]))
```

```
48967.88888888889
```

In [19]:
```
#average income of all states in 2012

sum=0
for row in incomedf.values:
    sum+=row[-2]
sum/len(incomedf.values)
```

Out[19]: 50038.8

In [ ]:
```

```