Numpy

- General purpose array processing package
- Fundamental package of scientific computation with python
- Considered as an efficient ND array container of python
- Array
  - Talbe of elements
  - No.of dimensions of the array is called as rank of the array
  - Arrays can be processed by using lists and tuples in python

Installation of numpy using cmd:

- pip install numpy
- import numpy

Working with Numpy

In [1]:
```
import numpy as np
a=np.array([1,2,3,4,8])  # creation of array using list
print(a)
```

```
[1 2 3 4 8]
```

In [29]:
```
c=np.array((34,7,7))  # Creation using tuple
print(c)
```

```
[34  7  7]
```

In [2]:
```
b=np.array([[7,786,566],[54,56,34]])  # Using nested lists
print(b)
```

```
[[  7 786 566]
 [ 54  56  34]]
```

In [3]:
```
print(a+1)
```

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

In [4]:
```
print(b-2)
```

```
[[  5 784 564]
 [ 52  54  32]]
```

In [7]:
```
print(a)
```

```
[1 2 3 4 8]
```

In [8]:
```
print(b)
```

```
[[  7 786 566]
 [ 54  56  34]]
```

In [70]:
```
print(a+b)
```

```
[[ 12 831 655]
 [ 99 132  68]]
```

In [10]:
```
a=np.array([[5,45,89],[45,76,34]])
print(a)
```

```
[[ 5 45 89]
 [45 76 34]]
```

In [11]:
```
print(a+b)  # adding two arrays in general
```

```
[[ 12 831 655]
 [ 99 132  68]]
```

In [12]:
```
print(a.sum())   # summing the all elements in an array
```

```
294
```

In [13]:
```
print(b.sum())   # using pre-defined fun()
```

```
1503
```

In [71]:
```
x=np.array([1,2,4])   # Printing data type of array
print(x.dtype)
```

```
int32
```

In [ ]:
```

```

Array Slicing()

In [14]:
```
print(a[1:])
```

```
[[45 76 34]]
```

In [18]:
```
print(b[::])
```

```
[[  7 786 566]
 [ 54  56  34]]
```

In [20]:
```
print(b[:1])
```

```
[[  7 786 566]]
```

In [22]:
```
x=np.array([1,2,4])
print(x.dtype)
```

```
int32
```

In [24]:
```
y=np.array([67.0,56.78,34.67,56])
print(y.dtype)
```

```
float64
```

In [26]:
```
z=np.array(["name","age","des",8])
print(z.dtype)
```

```
<U4
```

In [34]:
```
q = np.array([1, 2], dtype = np.int64)
print(q.dtype)
print(q)
```

```
int64
[1 2]
```

Math opeations in Numpy

In [39]:
```
ar=np.array([[1,2,3],[4,5,6]],dtype=np.int64)
print(ar)
```

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

In [41]:
```
ar2=np.array([[56.5,78.45,45],[78.34,67.5,56.90]],dtype=np.float64)
print(ar2)
```

```
[[56.5  78.45 45.  ]
 [78.34 67.5  56.9 ]]
```

In [42]:
```
print(ar.sum())
```

```
21
```

In [43]:
```
print(ar2.sum())
```

```
382.68999999999994
```

In [46]:
```
print(np.sum(ar2))
```

```
382.68999999999994
```

In [44]:
```
print(ar+ar2)
```

```
[[57.5  80.45 48.  ]
 [82.34 72.5  62.9 ]]
```

In [45]:
```
print(np.add(ar,ar2))
```

```
[[57.5  80.45 48.  ]
 [82.34 72.5  62.9 ]]
```

In [47]:
```
srt=np.sqrt(ar)
print(srt)
```

```
[[1.         1.41421356 1.73205081]
 [2.         2.23606798 2.44948974]]
```

In [48]:
```
h=np.array([34.67,78.78])
print(h)
```

```
[34.67 78.78]
```

In [49]:
```
print(np.sqrt(h))
```

```
[5.88812364 8.87580982]
```

In [53]:
```
print(ar.T)
```

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

In [57]:
```
r=np.random.rand(5)
print(r)
```

```
[0.16097096 0.48522677 0.68997848 0.49274216 0.03891745]
```

In [61]:
```
r2=np.random.randn(4)
print(r2)
```

```
[-0.37981751 -0.11254133  0.09139713  0.22432151]
```

In [66]:
```
r3=np.random.rand(3,2)
print(r3)
```

```
[[0.19434366 0.86584298]
 [0.35164976 0.95592323]
 [0.36491948 0.05332592]]
```

In [69]:
```
r4=np.random.rand(8,2,3)
print(r4)
```

```
[[[0.5412761  0.70884446 0.57074888]
  [0.79261831 0.4208052  0.2444365 ]]

 [[0.3097922  0.41684785 0.34229438]
  [0.04443333 0.27267997 0.33621945]]

 [[0.35921192 0.23332446 0.454282  ]
  [0.56572008 0.78227919 0.69606159]]

 [[0.93084507 0.74570367 0.16995026]
  [0.10988689 0.79088305 0.60303662]]

 [[0.8216267  0.12243226 0.82296121]
  [0.66017546 0.87923151 0.46268386]]

 [[0.65720697 0.26437285 0.63806292]
  [0.74364828 0.89133821 0.03381354]]

 [[0.09122139 0.87956712 0.73841506]
  [0.95175159 0.69355787 0.64755512]]

 [[0.36711358 0.60232769 0.17068771]
  [0.80798573 0.25605996 0.21967618]]]
```

In [ ]:
```

```