In [2]:
```
import numpy
print(numpy.__version__)
```

```
1.19.5
```

Numpy DataTypes

1. Int
2. float
3. Complex
4. Bool(0/1)

Numpy array creation

1-D Arrays

In [3]:
```
import numpy as np

b = [5,7,12]
a = np.array(b)
a
```

Out[3]:
```
array([ 5,  7, 12])
```

In [4]:
```
# Instead of python range for lists there is arange for numpy array
a = np.arange(1,12,2)
```

In [5]:
```
a
```

Out[5]:
```
array([ 1,  3,  5,  7,  9, 11])
```

In [6]:

```
# Create a linearly spaced values numpy array

a = np.linspace(1,10,5)
```

In [7]:

```
a
```

Out[7]:

```
array([ 1.  ,  3.25,  5.5 ,  7.75, 10.  ])
```

2-D Arrays

In [8]:

```
a = np.array([2,4,5,7,8,9])
a

c = a.reshape(3,2)
```

In [9]:

```
c.shape
```

Out[9]:

```
(3, 2)
```

In [10]:

```
a.shape
```

Out[10]:

```
(6,)
```

In [11]:

```
c.size == a.size
```

Out[11]:

```
True
```

In [12]:

```
## create a np array with all zeros

a = np.zeros((3,2),np.int16)
a
```

Out[12]:

```
array([[0, 0],
       [0, 0],
       [0, 0]], dtype=int16)
```

In [13]:

```
## create a np array with all ones

a = np.ones((3,7))
a
```

Out[13]:

```
array([[1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1.]])
```

Accessing Elements of a numpy array

In [14]:

```
a = np.array([2,4,5,7,8,9])
a = a.reshape(3,2)
```

In [15]:

```
a[0,1]
```

Out[15]:

```
4
```

In [16]:

```
a = np.array([[2,4,5],[7,8,9],[1,3,11]])
```

In [17]:

```
a.shape
```

Out[17]:

```
(3, 3)
```

In [18]:

```
a[1,2]
```

Out[18]:

```
9
```

In [19]:

```
a[2,1]
```

Out[19]:

```
3
```

In [20]:

```
a = np.array([2,4,5,7,8,9,11,13])
a = a.reshape(2,2,2)
```

In [21]:

```
a.shape
```

Out[21]:

```
(2, 2, 2)
```

In [22]:

```
a
```

Out[22]:

```
array([[[ 2,  4],
        [ 5,  7]],

       [[ 8,  9],
        [11, 13]]])
```

In [23]:

```
a[1][0][1]
```

Out[23]:

```
9
```

Numpy Random function

In [78]:

```
## create random integer in range
np.random.randint(100)
```

Out[78]:

```
4
```

In [79]:

```
## Generate a random number between 0 and 1
np.random.rand()
```

Out[79]:

```
0.7641230873011903
```

In [81]:

```
np.random.randint(100,size = (3,2))
```

Out[81]:

```
array([[37, 42],
       [56, 88],
       [46, 96]])
```

In [82]:

```
np.random.rand(5)
```

Out[82]:

```
array([0.49867237, 0.28789183, 0.76651746, 0.5934329 , 0.52978275])
```

In [88]:

```
np.random.rand(2,5)
```

Out[88]:

```
array([[0.09376187, 0.71054587, 0.75925363, 0.35034312, 0.38294943],
       [0.51757716, 0.80439525, 0.23861353, 0.49783967, 0.08573015]])
```

In [86]:

```
np.random.choice([1,9,8,6,5])
```

Out[86]:

```
8
```

In [89]:

```
np.random.choice([1,9,8,6,5],size = (5,5))
```

Out[89]:

```
array([[5, 5, 6, 9, 9],
       [8, 9, 5, 6, 6],
       [6, 1, 1, 6, 5],
       [1, 9, 6, 8, 5],
       [1, 5, 1, 8, 8]])
```

Numpy Stacking

In [28]:

```
a = np.array([[1,2],
             [4,5]])
b = np.array([[6,7],
              [8,9]])


```

In [33]:

```
## Vertical Stacking
np.vstack((a,b))
```

Out[33]:

```
array([[1, 2],
       [4, 5],
       [6, 7],
       [8, 9]])
```

In [34]:

```
## Horizontal Stacking
np.hstack((a,b))
```

Out[34]:

```
array([[1, 2, 6, 7],
       [4, 5, 8, 9]])
```

In [39]:

```
## Column Stacking

np.column_stack((a,b))
```

Out[39]:

```
array([[1, 2, 6, 7],
       [4, 5, 8, 9]])
```

In [41]:

```
## Concatenate Function

np.concatenate((a,b),1)
```

Out[41]:

```
array([[1, 2, 6, 7],
       [4, 5, 8, 9]])
```

Splitting

In [42]:

```
a = np.array([[1,3,7,9,11,13],
             [2,4,8,10,12,14]])
```

In [43]:

```
a
```

Out[43]:

```
array([[ 1,  3,  7,  9, 11, 13],
       [ 2,  4,  8, 10, 12, 14]])
```

In [55]:

```
## Hortizontal Split
c = np.hsplit(a,2)
```

In [58]:

```
c
```

Out[58]:

```
[array([[1, 3, 7],
        [2, 4, 8]]),
 array([[ 9, 11, 13],
        [10, 12, 14]])]
```

In [57]:

```
np.array(c).shape
```

Out[57]:

```
(2, 2, 3)
```

In [61]:

```
## Vertical Split

c = np.vsplit(a,2)
```

In [64]:

```
np.array(c)
```

Out[64]:

```
array([[[ 1,  3,  7,  9, 11, 13]],

       [[ 2,  4,  8, 10, 12, 14]]])
```

Broadcasting

In [66]:

```
a = np.array([1,3,5,6])

b = 2

a*b
```

Out[66]:

```
array([ 2,  6, 10, 12])
```

In [67]:

```
a+b
```

Out[67]:

```
array([3, 5, 7, 8])
```

In [69]:

```
### Broadcasting
a = np.array([1,3,5,6])
b = np.array([2,2,2,2])

a*b
```

Out[69]:

```
array([ 2,  6, 10, 12])
```

In [70]:

```
a+b
```

Out[70]:

```
array([3, 5, 7, 8])
```

In [78]:

```
a = np.array([0,10,20,30,40])
b = np.array([0,1,2])

b + a[:,np.newaxis]
```

Out[78]:

```
array([[ 0,  1,  2],
       [10, 11, 12],
       [20, 21, 22],
       [30, 31, 32],
       [40, 41, 42]])
```

In [81]:
```
b[:,np.newaxis]  + a
```

Out[81]:
```
array([[ 0, 10, 20, 30, 40],
       [ 1, 11, 21, 31, 41],
       [ 2, 12, 22, 32, 42]])
```

Matrix Attributes

In [83]:
```
a  = np.array([[6,1,1],
             [4,-2,5],
             [2,8,7]])
```

In [85]:
```
np.linalg.matrix_rank(a)
```

Out[85]:
```
3
```

In [86]:
```
np.trace(a)
```

Out[86]:
```
11
```

In [87]:
```
np.linalg.det(a)
```

Out[87]:
```
-306.0
```

In [88]:
```
np.linalg.inv(a)
```

Out[88]:
```
array([[ 0.17647059, -0.00326797, -0.02287582],
       [ 0.05882353, -0.13071895,  0.08496732],
       [-0.11764706,  0.1503268 ,  0.05228758]])
```

In [ ]:
```

```