

```
In [1]: pwd #present working directory
```

```
Out[1]: 'C:\\Users\\ANKIT SINGH'
```

```
In [2]: ls #list files/directories here
```

Volume in drive C has no label.
Volume Serial Number is F84E-DF1D

Directory of C:\Users\ANKIT SINGH

20-Jun-19	09:52 PM	<DIR>	.
20-Jun-19	09:52 PM	<DIR>	..
05-Mar-19	02:39 PM	<DIR>	.anaconda
21-Mar-19	05:28 PM	<DIR>	.atom
20-Jun-19	09:53 PM	<DIR>	.conda
20-Jun-19	05:32 PM		43 .condarc
06-Mar-19	08:26 PM	<DIR>	.dia
14-Apr-19	01:25 AM		208 .gitconfig
30-Apr-19	07:32 PM	<DIR>	.idlerc
20-Jun-19	09:52 PM	<DIR>	.ipynb_checkpoints
15-Apr-19	05:51 PM	<DIR>	.ipython
08-Mar-19	12:50 PM	<DIR>	.jssc
15-Apr-19	07:05 PM	<DIR>	.jupyter
20-Jun-19	06:40 PM	<DIR>	.matplotlib
08-Mar-19	12:02 PM	<DIR>	.oracle_jre_usage
29-May-19	08:22 AM	<DIR>	.VirtualBox
18-Jun-19	06:37 PM	<DIR>	.vscode
14-Apr-19	01:07 AM	<DIR>	3D Objects
21-Apr-19	10:15 PM	<DIR>	Anaconda3
20-Jun-19	05:38 PM		6,765 Class and instances.ipynb
14-Apr-19	01:07 AM	<DIR>	Contacts
20-Jun-19	09:52 PM		72 DAY 1.ipynb
20-Jun-19	04:41 PM		56,495 DAY 2.ipynb
20-Jun-19	07:58 PM		88,452 DAY 3.ipynb
20-Jun-19	04:55 PM	<DIR>	Desktop
02-Jun-19	11:09 AM	<DIR>	Documents
18-Jun-19	05:25 PM	<DIR>	Downloads
14-Apr-19	01:07 AM	<DIR>	Favorites
20-Jun-19	04:43 PM		263,988 home practice.ipynb
14-Apr-19	01:07 AM	<DIR>	Links
21-Apr-19	08:24 PM	<DIR>	Music
19-Jun-19	03:47 PM		16,601 mylistnp.ipynb
18-Jun-19	05:24 PM		15,161 Mylistttest.ipynb
20-Jun-19	04:59 PM	<DIR>	OneDrive
28-Apr-19	02:06 AM	<DIR>	Pictures

```
14-Apr-19 01:07 AM <DIR> Saved Games
14-Apr-19 01:07 AM <DIR> Searches
20-Jun-19 09:52 PM 3,933 Untitled2.ipynb
16-May-19 07:29 PM <DIR> Videos
28-May-19 09:38 PM <DIR> VirtualBox VMs
      10 File(s)      451,718 bytes
      30 Dir(s)  83,889,623,040 bytes free
```

```
In [3]: a=5 #integer variable
        print(type(a))
```

<class 'int'>

```
In [4]: a="Ankit Singh" #string
        print(type(a))
```

<class 'str'>

```
In [5]: a=[11,12,13,14,15,16]
        b=(17,18,19,20)
        c=[1,2]
```

```
In [6]: print(a)
```

[11, 12, 13, 14, 15, 16]

```
In [9]: a[0] #first element
```

Out[9]: 11

```
In [10]: a[5] #last element
```

Out[10]: 16

```
In [11]: a[-1] #element from last
```

Out[11]: 16

```
In [12]: a[-3] #3rd element from last
```

```
Out[12]: 14
```

```
In [25]: #methods  
del a
```

```
In [27]: a=[11,12,13,14,15,16]
```

```
In [29]: a.append(17) #appends or add 17 at the end
```

```
In [30]: a
```

```
Out[30]: [11, 12, 13, 14, 15, 16, 17]
```

```
In [31]: a.extend(c) #extends or add c at the end  
a
```

```
Out[31]: [11, 12, 13, 14, 15, 16, 17, 1, 2]
```

```
In [32]: a.insert(2,222) #insert 222 at position 2  
a
```

```
Out[32]: [11, 12, 222, 13, 14, 15, 16, 17, 1, 2]
```

```
In [33]: a[: -1] #print till last not taken the last
```

```
Out[33]: [11, 12, 222, 13, 14, 15, 16, 17, 1]
```

```
In [34]: a[2:6] #prints from index 2 to 5 (always till n-1)
```

```
Out[34]: [222, 13, 14, 15]
```

```
In [35]: #a[start:end:step size]
a[::1]
```

```
Out[35]: [11, 12, 222, 13, 14, 15, 16, 17, 1, 2]
```

```
In [36]: a[:]
```

```
Out[36]: [11, 12, 222, 13, 14, 15, 16, 17, 1, 2]
```

```
In [37]: a[::4]
```

```
Out[37]: [11, 14, 1]
```

```
In [49]: #numpy provides a large set of numeric datatypes
#that you can use to construct arrays
#(basically used for scientific calculations)

#mynp is optional argument to explicitly specify the datatype

import numpy as mynp
import numpy as np
```

```
In [50]: a=[11,12,13,14,15]
```

```
In [51]: print(a)
print(type(a))
print(len(a))

[11, 12, 13, 14, 15]
<class 'list'>
5
```

```
In [52]: xr=mynp.array(a) #this line converts list a into ndarray
print(type(xr))
print(len(xr))

<class 'numpy.ndarray'>
5
```

```
In [53]: xr
```

```
Out[53]: array([11, 12, 13, 14, 15])
```

```
In [54]: len(xr)
```

```
Out[54]: 5
```

```
In [55]: xr.ndim #used to find how many dimensions
```

```
Out[55]: 1
```

```
In [56]: xr.shape
```

```
Out[56]: (5,)
```

```
In [57]: xr.size
```

```
Out[57]: 5
```

```
In [58]: xr.dtype #show the data type
```

```
Out[58]: dtype('int32')
```

```
In [64]: d=[1.4,2.4,3.4,4.7,5.2,6.9] #data type is float (only single element needed to prove float)
e=[1,4,5,67,8] #data type is integer
z=np.array(d)
h=np.array(e)
print(type(d))
print(type(z))
print(z.dtype)

print(type(e))
print(type(h))
print(h.dtype)
```

```
<class 'list'>
<class 'numpy.ndarray'>
float64
<class 'list'>
<class 'numpy.ndarray'>
int32
```

```
In [67]: a=[[11,12,13],[14,15,16],[17,18,19],[20,21,22]]
a
```

```
Out[67]: [[11, 12, 13], [14, 15, 16], [17, 18, 19], [20, 21, 22]]
```

```
In [68]: ar=mynp.array(a) #ar is 2D array
ar
```

```
Out[68]: array([[11, 12, 13],
               [14, 15, 16],
               [17, 18, 19],
               [20, 21, 22]])
```

```
In [69]: len(ar) #shows Length
```

```
Out[69]: 4
```

```
In [70]: ar.size #shows n*m
```

```
Out[70]: 12
```

```
In [71]: #ar[row,column]
         #ar[start:end:stepSize,start:end:stepSize]  #1st for row 2nd for column
```

```
In [72]: ar[0]  #0th row
```

```
Out[72]: array([11, 12, 13])
```

```
In [74]: ar[0,:]  #0th row same o/p as above
```

```
Out[74]: array([11, 12, 13])
```

```
In [75]: ar[:,1]  #all rows but only index 1 column
```

```
Out[75]: array([12, 15, 18, 21])
```

```
In [76]: ar[2:4,1:3]  #2 to 4 index in row and 1 to 3 index in column
```

```
Out[76]: array([[18, 19],
               [21, 22]])
```

```
In [77]: ar[2:,1:]  #same as above
```

```
Out[77]: array([[18, 19],
               [21, 22]])
```

```
In [78]: ar[0:4:3,0:3:2]  #index 0 to 4 in row stepSize 3
                        #index 0 to 3 in column stepSize 2
```

```
Out[78]: array([[11, 13],
               [20, 22]])
```