In [2]:

```
def diff(a,b,c):
    dif1=abs(a-b)
    dif2=abs(b-c)
    dif3=abs(a-c)
    return dif1,dif2,dif3
diff(1,2,3)
```

Out[2]:

```
(1, 1, 2)
```

In [4]:

```
x=True
print(type(x))
```

```
<class 'bool'>
```

In [7]:

```
def can_vote(age):
    if age>18:
        return True
    else:
        return False
can_vote(17)
```

Out[7]:

```
False
```

In [8]:

```
3==3
```

Out[8]:

```
True
```

In [9]:

```
'3'==3
```

Out[9]:

```
False
```

In [11]:

```
3.0==3
```

Out[11]:

```
True
```

In [15]:

```
def can_vote(age,indian_citizen):
    if age>18 and indian_citizen==True:
        return 'Yes'
    else:
        return 'No'
can_vote(19,False)
```

Out[15]:

```
'No'
```

In [17]:

```
can_vote(12,True)
```

Out[17]:

```
'No'
```

In [18]:

```
can_vote(19,True)
```

Out[18]:

```
'Yes'
```

Data structures

List

In [19]:

```
list=[1,2,3,4,5]
list
```

Out[19]:

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

In [20]:

```
list.append([1,2,3])
```

In [21]:

```
list
```

Out[21]:

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

In [22]:

```
list.remove([1,2,3])
```

In [23]:

```
1 list
```

Out[23]:

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

In [25]:

```
list.insert(5,6)
```

In [26]:

```
list
```

Out[26]:

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

In [29]:

```
del list
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-29-604ec154886e> in <module>
----> 1 del list

NameError: name 'list' is not defined
```

In [31]:

```
l=[1,2,3,4,5]
l[3]
```

Out[31]:

```
4
```

In [32]:

```
l[::]
```

Out[32]:

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

In [38]:

```
l[-1:]
```

Out[38]:

```
[5]
```

In [39]:

```
l[::-1]
```

Out[39]:

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

In [40]:

```
l[0:3]
```

Out[40]:

```
[1, 2, 3]
```

In [41]:

```
len(l)
```

Out[41]:

```
5
```

In [43]:

```
sorted(l[::1])
```

Out[43]:

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

In [44]:

```
sum(l)
```

Out[44]:

```
15
```

In [46]:

```
l.pop()
```

Out[46]:

```
5
```

In [48]:

```
l.index(3)
```

Out[48]:

```
2
```

In [78]:

```
for i in l:
    print(i)
```

```
1
2
3
4
```

Tuples:

In [49]:

```
tup=(1,2,3)
tup
```

Out[49]:

```
(1, 2, 3)
```

In [53]:

```
#tuple is immutable
tup[1]=10
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-53-892b3932005a> in <module>
      1 #tuple is immutable
----> 2 tup[1]=10

TypeError: 'tuple' object does not support item assignment
```

In [54]:

```
tup[::-1]
```

Out[54]:

```
(3, 2, 1)
```

In [59]:

```
del tup
tup
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-59-cbde4a271b58> in <module>
----> 1 del tup
      2 tup

NameError: name 'tup' is not defined
```

Dictionary

In [60]:

```
#dict is key value pair
dict={'roll':1,
     'name':'Pratiksha'}
dict
```

Out[60]:

```
{'roll': 1, 'name': 'Pratiksha'}
```

In [62]:

```
len(dict)
```

Out[62]:

```
2
```

In [64]:

```
dict['s_name']='Katap'
dict
```

Out[64]:

```
{'roll': 1, 'name': 'Pratiksha', 's_name': 'Katap'}
```

In [65]:

```
dict.update({'marks1':19,"marks2":20})
dict
```

Out[65]:

```
{'roll': 1, 'name': 'Pratiksha', 's_name': 'Katap', 'marks1': 19, 'marks2':
20}
```

In [68]:

```
del dict['marks2']
dict
```

Out[68]:

```
{'roll': 1, 'name': 'Pratiksha', 's_name': 'Katap', 'marks1': 19}
```

In [72]:

```
del dict
dict
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-72-131e16ac4f38> in <module>
----> 1 del dict
      2 dict

NameError: name 'dict' is not defined
```

In [76]:

```
print(range(0,5))
```

```
range(0, 5)
```

String

In [81]:

```
x='Hello'
x
```

Out[81]:

```
'Hello'
```

In [82]:

```
x[0:3]
```

Out[82]:

```
'Hel'
```

In [83]:

```
x[::-1]
```

Out[83]:

```
'olleH'
```

In [85]:

```
y=x
y
```

Out[85]:

```
'Hello'
```

In [87]:

```
x+" "+y
```

Out[87]:

```
'Hello Hello'
```

In [88]:

```
x==y
```

Out[88]:

```
True
```

In [95]:

```
s="She said \'Hello'"
s
```

Out[95]:

```
"She said 'Hello'"
```

In [98]:

```
x=s.upper()
x
```

Out[98]:

```
"SHE SAID 'HELLO'"
```

In [99]:

```
x.lower()
```

Out[99]:

```
"she said 'hello'"
```

In [103]:

```
s.index('said')
```

Out[103]:

```
4
```

In [105]:

```
s.startswith('S')
```

Out[105]:

```
True
```

In [106]:

```
s.endswith('o')
```

Out[106]:

```
False
```

In [107]:

```
s.split()
```

Out[107]:

```
['She', 'said', "'Hello'"]
```

In [108]:

```
s.format()
```

Out[108]:

```
"She said 'Hello'"
```

working with external libraries

In [109]:

```
import math
print(math.pi)
```

```
3.141592653589793
```

In [113]:

```
math.log(1,2)
```

Out[113]:

```
0.0
```

In [114]:

```
dir()
```

```
 '_i98',
 '_i99',
 '_ih',
 '_ii',
 '_iii',
 '_oh',
 'can_vote',
 'diff',
 'exit',
 'get_ipython',
 'i',
 'l',
 'list',
 'math',
 'quit',
 's',
 'x',
 'y']
```

Python libraries

Numpy

contains functions for maths operations

-Helps work with N-dimensional array

In [116]:

```
import numpy as np  #here np is alise
```

In [117]:

```
np.__version__
```

Out[117]:

```
'1.19.2'
```

In [118]:

```
np.array([1,2,3,4,5,6])
```

Out[118]:

```
array([1, 2, 3, 4, 5, 6])
```

In [120]:

```
#multi dimensional array
a=[[1,2,3],
[4,5,6],
  [7,8,9]]
a
```

Out[120]:

```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

In [121]:

```
np.array(a)
```

Out[121]:

```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

In [125]:

```
np.random.randint(0,100)
```

Out[125]:

```
9
```

In [169]:

```
np.random.randint(10000,90000)
```

Out[169]:

```
89923
```

In [201]:

```
#seed() lets you fix the random number
np.random.seed(1)
np.random.randint(10,100)
```

Out[201]:

```
47
```

In [205]:

```
#zeros() to create array with all zerors
np.zeros((2,2),dtype=int)
```

Out[205]:

```
array([[0, 0],
       [0, 0]])
```

In [206]:

```
np.ones((4,4),dtype=int)
```

Out[206]:

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

In [207]:

```
np.full((2,2),4)
```

Out[207]:

```
array([[4, 4],
       [4, 4]])
```

In [225]:

```
a1=np.zeros((2,2),dtype=int)
a2=np.ones((2,2),dtype=int)
a2
np.concatenate((a1,a2),axis=1)#concat column wise
```

Out[225]:

```
array([[0, 0, 1, 1],
       [0, 0, 1, 1]])
```

In [227]:

```
np.concatenate((a1,a2),axis=0)#concat row wise
```

Out[227]:

```
array([[0, 0],
       [0, 0],
       [1, 1],
       [1, 1]])
```

In [232]:

```
a1.ndim
```

Out[232]:

```
2
```

In [233]:

```
a1.shape
```

Out[233]:

```
(2, 2)
```

In [234]:

```
a1.size
```

Out[234]:

```
4
```

In [235]:

```
a1.dtype
```

Out[235]:

```
dtype('int32')
```

In [236]:

```
a1.itemsize
```

Out[236]:

```
4
```

In [237]:

```
a1.data
```

Out[237]:

```
<memory at 0x008D6D48>
```

In [247]:

```
#numpy empty function creates an array with random elements
np.empty((2,2))
```

Out[247]:

```
array([[1.42137876e-076, 1.05901196e+218],
       [1.61410559e+132, 9.89803615e+164]])
```

In [249]:

```
np.arange(10,30,20)
```

Out[249]:

```
array([10])
```

In [253]:

```
from numpy import pi
a=12*pi
a
```

Out[253]:

```
37.69911184307752
```

In [255]:

```
np.sin(1)
```

Out[255]:

```
0.8414709848078965
```

In [257]:

```
np.exp(1)
```

Out[257]:

```
2.718281828459045
```

In [258]:

```
np.sqrt(9)
```

Out[258]:

```
3.0
```

In [259]:

```
np.add(10,20)
```

Out[259]:

```
30
```

In [262]:

```
np.floor((1,2))
```

Out[262]:

```
array([1., 2.])
```

In [264]:

```
a1.view()
```

Out[264]:

```
array([[0, 0],
       [0, 0]])
```

In [270]:

```
arr=np.array((1,2,3))
type(arr)
```

Out[270]:

```
numpy.ndarray
```

In [274]:

```
a2.ndim
```

Out[274]:

```
2
```

In [275]:

```
x=arr.copy()
```

In [276]:

```
x
```

Out[276]:

```
array([1, 2, 3])
```

In [277]:

```
x.view()
```

Out[277]:

```
array([1, 2, 3])
```

In [282]:

```
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])

newarr = arr.reshape(4, 3)

print(newarr)
```

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

In [289]:

```
np.array_split(arr,20)
```

Out[289]:

```
[array([1]),
 array([2]),
 array([3]),
 array([4]),
 array([5]),
 array([6]),
 array([7]),
 array([8]),
 array([9]),
 array([10]),
 array([11]),
 array([12]),
 array([], dtype=int32),
 array([], dtype=int32),
 array([], dtype=int32),
 array([], dtype=int32),
 array([], dtype=int32),
 array([], dtype=int32),
 array([], dtype=int32),
 array([], dtype=int32)]
```

In [292]:

```
np.where(arr==44)
```

Out[292]:

```
(array([], dtype=int32),)
```

In [294]:

```
np.searchsorted(arr,3)
```

Out[294]:

```
2
```

In [296]:

```
x=np.array([1,4,3,5,2])
np.sort(x)
```

Out[296]:

```
array([1, 2, 3, 4, 5])
```