Loops

In [2]:
```
# repeatation of group of statements
```

In [3]:
```
for i in range(10):
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
```

In [4]:
```
range(10)
```

Out[4]:
```
range(0, 10)
```

In [5]:
```
# table of 2
for i in range(10):
    print(i*2)
```

```
0
2
4
6
8
10
12
14
16
18
```

In [6]:
```
# syntax of range() --->
# range(start, end-1, steps)

for i in range(1, 11, 3):
    print(i)
```

```
1
4
7
10
```

In [7]:
```
for i in range(1,16,5):
    print(i)
```

```
1
6
11
```

In [9]:
```
for i in range(10, 0, -1):
    print(i)
```

```
10
9
8
7
```

```
6
5
4
3
2
1
```

In [12]:
```
for i in range(-4, -6, -1):
    print(i)
```

```
-4
-5
```

While loop

In [14]:
```
value = 10
value > 0
```

Out[14]:
```
True
```

In [15]:
```
value = 10
# print till 10
while (value > 0):
    print(value)
    value = value -1
```

```
10
9
8
7
6
5
4
3
2
1
```

String datatype

In [16]:
```
name = 'mehul '
name[0]
```

Out[16]:
```
'm'
```

In [17]:
```
name[5]
```

Out[17]:
```
' '
```

In [18]:
```
name[2]
```

Out[18]:
```
'h'
```

In [21]:
```
name = 'mehul wankhede'
name[6]
```

Out[21]:
```
'w'
```

In [22]:
```
name[-4]
```

```
'h'
```

Out[22]:

string functions

In [23]:
```
name
```

Out[23]:
```
'mehul wankhede'
```

In [24]:
```
name.upper() # returns all charector in caps
```

Out[24]:
```
'MEHUL WANKHEDE'
```

In [25]:
```
name.capitalize()
```

Out[25]:
```
'Mehul wankhede'
```

In [30]:
```
name.split(' ')
```

Out[30]:
```
['mehul', 'wankhede']
```

In [31]:
```
# write all the string related functions  home work
```

data structures

- list
- tuple
- set
- dictionary

List

- List is a data structure which is heterogeneous in nature and mutable

In [33]:
```
my_list = [1, 2, 3, 4]
my_list
```

Out[33]:
```
[1, 2, 3, 4]
```

In [34]:
```
len(my_list)
```

Out[34]:
```
4
```

In [35]:
```
my_list[1]
```

Out[35]:
```
2
```

In [37]:
```
my_list
```

Out[37]:
```
[1, 2, 3, 4]
```

In [38]:
```
my_list[-3]
```

```
2
```

Out[38]:

In [39]:
```
my_list
```

Out[39]:
```
[1, 2, 3, 4]
```

In [40]:
```
# we can change value at given index in list
# [1,5, 3, 4]
my_list[1] = 5
my_list
```

Out[40]:
```
[1, 5, 3, 4]
```

In [41]:
```
my_list[3] = 8
my_list
```

Out[41]:
```
[1, 5, 3, 8]
```

In [42]:
```
my_list[2] = 5
my_list
```

Out[42]:
```
[1, 5, 5, 8]
```

tuple

- tuples are heterogeneous data structures which are immutable

In [43]:
```
my_tuple = (1,2,3,4)
my_tuple
```

Out[43]:
```
(1, 2, 3, 4)
```

In [44]:
```
my_tuple[1]
```

Out[44]:
```
2
```

In [45]:
```
my_tuple[1] = 5
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [45], in <cell line: 1>()
----> 1 my_tuple[1] = 5

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

In [49]:
```
my_list
```

Out[49]:
```
[1, 5, 5, 8]
```

In [51]:
```
my_list.index(8)
```

Out[51]:
```
3
```

In [53]:
```
# write the all function of list and tuples  --->home work
```

set

- set are heterogeneous data structure

In [56]:
```
my_set = {1,2,2,3,3,4,4,4, 'mehul', 1.2}
```

In [61]:
```
set_a = {1,2,3}
set_b = {1,5,6}

set_a - set_b  # a -b
```

Out[61]:
```
{2, 3}
```

In [62]:
```
set_b - set_a  # b -a
```

Out[62]:
```
{5, 6}
```

In [65]:
```
set_a.union(set_b)
```

Out[65]:
```
{1, 2, 3, 5, 6}
```

dictionary

In [66]:
```
my_dict = {'name':'mehul', 'age':27, 'percentage':93.2}
```

In [67]:
```
my_dict
```

Out[67]:
```
{'name': 'mehul', 'age': 27, 'percentage': 93.2}
```

In [68]:
```
my_dict['name']
```

Out[68]:
```
'mehul'
```

In [69]:
```
my_dict['age']
```

Out[69]:
```
27
```

In [70]:
```
my_dict['percentage']
```

Out[70]:
```
93.2
```

In [71]:
```
my_dict['name'] = 'rahul'
my_dict
```

Out[71]:
```
{'name': 'rahul', 'age': 27, 'percentage': 93.2}
```

In [72]:
```
my_dict.keys()
```

Out[72]:
```
dict_keys(['name', 'age', 'percentage'])
```

In [73]:
```
my_dict.values()
```

Out[73]:
```
dict_values(['rahul', 27, 93.2])
```

In [74]:
```
my_dict['address']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Input In [74], in <cell line: 1>()
```

```
----> 1 my_dict['address']

KeyError: 'address'
```

In [76]:
```
my_dict.get('address','key not present')
```

Out[76]:
```
'key not present'
```

In [77]:
```
my_dict.get('percentage','key not present')
```

Out[77]:
```
93.2
```

In [79]:
```
my_dict.get('address','hi key nahi re')
```

Out[79]:
```
'hi key nahi re '
```

In [ ]:
```
# write all function related to set and dictionary  ---home work
```

List advance indexing and slacing

In [80]:
```
my_list = [[1,2,3], [4,5,6],[7,8,9]]
```

In [81]:
```
my_list
```

Out[81]:
```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

In [90]:
```
# access 2nd and 3rd row and all columns
# [row, columns] ===> [row_1:row_n, col_1:col_m]

my_list[0:2]
```

Out[90]:
```
[[1, 2, 3], [4, 5, 6]]
```

In [92]:
```
import numpy as np
np.array(my_list)[1:3,:]
```

Out[92]:
```
array([[4, 5, 6],
       [7, 8, 9]])
```

In [94]:
```
my_list
```

Out[94]:
```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

In [95]:
```
# my_list[start:end: steps]
my_list[0:3:2]
```

Out[95]:
```
[[1, 2, 3], [7, 8, 9]]
```

In [102…
```
# append

my_list = [2,3,4]
my_list.append(5)
```

In [103…
```
my_list
```

Out[103]:
```
[2, 3, 4, 5]
```

In [105…
```
my_list.insert(0,1)
```

In [106…
```
my_list
```

Out[106]:
```
[1, 2, 3, 4, 5]
```

In [107…
```
my_list.insert(3,7)
my_list
```

Out[107]:
```
[1, 2, 3, 7, 4, 5]
```

In [108…
```
my_list.insert(4,7)
my_list
```

Out[108]:
```
[1, 2, 3, 7, 7, 4, 5]
```

In [109…
```
my_list.insert(6,7)
my_list
```

Out[109]:
```
[1, 2, 3, 7, 7, 4, 7, 5]
```

In [110…
```
my_list.remove(7)
```

In [115…
```
my_list
```

Out[115]:
```
[1, 2, 3, 4, 5]
```

In [119…
```
my_list.append(6)
my_list
```

Out[119]:
```
[1, 2, 3, 4, 5, 6]
```

In [127…
```
first_list  = []  # divisible by 7
second_list =[]   # divisible by 9

# number divisible by 8
for i in range(0, 101):
    if i % 8 == 0:
        first_list.append(i)
    else:
        second_list.append(i)
```

In [123…
```
second_list
```

Out[123]:
```
[1,
 2,
 3,
 4,
 5,
 6,
 7,
 9,
 10,
 11,
 12,
 13,
 14,
 15,
 17,
 18,
 19,
 20,
 21,
```

22,
23,
25,
26,
27,
28,
29,
30,
31,
33,
34,
35,
36,
37,
38,
39,
41,
42,
43,
44,
45,
46,
47,
49,
50,
51,
52,
53,
54,
55,
57,
58,
59,
60,
61,
62,
63,
65,
66,
67,
68,
69,
70,
71,
73,
74,
75,
76,
77,
78,
79,
81,
82,
83,
84,
85,
86,
87,
89,
90,
91,
92,
93,
94,
95,
97,

```
 98,
 99,
 100]
```

In [124…
```
100%11
```

Out[124]:
```
1
```

In [125…
```
100%10
```

Out[125]:
```
0
```

In [ ]:
```

```