In [1]:
```
#This is a supplementary material to the lecture "Lists and Dictionaries" to quickly revise, whenever
 needed
```

In [2]:
```
#list in python are collection of data (may not be of the same type)
#creating a list
l1 = []      #will create empty list
l2 = list()  #will create empty list
l3 = [1, 5, 7.6, 'hello']
#adding element to the list
l1.append(4)
l1.append(3)
l2.append(10)
l3.append(50)
print(l1, len(l1))
print(l2, len(l2))
print(l3, len(l3))              #len(list_name) will return the length of list i.e. number of elements
in it
```

```
[4, 3] 2
[10] 1
[1, 5, 7.6, 'hello', 50] 5
```

In [3]:
```
#elements of list can be accessed by specifying the index
#indexing in list goes like 0, 1, .... (from left to right) and  (..., -3, -2, -1) from right to left
print(l1[1], l1[-1])
```

```
3 3
```

In [4]:
```
#slicing can be done also
l3_slice = l3[1:3]    #will give you a list containing elements from index 1 to index 2 of list l3
print(l3_slice)
```

```
[5, 7.6]
```

In [5]:
```
#we can do fast iterate over list
for item in l3:
  print(item, end=' ')
```

```
1 5 7.6 hello 50
```

In [6]:
```
#to cheeck if an element exist in the list
if 'hello' in l3:
  print('Hello')
```

```
Hello
```

In [7]:
```
#item can be removed from a list in two ways
#specify the item to be removed (using remove function) or specify the index of the item, which is to
 be removed (using pop function)
l3.remove('hello')          #removes element 'hello' from list l3
print(l3)
l3.pop(2)                   #removes element at index 2 of list l3
print(l3)
l3.pop()                    #removes last element of the list l3
print(l3)
```

```
[1, 5, 7.6, 50]
[1, 5, 50]
[1, 5]
```

In [8]:
```
#deletion can also be done using del
del l3[1]           #element at index 1 will be removed
print(l3)
```

```
[1]
```

In [9]:
```
#to create a copy of the list
l2_copy = l2.copy()
print(l2, l2_copy)
```

```
[10] [10]
```

In [10]:
```
#to empty the complete list in one go
l1.clear()
print(l1)
```

```
[]
```

In [11]:
```
#joining multiple lists
l1 = [1, 2, 3]
l2 = [4, 5, 6]
l3 = [7, 8, 9]
l4 = l1 + l2 + l3
print(l4)
```

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

In [12]:
```
#dictionary in python can be thought same as the unordered map in java
#collection of (key, value) pairs
#keys of the dictionary has to immutable, for example list can not be used as key
```

In [13]:
```
#creating a dictionary
d1 = {
    'name': 'abc',
    'rollNo': 123,
    'marks': 95.5,
    'subjects': ['english', 'maths', 'science']
}
#on left, we have keys and on the right we have values for those keys
#all (key, value) pairs are separated by commas inside curly brackets
#in above example, we have used string as keys, which are immutable
```

In [14]:
```
#creating empty dictionary
d2 = {}
#accessing elements from dictionary
name = d1['name']       #will give the value corresponding to the key 'name'
name = d1.get('name')   #alternative way to get the value for a key
print(name)
```

```
abc
```

In [15]:
```
#adding an item or updating the value for a key
d1['name'] = 'def'        #if key 'name' is already present in the dictionary, then it's value would b
e updated, otherwise new key, value pair will be added to the dictionary
d2['name'] = 'ghi'
d2['rollNo'] = 124
print(d1)
print(d2)
```

```
{'name': 'def', 'rollNo': 123, 'marks': 95.5, 'subjects': ['english', 'maths', 'science']}
{'name': 'ghi', 'rollNo': 124}
```

In [16]:
```
#iterating over dictionary
for key in d1:
  print(key, ':', d1.get(key))
```

```
name : def
rollNo : 123
marks : 95.5
subjects : ['english', 'maths', 'science']
```

In [17]:
```
#alternative way for iterating
for key, value in d1.items():
  print(key, ':', value)
```

```
name : def
rollNo : 123
marks : 95.5
subjects : ['english', 'maths', 'science']
```

In [18]:
```
#check if a key is present or not
if 'school' in d1:
  print("key 'school' is found in the dictionary")
else:
  print("key 'school' is not there, let's add it!")
  d1['school'] = 'xyz'
  print(d1)
```

```
key 'school' is not there, let's add it!
{'name': 'def', 'rollNo': 123, 'marks': 95.5, 'subjects': ['english', 'maths', 'science'], 'school':
'xyz'}
```

In [19]:
```
#length of dictionary
print(len(d2))
#to delete items from dictionary
d1.popitem()            #last added item i.e. (key, value) pair would be deleted, in this case 'schoo
l' will be deleted
d1.pop('subjects')      #key, value pair for the key 'subjects' would be deleted
del d1['marks']         #key, value pair for the key 'marks' would be deleted
print(d1, 'length: ', len(d1))
```

```
2
{'name': 'def', 'rollNo': 123} length:  2
```

In [20]:
```
#making a copy of dictionary
d1_copy = d1.copy()
#alternate way
d2_copy = dict(d2)
print(d1_copy)
print(d2_copy)
```

```
{'name': 'def', 'rollNo': 123}
{'name': 'ghi', 'rollNo': 124}
```

In [21]:
```
#Thanks, Happy coding!
```

In [ ]:
```
#To download .ipynb notebook, right click the following link and click save as
https://ninjasfiles.s3.amazonaws.com/0000000000003218.ipynb
```