List Functions

Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

```
cars = ['Ford', 'BMW', 'Volvo']
cars.sort()

#['BMW', 'Ford', 'Volvo']
```

```
fruits = ['apple', 'banana', 'cherry']
fruits.reverse()

#['cherry', 'banana', 'apple']
```

```
fruits = ['apple', 'banana', 'cherry']
fruits.remove("banana")

['apple', 'cherry']
```

```
fruits = ['apple', 'banana', 'cherry']
fruits.pop(1)

#['apple', 'cherry']
```

```
fruits = ['apple', 'banana', 'cherry']
fruits.insert(1, "orange")

#['apple', 'orange', 'banana', 'cherry']
```

```
fruits = ['apple', 'banana', 'cherry']
cars = ['Ford', 'BMW', 'Volvo']
fruits.extend(cars)

#['apple', 'banana', 'cherry', 'Ford',
'BMW', 'Volvo']
```

```
fruits = ['apple', 'banana', 'cherry', 'orange']
fruits.clear()

#[]
```

```
fruits = ['apple', 'banana', 'cherry']
fruits.append("orange")

#['apple', 'banana', 'cherry', 'orange']
```

Tuple Methods

Method	Description
count()	Returns the number of times a specified value occurs in a tuple
index()	Searches the tuple for a specified value and returns the position of where it was found

```
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)
x = thistuple.count(5)
print(x)
```

#2

```
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)
x = thistuple.index(8)
print(x)
```

#3

- len() amp=(1,2,3,4) len(amp) #4
- max() amp=(1,2,3,5,3,4,6,21,99) max(amp) #99
- min() amp=(1,2,3,44,2,39,99) min(amp) #1
- tuple() t=tuple(“abc”) print(t) #(‘a’ , ’b’ , ’c’)

Python Dictionaries

```
thisdict =      {"brand": "Ford", "model": "Mustang","year": 1964}
print(thisdict)

thisdict =      {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
```

Accessing Elements of a Dictionary

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(thisdict)
```

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(thisdict["brand"])
```

```
thisdict =              {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
x = thisdict.get("model")
print(x)
```

Characteristics of a Dictionary

- A dictionary is unorder set of **KEY : VALUES** pairs.
- Unlike the strings, list and Tuple, a dictionary is not a sequence because it is unordered set of elements.
- dictionaries are indexed by Keys and its Keys must be of any non-mutable type.
- each of the Keys within dictionary must be unique
- like list, dictionary is also mutable. we can change the value of certain key “in place” using the assignment statement as per Syntex.

Traversing a Dictionary

```
statesAndCapitals = {
    'Gujarat': 'Gandhinagar',
    'Maharashtra': 'Mumbai',
    'Rajasthan': 'Jaipur',
    'Bihar': 'Patna'
}

print('List Of given states:\n')

# Iterating over keys
for state in statesAndCapitals:
    print(state)
```

List Of given states:

Gujarat
Maharashtra
Rajasthan
Bihar

Adding, Updating and Deleting Elements to Dictionary

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict["year"] = 2018
```

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.update({"year": 2020})
```

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict["color"] = "red"
print(thisdict)
```

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.pop("model")
print(thisdict)
```

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
del thisdict["model"]
print(thisdict)
```

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.clear()
print(thisdict)
```

Nested Dictionaries

```
myfamily = {
  "child1" : {
    "name" : "Emil",
    "year" : 2004
  },
  "child2" : {
    "name" : "Tobias",
    "year" : 2007
  },
  "child3" : {
    "name" : "Linus",
    "year" : 2011
  }
}
print(myfamily)
```

```
child1 = {
  "name" : "Emil",
  "year" : 2004
}
child2 = {
  "name" : "Tobias",
  "year" : 2007
}
child3 = {
  "name" : "Linus",
  "year" : 2011
}
myfamily = {
  "child1" : child1,
  "child2" : child2,
  "child3" : child3
}
print(myfamily)
```

```
#{'child1': {'name': 'Emil', 'year': 2004}, 'child2': {'name': 'Tobias', 'year': 2007}, 'child3': {'name': 'Linus', 'year': 2011}}
```

Dictionary Methods and Functions

Method	Description
clear()	Removes all the elements from the dictionary
copy()	Returns a copy of the dictionary
fromkeys()	Returns a dictionary with the specified keys and value
get()	Returns the value of the specified key
items()	Returns a list containing a tuple for each key value pair
keys()	Returns a list containing the dictionary's keys
pop()	Removes the element with the specified key
popitem()	Removes the last inserted key-value pair
setdefault()	Returns the value of the specified key. If the key does not exist: insert the key, with the specified value
update()	Updates the dictionary with the specified key-value pairs
values()	Returns a list of all the values in the dictionary

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
x = car.get("model")
print(x)
```

```
x = ('key1', 'key2', 'key3')
y = 0
thisdict = dict.fromkeys(x, y)
print(thisdict)

#['key1': 0, 'key2': 0, 'key3': 0]
```

```
x = ('key1', 'key2', 'key3')
y = (1,2,3)

for z in y:
    thisdict = dict.fromkeys(x,
z)
print(thisdict)
#{'key1': 3, 'key2': 3, 'key3': 3}
```

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
x = car.items()
print(x)

#dict_items([('brand', 'Ford'), ('model', 'Mustang'), ('year', 1964)])
```

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
x = car.keys()
car["color"] = "white"
print(x)

#dict_keys(['brand', 'model', 'year', 'color'])
```

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
x = car.values()
print(x)

#dict_values(['Ford', 'Mustang', 1964])


car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
x = car.values()
car["year"] = 2018
print(x)

#dict_values(['Ford', 'Mustang', 2018])
```

Bubble Sort

```
alist=[12,23,1,4,14,2,3]
print("OG",alist)
n=len(alist)
for i in range(n):
    for j in range(0,n-i-1):
        if  alist[j] > alist[j+1]:
            alist[j],alist[j+1]=alist[j+1],alist[j]
print("sorting :",alist)
```

```
OG [12, 23, 1, 4, 14, 2, 3]
sorting : [1, 2, 3, 4, 12, 14, 23]
```

Insertion Sort

```
alist=[12,23,1,4,14,2,3]
print("OG",alist)
n=len(alist)
for i in range(1,n):
    key=alist[i]
    j=i-1
    while j>=0 and key< alist[j]:
        alist[j+1]=alist[j]
        j=j-1
    else:
        alist[j+1]=key
print("sorting :",alist)
```

```
OG [12, 23, 1, 4, 14, 2, 3]
sorting : [1, 2, 3, 4, 12, 14, 23]
```