# Dictionary

- Dictionaries are used to store data values in key:value pairs.
- We use the symbol '{}' braces to create a dictionary.
- We use keys for operations.
- An empty {} will always represent a dictionary but not a set.
- To represent an empty set we should use 's = set()' function.
- Dictionaries are the mapping concept, just like 'Maps' in java.

- A dictionary is a collection which is ordered*, changeable and do not allow duplicates.

As of Python version 3.7, dictionaries are *ordered*. In Python 3.6 and earlier, dictionaries are *unordered*.

```python
# Different ways of creating a dictionary

# empty dictionary
my_dict = {}
print(my_dict)

# dictionary with integer keys
my_dict = {1: 'apple', 2: 'ball'}
print(my_dict)

# dictionary with mixed keys
my_dict = {'name': 'John', 1: [2, 4, 3]}
print(my_dict)

# using dict()
my_dict = dict({1:'apple', 2:'ball'})
print(my_dict)

# from sequence having each item as a pair
my_dict = dict([(1,'apple'), (2,'ball')])
print(my_dict)
```

```python
# get vs [] for retrieving elements
my_dict = {'name': 'Jack', 'age': 26}

# Output: Jack
print(my_dict['name'])

# Output: 26
print(my_dict.get('age'))

# Trying to access keys which doesn't exist throws error
# Output None
print(my_dict.get('address'))

# KeyError
# print(my_dict['address'])




# Changing and adding Dictionary Elements
my_dict = {'name': 'Jack', 'age': 26}

# update value
my_dict['age'] = 27

#Output: {'age': 27, 'name': 'Jack'}
print(my_dict)

# add item
my_dict['address'] = '2nd street, Bapuji Nagar'

# Output: {'address': '2nd street, Bapuji Nagar', 'age': 27,
'name': 'Jack'}
print(my_dict)


dic = {} #create an empty dict

#insertion of elements in an empty dict
```

```
dic[100] = 'abc'
print(dic)

dic[99] = 'xyz'
print(dic)


# Dict with loops
dict= {'name':'abcd', 'age':'27 years','location':'vskp'}

# prints only the keys
for i in dict:
    print(i)

# prints only the keys
for i in dict.keys():
    print(i)

# prints only the values
for i in dict.values():
    print(i)


dict= {'name':'abcd', 'age':'27 years','location':'vskp'}

# items() method gives both key and values
print(dict.items())

# loop on items() method
for i in dict.items():
    print(i)
print("-----------------")

for i,j in dict.items():
    print(i, '--', j)
print("-----------------")

for i,j in dict.items():
    print(i)
print("-----------------")
```

```
for i,j in dict.items():
    print(j)


# Comparing two dict's
dict1 = {'name':'abcd', 'age':'27 years','location':'vskp'}
dict2 = {'age':'27 years','name':'abcd' ,'location':'vskp'}

# The keys of a dictionary may alter, but still they are
same
if(dict1 == dict2):
    print("same")
else:
    print("different")




# Removing elements from a dictionary

# create a dictionary
squares = {1: "one", 2: "two", 3: "three", 4: "four", 5:
"five"}

# remove a particular item, returns its value
print(squares.pop(4))
print(squares)

# Python dictionary popitem() method removes the last
inserted key-value pair from the dictionary and returns it
as a tuple.
print(squares.popitem())
print(squares)

# remove all items
squares.clear()
print(squares)
```

```
# delete the dictionary itself
del squares

# Throws Error
# print(squares)




# To get a copy of the dict
car = {"brand": "Maruti", "model": "800", "year": 1984}
print(car)
x = car.copy()
print(x)

# modification on copy
x['color'] = 'white'
print(x)
print(car)

# Using 'fromkeys()' fun to auto assign the given values to
dict
x = ('key1', 'key2', 'key3')
y = 0
my_dict = dict.fromkeys(x, y)
print(my_dict)


# If key value not given then 'None' will be assigned
x = ('k1', 'k2', 'k3')
my_dict = dict.fromkeys(x)
print(my_dict)

# Python dictionary method setdefault() is similar to get(),
but will set dict[key]=default if key is not already in
dict.
d = {'Name': 'SaiRam', 'Age': 7}
d.setdefault('Age', None)
```

```
d.setdefault('Sex', 'Male')
print(d['Name'])
print(d['Age'])
print(d['Sex'])

# update() adds dictionary dict2's key-values pairs in to
dict. This function does not return anything.
dict = {'Name': 'Sai Ram', 'Age': 7}
dict2 = {'Sex': 'M'}

dict.update(dict2)
print (dict)


# Sorting a dict
dict1 = {'name':'abcd', 'age':'27 years','location':'vskp'}

# Sorts and arranges the elements of the dictionary
for i in sorted(dict1.items()):
    print(i)




# Iterating through a Dictionary
squares = {1: 1, 3: 9, 5: 25, 7: 49, 9: 81}
for i in squares:
    print(squares[i])

# Missilenious
# Set Comprehension
myList = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
newSet = {element*3 for element in myList}
print("The existing list is:")
print(myList)
print("The Newly Created set is:")
print(newSet)

# Dictionary Comprehension
```

```python
squares = {x: x*x for x in range(6)}
print(squares)

# Above code can also be written as
squares = {}
for x in range(6):
    squares[x] = x*x
print(squares)


# Dictionary Comprehension with if conditional
odd_squares = {x: x*x for x in range(11) if x % 2 == 1}
print(odd_squares)
```