

Dictionaries

```
In [ ]: 1. Built-in collection data type.  
2. Dictionaries are used to store data values in key:value pairs.  
3. Dictionaries are written with curly brackets and have key-value pairs.  
4. Dictionaries are defined as objects with the data type 'dict'
```

```
In [41]: #Example of a dictionaries
```

```
my_dict = {"Name":"Neenu","Age":"30","Place":"USA"}  
print(my_dict)
```

```
{'Name': 'Neenu', 'Age': '30', 'Place': 'USA'}
```

Characteristics of a Dictionaries

```
In [ ]: Ordered           : As of Python version 3.7, dictionaries are ordered.it means that the items have  
                           a defined order, and that order will not change.  
  
Changeable               : Dictionaries are changeable, meaning that we can change, add or remove items after  
                           the dictionary has been created.  
  
Dose not allow duplicates : Dictionaries cannot have two items with the same key.Duplicate values will  
                           overwrite existing values.  
  
Dictionary items are presented in key:value pairs, and can be referred to by using the key name.
```

```
In [14]: #When there is duplicate values(two items with same key),duplicate values will overwrite existing values.
```

```
my_dict = {"Name":"Neenu","Age":"30","Place":"USA","Place":"India"} #here, key 'Place' is duplicated.  
print(my_dict)
```

```
{'Name': 'Neenu', 'Age': '30', 'Place': 'India'}
```

Dictionary Methods

```
In [ ]: Set of built-in methods that you can use on dictionaries are given below:
```

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

Dictionary Length

`len()` :Method used to determine the no. of items in a dictionary.

In [8]: *#Length of a dictionary*

```
my_dict = {"Name": "Neenu", "Age": "30", "Place": "USA"} #Here, dictionary 'dict' has 3 items
print(len(my_dict))
```

3

Dictionary Items - Data Types

The values in dictionary items can be of any data type such integer,string,float,Boolean,list etc.

`type()` : Method to find the datatype of a given variable.

In [12]: *#Dictionary with different data items.*

```
my_dict = {
    "Brand": "Tesla",
    "Electric": True,
    "Year": 2003,
    "Colors": ["Black", "Silver", "Blue"]
}
print (dict)
```

```
{'Brand': 'Tesla', 'Electric': True, 'Year': 2003, 'Colors': ['Black', 'Silver', 'Blue']}
```

```
In [16]: #Type of a dictionary
```

```
my_dict = {"Name": "Neenu", "Age": "30", "Place": "USA"}  
print(type(my_dict))
```

```
<class 'dict'>
```

The dict() Constructor

dict() :The dict() constructor creates a dictionary.

```
In [16]: #Create dictionary using dict()
```

```
my_dict = dict(Name = "Smith", Age = 36, Country = "England")  
print(my_dict)  
print(type(my_dict))
```

```
{'Name': 'Smith', 'Age': 36, 'Country': 'England'}  
<class 'dict'>
```

```
In [20]: #Another way to create dictionary using dict()
```

```
my_dict = dict({"Name": "Smith", "Age": 36, "Country": "England"})  
print(my_dict)  
print(type(my_dict))
```

```
{'Name': 'Smith', 'Age': 36, 'Country': 'England'}  
<class 'dict'>
```

```
In [17]: #Create a empty dictionary using dict()
```

```
my_dict = dict()  
print(my_dict)  
print(type(my_dict))
```

```
{}  
<class 'dict'>
```

Access Dictionary Items

```
In [ ]: We can access the items of a dictionary by referring to its key name, inside square brackets.
```

```
get() : Method to access items in a dictinaory.
```

```
keys()      : Return a list of all the keys in the dictionary.  
values()    : Return a list of all the values in the dictionary.  
items()     : Return each item in a dictionary, as tuples in a list.
```

In [58]: *# Access Dictionary Items*

```
my_dict = {  
    "Brand": "Tesla",  
    "Electric": True,  
    "Year": 2003,  
    "Colors": ["Black", "Silver", "Blue"]  
}  
print (dict["Brand"])  
print (dict["Year"])
```

```
Tesla  
2003
```

In [59]: *# Access Dictionary Items using get()*

```
my_dict = {  
    "Brand": "Tesla",  
    "Electric": True,  
    "Year": 2003,  
    "Colors": ["Black", "Silver", "Blue"]  
}  
brand = my_dict.get("Brand")  
year = my_dict.get("Year")  
print (brand)  
print (year)
```

```
Tesla  
2003
```

In [61]: *#use of keys() method to access all the keys in the dictionary*

```
my_dict = {  
    "Brand": "Tesla",  
    "Electric": True,  
    "Year": 2003,  
    "Colors": ["Black", "Silver", "Blue"]  
}  
  
k = my_dict.keys()  
print(k)
```

```
dict_keys(['Brand', 'Electric', 'Year', 'Colors'])
```

In [62]: *#use of values() method to access all the values in the dictionary*

```
my_dict = {  
    "Brand": "Tesla",  
    "Electric": True,  
    "Year": 2003,  
    "Colors": ["Black", "Silver", "Blue"]  
}  
  
v = my_dict.values()  
print(v)
```

```
dict_values(['Tesla', True, 2003, ['Black', 'Silver', 'Blue']])
```

In [63]: *#use of items() method to get each item in a dictionary as tuples, in a list*

```
my_dict = {  
    "Brand": "Tesla",  
    "Electric": True,  
    "Year": 2003,  
    "Colors": ["Black", "Silver", "Blue"]  
}  
  
i = my_dict.items()  
print(i)
```

```
dict_items([('Brand', 'Tesla'), ('Electric', True), ('Year', 2003), ('Colors', ['Black', 'Silver', 'Blue'])])
```

Check if Key Exists

In []: **in** : Keyword used to check **if** a key present **or not**.

In [1]: *#Check if a specified key present or not*

```
my_dict = {  
    "Brand": "Tesla",  
    "Electric": True,  
    "Year": 2003,  
    "Colors": ["Black", "Silver", "Blue"]  
}  
  
if "Year" in my_dict:
```

```
print("Yes, Year present in my_dict ")  
else:  
    print("No, Year not present in my_dict ")
```

Yes, Year present in my_dict

Change Dictionary Items

If you want to change the values in the dictionary:

1. By referring to its key name, inside square brackets
2. update () : This method update the dictionary with the items from the given argument.

The argument must be a dictionary, or an iterable object with key:value pairs

In [3]: *#Change dictionary items by referring to its key name, inside square brackets.*

```
my_dict = {  
    "Brand": "Tesla",  
    "Electric": True,  
    "Year": 2003,  
    "Colors": ["Black", "Silver", "Blue"]  
}  
  
my_dict["Year"] = 2013 #Year value has been changed.  
print(my_dict)
```

```
{'Brand': 'Tesla', 'Electric': True, 'Year': 2013, 'Colors': ['Black', 'Silver', 'Blue']}
```

In [10]: *#Change dictionary items by update() method*

```
my_dict = {  
    "Brand": "Tesla",  
    "Electric": True,  
    "Year": 2003,  
    "Colors": ["Black", "Silver", "Blue"]  
}  
my_dict.update({"Year" : 2013})  
  
print(my_dict)
```

```
{'Brand': 'Tesla', 'Electric': True, 'Year': 2013, 'Colors': ['Black', 'Silver', 'Blue']}
```

Add Dictionary Items

Adding an item to the dictionary is done by using a new index key and assigning a value to it.

In [21]: *#Add a new item to the dictionary*

```
my_dict = {  
    "Brand": "Tesla",  
    "Electric": True,  
    "Year": 2003,  
    "Colors": ["Black", "Silver", "Blue"]  
}  
my_dict["Model"] = "S" # new item has been added to the dictionary.  
print(my_dict)
```

```
{'Brand': 'Tesla', 'Electric': True, 'Year': 2003, 'Colors': ['Black', 'Silver', 'Blue'], 'Model': 'S'}
```

In []: *#Add a new item to the dictionary using update()*

In [22]:

```
my_dict = {  
    "Brand": "Tesla",  
    "Electric": True,  
    "Year": 2003,  
    "Colors": ["Black", "Silver", "Blue"]  
}  
my_dict.update({"Model": "S"}) # new item has been added to the dictionary.  
print(my_dict)
```

```
{'Brand': 'Tesla', 'Electric': True, 'Year': 2003, 'Colors': ['Black', 'Silver', 'Blue'], 'Model': 'S'}
```

Remove Dictionary Items

In []: There are several methods to remove items **from** a dictionary:

1. `pop()` : Removes the item **with** the specified key name.
2. `popitem()` : Removes the last inserted item.
3. `del` : keyword used to remove the item **with** the specified key name.
The `del` keyword can also delete the dictionary completely.

4. `clear()` : Method used to empty the dictionary.

In [24]: *#use pop() method to remove an item with the specified key.*

```
my_dict = {
    "Brand": "Tesla",
    "Electric": True,
    "Year": 2003,
    "Colors": ["Black", "Silver", "Blue"],
    "Model" : "S"
}
my_dict.pop("Year") #item with key "Year" has been removed
print(my_dict)
```

```
{'Brand': 'Tesla', 'Electric': True, 'Colors': ['Black', 'Silver', 'Blue'], 'Model': 'S'}
```

In [26]: *#use popitem() method to remove the last item from the dictionary.*

```
my_dict = {
    "Brand": "Tesla",
    "Electric": True,
    "Year": 2003,
    "Colors": ["Black", "Silver", "Blue"],
    "Model" : "S"
}
my_dict.popitem() #removes the last item.
print(my_dict)
```

```
{'Brand': 'Tesla', 'Electric': True, 'Year': 2003, 'Colors': ['Black', 'Silver', 'Blue']}
```

In [27]: *#use del keyword to remove the item with specified key from the dictionary.*

```
my_dict = {
    "Brand": "Tesla",
    "Electric": True,
    "Year": 2003,
    "Colors": ["Black", "Silver", "Blue"],
    "Model" : "S"
}
del my_dict["Electric"]
print(my_dict)
```

```
{'Brand': 'Tesla', 'Year': 2003, 'Colors': ['Black', 'Silver', 'Blue'], 'Model': 'S'}
```


In [28]: *#use del keyword to delete the dictionary completely.*

```
my_dict = {
    "Brand": "Tesla",
    "Electric": True,
    "Year": 2003,
    "Colors": ["Black", "Silver", "Blue"],
    "Model" : "S"
}
del my_dict
print(my_dict) # Throws error since the dictionary has been deleted.
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[28], line 11
      3 my_dict = {
      4     "Brand": "Tesla",
      5     "Electric": True,
    (...)
      8     "Model" : "S"
      9 }
     10 del my_dict
--> 11 print(my_dict)

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

In [29]:

```
my_dict = {
    "Brand": "Tesla",
    "Electric": True,
    "Year": 2003,
    "Colors": ["Black", "Silver", "Blue"],
    "Model" : "S"
}
my_dict.clear() #Empties the dictionary
print(my_dict)

{}
```

Loop Dictionaries

In []: We can use **for** loop to iterate through a dictionary:

for loop : When looping through a dictionary, the **return** value are the keys of the dictionary, but there are methods to **return** the values **as** well.

Methods used **with for** loop to **return** different results:

values() :Return values of a dictionary.
keys() :Return keys of a dictionary.
items() :Return key:value pairs of a dictionary.

In [31]: *#Using for loop to iterate through keys*

```
my_dict = {  
    "Brand": "Tesla",  
    "Electric": True,  
    "Year": 2003,  
    "Colors": ["Black", "Silver", "Blue"],  
    "Model" : "S"  
}  
for k in my_dict:  
    print(k) #print all keys
```

Brand
Electric
Year
Colors
Model

In [32]: *#Using for loop to iterate through values*

```
my_dict = {  
    "Brand": "Tesla",  
    "Electric": True,  
    "Year": 2003,  
    "Colors": ["Black", "Silver", "Blue"],  
    "Model" : "S"  
}  
for k in my_dict:  
    print(my_dict[k]) #print all values
```

Tesla
True
2003
['Black', 'Silver', 'Blue']
S

In [33]: *#Using for loop to iterate through values using values() method*

```
my_dict = {  
    "Brand": "Tesla",  
    "Electric": True,  
    "Year": 2003,  
    "Colors": ["Black", "Silver", "Blue"],  
    "Model" : "S"  
}  
for v in my_dict.values():  
    print(v) #print all values
```

Tesla
True
2003
['Black', 'Silver', 'Blue']
S

In [34]: *#Using for loop to iterate through keys using keys() method*

```
my_dict = {  
    "Brand": "Tesla",  
    "Electric": True,  
    "Year": 2003,  
    "Colors": ["Black", "Silver", "Blue"],  
    "Model" : "S"  
}  
for k in my_dict.keys():  
    print(k) #print all keys
```

Brand
Electric
Year
Colors
Model

In [38]: *#Using for loop to iterate through items using items() method*

```
my_dict = {  
    "Brand": "Tesla",  
    "Electric": True,  
    "Year": 2003,  
    "Colors": ["Black", "Silver", "Blue"],  
    "Model" : "S"  
}  
for k,v in my_dict.items():  
    print(k," ",v) #print all keys
```

```
Brand      Tesla
Electric    True
Year        2003
Colors      ['Black', 'Silver', 'Blue']
Model       S
```

Copy Dictionaries

In []: We can make a copy of a dictionary using the following built-in methods:

1. `copy()`
2. `dict()`

In [39]: *#Make copy of dictionary using copy() method*

```
my_dict = {
    "Brand": "Tesla",
    "Electric": True,
    "Year": 2003,
    "Colors": ["Black", "Silver", "Blue"],
    "Model" : "S"
}
```

```
my_dict_copy = my_dict.copy()
print(my_dict_copy)
```

```
{'Brand': 'Tesla', 'Electric': True, 'Year': 2003, 'Colors': ['Black', 'Silver', 'Blue'], 'Model': 'S'}
```

In [40]: *#Make copy of dictionary using dict() method*

```
my_dict = {
    "Brand": "Tesla",
    "Electric": True,
    "Year": 2003,
    "Colors": ["Black", "Silver", "Blue"],
    "Model" : "S"
}
```

```
my_dict_copy = dict(my_dict)
print(my_dict_copy)
```

```
{'Brand': 'Tesla', 'Electric': True, 'Year': 2003, 'Colors': ['Black', 'Silver', 'Blue'], 'Model': 'S'}
```

Nested Dictionaries

A dictionary can contain dictionaries, this is called nested dictionaries.

In [44]: *#Example of a nested dictionary.*

```
fruit1 = {  
    "name" : "Apple",  
    "color" : "Red"  
}  
fruit2 = {  
    "name" : "Mango",  
    "color" : "Yellow"  
}  
fruit3 = {  
    "name" : "Blueberry",  
    "color" : "Blue"  
}  
  
myFruits = {  
    "f1" : fruit1,  
    "f2" : fruit2,  
    "f3" : fruit3  
}  
  
print(myFruits) #Nested dictionary  
print("Acessing items: ", myFruits["f1"]["name"]) #Acessing items from nested dictionary
```

```
{'f1': {'name': 'Apple', 'color': 'Red'}, 'f2': {'name': 'Mango', 'color': 'Yellow'}, 'f3': {'name': 'Blueberry', 'color': 'Blue'}}
```

```
Acessing items: Apple
```

END