

Python

Dictionary

Class IX

Lab 12

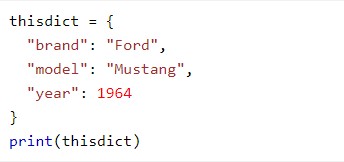
**Lab Objectives:**

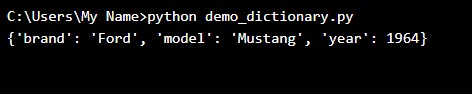
* What is dictionary
* Access values
* Change values
* Delete, Clear, Loop and Constructor

Dictionary

A dictionary is a collection which is unordered, changeable and indexed. In Python dictionaries are written with curly brackets, and they have keys and values.

Example:



Result:

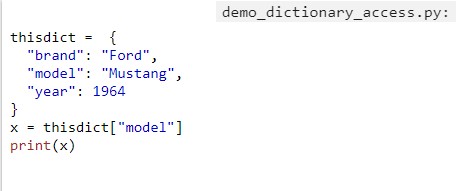
Every dictionary has three parts. First one is key, last one is value and there is a colon to separate the key and value in middle.

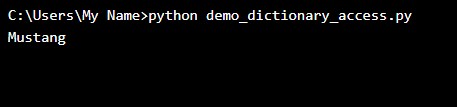
Here, thisdict is a variable. ‘brand’ is the key, ‘Ford’ is the value and colon ( : ) sign is used for separating the key and value. Every key and value contains a single item, there are coma (,) sign to separate the item. You may use as much item as you can.

Suppose your name is Ariyan, your class roll is 2. If teacher says, roll number 2 please stand up. You will must stand up and there is no need to mention your name. if you think it like a dictionary then your roll is the key and your name Aryan is the value. Dictionary in python is the absolutely same thing. Lots of keys and their values are stored there. When you call something by its key you will see the value of it.

Accessing Items

You can access the items of a dictionary by referring to its key name:

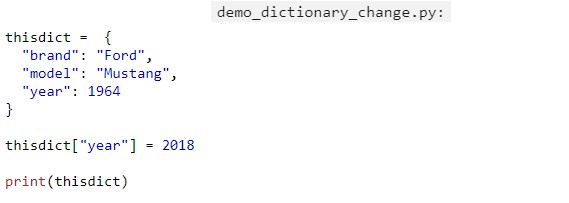
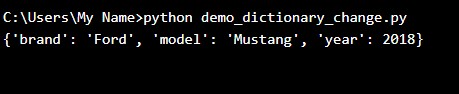




Here you are calling through the key named ‘model’ and store it in a variable X. So when you print it you have seen the value ‘mustang’ of that key.

Change Values

You can change the value of a specific item by referring to its key name:



You have changed the value of the key ‘year’. So when you print again you see the current value of year 2018 instead of 1964.

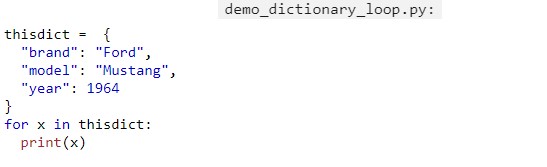
Crete your personal dictionary including your name, roll, favorite subject. Access values on it and try to change values.

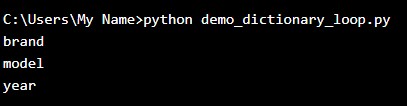
Loop Through a Dictionary

You can loop through a dictionary by using a for loop.

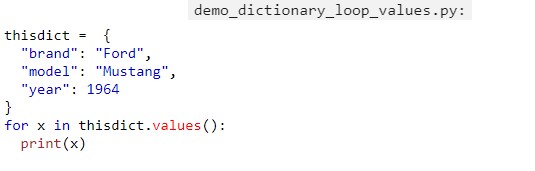
When looping through a dictionary, the return value are the keys of the dictionary, but there are methods to return the values as well.

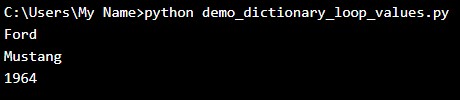
Print all key names in the dictionary, one by one:



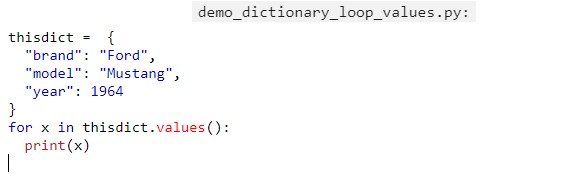


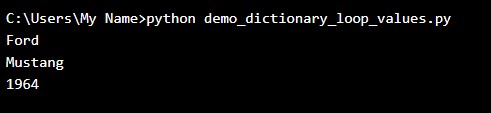
Print all values in the dictionary, one by one:



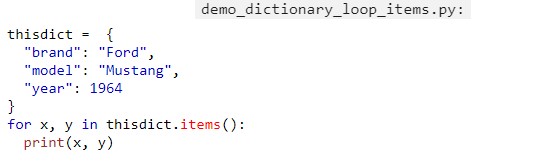


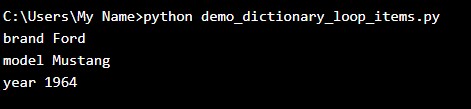
You can also use the values () function to return values of a dictionary:





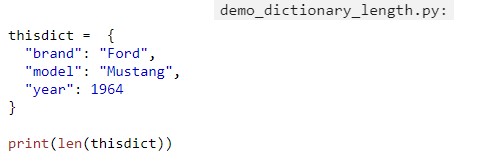
Loop through both keys and values, by using the items() function:

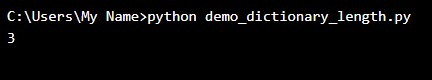




Dictionary Length

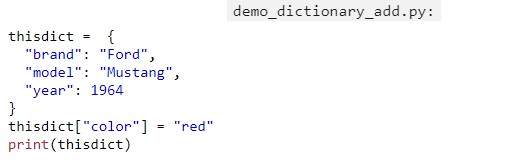
To determine how many items (key-value pairs) a dictionary have, use the len() method.

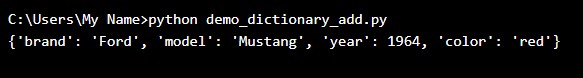




Adding New Items

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

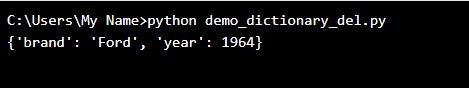
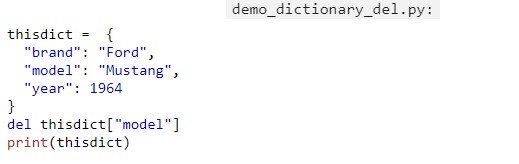




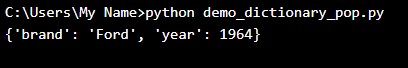
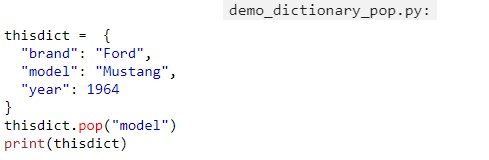
Removing Items

There are several methods to remove items from a dictionary.

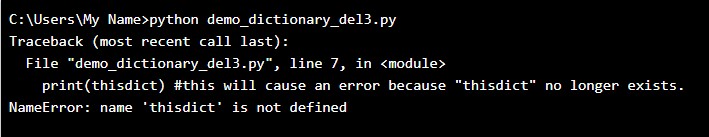
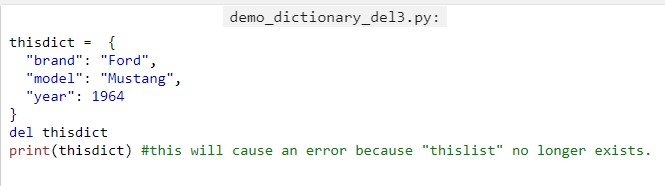
The *del* keyword removes the item with the specified key name.



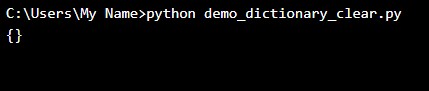
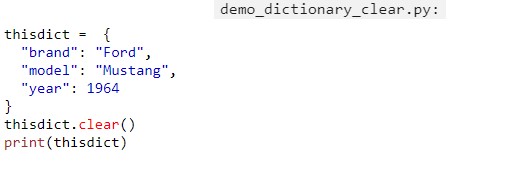
The *pop()*method removes the item with the specified key name:



The del keyword can also delete the dictionary completely:



The clear() keyword empties the dictionary:



This time you’ll not see the error.

The dict() Constructor

It is also possible to use the dict() constructor to make a dictionary:

