Python Dictionary:

Dictionaries are a useful data structure for storing data in Python because they are capable of imitating real-world data arrangements where a certain value exists for a given key.

The data is stored as key-value pairs using a Python dictionary.

* This data structure is mutable
* The components of dictionary were made using keys and values.
* Keys must only have one component.
* Values can be of any type, including integer, list, and tuple.

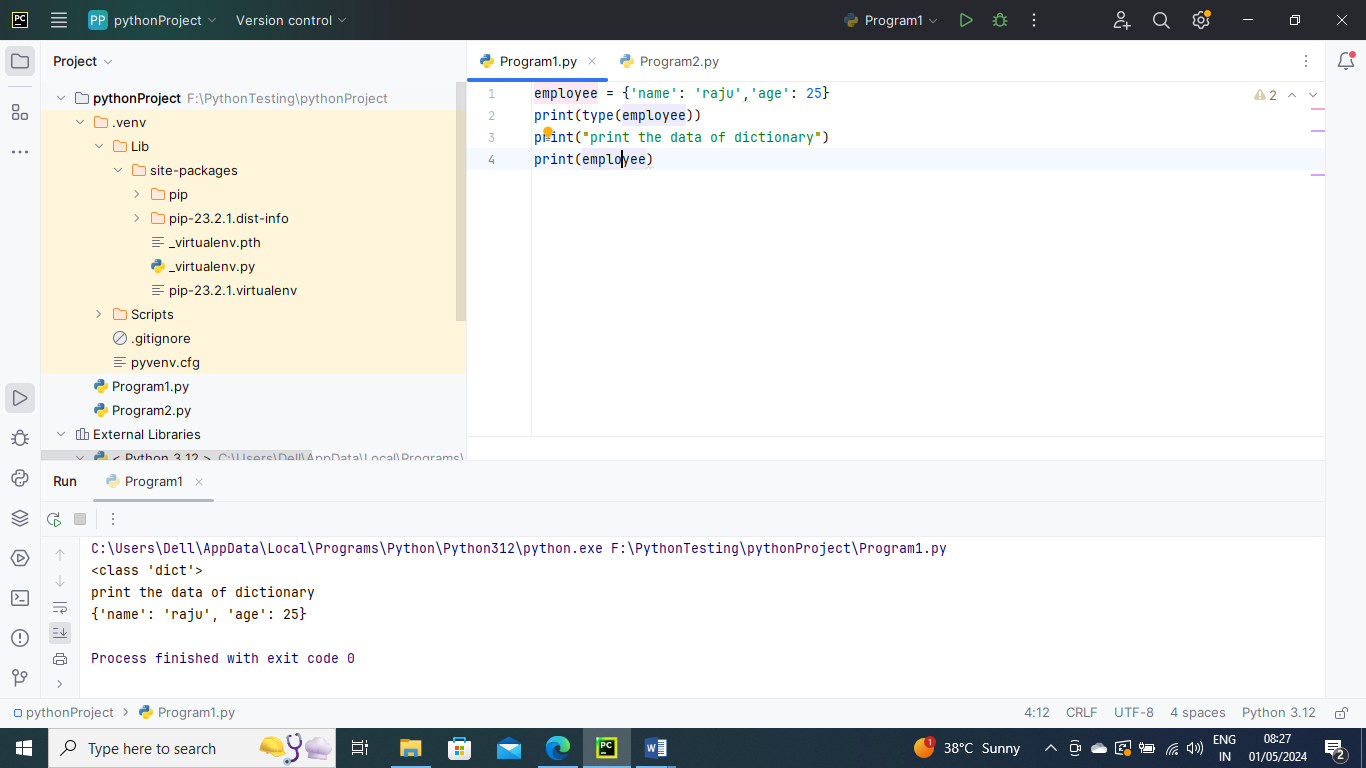
Creating Dictionary:

Curly brackets are the simplest way to generate a Python dictionary, although there are other approaches as well. With many key-value pairs surrounded in curly brackets and a colon separating each key from its value, the dictionary can be built. (:). The following provides the syntax for defining the dictionary.

Synatx:

employee = {‘name’ : ‘raju’, ‘age’ : 25}

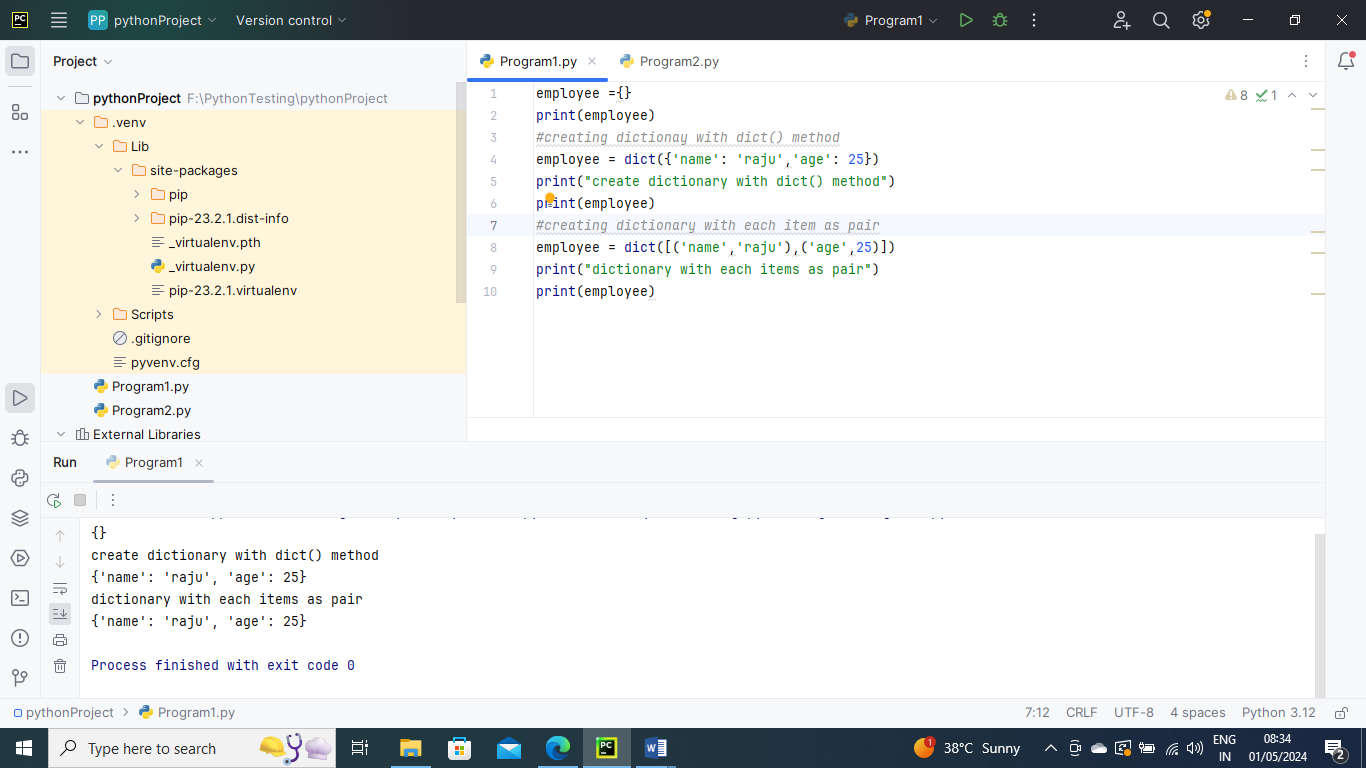
In the above dictionary **employee**, The keys **Name** and **Age** are the strings which comes under the category of an immutable object.



-----------

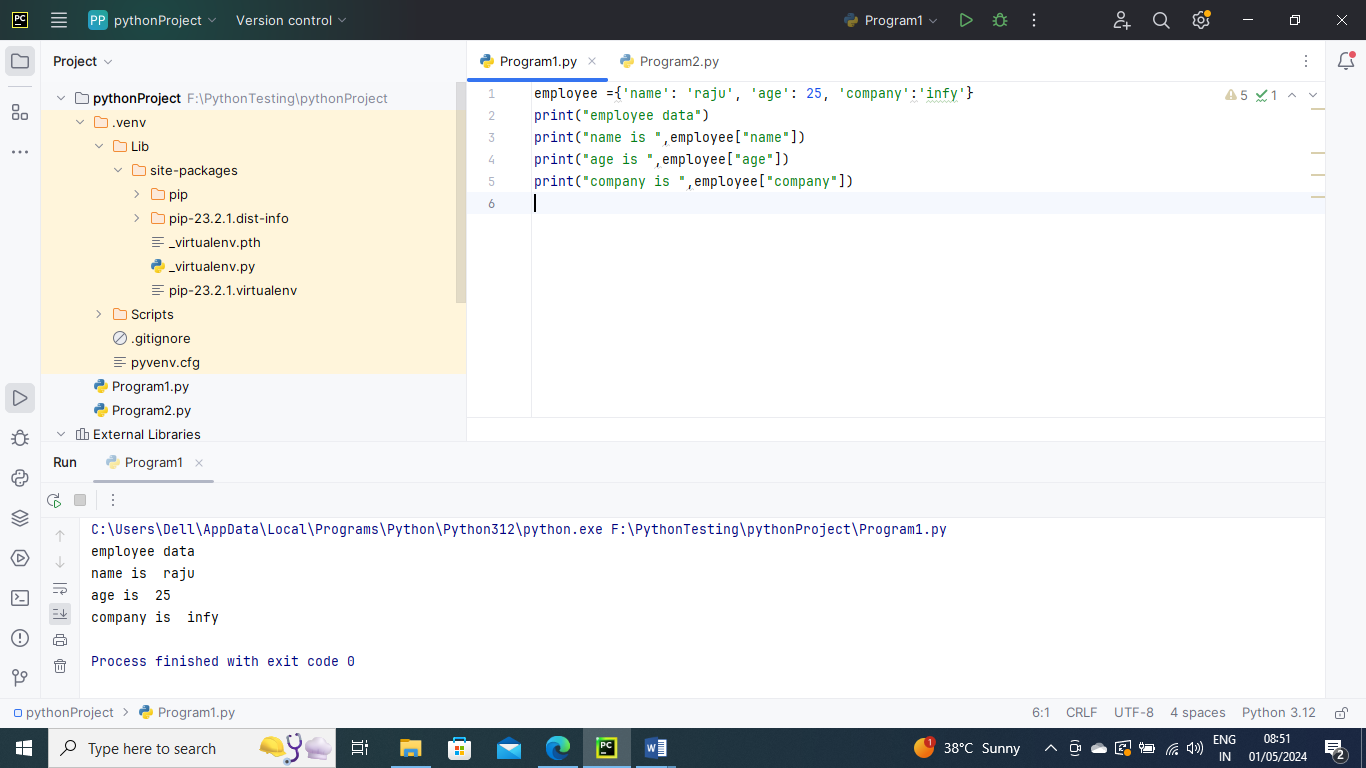
Python provides built in function **dict()** method which is also used to create dictionary

The empty curly braces used to create empty dictionary



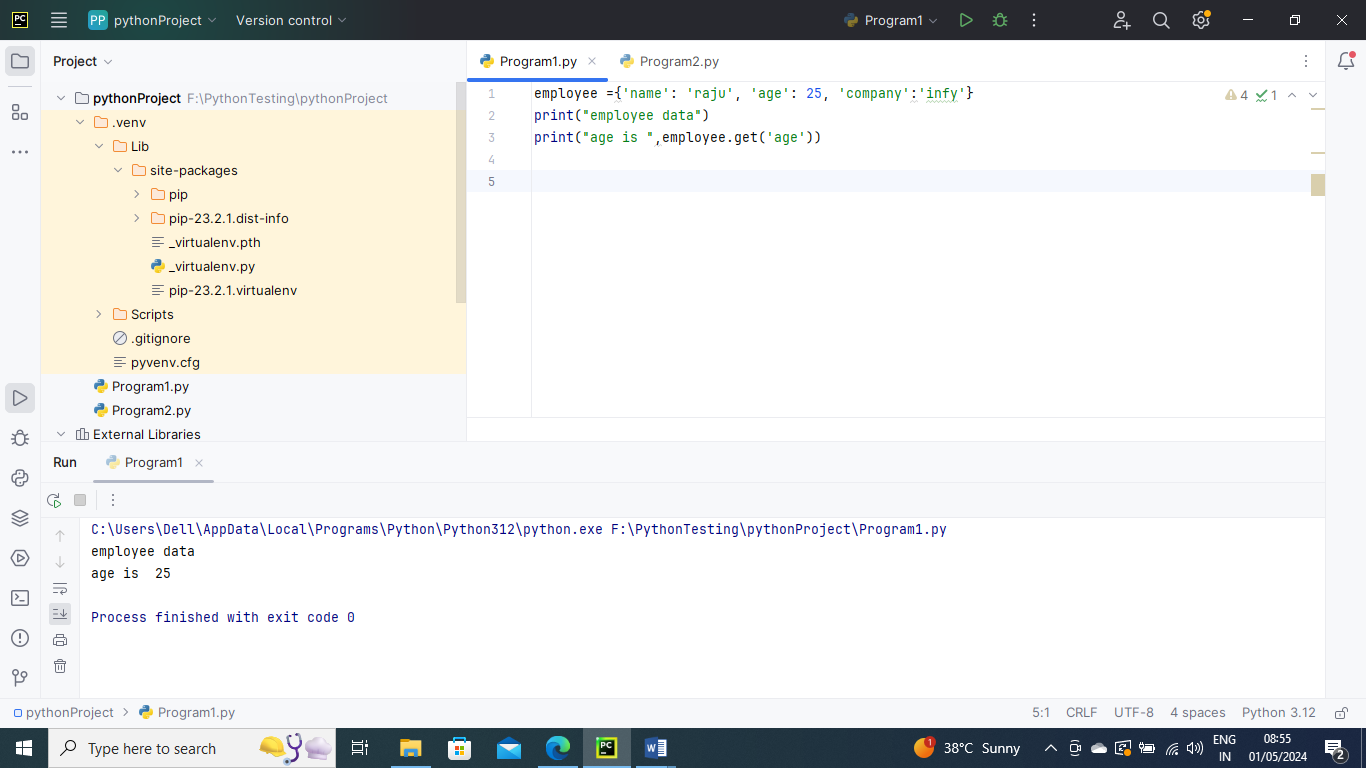
Accessing the dictionary values:

To access data contained in lists and tuples, indexing has been studied. The keys of the dictionary can be used to obtain the values because they are unique from one another. The following method can be used to access dictionary values.



---------------

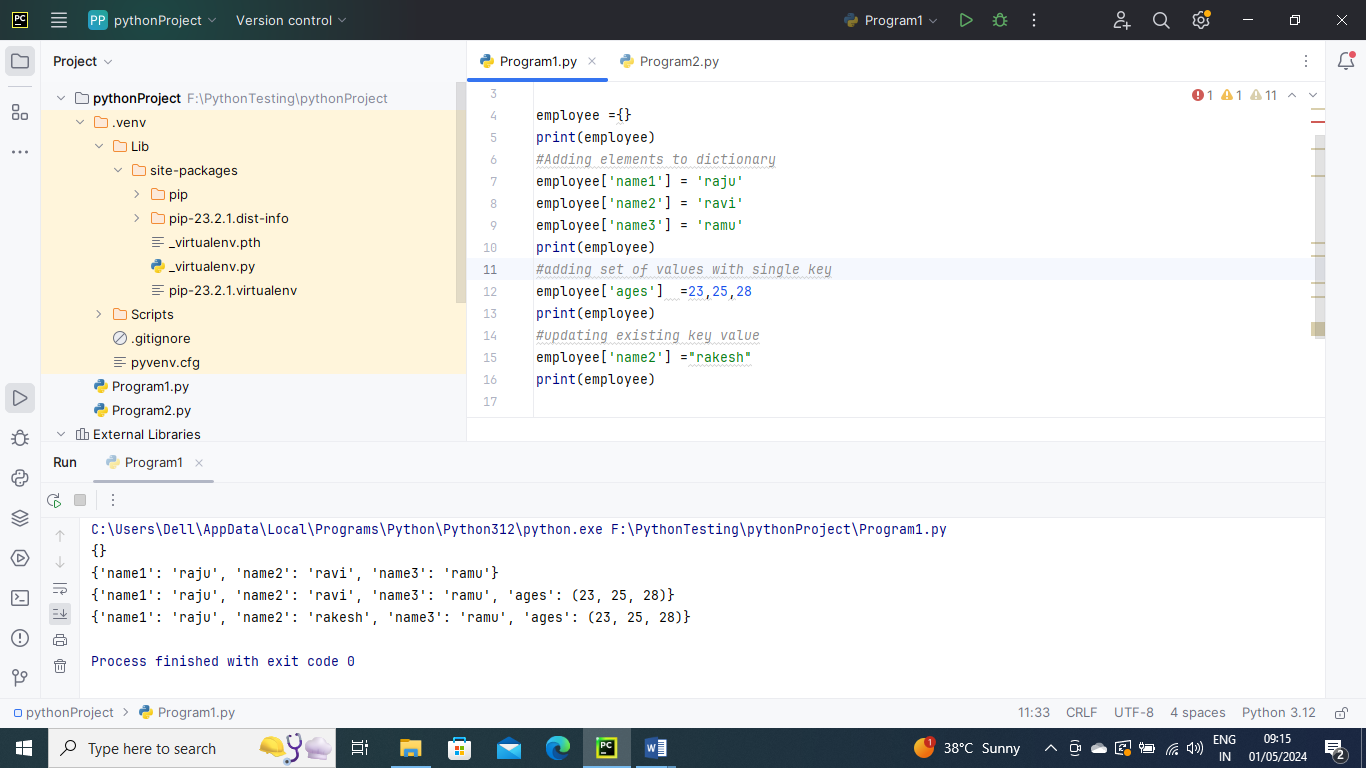
Python provides us with an alternative to use the get() method to access the dictionary values. It would give the same result as given by the indexing.



---------

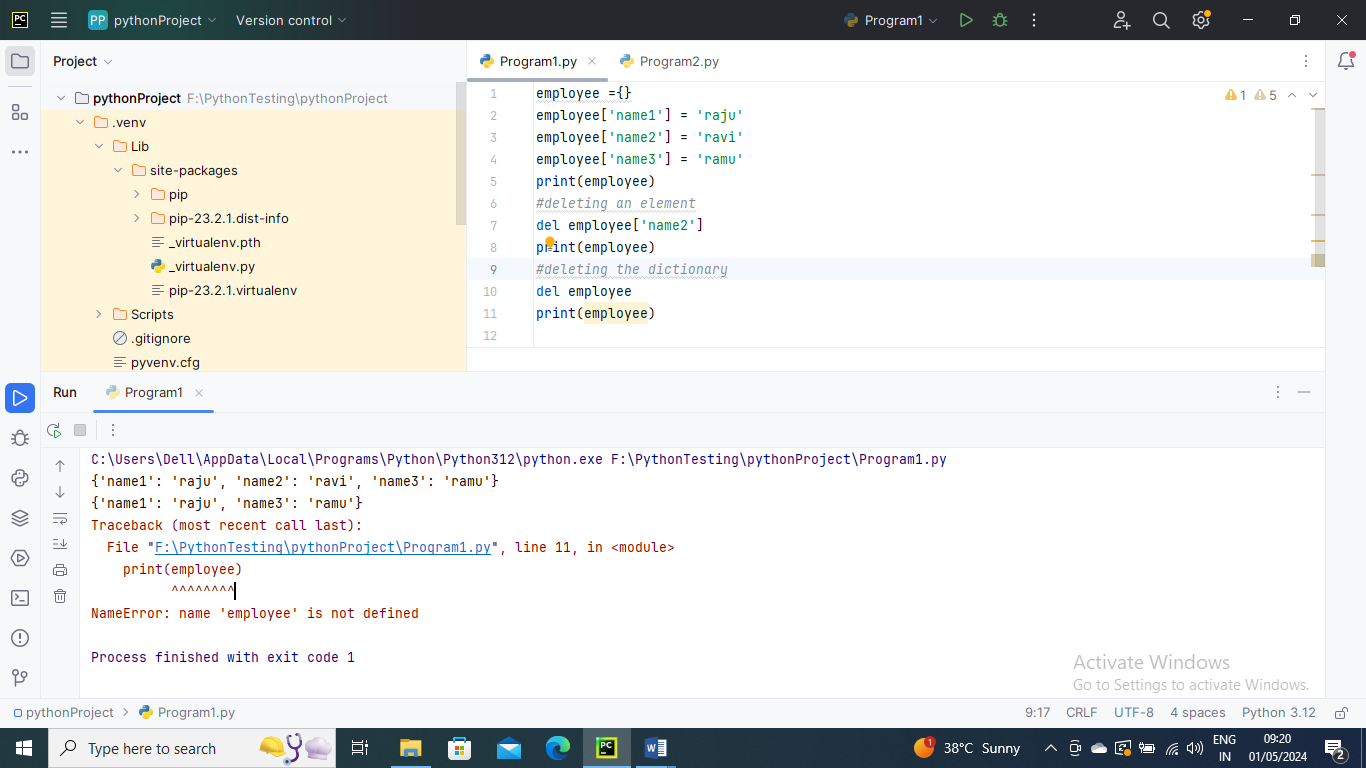
Adding Dictionary values

The dictionary is a mutable data type, and utilising the right keys allows you to change its values. Dict[key] = value and the value can both be modified. An existing value can also be updated using the update() method



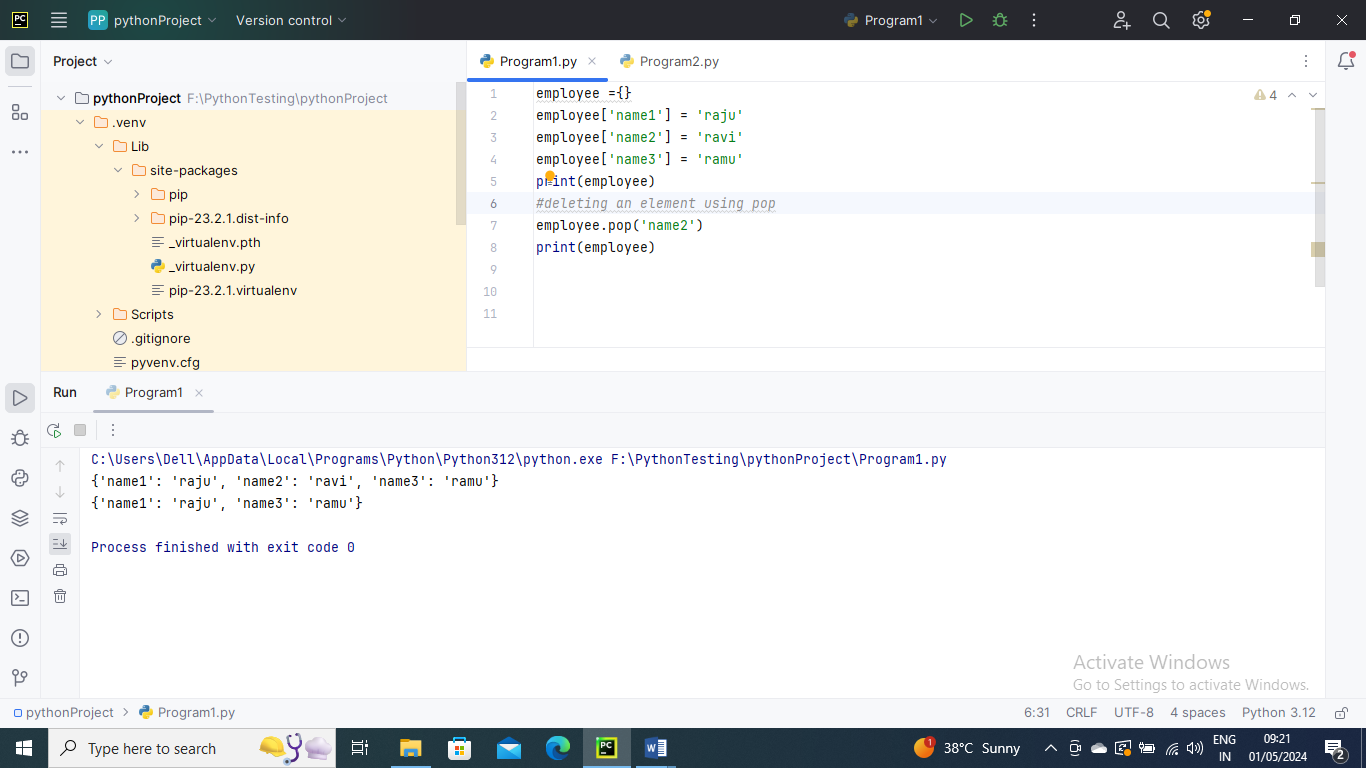
-----------------

Deleting elements using del keyword



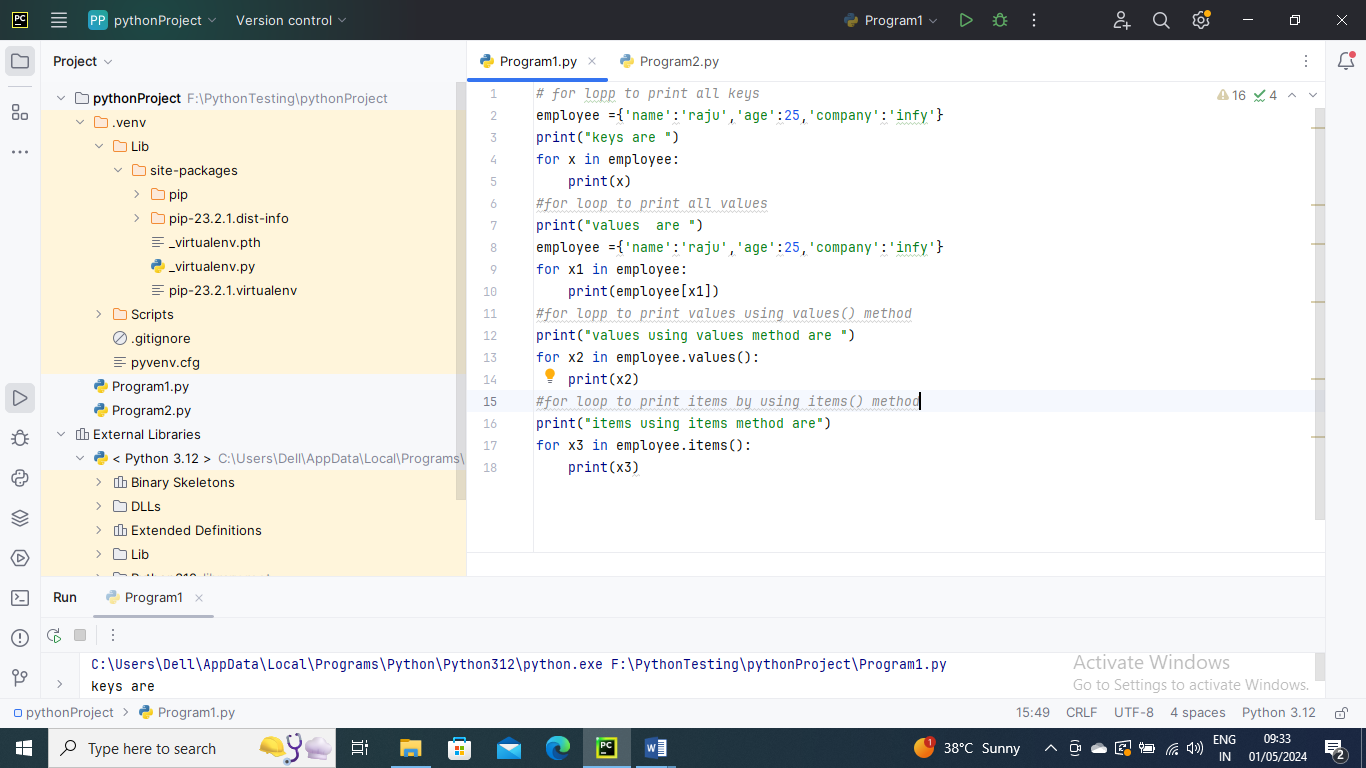
---------------

Deleting elements using pop() method

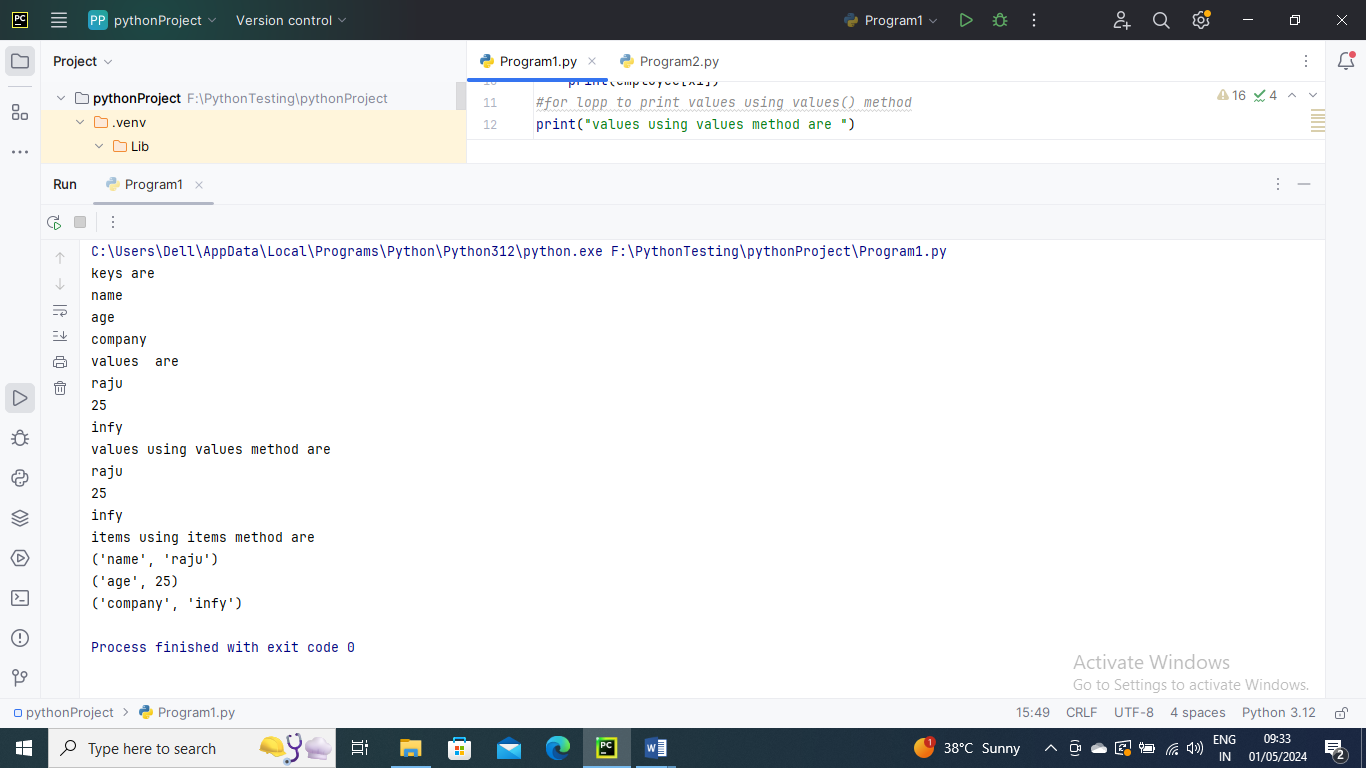


---------------

Iterating Dictionary



Outputs:



-----------

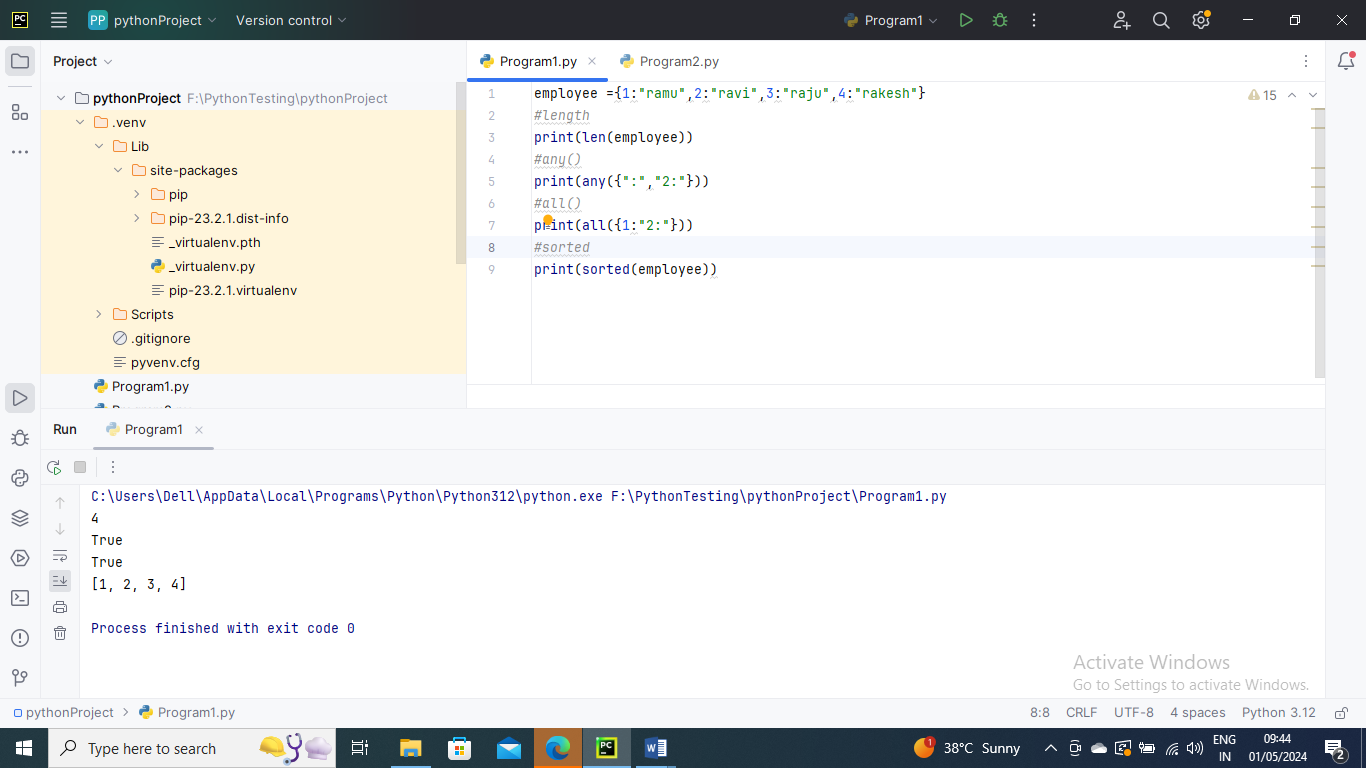
Built in Dictionary Functions

len() – It gives length of dictionary

any() – it returns true indeed if one dictionary key does have a Boolean expression that evaluates to true

all() – Unlike in any() method, all() only returns rtrue if each of the dictionary’skey contain a true Boolean value

sorted() – it returns ordered series of dictionary keys. The ascending sorting has no effect on original python dictionary



--------------------

Built in Dictionary Methods

clear() – delete all items in dictionary

copy() – returns a shallow copy of the dictionary which is created

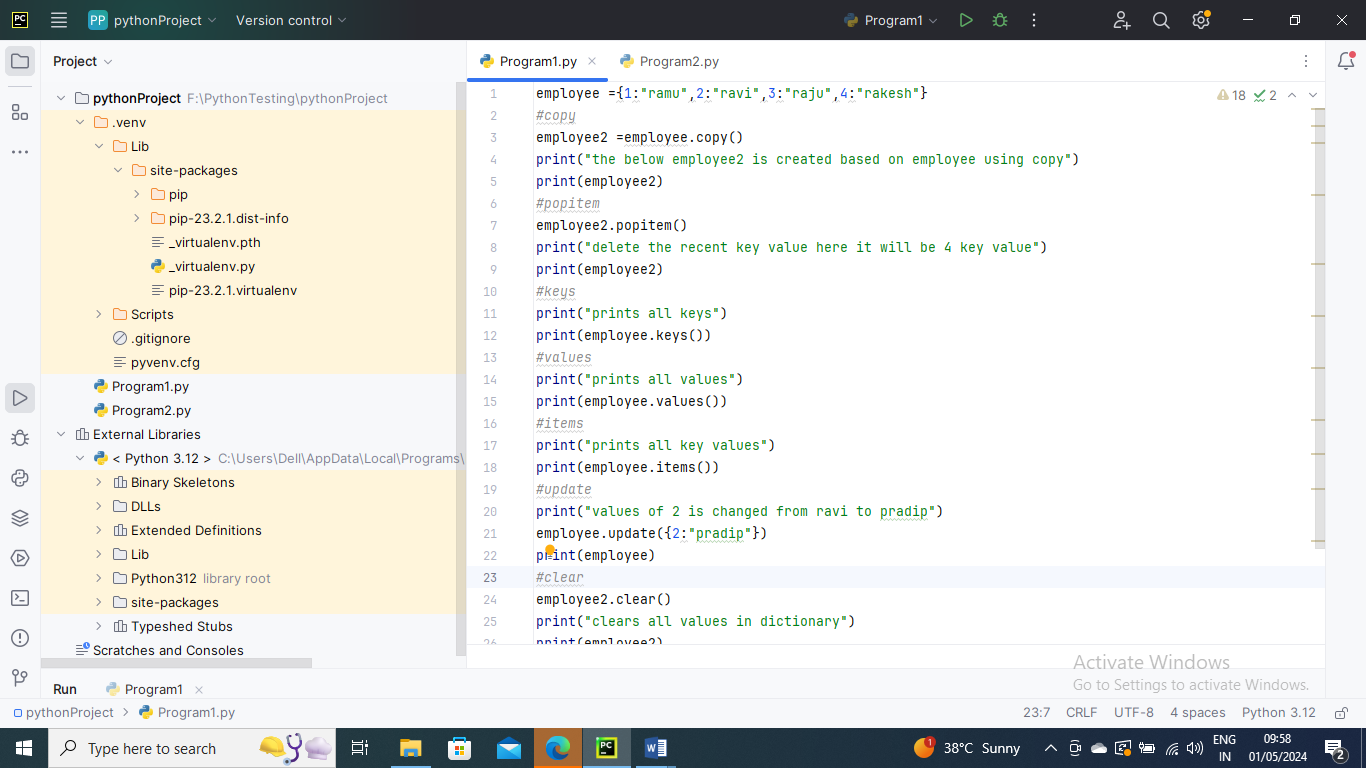
popitem() – removes the recent key-value pair

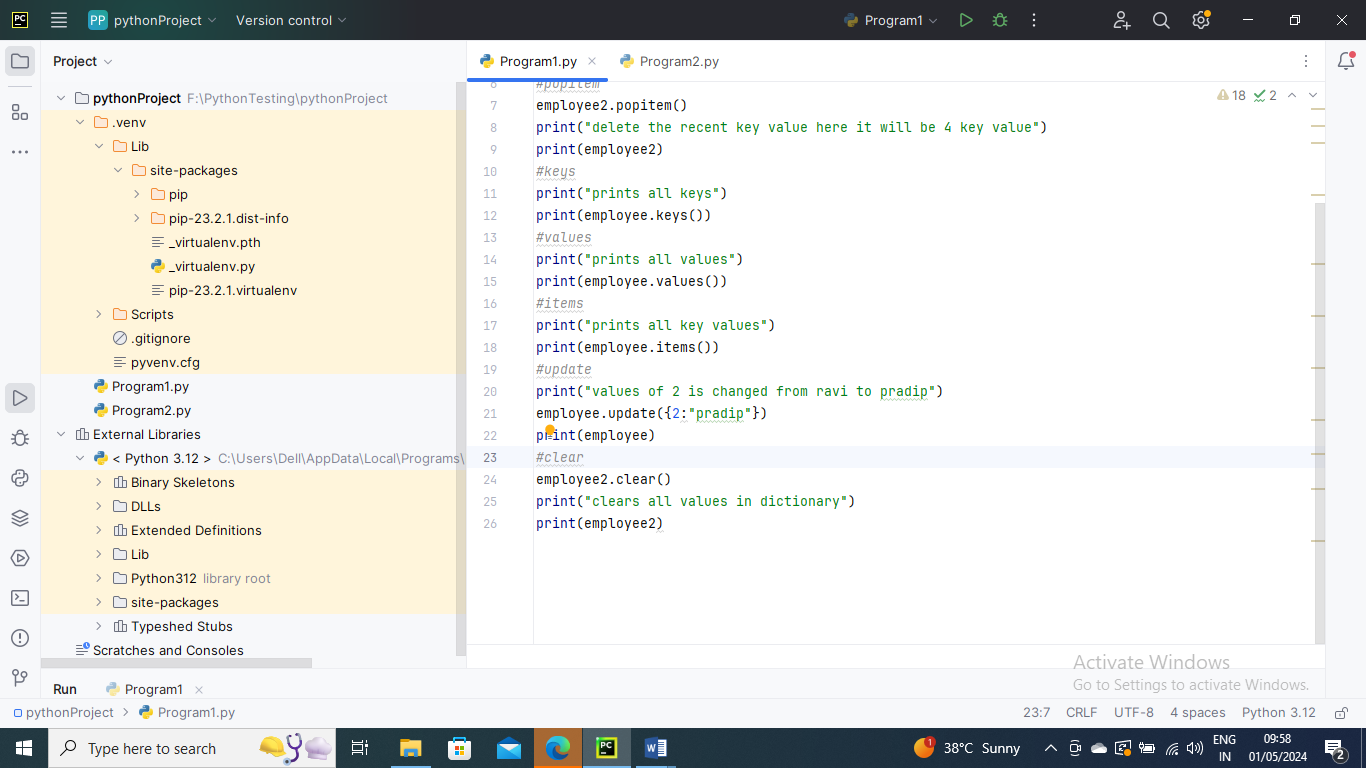
keys() – returns all the keys of dictionary

values() – returns all the vaues of dictionary

items() – returns all key-values pairs as a tuple

update() – for updating a value of a key





Outputs:

