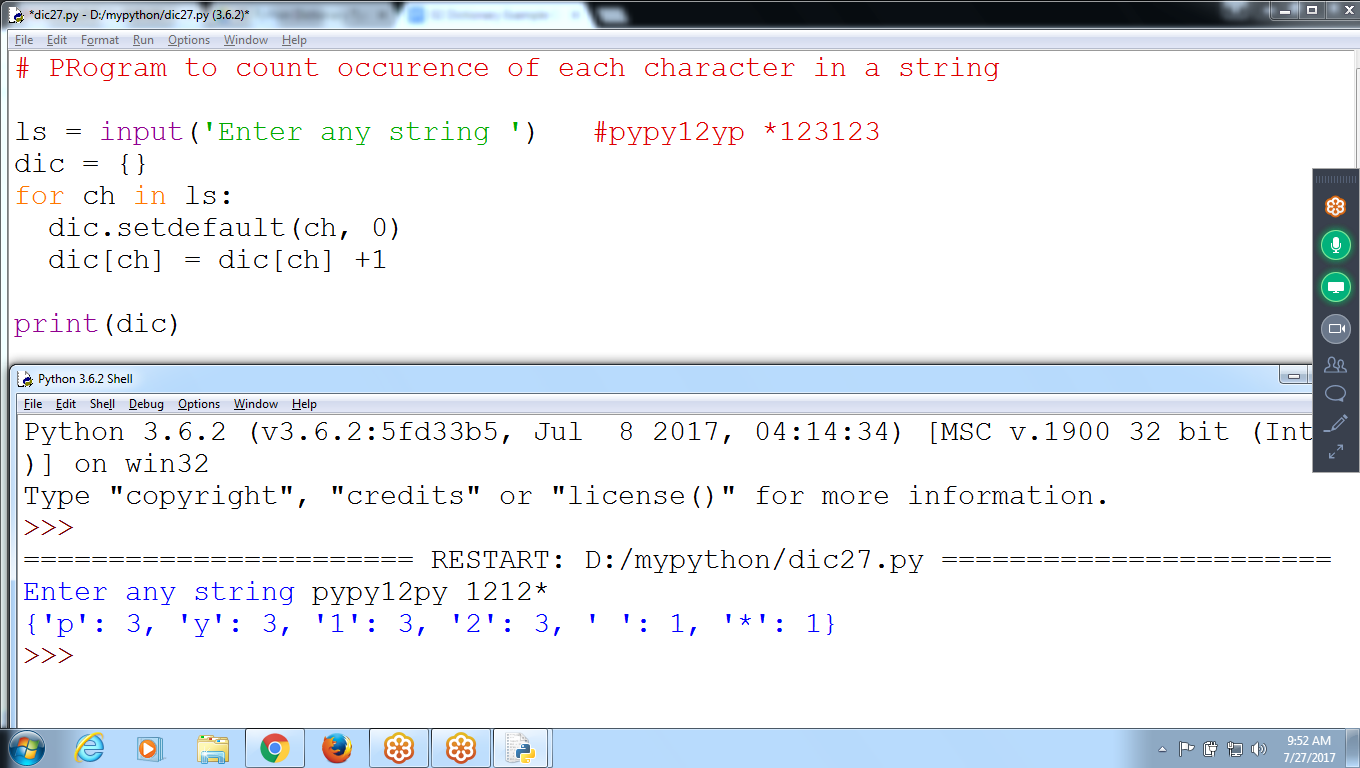
program that counts the number of occurrences of each letter in a string.

setdefault() : checks for key, if not existing creates new key and initialize with 0



# PRogram to count occurrence of each character in a string

ls = input('Enter any string ') #pypy12yp \*123123

dic = {}

for ch in ls:

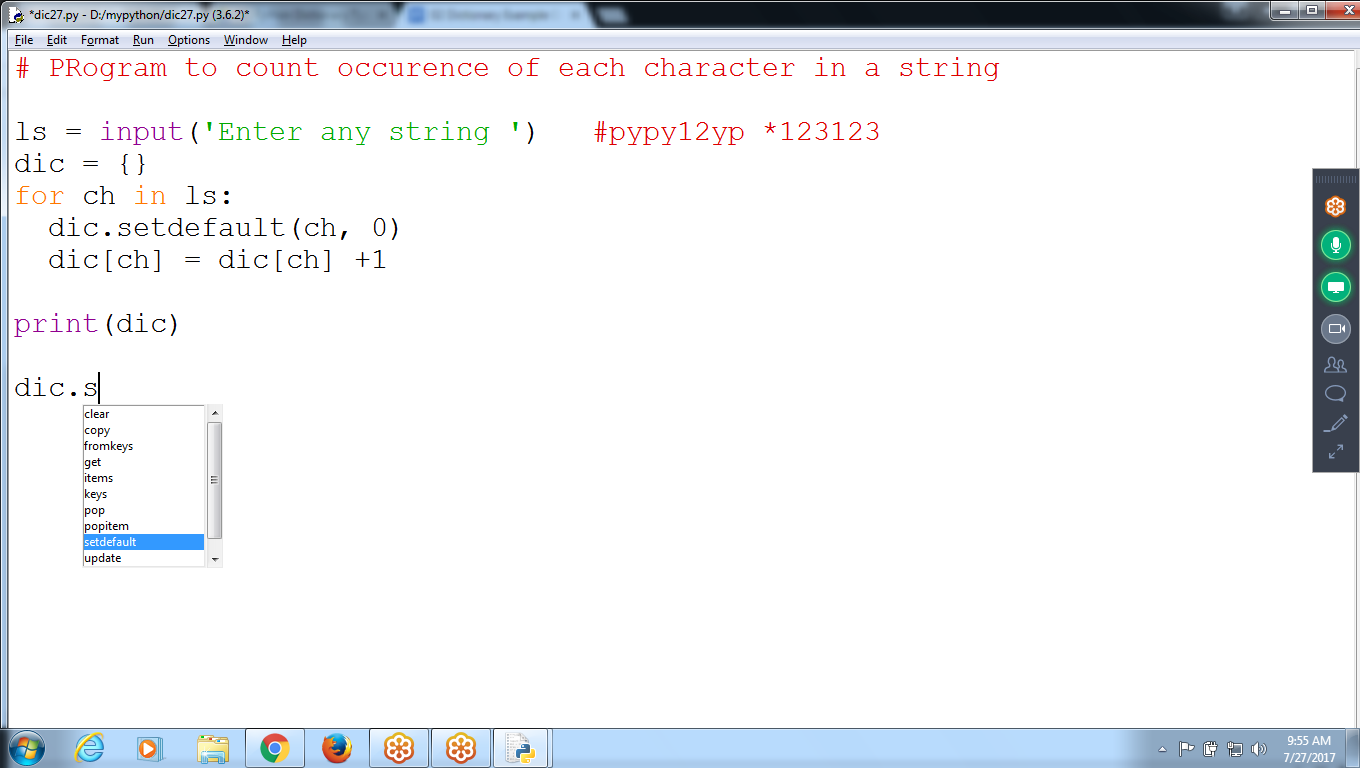
dic.setdefault(ch, 0)

dic[ch] = dic[ch] +1

print(dic)

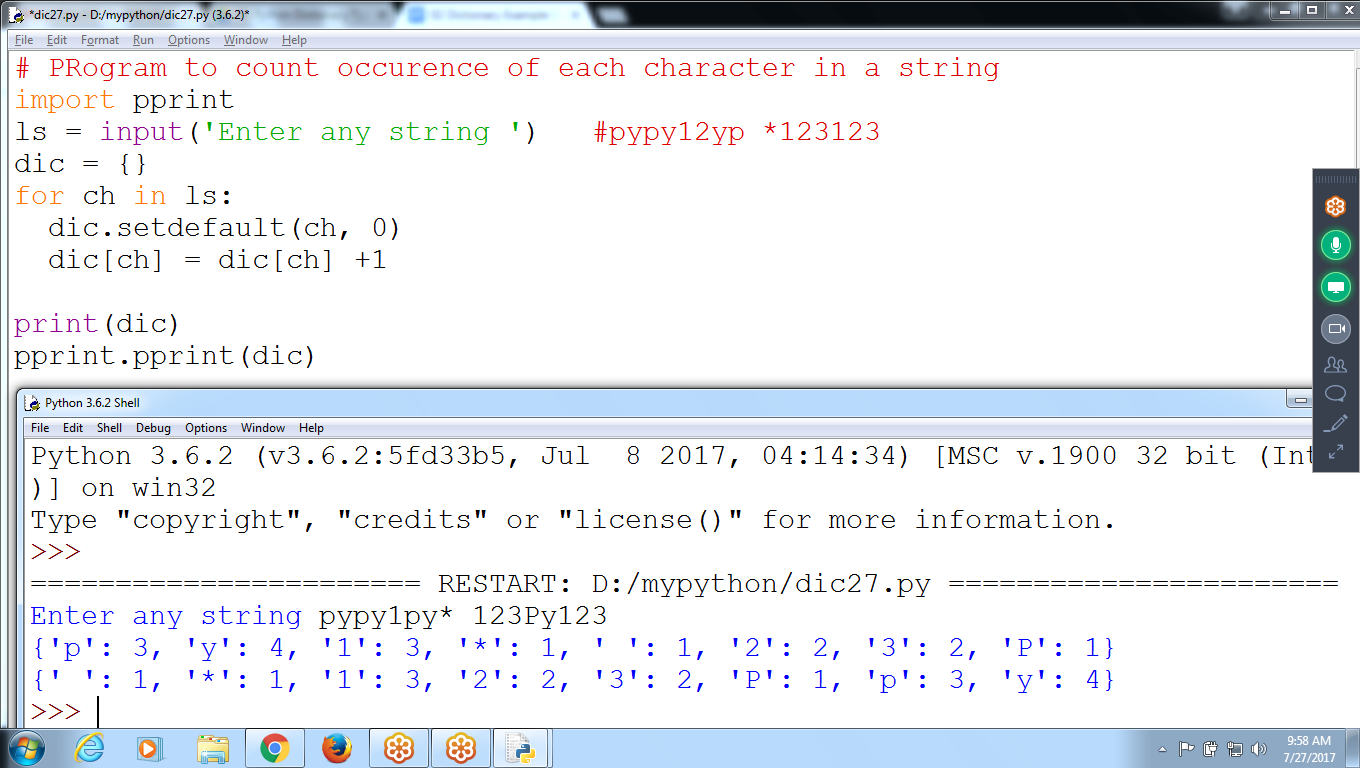
In Dictionary don’t have SORT() method

To sort values in dictionary can use “pprint” Module



# **using pprint module can sort dictionary values**

import the **pprint** module can access to the **pprint() and pformat()** functions. A cleaner display of the items in a dictionary than what print() provides



# PRogram to count occurrence of each character in a string

import pprint

ls = input('Enter any string ') #pypy12yp \*123123

dic = {}

for ch in ls:

dic.setdefault(ch, 0)

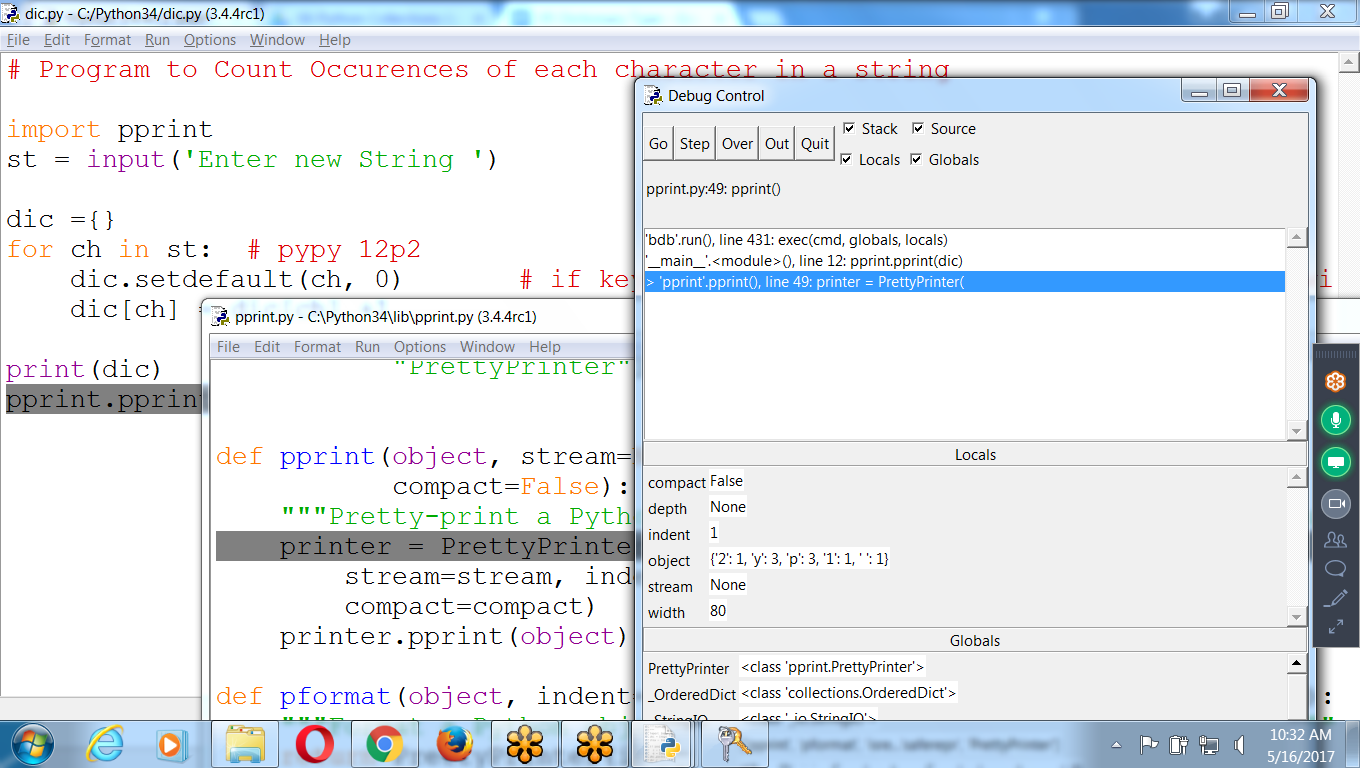
dic[ch] = dic[ch] +1

print(dic)

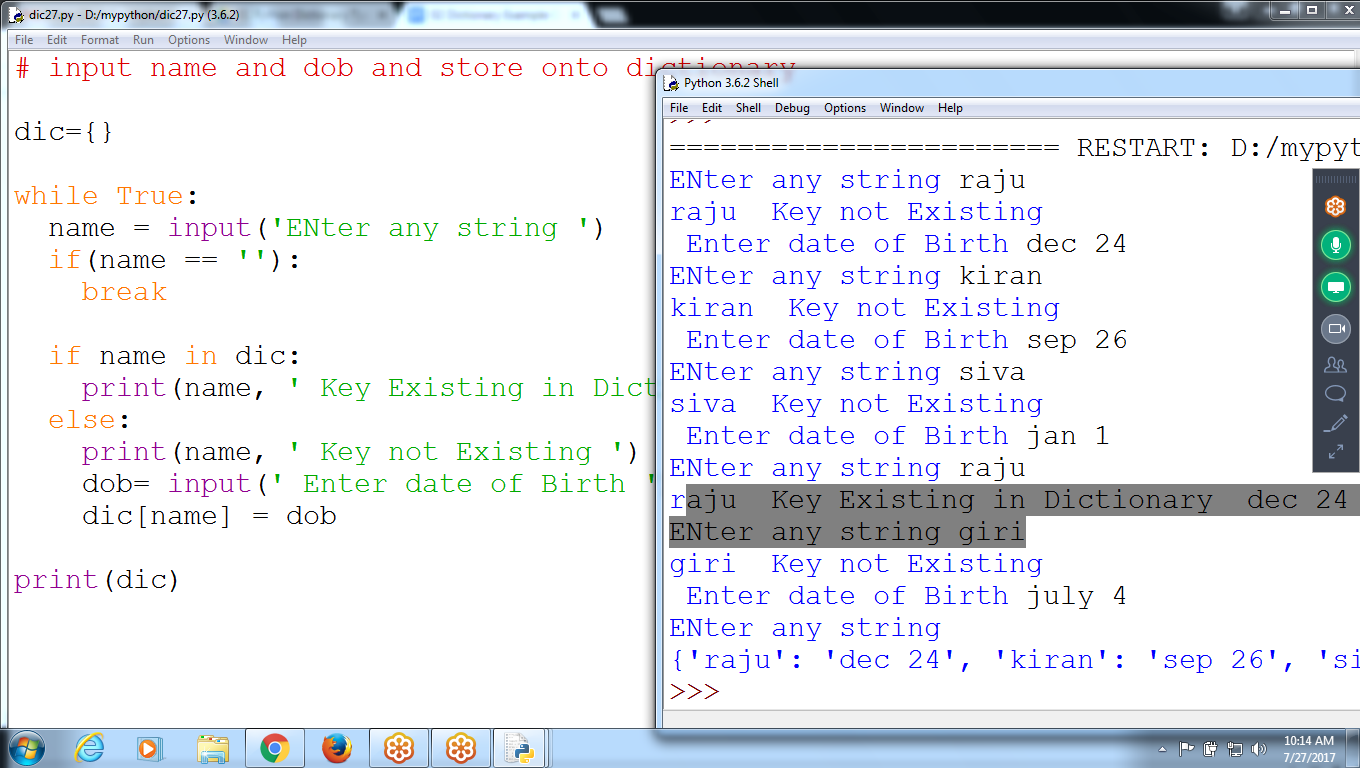
pprint.pprint(dic)

**Debugger ON can see logic for pprint module**

On reaching pprint() click “STEP” : moves function



Write Python program to accept names and store onto Dictionary as key



# input name and dob and store onto dictionary

dic={}

while True:

name = input('ENter any string ')

if(name == ''):

break

if name in dic:

print(name, ' Key Existing and dob is ', dic[name])

else:

print(name, ' Key not Existing ')

dob= input(' Enter date of Birth ')

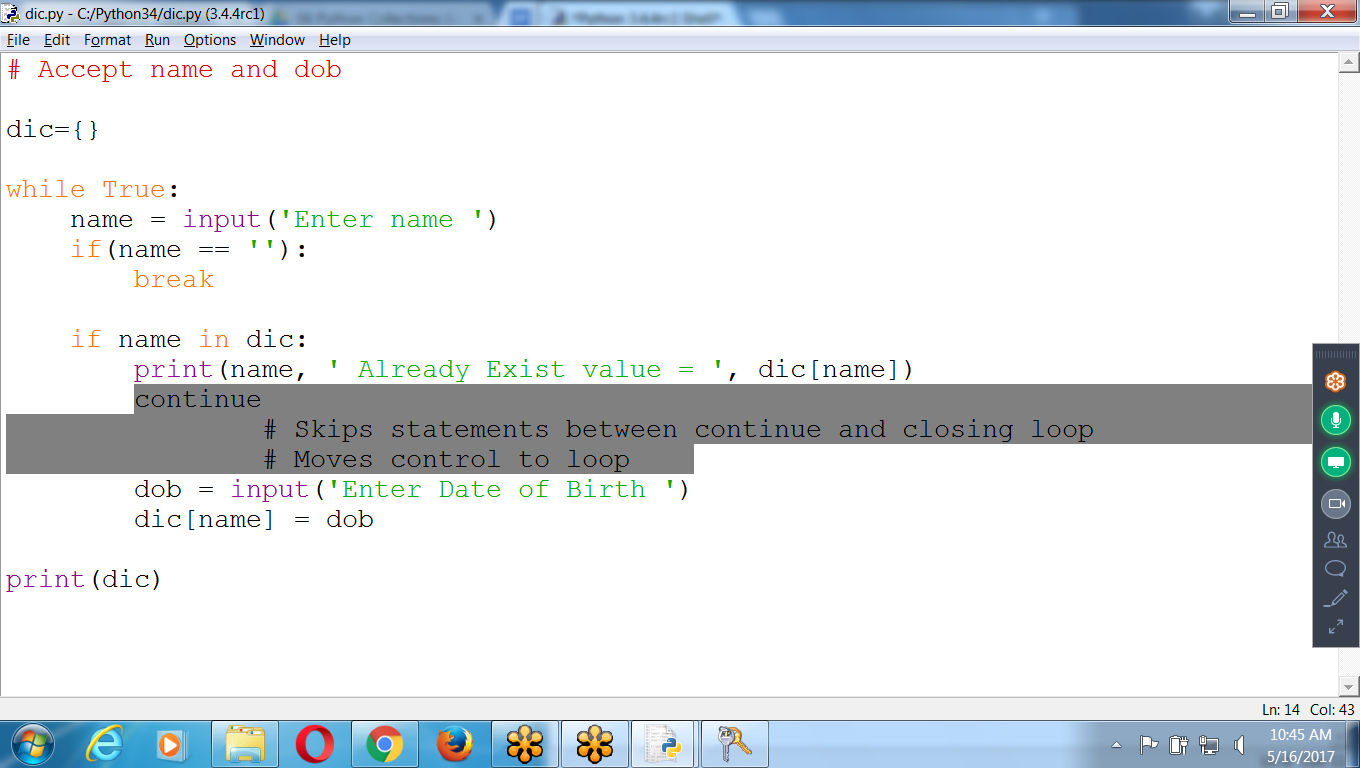
dic[name] = dob

print(dic)

continue

# Skips statements between continue and closing loop

# Moves control to starting of the loop



# Accept name and dob

dic={}

while True:

name = input('Enter name ')

if(name == ''):

break

if name in dic:

print(name, ' Already Exist value = ', dic[name])

continue

# Skips statements between continue and closing loop

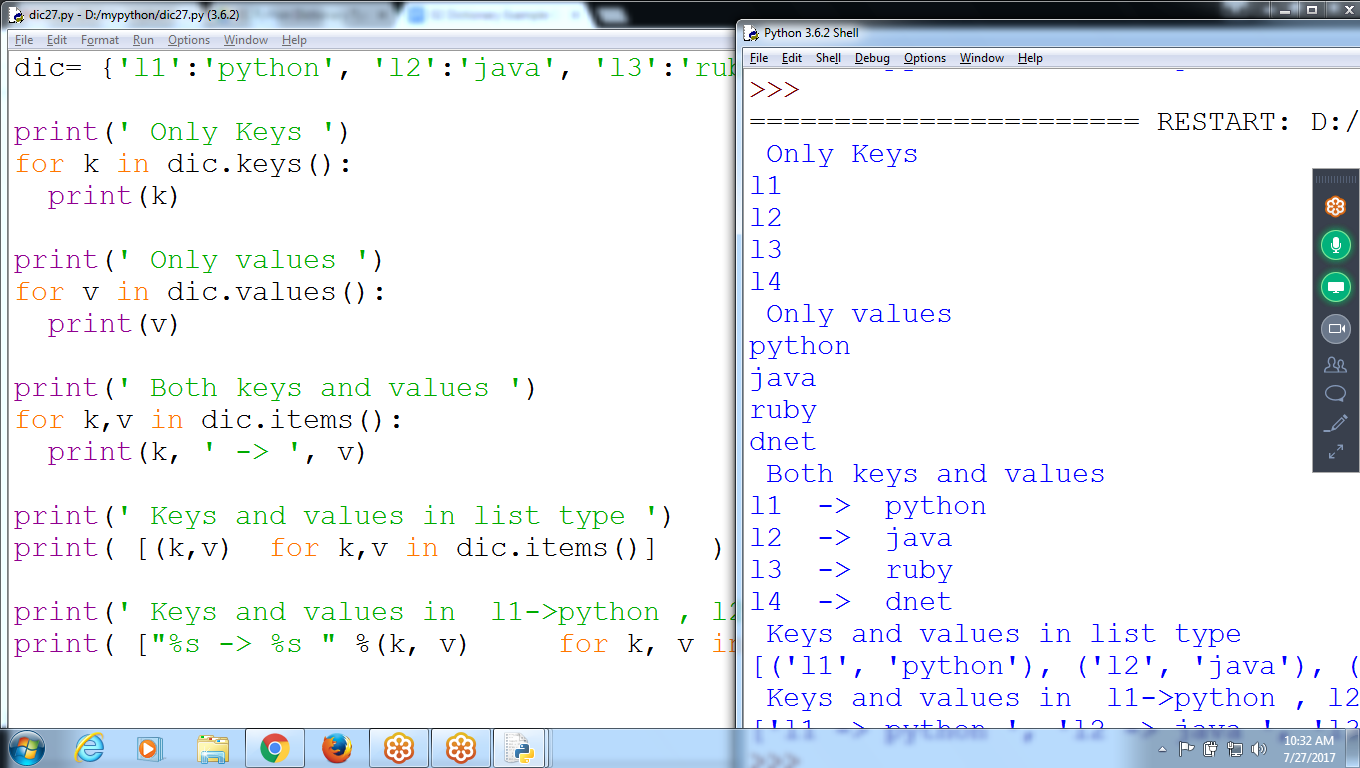
# Moves control to loop

dob = input('Enter Date of Birth ')

dic[name] = dob

print(dic)

**List Comprehensions with Dictionary**

****

dic= {'l1':'python', 'l2':'java', 'l3':'ruby', 'l4':'dnet'}

print(' Only Keys ')

for k in dic.keys():

print(k)

print(' Only values ')

for v in dic.values():

print(v)

print(' Both keys and values ')

for k,v in dic.items():

print(k, ' -> ', v)

print(' Keys and values in list type ')

print( [(k,v) for k,v in dic.items()] )

print(' Keys and values in l1->python , l2->java ')

**print( ["%s -> %s " %(k, v) for k, v in dic.items()] )**

>>> s = {'a':1, 'b':2, 'c':3}

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Explanation** | **Example** | **Result on s** |
| s.keys() | return a list of keys of ‘s’ | s.keys() | [‘a’, ‘b’, ‘c’] |
| s.values() | return a list of values of ‘s’ | s.values() | [1, 2, 3] |
| s.items() | return key-value pairs of ‘s’ | s.items() | [(‘a’, 1), (‘b’, 2), (‘c’, 3)] |
| s.get(k [, x]) | s[k] if k in ‘s’ else x | s.get(‘e’, 5) | 5 (no change in ‘s’) |
| s.setdefault(k [, x]) | s[k] if k in ‘s’, else x (also setting it) | s.setdefault({‘e’, 5}) | 5 (‘s’ contains {“e”:5}) |
| s.update([b]) | update ‘s’ with key-value pairs from ‘b’, return None | s.update({‘d’:4}) | {‘a’:1, ‘b’:2, ‘c’:3, ‘d’:4} |
| del s[k] | remove s[k] from ‘s’ | del s[‘b’] | {‘a’:1, ‘c’:3} |
| s.pop(k [, x]) | s[k] if k in ‘s’, else x (and remove s[k]) | s.pop(‘c’, 5) | {‘a’:1, ‘b’:2} |