DICTIONARIES

WHAT IS DICTIONARY?

WHAT IS DICTIONARY?

A dictionary is like a list, but more in general. In a list, index value is an integer, while in a dictionary index value can be any other data type and are called keys. The key will be used as a string as it is easy to recall. A dictionary is an extremely useful data storage construct for storing and retrieving all key value pairs, where each element is accessed (or indexed) by a unique key. However, dictionary keys are not in sequences and hence maintain no left-to right order.

KEY VALYE PAIR

We can refer to a dictionary as a mapping between a set of indices (which are called keys) and a set of values. Each key maps a value. The association of a key and a value is called a key-value pair.

Syntax:

```
my_dict = {'key1': 'value1','key2': 'value2','key3': 'value3'…'keyn': 'valuen'}
```

DICTIONARIES

- ✓Curley brackets are used to represent a dictionary.
- ✓Each pair in the dictionary is represented by a key and value separated by a colon.
- ✓Multiple pairs are separated by vcomas

DICTIONARIES

✓A dictionary is an unordered collection of key-value pairs.

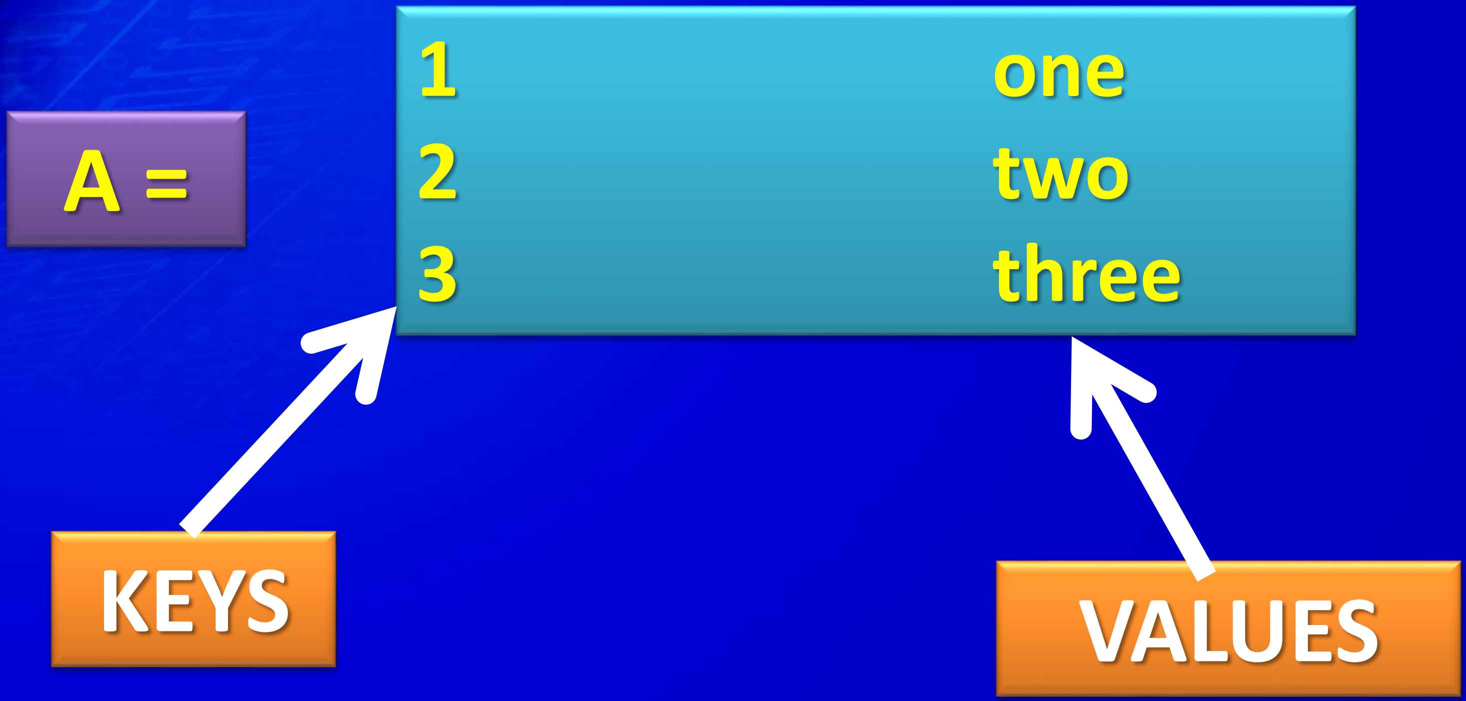
✓A dictionary has a length, specifically the numb er of keyvalue pairs.

✓A dictionary provides fast look up by key.

✓The keys must be immutable object types.

STATE DIAGRAM

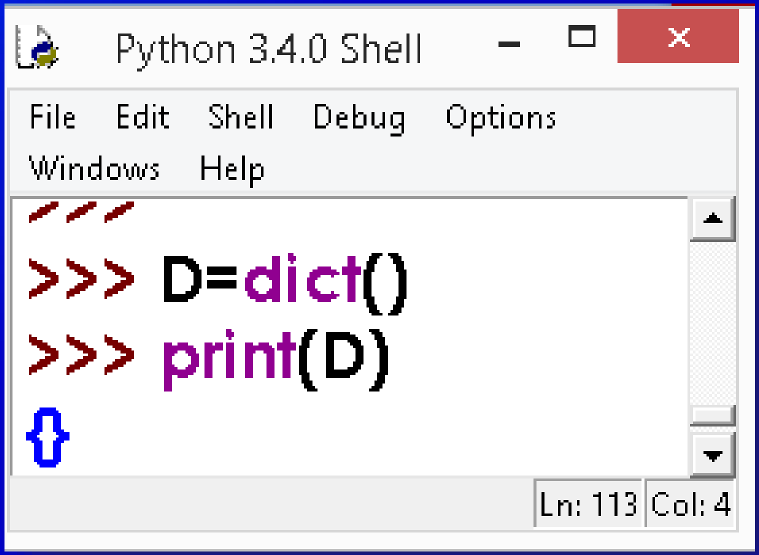
```
>>> A={1:"one",2:"two",3:"three"}
```



CREATING A DICTIONARY – dict()

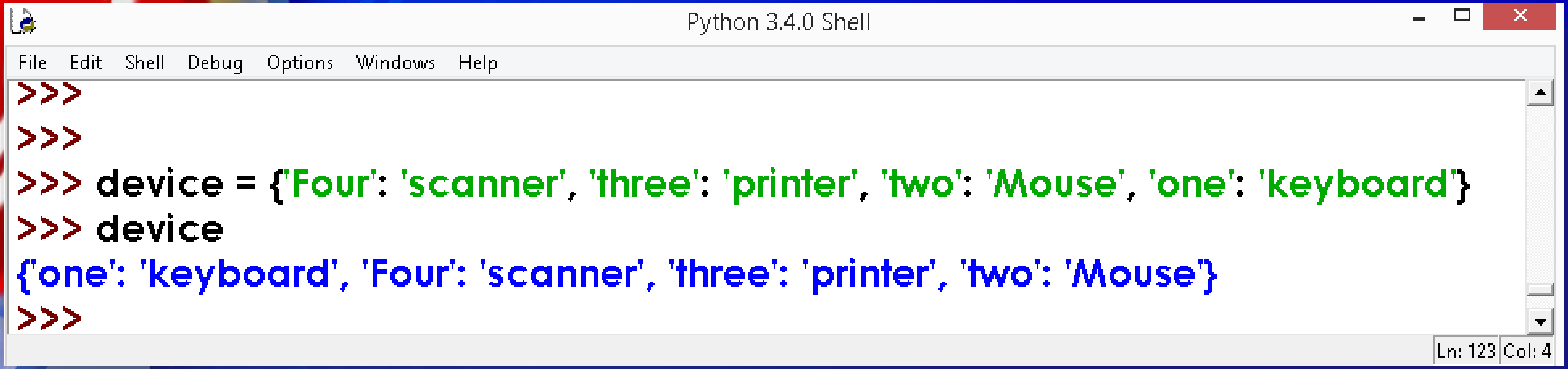
CREATING DICTIONARAY – dict()

The function dict ( ) is used to create a new dictionary with no items. This function is called built-in function. We can also create dictionary using {}.



```
>>> D=dict()
>>> print(D)
{}
```

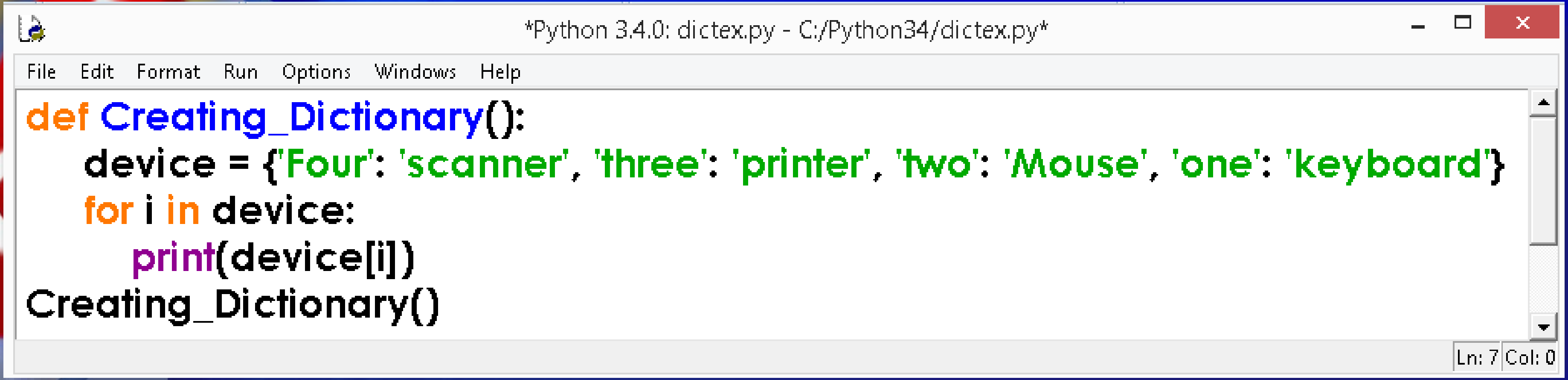
CREATING DICTIONARAY – Example



```
>>>
>>>
>>> device = {'Four': 'scanner', 'three': 'printer', 'two': 'Mouse', 'one': 'keyboard'}
>>> device
{'one': 'keyboard', 'Four': 'scanner', 'three': 'printer', 'two': 'Mouse'}
>>>
```

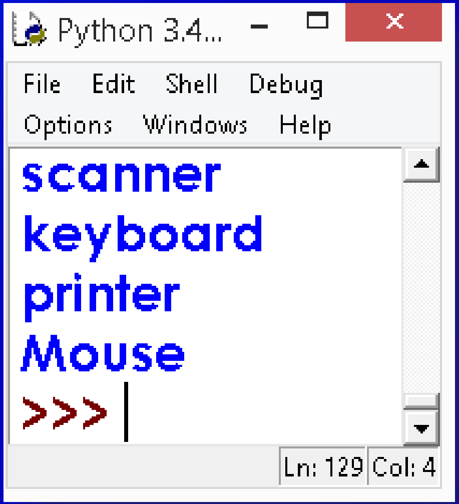
CREATING AND TRAVERSING DICTIONARAY

CREATING AND TRAVERSING DICTIONARAY


Python 3.4.0: dictex.py - C:/Python34/dictex.py
File Edit Format Run Options Windows Help
Ln: 7 Col: 0

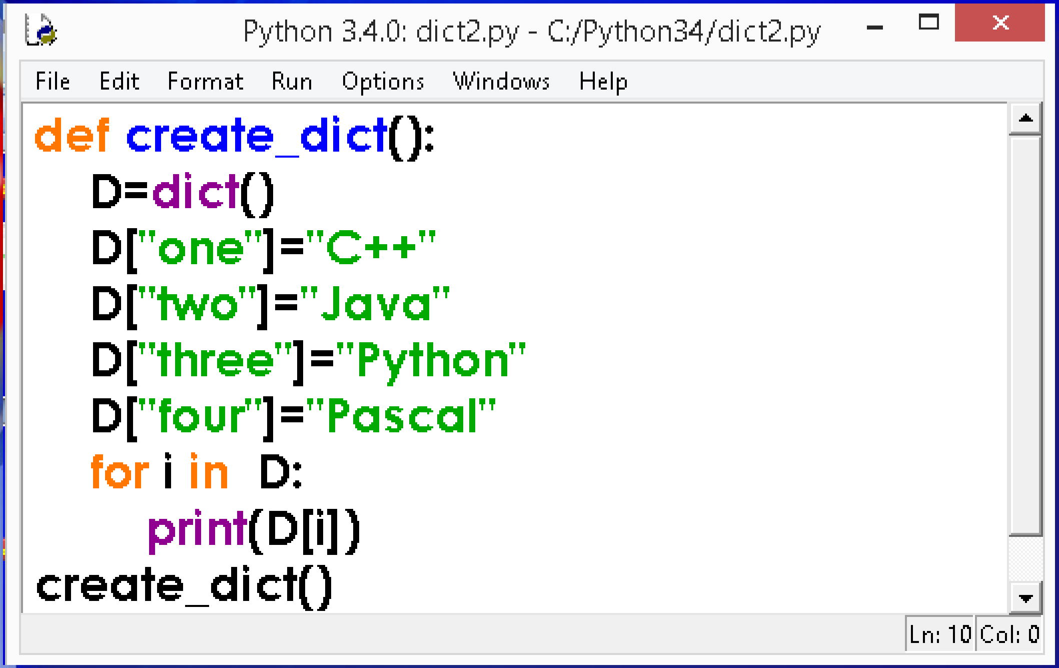
```
def Creating_Dictionary():
    device = {'Four': 'scanner', 'three': 'printer', 'two': 'Mouse', 'one': 'keyboard'}
    for i in device:
        print(device[i])
Creating_Dictionary()
```

OUT PUT



```
scanner
keyboard
printer
Mouse
>>> 
```

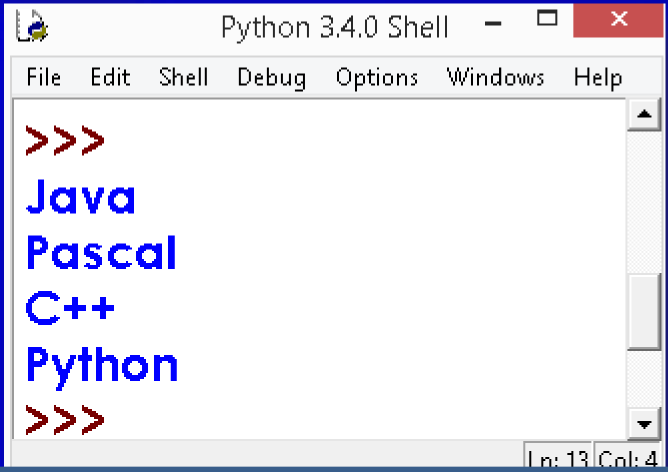
CREATING AND TRAVERSING DICTIONARAY



Python 3.4.0: dict2.py - C:/Python34/dict2.py
File Edit Format Run Options Windows Help
Ln: 10 Col: 0

```python
def create_dict():
    D=dict()
    D["one"]="C++"
    D["two"]="Java"
    D["three"]="Python"
    D["four"]="Pascal"
    for i in  D:
        print(D[i])
create_dict()
```

OUT PUT



```
>>>
Java
Pascal
C++
Python
>>>
```

DICTIONARAY – BUILT IN METHODS

DICTIONARAY – BUILT IN METHODS

Dictionary Method	Meaning
dict.clear()	Removes all the elements of the dictionary
dict.copy()	Returns (shallow)copy of dictionary.
dict.get(key, default=None)	for key key, returns value or default if key not in dictionary (note that default's default is None)
dict.items()	returns a list of dict's (key, value) tuple pairs

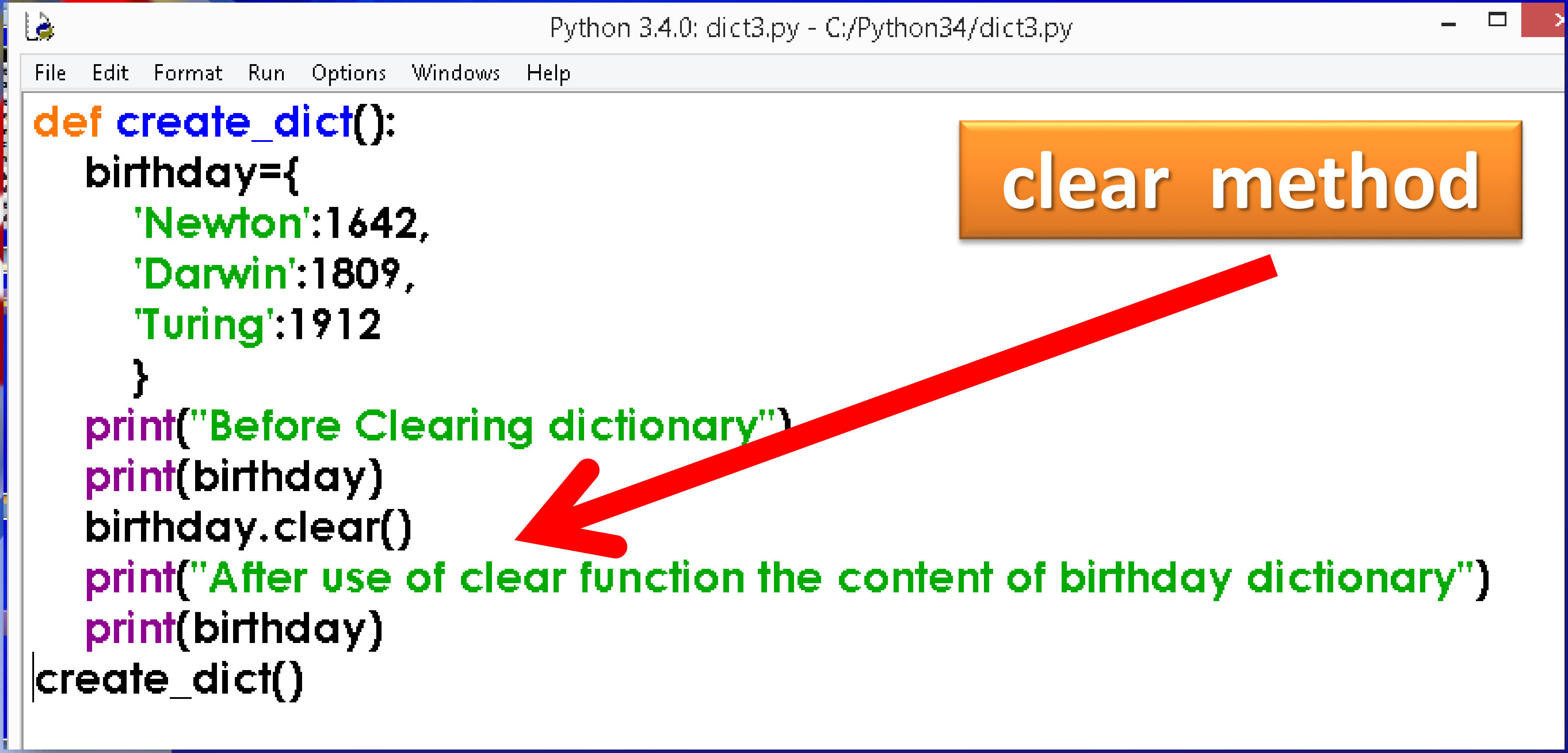
DICTIONARAY – BUILT IN METHODS

Dictionary Method	Meaning
dict.keys()	returns list of dictionary dict's keys
dict.setdefault key, default=None	similar to get(), but will set dict[key]=default if key is not already in dict
dict.update(dict2)	adds dictionary dict2's key-values pairs to dict
dict.values()	returns list of dictionary dict's values

DICTIONARAY – BUILT IN METHODS

Dictionary Method	Meaning
dict.pop()	returns list of dictionary dict's keys
dict.popitem()	similar to get(), but will set dict[key]=default if key is not already in dict

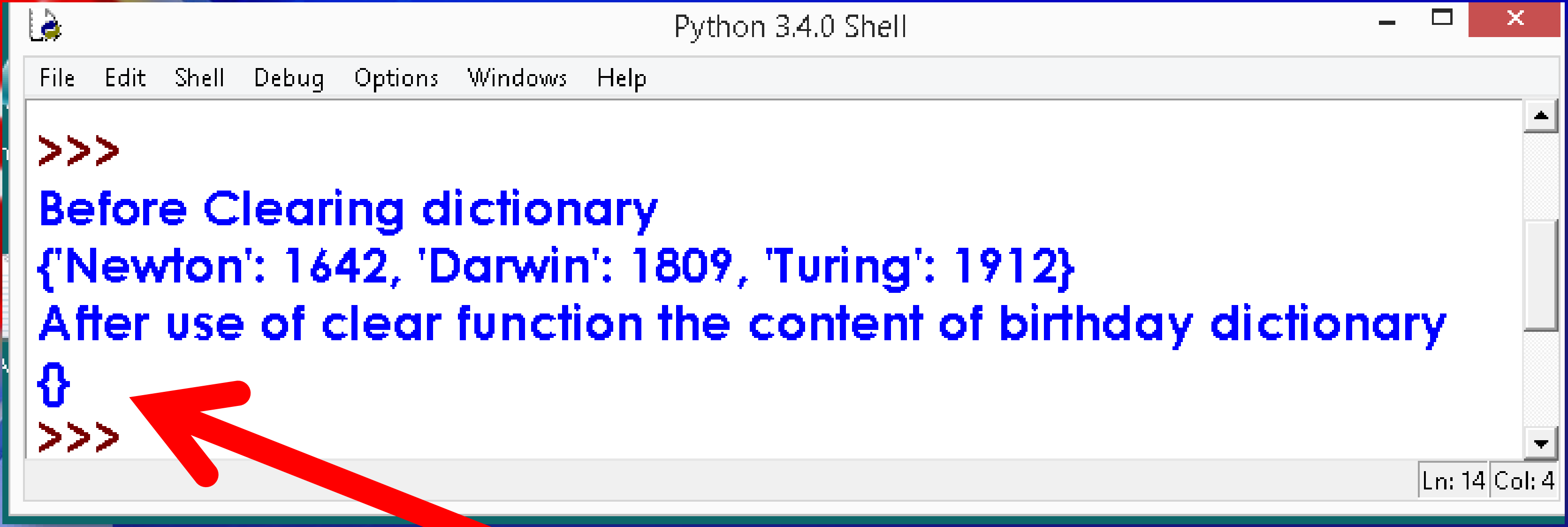
dict.clear() METHOD


Python 3.4.0: dict3.py - C:/Python34/dict3.py
File Edit Format Run Options Windows Help
clear method

```
def create_dict():
    birthday={
        'Newton':1642,
        'Darwin':1809,
        'Turing':1912
        }
    print("Before Clearing dictionary")
    print(birthday)
    birthday.clear()
    print("After use of clear function the content of birthday dictionary")
    print(birthday)
create_dict()
```

OUTPUT is in next slide!

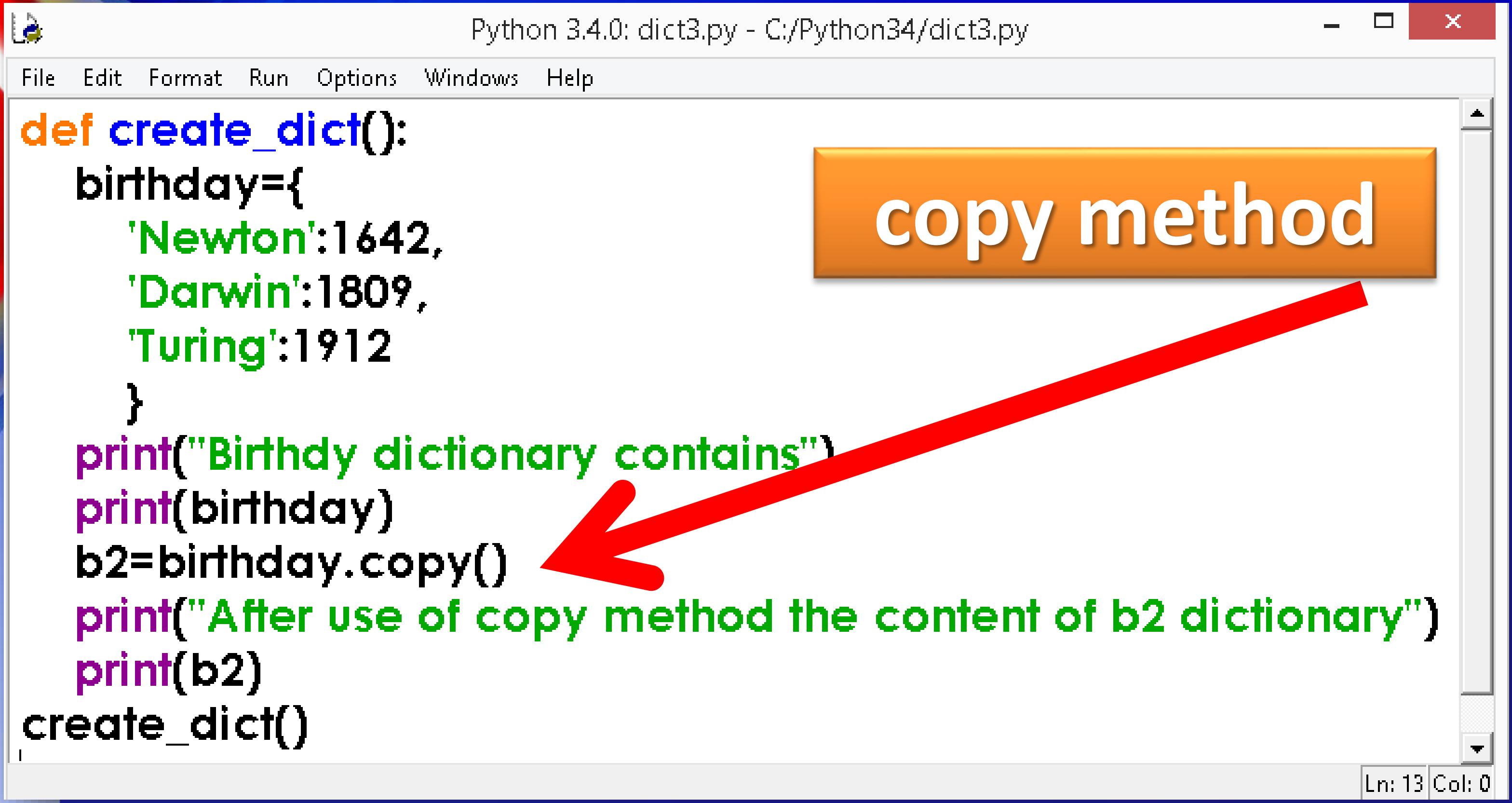
dict.clear() METHOD - OUTPUT



```
>>>
Before Clearing dictionary
{'Newton': 1642, 'Darwin': 1809, 'Turing': 1912}
After use of clear function the content of birthday dictionary
{}
>>>
```

Birthday dictionary cleared

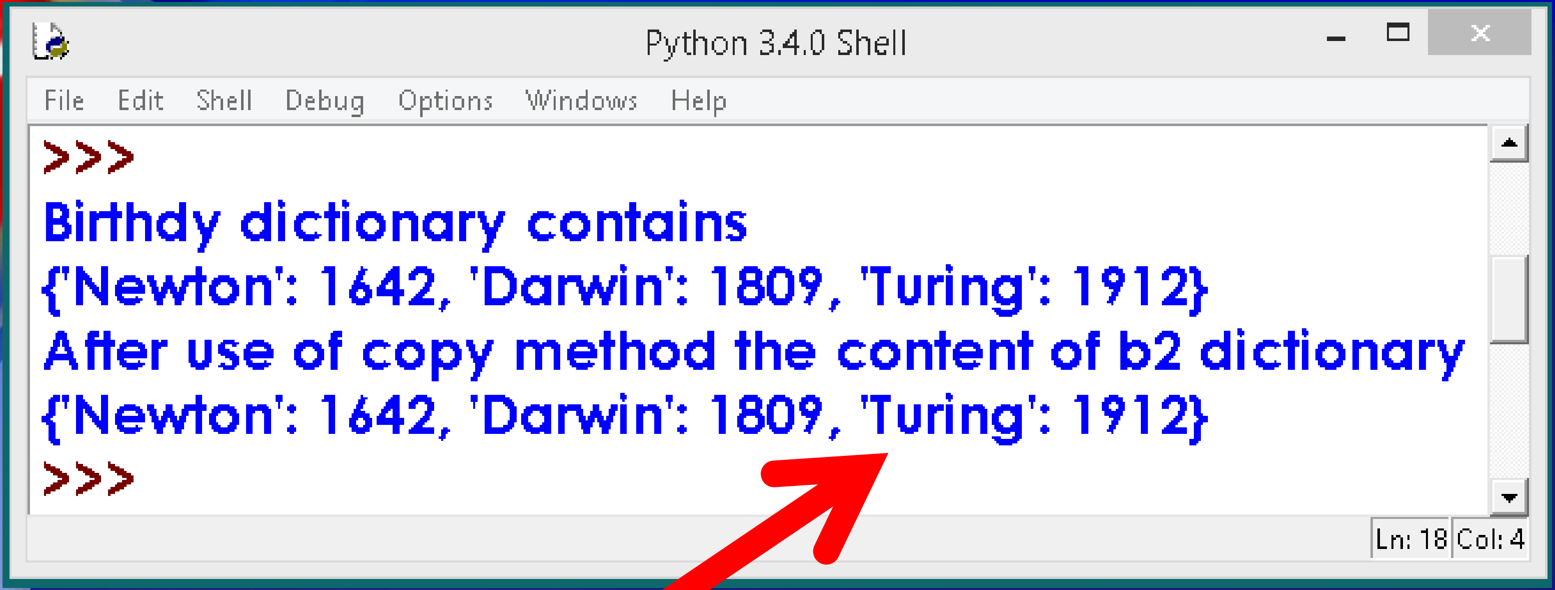
dict.copy() METHOD


Python 3.4.0: dict3.py - C:/Python34/dict3.py
File Edit Format Run Options Windows Help
copy method
Ln: 13 Col: 0

```
def create_dict():
    birthday={
        'Newton':1642,
        'Darwin':1809,
        'Turing':1912
        }
    print("Birthdy dictionary contains")
    print(birthday)
    b2=birthday.copy()
    print("After use of copy method the content of b2 dictionary")
    print(b2)
create_dict()
```

OUTPUT is in next slide!

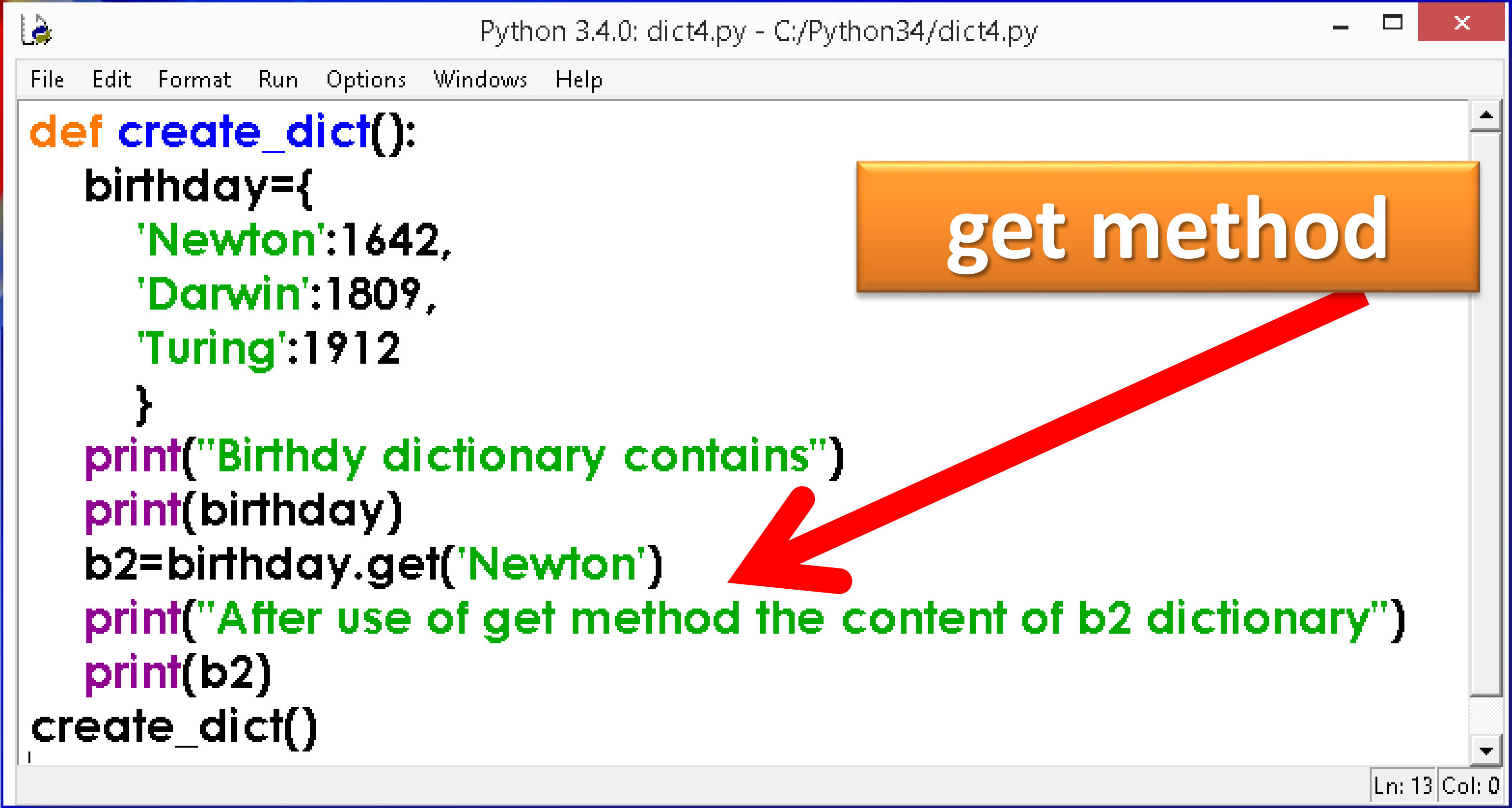
dict.copy() METHOD - OUTPUT



```
>>>
Birthdy dictionary contains
{'Newton': 1642, 'Darwin': 1809, 'Turing': 1912}
After use of copy method the content of b2 dictionary
{'Newton': 1642, 'Darwin': 1809, 'Turing': 1912}
>>>
```

copy method creates b2 dictionary

dict.get() METHOD

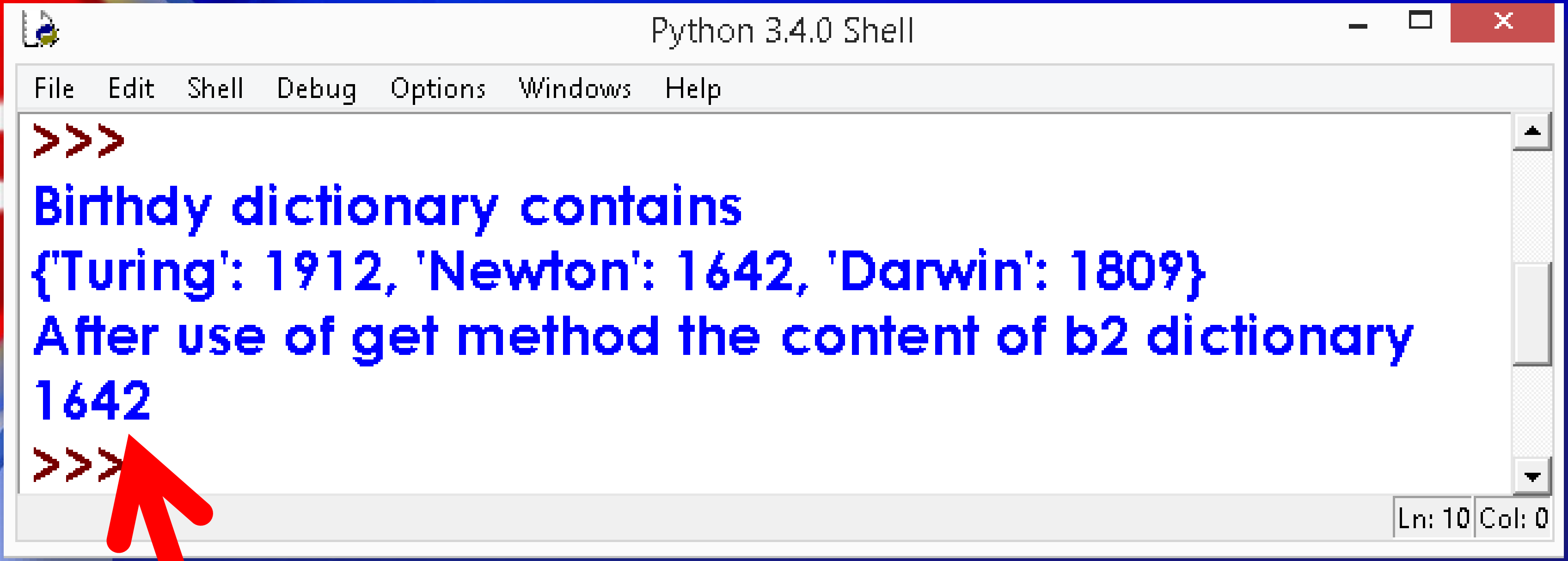

Python 3.4.0: dict4.py - C:/Python34/dict4.py
File Edit Format Run Options Windows Help
get method
Ln: 13 Col: 0

```
def create_dict():
    birthday={
        'Newton':1642,
        'Darwin':1809,
        'Turing':1912
        }
    print("Birthdy dictionary contains")
    print(birthday)
    b2=birthday.get('Newton')
    print("After use of get method the content of b2 dictionary")
    print(b2)
create_dict()
```

OUTPUT is in next slide!

fppt.com

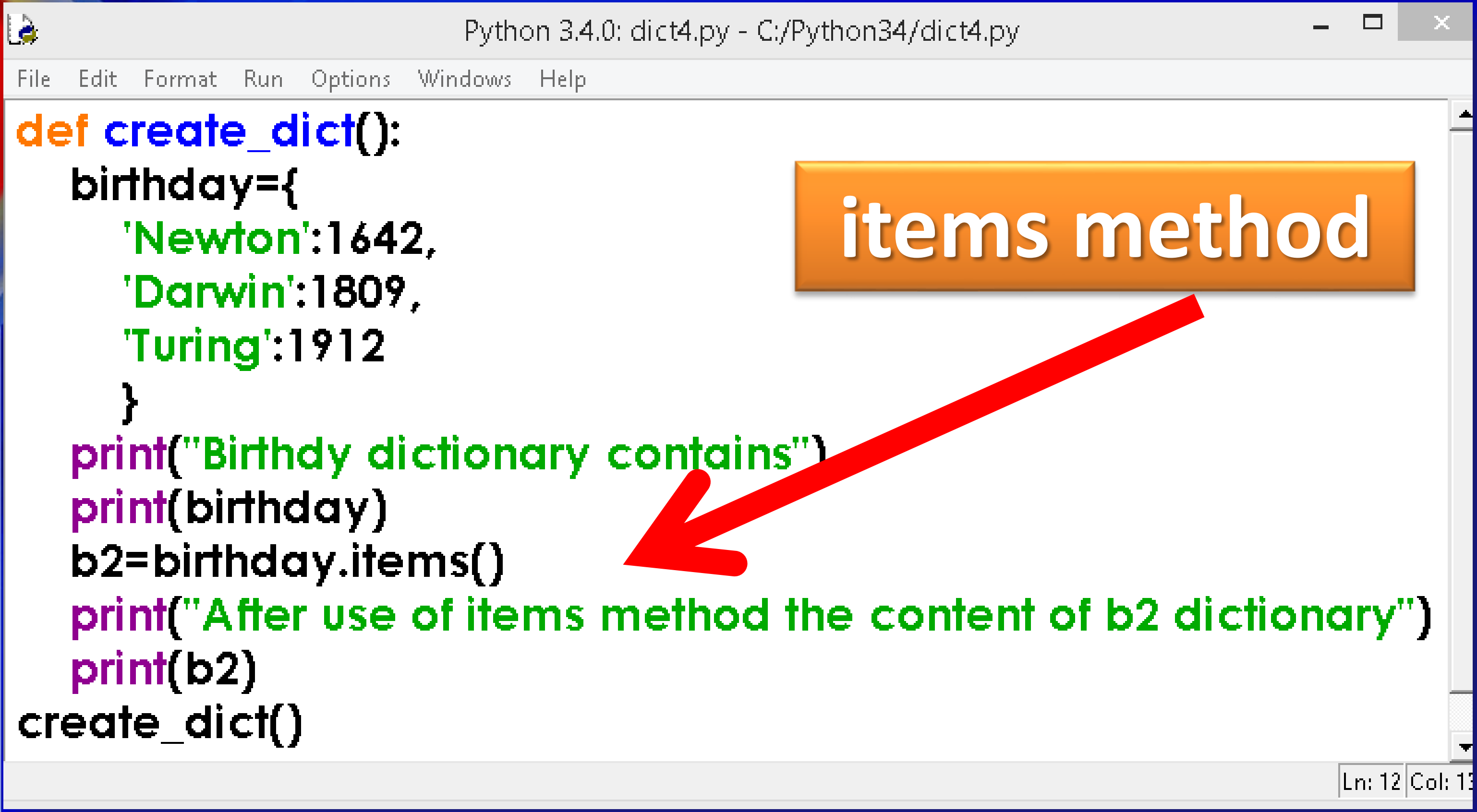
dict.get() METHOD - OUTPUT



```
>>>
Birthdy dictionary contains
{'Turing': 1912, 'Newton': 1642, 'Darwin': 1809}
After use of get method the content of b2 dictionary
1642
>>>
```

Creating a b2 dictionary using get method

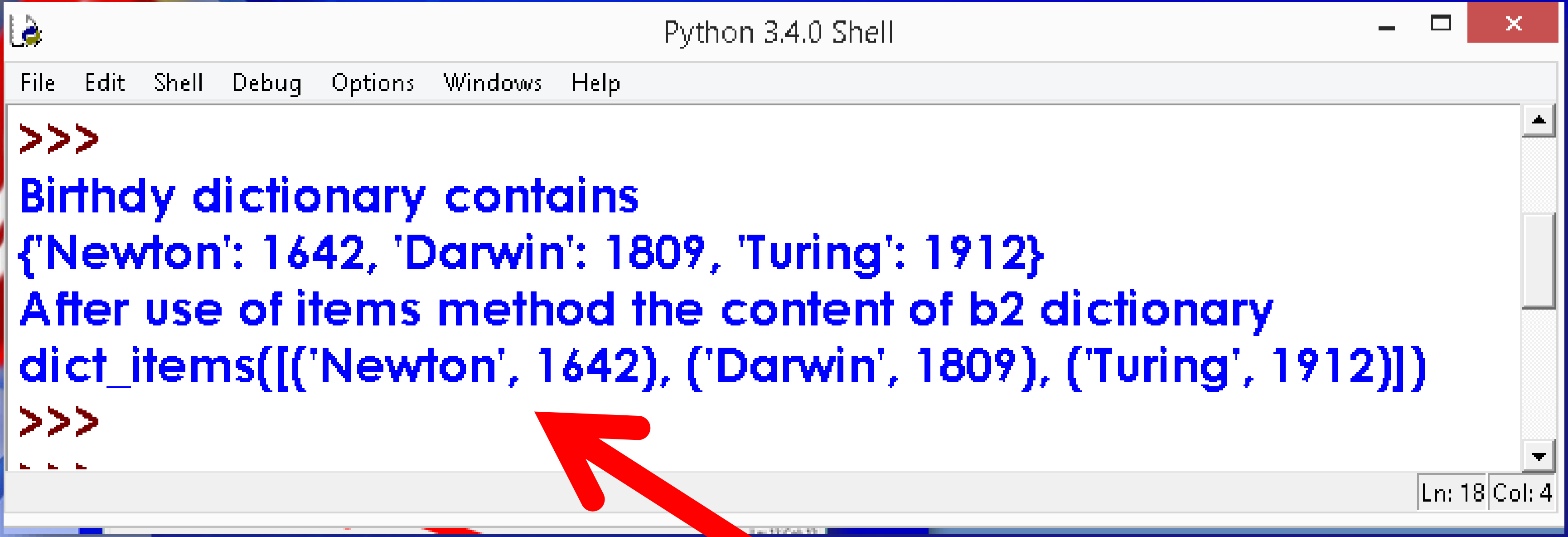
dict.items() METHOD


Python 3.4.0: dict4.py - C:/Python34/dict4.py
File Edit Format Run Options Windows Help
items method
Ln: 12 Col: 13

```
def create_dict():
    birthday={
        'Newton':1642,
        'Darwin':1809,
        'Turing':1912
        }
    print("Birthdy dictionary contains")
    print(birthday)
    b2=birthday.items()
    print("After use of items method the content of b2 dictionary")
    print(b2)
create_dict()
```

OUTPUT is in next slide!

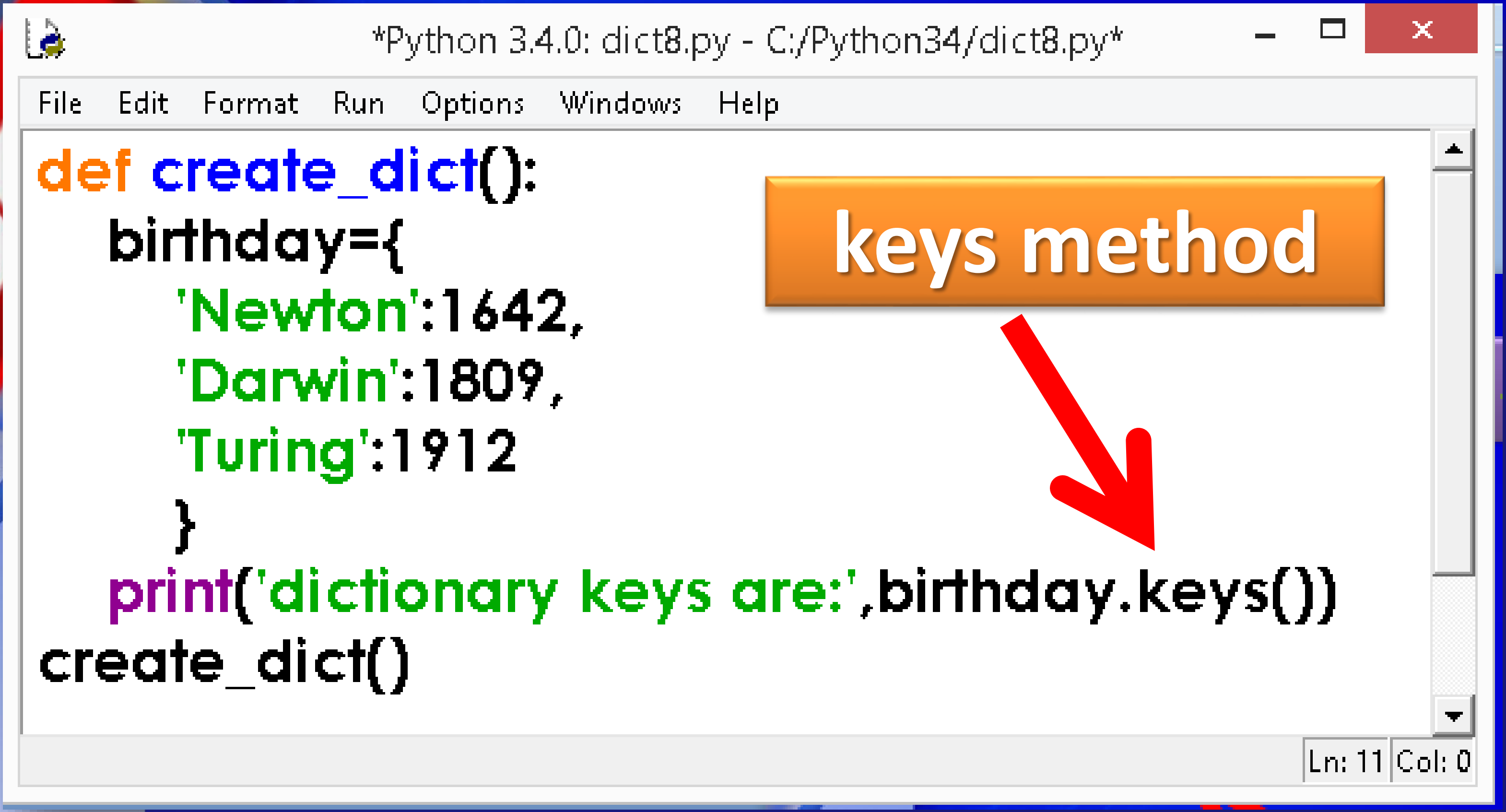
dict.items() METHOD - OUTPUT



```
>>>
Birthdy dictionary contains
{'Newton': 1642, 'Darwin': 1809, 'Turing': 1912}
After use of items method the content of b2 dictionary
dict_items([('Newton', 1642), ('Darwin', 1809), ('Turing', 1912)])
>>>
```

items method returns dictionary content

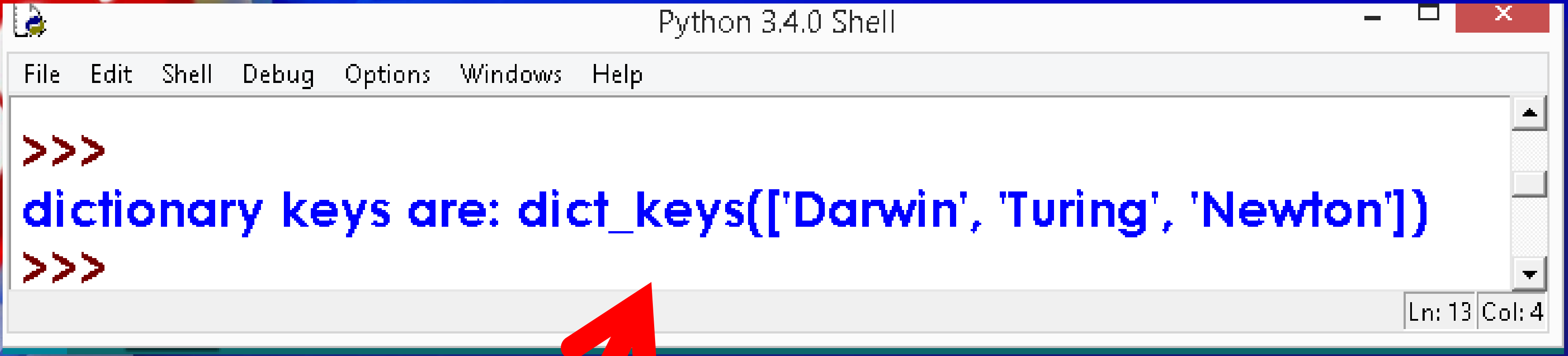
dict.keys() METHOD


Python 3.4.0: dict8.py - C:/Python34/dict8.py
File Edit Format Run Options Windows Help
keys method
Ln: 11 Col: 0

```
def create_dict():
    birthday={
        'Newton':1642,
        'Darwin':1809,
        'Turing':1912
        }
    print('dictionary keys are:',birthday.keys())
create_dict()
```

OUTPUT is in next slide!

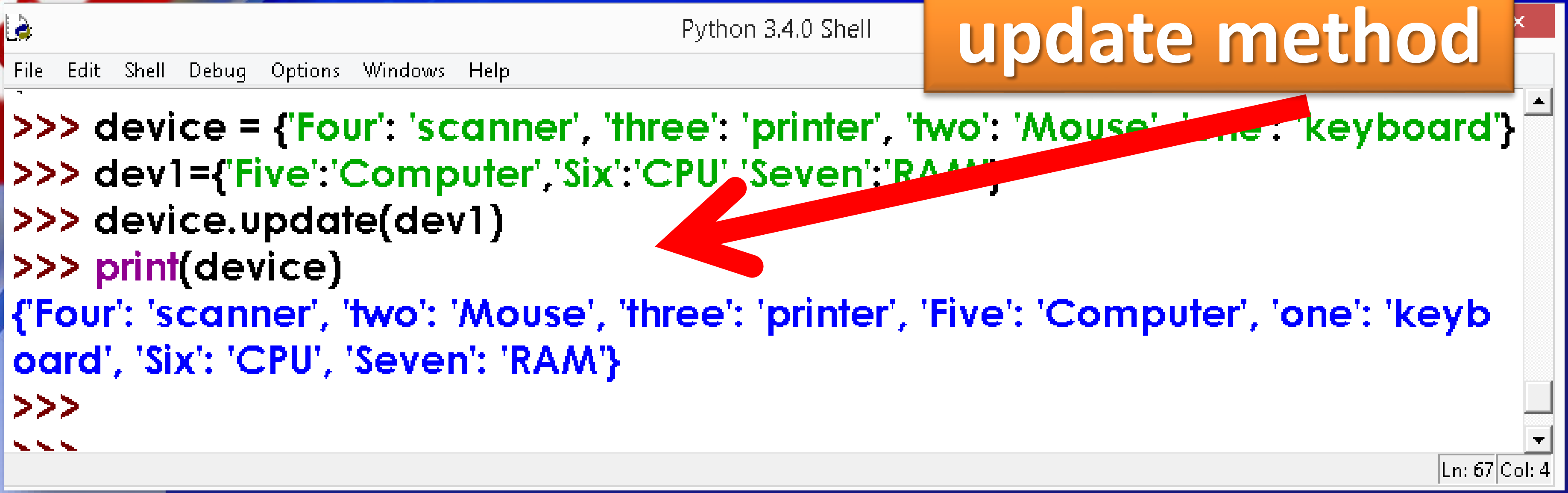
dict.keys() METHOD - OUTPUT



```
>>>
dictionary keys are: dict_keys(['Darwin', 'Turing', 'Newton'])
>>>
```

keys method returns dictionary keys

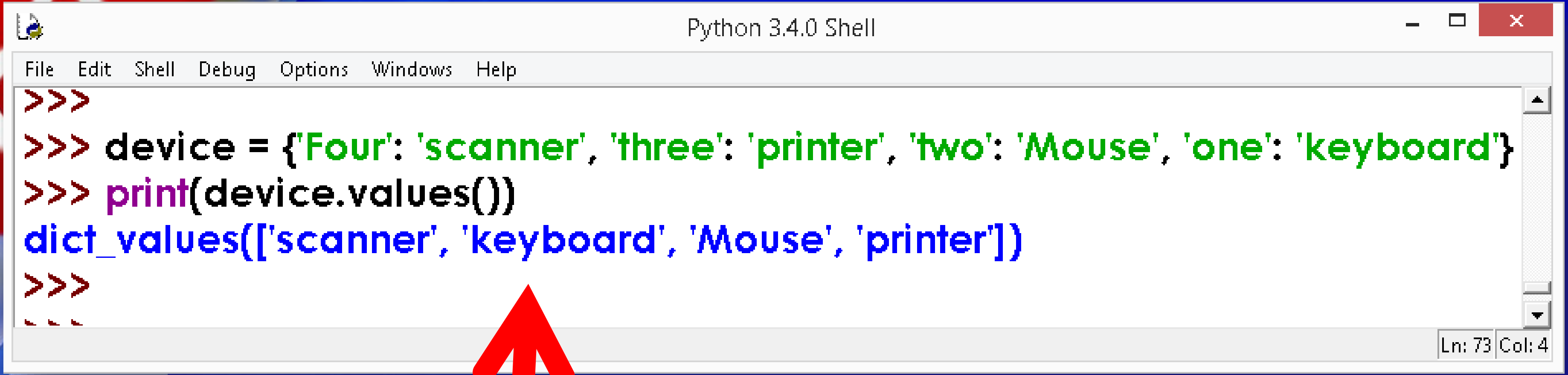
dict.update() METHOD



update method

```
>>> device = {'Four': 'scanner', 'three': 'printer', 'two': 'Mouse', 'one': 'keyboard'}
>>> dev1={'Five':'Computer','Six':'CPU','Seven':'RAM'}
>>> device.update(dev1)
>>> print(device)
{'Four': 'scanner', 'two': 'Mouse', 'three': 'printer', 'Five': 'Computer', 'one': 'keyb
oard', 'Six': 'CPU', 'Seven': 'RAM'}
>>>
```

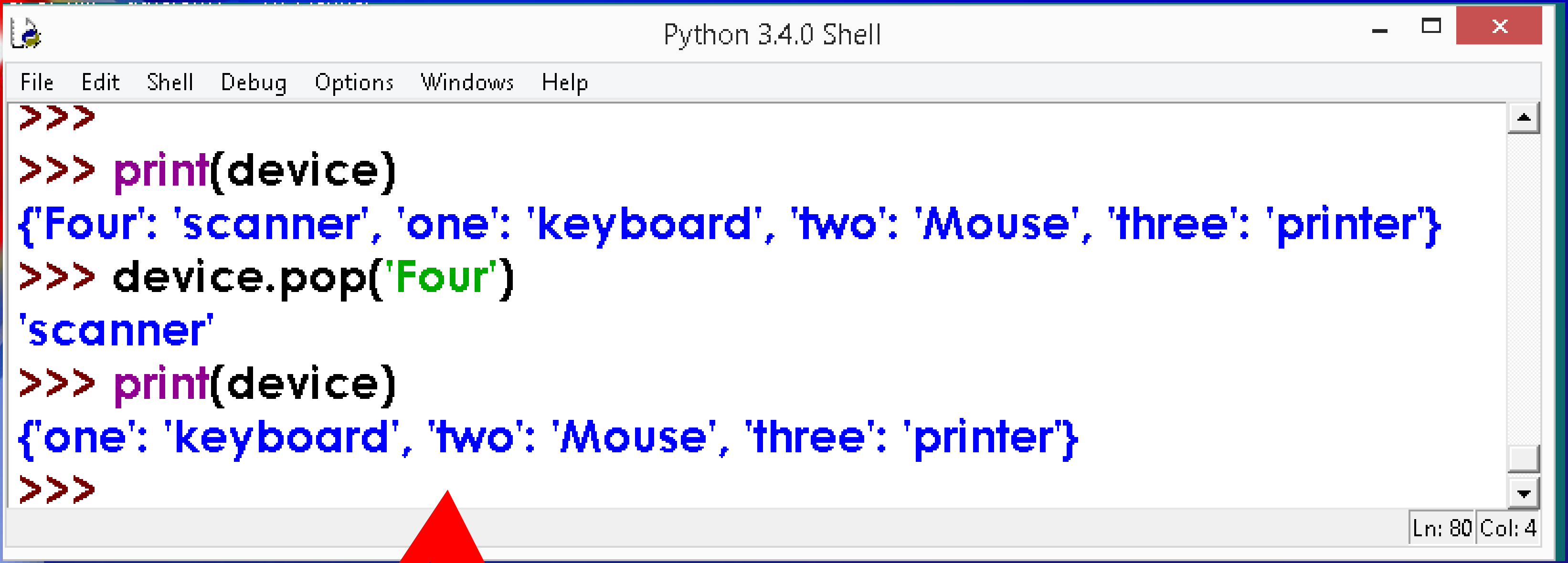
dict.values() METHOD



```
>>>
>>> device = {'Four': 'scanner', 'three': 'printer', 'two': 'Mouse', 'one': 'keyboard'}
>>> print(device.values())
dict_values(['scanner', 'keyboard', 'Mouse', 'printer'])
>>>
```

values method returns dictionary values

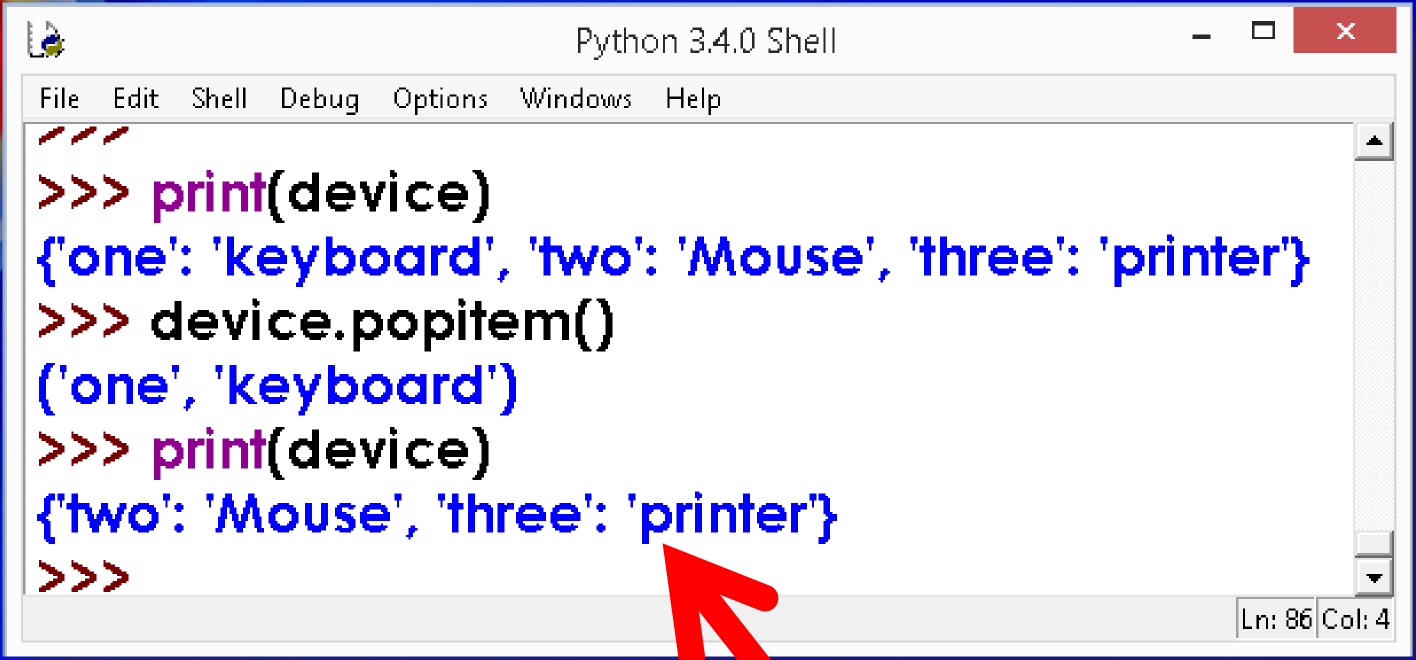
dict.pop() METHOD



```
>>>
>>> print(device)
{'Four': 'scanner', 'one': 'keyboard', 'two': 'Mouse', 'three': 'printer'}
>>> device.pop('Four')
'scanner'
>>> print(device)
{'one': 'keyboard', 'two': 'Mouse', 'three': 'printer'}
>>>
```

pop method removes specified key values from the dictionary

dict.popitem() METHOD



```
>>> print(device)
{'one': 'keyboard', 'two': 'Mouse', 'three': 'printer'}
>>> device.popitem()
('one', 'keyboard')
>>> print(device)
{'two': 'Mouse', 'three': 'printer'}
>>>
```

popitem method removes values/items from the dictionary

K.V.Sesha Sain [Kasyap]

Corporate Trainer| Technical Trainer|

Company Specific Trainer |Certified Skilled Trainer

M.sc Comp, OCP, JCP, MDSE

Oracle Certified Professional (2) |

Java Certified Professional

16+ Years of Experience

Email:brainsncrowns@gmail.com

Mobile: 6300340023