Python dictionary Method

Python has a set of built-in methods that you can use on dictionaries.

Method	Description
clear()	Removes all the elements from the dictionary
copy()	Returns a copy of the dictionary
fromkeys()	Returns a dictionary with the specified keys and value
get()	Returns the value of the specified key
items()	Returns a list containing a tuple for each key value pair
keys()	Returns a list containing the dictionary's keys
pop()	Removes the element with the specified key
popitem()	Removes the last inserted key-value pair

setdefault()	Returns the value of the specified key. If the key does not exist: insert the key, with the specified value
update()	Updates the dictionary with the specified key-value pairs
values()	Returns a list of all the values in the dictionary

get() method

The method **get()** returns a value for the given key. If key is not available then returns default value None.

Syntax

Following is the syntax for **get()** method −

```
dict.get(key, default=None)
```

Parameters

- **key** − This is the Key to be searched in the dictionary.
- **default** − This is the Value to be returned in case key does not exist.

Return Value

This method return a value for the given key. If key is not available, then returns default value None.

Example

The following example shows the usage of get() method.

```
#!/usr/bin/python3

dict = {'Name': 'Zara', 'Age': 27}

print ("Value : %s" %  dict.get('Age'))
print ("Value : %s" %  dict.get('Location', "NA"))
```

Result

When we run above program, it produces the following result −

```
Value : 27
Value : NA
```

items() Method

The `items()` method returns a view object. The view object contains the key-value pairs of the dictionary, as tuples in a list.

The view object will reflect any changes done to the dictionary, see example below.

Syntax

```
dictionary.items()
```

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}

x = car.items()

print(x)
```

Output: `dict_items([('brand', 'Ford'), ('model', 'Mustang'), ('year', 1964)])`

fromkeys() Method

The `fromkeys()` method returns a dictionary with the specified keys and the specified value.

Syntax

```
dict.fromkeys(keys, value)
```

Parameter Values

Parameter	Description
keys	Required. An iterable specifying the keys of the new dictionary
value	Optional. The value for all keys. Default value is None

Example

Same example as above, but without specifying the value:

```
x = ('key1', 'key2', 'key3')

thisdict = dict.fromkeys(x)

print(thisdict)
```

Output: `['key1': 0, 'key2': 0, 'key3': 0]`

setdefault() Method

The `setdefault()` method returns the value of the item with the specified key.

If the key does not exist, insert the key, with the specified value, see example below

Syntax

```
dictionary.setdefault(keyname, value)
```

Parameter Values

Parameter	Description
keyname	Required. The keyname of the item you want to return the value from
value	Optional. If the key exist, this parameter has no effect. If the key does not exist, this value becomes the key's value Default value None

Example

Get the value of the "color" item, if the "color" item does not exist, insert "color" with the value "white":

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}

x = car.setdefault("color", "white")

print(x)
```

Output: white