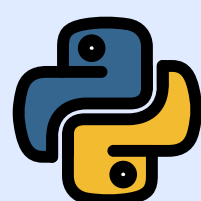


PYTHON

Dictionary Methods



@AbzAaron

CLEAR

Remove all elements from Dictionary

```
>> Office = {1 : "Dwight", 2 : "Pam"}
>> Office.clear()
>> Office
{}

```

COPY

Return copy of Dictionary

```
>> Office = {1 : "Dwight", 2 : "Pam"}
>> Office_two = Office.copy()
>> Office_two
{1 : "Dwight", 2 : "Pam"}

```

FROMKEYS

Return Dictionary with specified keys & values

Optional Parameter: Value. This is what's set for all keys.
Default is None

```
>> key = ("USA", "UK")
>> value = "Country"
>> countries = dict.fromkeys(key, value)
>> countries
{'UK': 'Country', 'USA': 'Country'}

```

GET

Return value of specified key

Optional Parameter: Value. Value returned if no key exists.
Default is None

```
>> Office = {1 : "Jim", 2 : "Pam"}
>> x = Office.get(2)
>> x
Pam

```

ITEMS

Return view object containing list of key value pairs

```
>> Office = {1 : "Jim", 2 : "Pam"}
>> x = Office.items()
>> x
dict_items([(1, 'Jim'), (2, 'Pam')])

```

KEYS

Return view object containing list of keys

```
>> Office = {1 : "Jim", 2 : "Pam"}
>> x = Office.keys()
>> x
dict_keys([1, 2])

```

POP

Remove element with specified key and return it

Optional Parameter: Defaultvalue. This is value to return if no key is found in dictionary

```
>> Office = {1 : "Jim", 2 : "Pam"}
>> x = Office.pop(1)
>> Office
{2 : "Pam"}
>> x
Jim

```

POPITEM

Remove last inserted key value pair and return as tuple

Python < 3.7 method returns random item

```
>> Office = {1 : "Jim", 2 : "Pam"}
>> x = Office.popitem()
>> Office
{1 : "Jim"}
>> x
(2, 'Pam')

```

SETDEFAULT

Return value of item with specified key. If key doesn't exist, insert with specified value

Optional Parameter: value. Default is None

```
>> office = {1 : "Jim", 2 : "Pam"}
>> x = office.setdefault(5, "Dwight")
>> x
Dwight
>> office
{1: 'Jim', 2: 'Pam', 5: 'Dwight'}

```

UPDATE

Update dictionary with specified key value pairs

```
>> Office = {1 : "Jim", 2 : "Pam"}
>> Office.update({3 : "Dwight", 4 : "Andy"})
>> Office
{1: 'Jim', 2: 'Pam', 3: 'Dwight', 4: 'Andy'}

```

VALUES

Return view objects containing list of values

```
>> Office = {1 : "Jim", 2 : "Pam"}
>> x = Office.values()
>> x
dict_values(['Jim', 'Pam'])

```