Python Dictionary

Dictionary

1. A dictionary is **mutable** and unordered collection of items each item is the pair of keys and values.
2. Dictionaries consist of **pairs** (called items) of **keys** and their corresponding **values**.
3. Think of the **key as a Index in ordered Collections.**
4. Python dictionaries are also known as associative arrays or hash tables.
5. dict = {'Car': ['Nano', 'Alto'] , 'Color': 'White', 'Speed': 100}

Dict…

- Each **key** is separated from its value by a colon(**:**), the items are separated by commas(,) and the whole thing is enclosed in **curly braces**.
- An **empty dictionary** without any items is written with just two curly braces, like this: **{}**.
- **Keys** are **unique** within a dictionary while values may not be. The **values** of a dictionary can be of **any type**, but the **keys** must be of an **immutable** data type such as **strings, numbers, or tuples**.

Examples

```
# dictionary creation empty
d = {}
print(d, type(d))  # {} <class 'dict'>

d = dict()  # dict constructor
print(d, type(d))  # {} <class 'dict'>


# dictionary operation

dct = { 'rolln':30, 'cpi':8.0, 'section':'A', 'name':'akshay'}
print(dct, type(dct))
# accessing the element
d = dct['name']  # element at given key
print(d) # 'akshay'

# keys() : get all the keys of dict
dct = {'name':'Govind', 'rolln':30, 'cpi':7.0, 'section':'E'}
out1 = dct.keys()
print(out1)  # dict_keys(['name', 'rolln', 'cpi', 'section'])

out2 = dct.values()
print(out2, type(out2))  # dict_values(['Govind', 30, 7.0, 'E']) <class 'dict_values'>
```

Keys – must be unique

- Keys must be **unique**
- One key accesses **one value**.
- Keys are unique within a dictionary while values may not be
- The keys() method returns a view object that displays a list of all the keys in the dictionary and similarly values() does for all values()
- Which are possible keys(All immutable objects)
  - Student ID number,
  - date of birth,
  - email address,
  - car registration,
  - full name

Accessing Values in Dictionary

1. To access dictionary elements, you can use the familiar square brackets.
2.
```
dict = {'Name': 'Era', 'Age': 5, 'Class': 'First'}
print ("dict['Name']: ", dict['Name'])
print ("dict['Age']: ", dict['Age'])
```

it produces **the following result**:

```
dict['Name']: Era
dict['Age']: 5
```

❑ For all values()

```
print(dict.values())
```

Result: dict_values(['Era', 5, 'First'])

Updating Dictionary

You can **update** a dictionary by **adding** a new entry or item (i.e., a key-value pair), **modifying** an existing entry, or **deleting** an existing entry as shown below in the simple example:

```
    dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'};
    dict['Age'] = 8; # update existing entry
    dict['School'] = "DPS School"; # Add new entry
    print ("dict['Age']: ", dict['Age'])
    print ("dict['School']: ", dict['School'])
```

it produces **the following result**:

```
    dict['Age']: 8
    dict['School']: DPS School
```

Delete Dictionary Elements:

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'};

deletion statement:

- del dict['Name']; # remove entry with key 'Name'

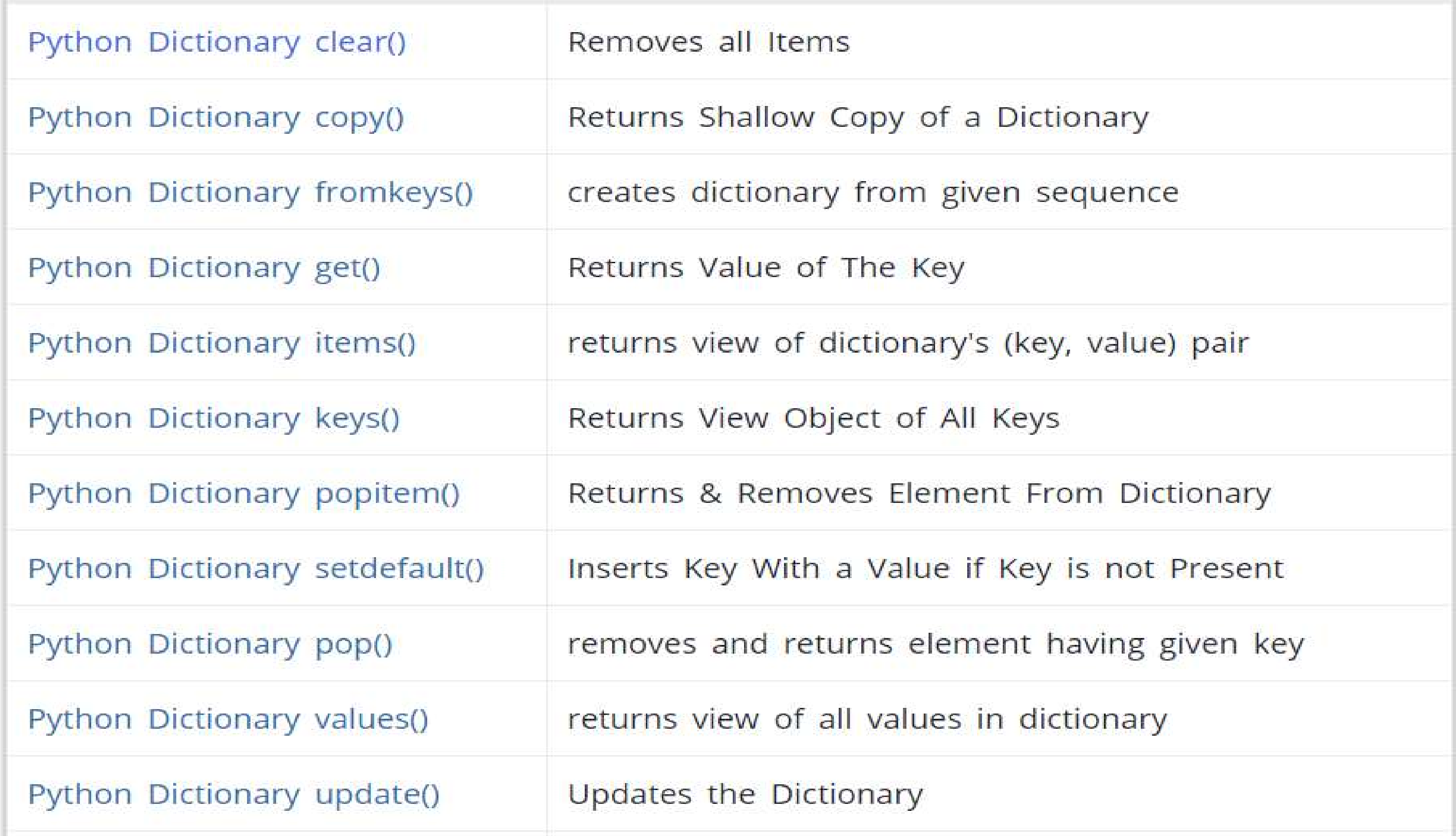
dictionary method:

- dict.pop('Name') # remove the key from dictionary and return the value
- dict.popitems() # It removes the arbitrary key-value pair from the dictionary and returns it as tuple.
- dict.clear(); # remove all entries in dict
- del dict ; # delete entire dictionary

Properties of Dictionary:

- Dictionary **values** have no restrictions.
- They can be any **arbitrary Python object**, either standard objects or user-defined objects.
- However, same is not true for the keys.
- Keys are unique (**no duplicate key is allowed**)
- Keys should be immutable type objects
- The collections is unordered of items
- Generally dictionary is the a list of (key, value) tuples of the dictionary. We can use it iterate/loop over the dictionary.

Methods



Python Dictionary clear()	Removes all Items
Python Dictionary copy()	Returns Shallow Copy of a Dictionary
Python Dictionary fromkeys()	creates dictionary from given sequence
Python Dictionary get()	Returns Value of The Key
Python Dictionary items()	returns view of dictionary's (key, value) pair
Python Dictionary keys()	Returns View Object of All Keys
Python Dictionary popitem()	Returns & Removes Element From Dictionary
Python Dictionary setdefault()	Inserts Key With a Value if Key is not Present
Python Dictionary pop()	removes and returns element having given key
Python Dictionary values()	returns view of all values in dictionary
Python Dictionary update()	Updates the Dictionary

for examples go to this link
https://github.com/amirkhan1092/python38-GLA/blob/master/dict_example.py

Practice Questions

1. Write a Python script to merge two Python dictionaries.
2. Write a Python program to iterate over dictionaries using for loops.
3. Write a Python program to sum all the items in a dictionary.
4. Write a Python program to multiply all the items in a dictionary.
5. Write a Python program to remove a key from a dictionary.
6. Write a Python program to map two lists into a dictionary.
7. Write a Python program to sort a dictionary by key.
8. Write a Python program to get the maximum and minimum value in a dictionary.
9. Write a Python program to get a dictionary from an object's fields.
10. Write a Python program to remove duplicates from Dictionary.
11. Write a Python program to check a dictionary is empty or not.