Python Dictionary

Dictionary is an ordered set of a key-value pair of items. They work like associative arrays or hashes and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).

Example	Output

d = {1:'abc', 2:'xyz', 3:'pqr'};
print(d[1]);
print(d[3]);
print (d);
print (d.keys());
print (d.values());
	abc
	pqr
	{1: 'abc', 2: 'xyz', 3: 'pqr'}
	[1, 2, 3]
	['abc', 'xyz', 'pqr']

Accessing Dictionary Values

Dictionaries values can be accessed by using the square braces, i.e. [ ] along with the key name to obtain the value. This is multiple item declaration procedures used to declare keys along with their values in Python's dictionary.

Example	Output

d = {1:'abc', 2:'xyz', 3:'pqr'};
print(d[1]);
print(d[3]);
print (d);
	abc
	pqr
	{1: 'abc', 2: 'xyz', 3: 'pqr'}

Update Dictionary

Programmers can update or modify the existing dictionary by simply adding a new entry or a key-value pair or by deleting an item or entry.

Example	Output

d = {1:'abc', 2:'xyz', 3:'pqr'};
#update dictionary
d[1]='hi'
print (d)
 	
#add element into dictionary
d['name'] = 'jack'
d[4] = 'how'
print(d)
	{1: 'hi', 2: 'xyz', 3: 'pqr'}
	{1: 'hi', 2: 'xyz', 3: 'pqr', 4: 'how', 'name': 'jack'}

Deleting Python Dictionary Elements

del statement is used for performing deletion operation. An item can be deleted from a dictionary using the key only.

Example	Output
d = {1:'abc', 2:'xyz', 3:'pqr'}; del d[1] print d #delete full dictionary del d print d #print error dictionary is deleted	{2: 'xyz', 3: 'pqr'}

len(dict) Method

Gives the total length of the dictionary. This would be equal to the number of items in the dictionary.

Example	Output
d = {1:'abc', 2:'xyz', 3:'pqr'}; print(len(d))	3

cmp(dict1, dict2) Method

Compares elements of both dictionary.

Example	Output
d = {1:'abc', 2:'xyz', 3:'pqr'}; d1 = {'name':'alice' , 'age':20, 'country':'US'} print(cmp(d,d1)) print(cmp(d1,d))	-1 1

str(dict) Method

This method returns string formation of the value.

Example	Output
d = {1:'abc', 2:'xyz', 3:'pqr'}; print(str(d))	{1: 'abc', 2: 'xyz', 3: 'pqr'}

clear()Method

Removes all the elements from the dictionary

Example	Output
d = {1:'abc', 2:'xyz', 3:'pqr'}; d.clear() print d	{}

copy()Method

Returns a copy of the dictionary

Example	Output
d = {1:'abc', 2:'xyz', 3:'pqr'}; d1=d.copy() print d1	{1: 'abc', 2: 'xyz', 3: 'pqr'}

get()Method

Returns the value of the specified key

Example	Output
d = {1:'abc', 2:'xyz', 3:'pqr'}; print(d.get(1))	abc

items()Method

Returns a list containing the a tuple for each key value pair

Example	Output
d = {1:'abc', 2:'xyz', 3:'pqr'}; print(d.items())	[(1, 'abc'), (2, 'xyz'), (3, 'pqr')]

keys()Method

Returns a list contianing the dictionary's keys

Example	Output
d = {1:'abc', 2:'xyz', 3:'pqr'}; print(d.keys())	[1, 2, 3]

pop()Method

Removes the element with the specified key

Example	Output
d = {1:'abc', 2:'xyz', 3:'pqr'}; print(d.pop(1)) print d	abc {2: 'xyz', 3: 'pqr'}

popitem()Method

Removes the last inserted key-value pair

Example	Output
d = {1:'abc', 2:'xyz', 3:'pqr'}; print(d.popitem()) print d	(1, 'abc') {2: 'xyz', 3: 'pqr'}

update()Method

Updates the dictionary with the specified key-value pairs

Example	Output
d = {1:'abc', 2:'xyz', 3:'pqr'}; d.update({4:'hi'}) print d d.update({2:'how'}) print d	{1: 'abc', 2: 'xyz', 3: 'pqr', 4: 'hi'} {1: 'abc', 2: 'how', 3: 'pqr', 4: 'hi'}

values() Method

Returns a list of all the values in the dictionary

Example	Output
d = {1:'abc', 2:'xyz', 3:'pqr'}; print(d.values())	['abc', 'xyz', 'pqr']