PYTHON PROGRAMMING

MODULE 4 – PART 4

DICTIONARY AND ASSOCIATED OPERATIONS

OUTLINE

- Dictionary
- Create and Assign
- Access Elements
- Change/Add Elements
- Delete/Remove Elements
- Basic Operations
- Built-in Functions and Methods

Dictionary

- Mutable
- Unordered Key-Value pairs
- Indexed according to keys

Keys and Values

A dictionary is a set of key-value pairs enclosed in curly braces {}, where each key is separated from value using a colon (:) and each pair is separated from each other by comma (,)

Keys must be unique, immutable type (integer, string, tuple etc.) and are case sensitive.

Values may not be unique and hence, can be repeated.

Creating Dictionaries

```
#empty dictionary
d={}

#can have different types
d1={1:2,3:4,'a':5,'b':6}

#keys are immutable, list cannot be
used as a key
d2={('a','b'):[1,2] }

#nested dictionary
d3={1:{2:3,4:5},2:6}
print(d1,"\n",d2,"\n",d3)
```

OUTPUT

```
{1: 2, 3: 4, 'a': 5, 'b': 6}
{('a', 'b'): [1, 2]}
{1: {2: 3, 4: 5}, 2: 6}
```

Creating dictionaries using dict() function

```
d4=dict({1:2,3:4})

'''dict() takes only one argument, hence
pass key-val pairs inside list or tuple'''
d5=dict([(1,2),(3,4)])

print(d4,"\n",d5)
```

OUTPUT

```
{1: 2, 3: 4}
{1: 2, 3: 4}
```

Accessing dictionary elements

Keys are used as an index to access values. Slicing operation is not supported in dictionary.

```
d1={1:2,3:4,'a':5,'b':6}
d2={(1,2):'numbers',('a','b'):'alphabets'}
d3={1:{2:3,4:5},2:6}
print(d1['a'],"\n",d2[(1,2)])
print(d3[1],"\n",d3[2],"\n",d3[1][2])

#using get() function
print(d1.get(1),"\n",d1.get('a'))
```

OUTPUT

```
5
numbers
{2: 3, 4: 5}
6
3
2
5
```

Change or add elements

If the key is already present its value gets updated otherwise a new key-value pair is added.

```
d1={1:2,3:4,'a':5,'b':6}
d1[1]=10
d1[2]=20
print(d1)
```

OUTPUT

```
{1: 10, 3: 4, 'a': 5, 'b': 6, 2: 20}
```

Delete or Remove elements

```
d1={1:2,3:4,'a':5,'b':6}
d2={1:2,3:4,5:6}

#deletes element with given key & stores the value
print(d1.pop(1))

#deletes an arbitrary item & stores the value
print(d1.popitem())

#deletes element with given key & cannot store value
del d1[3]
print("Dictionary after deleting 3:4 is --- ",d1)

#deletes entire dictionary
del d1

#deletes entire dictionary
d2.clear()
```

OUTPUT

```
2
('b', 6)
Dictionary after deleting 3:4 is ---  {'a': 5}
```

Basic operations (+ and * cannot be applied)

```
d1={1:2,3:4,'a':5,'b':6}

#returns length
print(len(d1))

#membership operators are for keys only
print(1 in d1,"\n",2 in d1)

for i in d1:
    # i prints key and d1[i] prints value
    print(i,":",d1[i])
```

OUTPUT

```
4
True
False
1 : 2
3 : 4
a : 5
b : 6
```

Functions Discussed Earlier

1) dict(key_value_pairs)
2) get(key)
3) pop(key)
4) popitem( )
5) clear( )
6) len(dictionary)

Functions Discussed Next

1) any(dictionary)
2) all(dictionary)
3) sorted(list, reverse=True/False, key=myFunc)
4) keys()
5) values()
6) items()
7) copy()
8) fromkeys(dictionary, value_for_all_keys)
9) update(key_value_pair)
10) setdefault(key, value)

Built in functions and methods

```
d={'':'',2:3}

#True if even one key in dictionary has Boolean value of True
print(any(d))

#True only if all keys in dictionary have Boolean value of True
print(all(d))

d1={2:1,3:1,1:1,4:1}
#sorts keys in ascending order & returns as list
print(sorted(d1))
#sorted() doesn't modify original dictionary
print(d1)
```

OUTPUT

```
True
False
[1, 2, 3, 4]
{2: 1, 3: 1, 1: 1, 4: 1}
```

```
d1={1:2,3:4,'a':5,'b':6}

#returns keys as list
print(d1.keys())

#returns values as list
print(d1.values())

#returns tuple of key-value pairs
print(d1.items())

#copies d1 to d
d=d1.copy()
print(d)
d1[1]=3
'''updates not reflected in
copied dictionary'''
print(d1,d)
```

OUTPUT

```
dict_keys([1, 3, 'a', 'b'])
dict_values([2, 4, 5, 6])
dict_items([(1, 2), (3, 4), ('a', 5), ('b', 6)])
{1: 2, 3: 4, 'a': 5, 'b': 6}
{1: 3, 3: 4, 'a': 5, 'b': 6} {1: 2, 3: 4, 'a': 5, 'b': 6}
```

Continued...

```
'''creates new dictionary from another dictionary,
2nd argument is corresponding values for keys, default is None '''
newdict=d1.fromkeys(d1,0)
print(newdict)
newdict=d1.fromkeys(d1)
print(newdict)

#updates dictionary with given pair
d1.update({'c':7})
print(d1)

#returns original value if key is present
print(d1.setdefault('a',10))
'''returns given value for new key and
updates original dictionary with given pair '‘’
print(d1.setdefault('abc',10))
print(d1)
```

```
{1: 0, 3: 0, 'a': 0, 'b': 0}
{1: None, 3: None, 'a': None, 'b': None}
{1: 3, 3: 4, 'a': 5, 'b': 6, 'c': 7}
5
10
{1: 3, 3: 4, 'a': 5, 'b': 6, 'c': 7, 'abc': 10}
```

Study at Home

Notes

Learn topics better with a little less effort utilizing notes by subject experts.

Syllabus

View or download syllabus for Bsc Computer Science under Calicut University.

Video Lessons

Better learning experience through video lectures from skilled personnel.

Question Papers

Familiarize with exam patterns through our library of previous year question papers.

Quiz

Test your knowledge of topics through various quizzes.

Solved Questions

Vast collection of questions and answers related to each topic.

Programming Laboratory

Practicals made easy with complete list of tested programs.

Software Downloads

Practice programming by downloading and installing respective softwares.

Teaching Resources

Find PPTs and related resources for teaching aid here.

teachics.

The Computer Science Learning Platform.

Thank You