



Dictionaries

Presented By
M.Malarmathi
AP/IT



A dictionary is like a list

- dictionary is a mapping between a set of indices (which are called **keys**) and a set of values.
- Each key maps to a value. The association of a key and a value is called a **key-value pair or sometimes an item**
- In a list, the indices have to be integers; in a dictionary they can be (almost) any type
- The function **dict** creates a new dictionary with no items
- dict is the name of a built-in function, avoid using as a variable name

```
>>> eng2sp = dict()
>>> print eng2sp
{}-----> empty dictionary
```



- Ex1:

```
>>> eng2sp['one'] = 'uno'
```

```
>>> print eng2sp
```

```
{'one': 'uno'}
```

- The output format is also an input format

- Ex2:

```
>>> eng2sp = {'one': 'uno', 'two': 'dos', 'three': 'tres'}
```

```
>>> print eng2sp
```

```
{'one': 'uno', 'three': 'tres', 'two': 'dos'}
```



Accessing Values in Dictionary



- To access dictionary elements, use square bracket

```
dict = {'Name': 'python', 'Age': 7, 'Class': 'First'}  
print "dict['Name']: ", dict['Name']  
print "dict['Age']: ", dict['Age']
```

- Output

```
dict['Name']: python  
dict['Age']: 7
```



Updating Dictionary



- update a dictionary by adding a new entry or a key-value pair

```
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']
```

OUTPUT:

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



Delete Dictionary Elements



- removes individual dictionary elements or clears the entire contents of a dictionary

```
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}  
del dict['Name']; # remove entry with key 'Name'  
dict.clear(); # remove all entries in dict  
del dict ; # delete entire dictionary  
print "dict['Age']: ", dict['Age']  
print "dict['School']: ", dict['School']
```

- OUTPUT**

```
dict['Age']: Traceback (most recent call last):  
File "test.py", line 8, in <module>  
print "dict['Age']: ",  
dict['Age'];  
TypeError: 'type' object is unsubscriptable
```








Built-in Dictionary Functions



Sr.No.	Function with Description
1	cmp(dict1, dict2) Compares elements of both dict.
2	len(dict) Gives the total length of the dictionary. This would be equal to the number of items in the dictionary.
3	str(dict) Produces a printable string representation of a dictionary
4	type(variable) Returns the type of the passed variable. If passed variable is dictionary, then it would return a dictionary type.



Dictionary Methods

Sr.No.	Methods with Description
1	<code>dict.clear()</code>  Removes all elements of dictionary <i>dict</i>
2	<code>dict.copy()</code>  Returns a shallow copy of dictionary <i>dict</i>
3	<code>dict.fromkeys()</code>  Create a new dictionary with keys from <i>seq</i> and values set to <i>value</i> .
4	<code>dict.get(key, default=None)</code>  For key <i>key</i> , returns value or default if key not in dictionary
5	<code>dict.has_key(key)</code>  Returns <i>true</i> if key in dictionary <i>dict</i> , <i>false</i> otherwise



6	dict.items() Returns a list of <i>dict</i> 's (key, value) tuple pairs
7	dict.keys() Returns list of dictionary <i>dict</i> 's keys
8	dict.setdefault(key, default=None) Similar to <code>get()</code> , but will set <code>dict[key]=default</code> if <i>key</i> is not already in <i>dict</i>
9	dict.update(dict2) Adds dictionary <i>dict2</i> 's key-values pairs to <i>dict</i>
10	dict.values() Returns list of dictionary <i>dict</i> 's values



`dict.clear()`

```
dict = {'Name': 'Zara', 'Age': 7};  
print "Start Len : %d" , len(dict)  
dict.clear()  
print "End Len : %d" , len(dict)
```

o/p

Start Len : 2

End Len : 0

- `dict.copy()`

```
dict1 = {'Name': 'Zara', 'Age': 7};  
dict2 = dict1.copy()  
print "New Dictionary : %s" (dict2)
```

O/p

Dictionary : {'Name': 'Zara', 'Age': 7}





- **fromkeys()**
- Parameters

seq – list of values which would be used for dictionary keys preparation.

value – optional, if provided then value would be set to this value

```
dict.fromkeys(seq[, value])  
seq = ('name', 'age', 'sex')  
dict = dict.fromkeys(seq)  
print " Dictionary : %s" % str(dict)  
dict = dict.fromkeys(seq, 10)  
print "New Dictionary : %s" % str(dict)
```

- O/p
Dictionary : {'age': None, 'name': None, 'sex': None}
New Dictionary : {'age': 10, 'name': 10, 'sex': 10}



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

```
dict = {'Name': 'Zabra', 'Age': 7}  
print "Value : %s" % dict.get('Age')  
print "Value : %s" % dict.get('Education', "Never")
```

- O/p
Value : 7
Value : Never



- **has_key()**

returns true if a given *key* is available in the dictionary, otherwise it returns a false

```
dict = {'Name': 'Zara', 'Age': 7}  
print "Value : %s" % dict.has_key('Age')  
print "Value : %s" % dict.has_key('Sex')
```

- O/p

Value : True

Value : False



• items()

returns a list of tuple pairs

```
dict = {'Name': 'Zara', 'Age': 7}
```

```
print "Value : %s" % dict.items()
```

• O/p

```
Value : [('Age', 7), ('Name', 'Zara')]
```





- **dict.keys()**

returns a list of all the available keys in the dictionary

```
dict = {'Name': 'Zara', 'Age': 7}
print "Value : %s" % dict.keys()
```

- **O/p**

Value : ['Age', 'Name']

- **setdefault()**

```
dict = {'Name': 'Zara', 'Age': 7}
print "Value : %s" % dict.setdefault('Age', None)
print "Value : %s" % dict.setdefault('Sex', None)
```

- **O/p**

Value : 7

Value : None



- **update()**
- adds dictionary dict2's key-values pairs in to dict

```
dict = {'Name': 'Zara', 'Age': 7}
```

```
dict2 = {'Sex': 'female' }
```

```
dict.update(dict2)
```

```
print "Value : %s" % dict
```

- O/p
Value : {'Age': 7, 'Name': 'Zara', 'Sex': 'female'}



- **values()**

returns a list of all the values available in a given dictionary.

```
dict = {'Name': 'Zara', 'Age': 7}  
print "Value : %s" % dict.values()
```

- O/p

```
Value : [7, 'Zara']
```



Thank You