

Python Dictionary

What is Python Dictionary?

- A **data** structure that allows retrieval by **keyword**
- Table of **key-value** pairs
- Commonly called *Associative array* (in general e.g. JavaScript, C++)
- Implemented using hashtable

Why the need to learn about dictionary?

Python Dictionary

Python dictionaries use the curly braces { } :

{**key1**:**value1**, **key2**:**value2**, ...}

e.g. { '**Name**' : '**James**' , '**Age**' : **18** , '**Gender**' : '**M**' }

Key:	Value
'Name'	'James'
'Age'	18
'Gender'	'M'

**Unique,
Immutable**

**One value per key,
Any data type**

Create Python Dictionary

```
>>> menu = { 'soup': [ ('mushroom', 4), ('tomato', 3) ],  
'mains': ('chicken', 10), 'dessert': { 'weekdays': 0,  
    'weekends': ('lava cake', 7) } }
```

Key:	Value
'soup'	[('mushroom', 4), ('tomato', 3)]
'mains'	('chicken', 10)
'dessert'	{ 'weekdays': 0, 'weekends': ('lava cake', 7) }

Create Python Dictionary

```
>>> {}    #empty dict using curly braces
```

```
{}
```

```
>>> dict()    #empty dict using built-in function
```

```
{}
```

Create Python Dictionary

Given a sequence (e.g. tuple, list) that contains **key-value** pair tuples:

```
>>> t = [ ('boys', 11), ('girls', 13)] #list  
        containing tuples of key-value pairs)
```

```
>>> s = [ ('soup', [ ('mushroom', 4), ('tomato', 3)]),  
          ('mains', ('chicken', 10)), ('dessert', { 'weekdays': 0  
          , 'weekends': ('lava cake', 7) }) ]
```

Create Python Dictionary

```
>>> dict(t)
```

```
{ 'boys': 11, 'girls', 13 }
```

```
>>> menu = dict(s)
```

```
>>> menu
```

```
{ 'soup': [ ('mushroom', 4), ('tomato', 3) ],
```

```
  'mains': ('chicken', 10), 'dessert': { 'weekdays': 0,  
    'weekends': ('lava cake', 7) } }
```

Create Python Dictionary

`dict()` applied to a sequence of key-value pairs

```
>>> dict([(1, 2), (2, 4), (3, 6)]) #list of kv tuples  
{1:2, 2:4, 3:6}
```

Alternatively,

```
>>> dict((1, 2), (2, 4), (3, 6)) #tuple of kv tuples  
>>> dict([1, 2], [2, 4], [3, 6]) #list of kv list
```


Access value using key

```
>>> menu = { 'soup': [ ('mushroom', 4), ('tomato', 3) ],  
             'mains': ('chicken', 10), 'dessert': { 'weekdays': 0,  
             'weekends': ('lava cake', 7) } }
```

```
>>> menu[ 'mains' ]  
  
( 'chicken', 10)
```

```
>>> menu[ 'sides' ]
```

KeyError

Check whether entry exist

```
>>> menu = { 'soup' : [ ('mushroom' , 4) , ('tomato' , 3) ] ,  
             'mains' : ('chicken' , 10) , 'dessert' : { 'weekdays' : 0 ,  
               'weekends' : ('lava cake' , 7) } }
```

##check for entry using in

```
>>> 'dessert' in menu    #only works for key
```

```
True
```

```
>>> 'chicken' in menu
```

```
False                #since 'chicken' is not a key
```

Add/update an entry to dictionary

```
>>> menu['sides'] = ('salad', 4) #add sides entry
```

```
>>> menu
```

```
{ 'soup': [ ('mushroom', 4), ('tomato', 3) ],  
  'mains': ('chicken', 10), 'dessert': { 'weekdays': 0,  
    'weekends': ('lava cake', 7) }, 'sides': ('salad', 4) }
```

#Note that dictionary is mutable, and supports key-value assignment/re-assignment

Add/update an entry to dictionary

```
>>> menu[ 'sides' ]= ( 'fries' ,4)      #update sides entry
```

```
>>> menu
```

```
{ 'soup' : [ ( 'mushroom' , 4) , ( 'tomato' ,3) ] ,  
  'mains' : ( 'chicken' ,10) ,  'dessert' : { 'weekdays' :0 ,  
        'weekends' : ( 'lava cake' ,7) } ,  'sides' : ( 'fries' ,4) }
```

```
#sides value has been updated, in the same way sides  
was created
```

Delete entry

```
>>> del menu['soup']    #delete sides entry using key
```

```
>>> menu
```

```
{ 'mains': ('chicken',10), 'dessert':{ 'weekdays':0,  
    'weekends':('lava cake',7)}, 'sides':('fries',4) }
```

```
>>> menu.clear() #delete all entries
```

List keys, values of dictionary

```
menu = { 'mains' : ( 'chicken',10),  
        'dessert' : { 'weekdays':0, 'weekends': ( 'lava  
cake',7) }, 'sides' : ( 'fries',4) }
```

```
>>> list(menu.keys())    #keys of menu
```

```
['mains', 'dessert', 'sides']
```

```
>>> list(menu.values())  #values of menu
```

```
[ ( 'chicken',10), { 'weekdays':0, 'weekends': ( 'lava  
cake',7) }, ( 'fries',4) ]
```

Looping construct

```
>>> for key in menu:  
    print menu[key]
```

```
('chicken',10)
```

```
{ 'weekdays':0, 'weekends': ('lava cake',7) }
```

```
('fries',4)
```

```
>>> for key, value in menu.items():  
    print (key, value)
```

```
'mains' ('chicken',10)
```

```
'dessert' { 'weekdays':0, 'weekends': ('lava cake',7) }
```

```
'sides' ('fries',4)
```