Python 元组 Python 日期和时间

Python 字典(Dictionary)

字典是另一种可变容器模型，且可存储任意类型对象。

字典的每个键值 `key=>value` 对用冒号 `:` 分割，每个键值对之间用逗号 `,` 分割，整个字典包括在花括号 `{}` 中 ,格式如下所示：

```
d = {key1 : value1, key2 : value2 }
```

键一般是唯一的，如果重复最后的一个键值对会替换前面的，值不需要唯一。

```
>>>dict = {'a': 1, 'b': 2, 'b': '3'}
>>> dict['b']
'3'
>>> dict
{'a': 1, 'b': '3'}
```

值可以取任何数据类型，但键必须是不可变的，如字符串，数字或元组。

一个简单的字典实例：

```
dict = {'Alice': '2341', 'Beth': '9102', 'Cecil': '3258'}
```

也可如此创建字典：

```
dict1 = { 'abc': 456 }
dict2 = { 'abc': 123, 98.6: 37 }
```

访问字典里的值

把相应的键放入熟悉的方括弧，如下实例:

实例

```
#!/usr/bin/python
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
print "dict['Name']: ", dict['Name']
print "dict['Age']: ", dict['Age']
```

以上实例输出结果：

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

如果用字典里没有的键访问数据，会输出错误如下：

实例

```
#!/usr/bin/python
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
print "dict['Alice']: ", dict['Alice']
```

以上实例输出结果：

```
dict['Alice']:
Traceback (most recent call last):
  File "test.py", line 5, in <module>
    print "dict['Alice']: ", dict['Alice']
KeyError: 'Alice'
```

修改字典

向字典添加新内容的方法是增加新的键/值对，修改或删除已有键/值对如下实例:

实例

```
#!/usr/bin/python
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
dict['Age'] = 8 # 更新
dict['School'] = "RUNOOB" # 添加
print "dict['Age']: ", dict['Age']
print "dict['School']: ", dict['School']
```

以上实例输出结果：

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

删除字典元素

能删单一的元素也能清空字典，清空只需一项操作。

显示删除一个字典用del命令，如下实例：

实例

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
del dict['Name'] # 删除键是'Name'的条目
dict.clear() # 清空词典所有条目
del dict # 删除词典
print "dict['Age']: ", dict['Age']
print "dict['School']: ", dict['School']
```

但这会引发一个异常，因为用del后字典不再存在：

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

注：del()方法后面也会讨论。

字典键的特性

字典值可以没有限制地取任何python对象，既可以是标准的对象，也可以是用户定义的，但键不行。

两个重要的点需要记住：

1）不允许同一个键出现两次。创建时如果同一个键被赋值两次，后一个值会被记住，如下实例：

实例

```
#!/usr/bin/python
dict = {'Name': 'Zara', 'Age': 7, 'Name': 'Manni'}
print "dict['Name']: ", dict['Name']
```

以上实例输出结果：

```
dict['Name']:  Manni
```

2）键必须不可变，所以可以用数字，字符串或元组充当，所以用列表就不行，如下实例：

实例

```
#!/usr/bin/python
dict = {['Name']: 'Zara', 'Age': 7}
print "dict['Name']: ", dict['Name']
```

以上实例输出结果：

```
Traceback (most recent call last):
  File "test.py", line 3, in <module>
    dict = {['Name']: 'Zara', 'Age': 7}
TypeError: list objects are unhashable
```

字典内置函数&方法

Python字典包含了以下内置函数：

序号	函数及描述
1	cmp(dict1, dict2) 比较两个字典元素。

2	len(dict) 计算字典元素个数，即键的总数。
3	str(dict) 输出字典可打印的字符串表示。
4	type(variable) 返回输入的变量类型，如果变量是字典就返回字典类型。

Python字典包含了以下内置方法：

序号	函数及描述
1	dict.clear() 删除字典内所有元素
2	dict.copy() 返回一个字典的浅复制
3	dict.fromkeys(seq[, val]) 创建一个新字典，以序列 seq 中元素做字典的键，val 为字典所有键对应的初始值
4	dict.get(key, default=None) 返回指定键的值，如果值不在字典中返回default值
5	dict.has_key(key) 如果键在字典dict里返回true，否则返回false
6	dict.items() 以列表返回可遍历的(键, 值) 元组数组
7	dict.keys() 以列表返回一个字典所有的键
8	dict.setdefault(key, default=None) 和get()类似, 但如果键不存在于字典中，将会添加键并将值设为default
9	dict.update(dict2) 把字典dict2的键/值对更新到dict里
10	dict.values() 以列表返回字典中的所有值
11	pop(key[,default])

	删除字典给定键 key 所对应的值，返回值为被删除的值。key值必须给出。 否则，返回default值。
12	popitem() 随机返回并删除字典中的一对键和值。

Python 元组 Python 日期和时间