

# Dictionary 字典

## 简介

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 **keys->values** 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：

In [ ]:

```
d = {'key_0':'value_0', 'key_1':'value_1'}
```

值得注意的是key值是唯一的，如果有两个相同的key，则后一个key对应的value会覆盖前一个：

In [35]:

```
flower = {'rose':1, 'iris':2, 'iris':3}  
print('iris对应的key值是: '+str(flower["iris"]))
```

iris对应的key值是: 3

## 初始化一个字典

你可以通过如下方法创建一个字典

In [ ]:

```
dic = {'a':0, 'b':1}
```

当然你也可以分两步

In [9]:

```
dic2 = dict() #先建立空字典  
dic2['a'] = 0 #然后添加数值  
dic2['b'] = 1
```

## 数值访问

我们可以通过**dict[key]**的形式访问字典中的value值，即把需要查找的key放进方括号里

In [21]:

```
fruit={"apple":1 , "pear":2} #创建字典
print("fruit['apple']:",fruit['apple'])
print("fruit['pear']:",fruit['pear'])
```

```
fruit['apple']: 1
fruit['pear']: 2
```

如果key值不存在，则会报错

In [22]:

```
print(fruit['citrus'])
```

```
-----
---
KeyError                                Traceback (most recent call last)
<ipython-input-22-884e8f983781> in <module>
----> 1 print(fruit['citrus'])

KeyError: 'citrus'
```

我们可以通过in或not in来判断一个字典中是否包含某个数值:

In [36]:

```
'iris' in flower
```

Out[36]:

True

In [37]:

```
'citrus' not in flower
```

Out[37]:

True

除此之外，字典还提供了三个接口可以进行数值访问，它们是：

**\*.items()**可以列出所有的key和value (需要配合list())

In [23]:

```
print(list(fruit.items()))
```

```
[('apple', 1), ('pear', 2)]
```

**\*.keys()**可以列出所有的key (需要配合list())

In [24]:

```
print(list(fruit.keys()))  
['apple', 'pear']
```

**\*.values()**可以列出所有的value (需要配合list())

In [25]:

```
print(list(fruit.values()))  
[1, 2]
```

因为以上三个接口要和列表一起配合使用，因而通过列表操作，我们可以实现判断dict长度等操作

## 编辑字典

### 添加字典

添加字典可以通过**dict[key]=value**的方式实现

In [28]:

```
void_dict = {}  
void_dict["new_item"] = 1  
print(void_dict)  
  
{'new_item': 1}
```

### 删除字典

你可以一次删除一个对象，也可以直接清空

In [31]:

```
french = {1:'un' , 2:'deux', 3:'trois', 4:'quatre', 5:'cinq'}  
deutsch = {1:'eine', 2:'zwei', 3:'drei', 4:'vier', 5 : 'fünf'}  
del french[5]  
deutsch.clear()  
print(french)  
print(deutsch)  
  
{1: 'un', 2: 'deux', 3: 'trois', 4: 'quatre'}  
{}
```

你也可以选择将整个字典对象全部删除，但删除后对象会彻底消失，不能对其进行操作，否则会报错

In [32]:

```
italian = {1:'uno', 2:'due', 3:'tre'}  
del italian  
italian[1]='uno'
```

```
-----  
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-32-aa7dfe01238f> in <module>  
      1 italian = {1:'uno', 2:'due', 3:'tre'}  
      2 del italian  
----> 3 italian[1]='uno'  
  
NameError: name 'italian' is not defined
```

## 字典中key值的特性

字典**value**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key不行!**  
两个重要的点需要记住：

- 1) 不允许同一个key出现两次。创建时如果同一个key被赋值两次，后一个值会被记住。(实例已在上文展示)
- 2) key必须不可变(不可变，唯一)，所以可以用数字，字符串或元组充当，但用列表就不行，如下实例：

In [39]:

```
foo = {'Name': 'Zara', 'Age': 7}  
  
print ("foo['Name']:", foo['Name'])
```

```
-----  
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-39-26454529164e> in <module>  
----> 1 foo = {'Name': 'Zara', 'Age': 7}  
      2  
      3 print ("foo['Name']:", foo['Name'])  
  
TypeError: unhashable type: 'list'
```