

# 字典类型

# 为什么需要字典类型？

```
>>> list1 = ["name", "age", "gender"]
```

```
>>> list2 = ["fentiao", 5, "male"]
```

```
>>> zip(list1, list2)
```

```
    //通过zip内置函数将两个列表结合，help(zip)
```

```
[('name', 'fentiao'), ('age', 5), ('gender', 'male')]
```

# 为什么需要字典类型？

```
>>> list2[0]
```

```
//在直接编程时，并不能理解第一个索引表示姓名
```

```
'fentiao'
```

```
>>> list2[name]
```

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
TypeError: list indices must be integers, not str
```

故字典是python中唯一的映射类型,key-value(哈希表)，字典对象是可变的，但key必须用不可变对象。

# 字典的定义

- 简单字典创建

```
In [13]: dic = {"name":"fentiao", "age":5, "gender":"male"}

In [14]: dic['name']
Out[14]: 'fentiao'

In [15]: dic[0]
-----
KeyError                                Traceback (most recent call last)
<ipython-input-15-c3244007d783> in <module>()
----> 1 dic[0]

KeyError: 0
```

# 字典的定义

- 内建方法:fromkeys

字典中的key有相同的value值，默认为None

```
In [16]: ddict = {}.fromkeys(('username','password'),'fentiao')
```

```
In [17]: ddict
```

```
Out[17]: {'password': 'fentiao', 'username': 'fentiao'}
```

```
In [18]: ddict = {}.fromkeys(('username','password'),)
```

```
In [19]: ddict
```

```
Out[19]: {'password': None, 'username': None}
```

# 字典值的访问

- 直接通过key访问

```
In [20]: dic = {"name":"fentiao", "age":5, "gender":"male"}
In [21]: dic['name']
Out[21]: 'fentiao'
In [22]: dic['age']
Out[22]: 5
In [23]: dic['male']
-----
KeyError                                Traceback (most recent call last)
<ipython-input-23-1bd3aa66f0f6> in <module>()
----> 1 dic['male']
KeyError: 'male'
```

# 字典值的访问

- 循环遍历访问

```
In [24]: dic = {"name":"fentiao", "age":5, "gender":"male"}  
  
In [25]: for key in dic:  
        ....:     print dic[key]  
        ....:  
male  
5  
fentiao
```

# 字典key-value的添加

- `dic[key] = value`

通过这个操作，我们会发现字典是无序的数据类型

```
In [27]: dic = {"name":"fentiao", "age":5, "gender":"male"}
```

```
In [28]: dic['kind'] = 'cat'
```

```
In [29]: dic
```

```
Out[29]: {'age': 5, 'gender': 'male', 'kind': 'cat', 'name': 'fentiao'}
```



# 字典的删除

- `dic.pop(key)`  
根据key值删除字典的元素；
- `dic.popitem()`  
随机删除字典元素，返回（key,value）
- `dic.clear()`  
删除字典中的所有元素
- `del dic`  
删除字典本身

# 字典的常用方法

- `dict.get()`

如果key存在于字典中，返回对应value值

- `dic.keys()`

返回字典的所有key值

- `dict.has_keys()`

字典中是否存在某个key值

over !