

# 11.字典类型

字典：字典是一种映射类型（mapping type），是由键值对组成的，字典的每个元素都有不同的键（key）与之对应，需要通过键来访问元素，字典里的元素排列是无序的，因此字典不是序列类型。因此，字典的键的类型必需是不可变类型（布尔类型、整数类型、浮点数类型、字符串类型、元组类型以及不可变集合类型）。但是字典也是可变类型，因此字典是不能作为字典的键的，不可变类型其实就是说这个类型的值是可哈希的，用python内置的hash函数就可以检验。

## 一、字典的创建

### 1. 使用{}创建字典

```
# 创建空字典
d1 = {}
# 创建有元素的字典
d2 = {'a': 1, 'b': 2, 'c': 3}
# python的键必需是不可变类型，值可以是任意的类型
d3 = {1: "hello", "name": "brooks", (1, 2): 3.1415, 1.414: [1, 2, 4, 5], True: {"help": 30}}
```

### 2. 使用dict函数创建字典

```
# 创建空字典
d1 = dict()
# 创建有元素的字典
d2 = dict([(1, 'a'), (2, 'b'), (3, 'c')])
# 注意：使用dict创建字典的时候，参数必需是可迭代类型，参数的元素必须是成对出现的，元素的第一个值作为字典的键，第二个值作为字典的值。这种类型的参数可以直接通过zip或者是map函数来快速创建
# 下面的这几种形式都是支持的
# 1. 两个元素组成的二维列表：[['a', 1], ['b', 2], ['c', 3], ['d', 4]]
# 2. 包含双值元组的列表：[('a', 'b'), ('c', 'd'), ('e', 'f')]
# 3. 包含双值列表的元组：([ 'a', 1], [ 'b', 2], [ 'c', 3], [ 'd', 4])
# 4. 包含双值元组的元组：(( 'a', 'b'), ( 'c', 'd'), ( 'e', 'f'))
# 5. 双字符串字符串组成的列表：['ab', 'cd', 'ef', 'gh']
# 6. 双字符串字符串组成的元组：('ab', 'cd', 'ef')
# 7. 上面的可以互相嵌套：[['a', 1], ('c', 'd'), 'a1', ((3, ), ['a', 12.4])] 猜猜这个会生成啥，写手写再用python运行验证
```

### 3. 使用fromkeys方法创建字典

```
# fromkeys(keys,[value]) 第一个参数是可迭代类型，类型的元素必须是可哈希的，
第二个值是默认值，默认是None
# 1.使用dict.fromkeys()创建字典，字典的键分别是a,b,c,d,e值都是None
d1 = dict.fromkeys('abcde')
# 字典的键分别是1, 2, 3, 4值都是0
d2 = dict.fromkeys([1,2,3,4], 0)
# 2. 使用{}.fromkeys()创建字典，参数的含义跟使用dict是一样的
d3 = {}.fromkeys('abcde')
# 3. 如果{}不是空的，那么使用fromkeys创建时会覆盖掉原来的内容
d4 = {'a': 2}.fromkeys(["name", "age", "class", "addr"], "")
```

## 二、字典的相关操作(增删改查)

### 1. 访问字典中的值

```
#字典中的值都是通过键名来直接访问的
dict1 = {'name': 'brooks', 'age': 18, 'score': 99.9, 'fav':
'program'}
>>> dict1['name']
'brooks'
# 使用内键方法get来访问，get方法接收两个参数，第一个是键名，第二个是默认值
# 默认值的意思就是说，假如键名不存在，就返回默认值，默认值默认是None
>>> dict1.get('name')
'brooks'
>>> dict1.get('haha') # 不存在，返回默认值None
None
>>> dict1.get("what", "hehe") # 不存在，返回自定义的默认值hehe
"hehe"
# 使用字典内建方法setdefault来访问字典中的键，该方法接收两个参数，第一个是键
名，
# 第二个是默认值None。如果键名存在，就返回键名对应的值，如果键名不存在，则
# 在字典中创建一个键名，并把默认值设为对应的值。
>>> dict1.setdefault('name')
'brooks'
>>> dict1.setdefault('you')
>>> dict1.setdefault('wait', 10)
# 通过for循环来访问字典中的元素，for循环默认迭代的是字典中的键(key)
for key in dict1:
    print dict1[key]
# items返回的是类集合对象的dict_items对象，需要使用for循环才能遍历获取里面的元
素
for key, val in dict1.items():
    print key, val
```

### 2. 修改字典中的值

```
#修改字典中的值也是通过键名来修改的
dict1 = {'name': 'brooks', 'age': 18, 'score': 99.9, 'fav':
'program'}
dict1['name'] = 'jonny'
```

### 3. 给字典添加新元素

```
# 当给字典一个不存在的键名赋值时，字典就会自动的创建对应的键名并把值给它
dict1 = {'name': 'brooks', 'age': 18, 'score': 99.9, 'fav':
'program'}
dict1['salary'] = 10000
# 将另一个字典中的元素添加进来
dict2 = {1: 'a', 2: 'b'}
dict1.update(dict2) # 将dict2中的所有键值对添加到dict1中
```

### 4. 删除字典中的元素和字典

```
# 删除指定键名的元素
dict1 = {'name': 'brooks', 'age': 18, 'score': 99.9, 'fav':
'program'}
del dict1['age'] # 把键名为age的元素删除
dict1.pop('name') # 把键名为name的元素删除并返回它的值
# 清空整个字典（删除字典里面的所欲元素）
dict1.clear()
# 删除字典
del dict1
```

### 5. 查询字典中是否存在某个键

```
# 使用成员操作符(in, not in)查询字典中是否存在某个键名
dict1 = {'name': 'brooks', 'age': 18, 'score': 99.9, 'fav':
'program'}
>>> 'name' in dict1
True
>>> 'name' not in dict1
False
```

## 三、字典的相关函数(方法)

### 1. python的内建函数

函数	说明
dict([container])	创建字典的工厂函数。如果提供了容器类（container），就用其中的条目填充字典，否则就创建一个空字典。
len(mapping)	返回映射类型的长度（键值对的个数）
hash(obj)	返回obj的哈希值

## 1. dict的内建方法

方法	说明
clear	清空字典中的所有元素
copy	返回字典的一个浅拷贝
dict.fromkeys(seq, val=None)	创建并返回一个新的字典，以seq（序列类型）中的元素做该字典，val做该字典中所有键值对应的初始值（如果不提供此值，则默认为None）
dict.get(key, default=None)	返回字典中键key对应的值，如果字典中不存在此键，则返回default的值，default的值默认为None
dict.items()	返回一个包含字典中键值对组成的类集合对象dict_items，只能通过迭代的方式获取里面的元素
dict.keys()	返回字典中所有键组成的一个类集合对象，这个对象叫dict_keys
dict.pop(key)	从字典中删除键key及其对应的值，并返回其对应的值
dict.popitem()	删除字典中的某个键值对，并返回有键值对组成的双元素元组
dict.setdefault(key, default=None)	如果字典中存在key，则返回key对应的值，如果不存在则在字典中新建一个key，并以default的值设为它的值，default的值默认是None
dict.update(dict2)	将字典dict2中的所有元素添加到dict中
dict.values()	返回由dict中所有值组成的dict_values对象

## 四、字典相关特性总结

---

1. 字典是一种映射类型，是由键值对组成的，其中字典的键必需是可哈希的（也就是不可变的），而且同一个字典内不能出现同样的键名；字典的值可以是任意类型，同时字典的值是可以重复的。
2. 字典也是一种容器类型，容器类型包括字符串、列表、元组、字典、集合，因为这些类型就像一个容器一样可以装一堆的东西，所以叫容器类型（不要求记住）。所有的容器类型都是可迭代的，也就是可以通过循环语句一个个的把里面的元素拿出来的。所有的容器类型都可以通过len()方法来得到里面元素的个数。
3. 字典是可变类型，因此可以对字典进行增删改查等操作。

## 作业

---

1. 使用字典的方式实现关键词的分类