

Python语言基础与应用

数据类型 / 容器类型：字典

陈斌 北京大学 gischen@pku.edu.cn



容器类型：字典

- › 贴标签的数据
- › 创建一个字典
- › 访问字典的数据项
- › 在字典中查找

贴标签的数据

› “标签收纳盒”

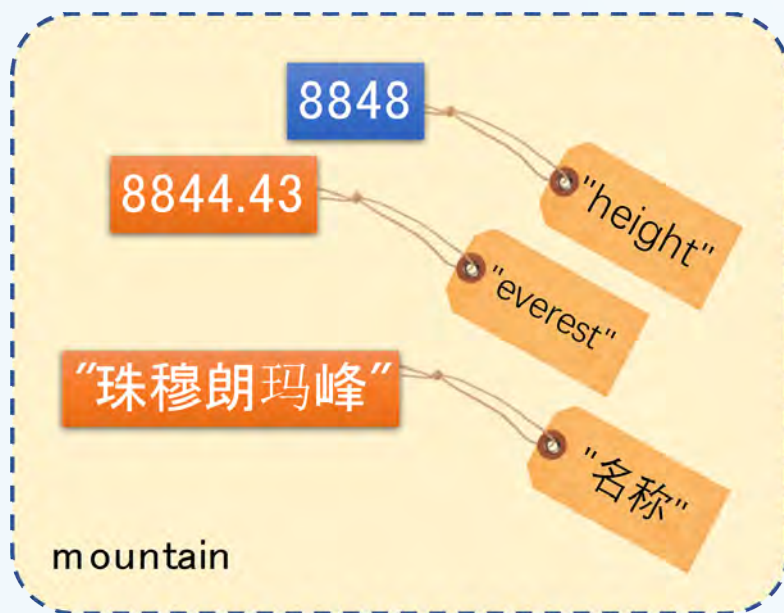
- 给数据贴上标签，就可以通过具有特定含义的名字或者别的记号来获取数据。



贴标签的数据

› 标签 (key) 和数据值 (value)

- 字典容器中保存着一系列的key-value对
- 通过键值key来索引元素value



创建一个字典

› 花括号法和指明类型法

```
student = {}  
student = dict()
```

› 数据项 (item)

- 字典中保存的各个标签-数据值 (key-value)
- 标签和数据值之间用冒号“:”连接

› 批量添加数据项

```
student = dict.fromkeys(("name", "age"))
```

› 字典是可变类型，可以添加、删除、替换元素

创建一个字典

- › 字典中的元素value没有顺序，可以是任意类型，甚至也可以是字典
- › 字典的键值key可以是任意不可变类型（数值 / 字符串 / 元组）

例如：用元组来作为坐标，索引元素

```
>>> bands={'Marxes':['Moe','Curly'],'KK':[True,'moon']}
>>> bands['KK'][0]
True
>>> poi={(100,100):'Zhongguancun',(123,23):'Pizza'}
>>> poi[(100,100)]
'Zhongguancun'
```

更新一个字典

› 合并字典

update方法

› 增长字典

“关联”操作

update操作：以key=value
的形式批量添加数据项

```
>>> student = {}
>>> student["name"] = "Tom"
>>> student["age"] = "20"
>>> student["gender"] = "male"
>>> student
{'name': 'Tom', 'age': '20', 'gender': 'male'}
>>> bar = {"course": ["数学", "英语"]}
>>> student.update(bar)
>>> student
{'name': 'Tom', 'age': '20', 'gender': 'male', 'course': ['数学', '英语']}
>>> student.update(friends=["Mike", "Alice"])
>>> student
{'name': 'Tom', 'age': '20', 'gender': 'male', 'course': ['数学', '英语'], 'friends': ['Mike', 'Alice']}
```


更新一个字典

› 缩减字典

`del`操作：删除指定标签的数据项

`pop`操作：删除指定标签的数据项并返回数据值

`popitem`操作：删除并返回任意一个数据项

`clear`操作：清空字典

› 字典大小

`len`函数

```
>>> del student["age"]
>>> student.pop('course')
['数学', '英语']
>>> student.popitem()
('friends', ['Mike', 'Alice'])
>>> student
{'name': 'Tom', 'gender': 'male'}
>>> student.clear()
>>> student
{}
```

访问字典的数据项

› 标签索引

`dict[key]`

- 获取字典中指定标签的数据值
- 更新指定标签的数据项

› get操作

```
>>> student = {"name": "Tom", "age": 20, "gender": "male"}
>>> student['age']
20
>>> student['age'] = 24
>>> student
{'name': 'Tom', 'age': 24, 'gender': 'male'}
>>> student.get('name')
'Tom'
```

访问字典的数据项

› 获取字典的标签、数据值和数据项

- `keys`函数：返回字典中的所有标签；
- `values`函数：返回字典中的所有数据值；
- `items`函数：将每个数据项表示为二元元组，返回所有的数据项。

```
>>> student = {"name": "Tom", "age": 20, "gender": "male"}
>>> student.keys()
dict_keys(['name', 'age', 'gender'])
>>> student.values()
dict_values(['Tom', 20, 'male'])
>>> student.items()
dict_items([('name', 'Tom'), ('age', 20), ('gender', 'male')])
```

在字典中查找

› in操作

判断字典中是否存在某个标签

› in操作和values函数的组合

判断字典中是否存在某个数据值

```
>>> student = {"name": "Tom", "age": 20, "gender":  
"male", "course": ['math', 'computer']}  
>>> 'name' in student  
True  
>>> 'city' in student  
False  
>>> 20 in student.values()  
True
```