

5. 集合

集合（Set）是用来存储多个数据的数据结构，它跟列表相似，是一个无序不重复元素的序列。其特点总结如下：

- 数据唯一，不重复
- 数据存储是无序的

小贴士

由于集合数据的唯一性，所以在实际开发中，集合常被用来作为过滤器使用或者用来统计数据的种类。

（1）创建集合

集合的创建语法，跟列表一样，但是元素使用大括号括{}起来。

【语法】

1) 创建含有元素的集合

集合名 = {数据 1, 数据 2, ……}

2) 创建空集合

集合名 = set()

想一想，为什么创建空集合使用 `set`，而不是直接采用如下代码的 `city` 呢？因为这段代码是创建一个空的字典！一定注意区分

```
city = {} #创建空字典
address = set() #创建空集合
colour = {'white', 'black', 'yellow', 'red', 'pink'} #创建含 5 个元素的集合
```

有一点值得说明一下，集合的元素不能重复，来看一下如下这段代码：

```
colour = {'white', 'black', 'yellow', 'red', 'pink', 'black'}
print(colour)
```

输出结果为：

```
{'red', 'pink', 'white', 'black', 'yellow'}
```

定义时集合中的元素“black”是重复的，尽管程序不会报错，但是进行实际运算时，

只输出了一个“black”，这显示了集合的去重功能。

(2) 访问集合元素

判断元素是否在集合中存在，使用如下方法。

【语法】

`e in s`

判断元素 `e` 是否在集合 `s` 中存在，存在则返回 `True`，否则返回 `False`。

(3) 更新集合

1) 添加集合元素，使用如下方法。

【语法】

`s.add(e)`

将元素 `e` 添加到集合 `s` 中，如果元素已存在，则不进行任何操作。

2) 删除集合元素，使用如下方法。

【语法】

`s.remove(e)`

将元素 `e` 从集合 `s` 中删除，如果元素不存在，则会发生错误。

示例：集合 `colour` 是各种颜色的集合，要求完成如下操作：

- 查询集合中是否有紫色“purple”，给出提示。如果没有，添加紫色
- 查询集合中是否有红色“red”，给出提示，如果有，删除之。

```

if 'purple' in colour:
    print('集合中存在紫色 purple')
else:
    colour.add('purple')
print('新的集合为:', colour)
#查询集合中是否有红色“red”，给出提示，如果有，删除之
if 'red' in colour:
    colour.remove('red')
    print('红色 red 已删除')
else:
    print('红色不存在')
print('新的集合为:', colour)

```

输出结果为：

原始集合为: {'red', 'black', 'yellow', 'pink', 'white'}

新的集合为: {'red', 'black', 'yellow', 'pink', 'white', 'purple'}

红色 red 已删除

新的集合为: {'black', 'yellow', 'pink', 'white', 'purple'}

(4) 集合运算

Python 中的集合与数学上的集合一样，可以计算两个集合的交集和并集。集合运算符如下表所示。

运算符	说明
	计算两个集合的并集
&	计算两个集合的交集

示例：员工 emp1 和 emp2 本周的加班时间设置成两个集合，要求计算：

- 两个员工总的加班时间。
- 两个员工都加班的时间。

```

emp1 = {'周三', '周四', '周五'}
emp2 = {'周一', '周四', '周六'}
overtime_totle = emp1|emp2
overtime_intersect = emp1&emp2
print('总加班时间:', overtime_totle)
print('都加班的时间:'.overtime_intersect)

```

输出结果为：

总加班时间： {'周三','周一','周四','周六','周五'}

都加班的时间： {'周四'}

小贴士

不知你发现没有，相同的程序，每次运行，集合元素显示的顺序都可能不同，这也证明了集合元素无序存储的特点。