

集合

- 目标
 - 创建集合
 - 集合数据的特点
 - 集合的常见操作

1、创建集合

创建集合使用 `{}` 或 `set()`，但是如果需要创建空集合只能使用 `set()`，因为 `{}` 用来创建空字典。

```
s1 = {10, 20, 30, 40, 50}
print(s1)

s2 = {10, 30, 20, 10, 30, 40, 30, 50}
print(s2)

s3 = set('abcdefg')
print(s3)

s4 = set()
print(type(s4)) # set

s5 = {}
print(type(s5)) # dict
```

```
D:\Anaconda3\python.exe "Z:/01 青灯教育课程/01 青灯教育-vip课程/
{40, 10, 50, 20, 30}
{40, 10, 50, 20, 30}
{'b', 'e', 'a', 'd', 'g', 'c', 'f'}
<class 'set'>
<class 'dict'>

Process finished with exit code 0
```

特点：

1. 集合可以去掉重复数据；
2. 集合数据是无序的，故不支持下标

2、集合常见操作方法

2、1 增加数据

- `add()`

```
s1 = {10, 20}
s1.add(100)
s1.add(10)
print(s1) # {100, 10, 20}
```

因为集合有去重功能，所以，当向集合内追加的数据是当前集合已有数据的话，则不进行任何操作。

- update(), 追加的数据是序列。

```
s1 = {10, 20}
# s1.update(100) # 报错
s1.update([100, 200])
s1.update('abc')
print(s1)
```

```
D:\Anaconda3\python.exe "Z:/01 青灯教育课程/01 青灯教育-vip课程/
{100, 200, 10, 'name', 'b', 20, 'a', 'c', '正心'}

Process finished with exit code 0
```

2、2 删除数据

- remove(), 删除集合中的指定数据，如果数据不存在则报错。

```
s1 = {10, 20}

s1.remove(10)
print(s1)

s1.remove(10) # 报错
print(s1)
```

- discard(), 删除集合中的指定数据，如果数据不存在也不会报错。

```
s1 = {10, 20}

s1.discard(10)
print(s1)

s1.discard(10)
print(s1)
```

2、3 查找数据

- in: 判断数据在集合序列
- not in: 判断数据不在集合序列

```
s1 = {10, 20, 30, 40, 50}
```

```
print(10 in s1)  
print(10 not in s1)
```

3、总结

- 创建集合

- 有数据集合

```
s1 = {数据1, 数据2, ...}
```

- 无数据集合

```
s1 = set()
```

- 常见操作

- 增加数据

- add()
 - update()

- 删除数据

- remove()
 - discard()