PYTHON
编程基础

集合的运算

交集和并集

集合中的intersection方法可以用于计算一个集合与另一个集合的交集，语法格式为：

```
s1.intersection(s2)
```

其作用是计算s1和s2的交集并返回。intersection方法不会修改s1和s2本身的值。

集合中的union方法可以用于计算一个集合与另一个集合的并集，语法格式为：

```
s1.union(s2)
```

其作用是计算s1和s2的并集并返回。union方法不会修改s1和s2本身的值。

差集

集合中的difference方法可以用于计算一个集合与另一个集合的差集，语法格式为：

```
s1.difference(s2)
```

其作用是计算s1和s2的差集并返回，差集是指由包含在s1中但不包含在s2中的元素组成的集合。difference方法不会修改s1和s2本身的值。

对称差集

集合中的symmetric_difference方法可以用于计算一个集合与另一个集合的对称差集，语法格式为：

```
s1. symmetric_difference(s2)
```

其作用是计算s1和s2的对称差集并返回，对称差集是指由只包含在s1中或只包含在s2中的元素组成的集合。symmetric_difference方法不会修改s1和s2本身的值。

示例

例：集合的交、并、差、对称差示例。

```
s1=set([1,2,3])
s2=set([2,3,4])
s3=s1.intersection(s2) #{2, 3}
s4=s1.union(s2) #{1, 2, 3, 4}
s5=s1.difference(s2) #{1}
s6=s1.symmetric_difference(s2) #{1, 4}
```

示例

例：集合的交、并、差、对称差示例。

```
print('s1和s2的值分别为：',s1,s2)
print('s1和s2的交集为：',s3)
print('s1和s2的并集为：',s4)
print('s1和s2的差集为：',s5)
print('s1和s2的对称差集为：',s6)
```

示例

例：集合的交、并、差、对称差示例。

s1和s2的值分别为： {1, 2, 3} {2, 3, 4}

s1和s2的交集为： {2, 3}

s1和s2的并集为： {1, 2, 3, 4}

s1和s2的差集为： {1}

s1和s2的对称差集为： {1, 4}

子集和父集

集合中的issubset方法用于判断一个集合是否是另一个集合的子集，语法格式为：

```
s1. issubset(s2)
```

其作用是判断s1是否是s2的子集。如果s1是s2的子集，则返回True；否则，返回False。

子集和父集

集合中的issuperset方法可以用于判断一个集合是否是另一个集合的父集，语法格式为：

```
s1.issuperset(s2)
```

其作用是判断s1是否是s2的父集（即判断s2是否是s1的子集）。如果s1是s2的父集，则返回True；否则，返回False。

子集和父集

例：子集和父集判断示例。

```
s1=set([1,2,3,4]) #创建集合对象并赋给变量s1
s2=set([2,3,4,5]) #创建集合对象并赋给变量s2
s3=set([1,3]) #创建集合对象并赋给变量s3
print('s3是s1的子集：',s3.issubset(s1)) #True
print('s1是s3的父集：',s1.issuperset(s3)) #True
print('s3是s2的子集：',s3.issubset(s2)) #False
print('s2是s3的父集：',s3.issuperset(s2)) #False
```

子集和父集

例：子集和父集判断示例。

```
s3是s1的子集： True
s1是s3的父集： True
s3是s2的子集： False
s2是s3的父集： False
```