

在 Python 语言中的交集符号是 $&$ ，另外，也可以使用 `intersection()` 方法完成这个工作。

程序实例 ch10_3.py：有数学与物理 2 个夏令营，这个程序会列出同时参加这 2 个夏令营的成员。

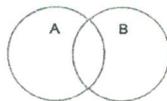
```
1 # ch10_3.py
2 math = {'Kevin', 'Peter', 'Eric'} # 设定参加数学夏令营的成员
3 physics = {'Peter', 'Nelson', 'Tom'} # 设定参加物理夏令营的成员
4 both1 = math & physics
5 print("同时参加数学与物理夏令营的成员 ",both1)
6 both2 = math.intersection(physics)
7 print("同时参加数学与物理夏令营的成员 ",both2)
```

执行结果

```
===== RESTART: D:\Python\ch10\ch10_3.py =====
同时参加数学与物理夏令营的成员 {'Peter'}
同时参加数学与物理夏令营的成员 {'Peter'}
```

10-2-2 并集 (union)

有 A 和 B 两个集合，如果想获得所有的元素，则可以使用并集。例如：你举办了数学（可想成 A 集合）与物理（可想成 B 集合）2 个夏令营，如果想统计参加数学或物理夏令营的全部成员，可以使用此功能。



并集的数学符号是 \cup ，若是以右图而言就是 $A \cup B$ 。

Python 语言中的并集符号是 `|`，另外，也可以使用 `union()` 方法完成这个工作。

程序实例 ch10_4.py：有数学与物理 2 个夏令营，这个程序会列出参加数学或物理夏令营的所有成员。

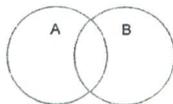
```
1 # ch10_4.py
2 math = {'Kevin', 'Peter', 'Eric'} # 设定参加数学夏令营的成员
3 physics = {'Peter', 'Nelson', 'Tom'} # 设定参加物理夏令营的成员
4 allmember1 = math | physics
5 print("参加数学或物理夏令营的成员 ",allmember1)
6 allmember2 = math.union(physics)
7 print("参加数学或物理夏令营的成员 ",allmember2)
```

执行结果

```
===== RESTART: D:\Python\ch10\ch10_4.py =====
参加数学或物理夏令营的成员 {'Eric', 'Tom', 'Peter', 'Nelson', 'Kevin'}
参加数学或物理夏令营的成员 {'Eric', 'Tom', 'Peter', 'Nelson', 'Kevin'}
```

10-2-3 差集 (difference)

有 A 和 B 两个集合，如果想获得属于 A 集合同时不属于 B 集合的元素，则可以使用差集 (A-B)。如果想获得属于 B 集合同时不属于 A 集合的元素，则可以使用差集 (B-A)。例如：你举办了数学（可想成 A 集合）与物理（可想成 B 集合）2 个夏令营，如果想统计参加数学夏令营但是没有参加物理夏令营的成员，可以使用此功能。



如果想统计参加物理夏令营但是没有参加数学夏令营的成员，也可以使用此功能。

在 Python 语言中差集符号是 `-`，另外，也可以使用 `difference()` 方法完成这个工作。

