

Python 基礎

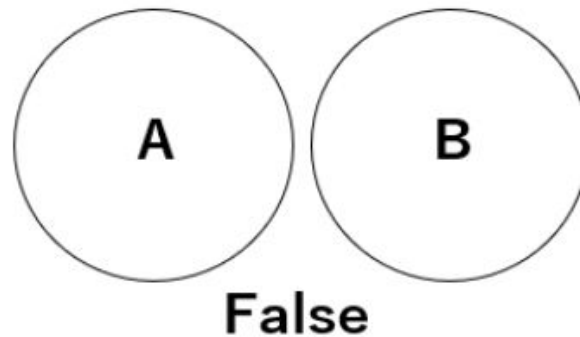
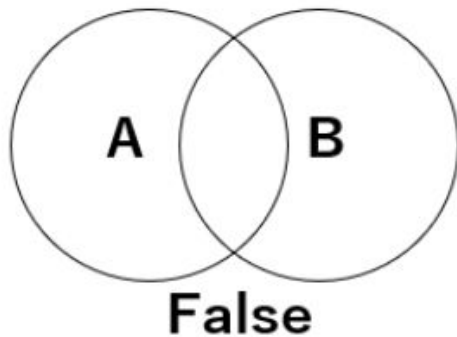
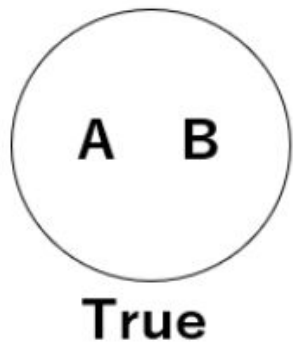
set - 要素の比較

「==」演算子

複数のsetの要素が**完全に一致**しているかどうかを調べるには「==」演算子を使います。

```
set == set
```

setには要素の順番はないので、並びは関係ありません。



「==」演算子例:

```
1 set1 = {'A', 'B', 'C'}
2 set2 = {'A', 'B', 'C'}
3 set3 = {'B', 'C', 'A'} # 順序は同じではありません
4 set4 = {'C', 'D', 'E'} # 部分的に同じ
5 set5 = {'D', 'E', 'F'} # 完全不一致
6
7 print(set1 == set2)
8 print(set1 == set3)
9 print(set1 == set4)
10 print(set1 == set5)
```

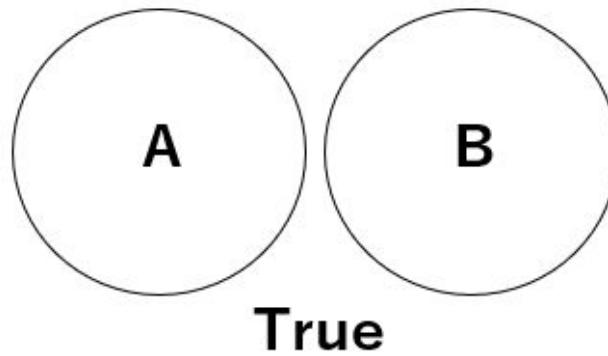
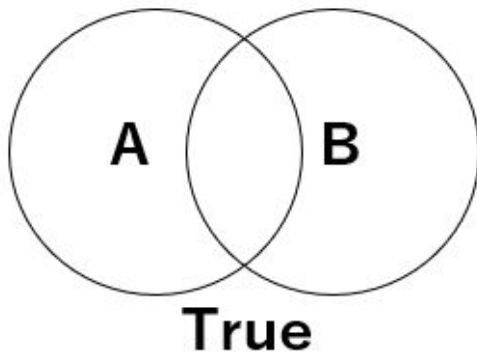
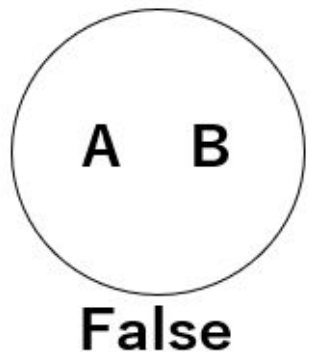
Results:

```
True
True
False
False
```

「!=」演算子

等しくないかどうかについては比較演算子の != で調べることができます。

```
set != set
```



「!=」演算子例:

```
1 set1 = {'A', 'B', 'C'}
2 set2 = {'A', 'B', 'C'}
3 set3 = {'B', 'C', 'A'} # 順序は同じではありません
4 set4 = {'C', 'D', 'E'} # 部分的に同じ
5 set5 = {'D', 'E', 'F'} # 完全不一致
6
7 print(set1 != set2)
8 print(set1 != set3)
9 print(set1 != set4)
10 print(set1 != set5)
```

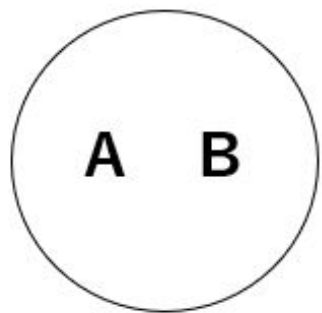
Results:

```
False
False
True
True
```

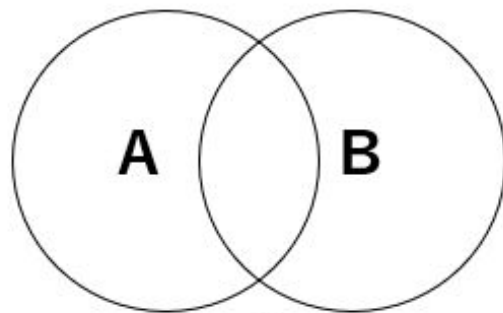
完全不一致 - isdisjoint()メソッド

複数のsetに共通の要素が1つでもあるかどうか(完全不一致かどうか)を判定するには、isdisjoint()メソッドを使います。

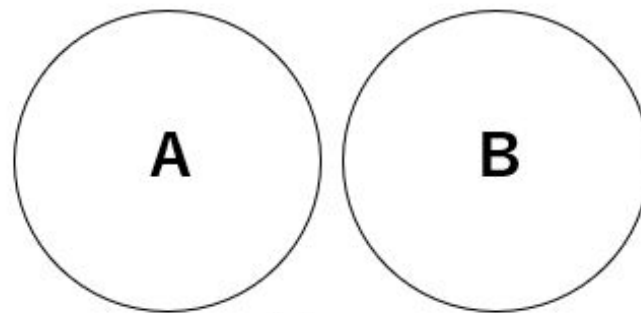
```
set.isdisjoint(set)
```



False



False



True

完全不一致 - isdisjoint()メソッド 例:

```
1 set1 = {'A', 'B', 'C'}
2 set2 = {'A', 'B', 'C'}
3 set3 = {'B', 'C', 'A'} # 順序は同じではありません
4 set4 = {'C', 'D', 'E'} # 部分的に同じ
5 set5 = {'D', 'E', 'F'} # 完全不一致
6
7 print(set1.isdisjoint(set2))
8 print(set1.isdisjoint(set3))
9 print(set1.isdisjoint(set4))
10 print(set1.isdisjoint(set5))
```

Results:

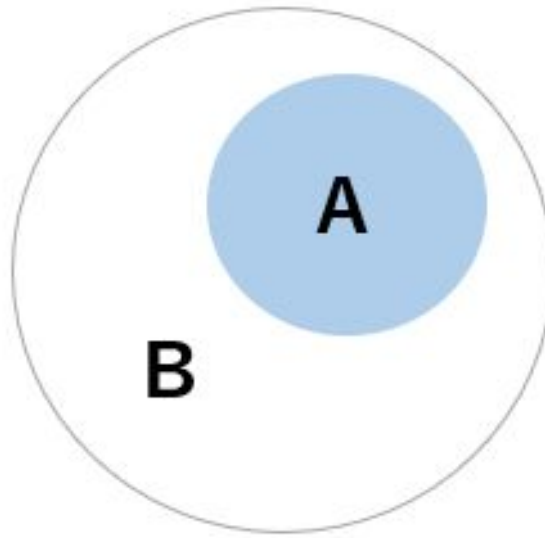
```
False
False
False
True
```

部分集合の判定 | \leq 演算子 and `issubset()`

Aの要素がBの要素の一部分から成り立っている場合、AはBの部分集合といいます。

```
set <= set
```

```
set.issubset(set)
```



部分集合の判定 | <=演算子 and issubset()例:

```
1  setA = {1, 2, 3}
2  setB = {1, 2, 3, 4, 5, 6}
3  print(setA <= setB)
4  print(setA.issubset(setB))
5
6  setA = {1, 2, 3, 4, 5, 6, 7, 8, 9}
7  setB = {1, 2, 3, 4, 5, 6}
8  print(setA <= setB)
9  print(setA.issubset(setB))
10
```

Results:

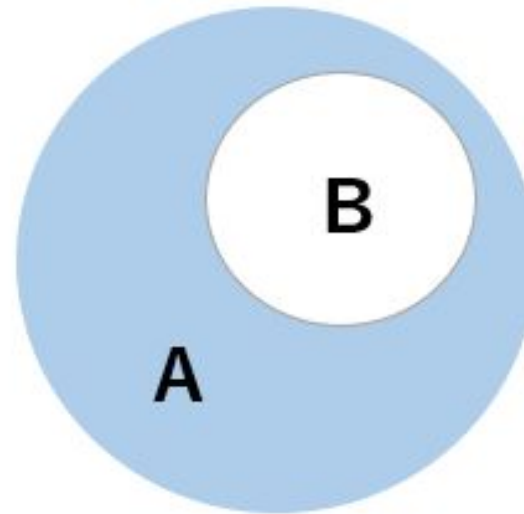
```
True
True
False
False
```

上位集合の判定 | \supseteq 演算子 and `issuperset()`

Aの要素の中にBの要素が全て含まれている場合、AはBの上位集合といいます。

```
set >= set
```

```
set.issuperset(set)
```



上位集合の判定 | \supseteq 演算子 and `issuperset()`

```
1 setA = {1, 2, 3, 4, 5, 6}
2 setB = {1, 2, 3}
3 print(setA >= setB)
4 print(setA.issuperset(setB))
5
6 setA = {1, 2, 3, 4, 5, 6}
7 setB = {1, 2, 3, 4, 5, 6, 7, 8, 9}
8 print(setA >= setB)
9 print(setA.issuperset(setB))
```

Results:

```
True
True
False
False
```