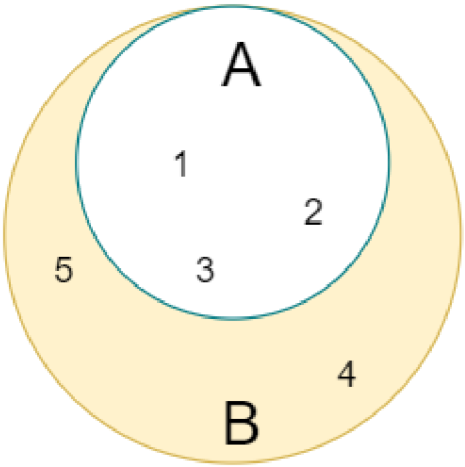
Python issubset

Summary: in this tutorial, you'll learn about how to use the Python `issubset()` method to check if a set (https://www.pythontutorial.net/python-basics/python-set/) is a subset of another set.

Introduction to the Python issubset() method

Suppose that you have two sets A and B. Set A is a subset of set B if all elements of A are also elements of B. Then, set B is a superset of set A.

The following Venn diagram illustrates that the set A is a subset of the set B:



Set A and set B can be equal. If set A and set B are **not** equal, A is a **proper** subset of B.

In Python, you can use the Set `issubset()` method to check if a set is a subset of another:

```
set_a.issubset(set_b)
```

If the `set_a` is a subset of the `set_b` , the `issubset()` method returns `True` . Otherwise, it returns `False` .

The following example uses the `issubset()` method to check if the `set_a` is a subset of the `set_b` :

```
numbers = {1, 2, 3, 4, 5}
scores = {1, 2, 3}

print(scores.issubset(numbers))
```

Output:

```
True
```

By definition, a set is also a subset of itself. The following example returns `True` :

```
numbers = {1, 2, 3, 4, 5}

print(numbers.issubset(numbers))
```

Output:

```
True
```

The following example returns `True` because some elements in the `numbers` set aren't in the `scores` set. In other words, the `numbers` set is not a subset of the `scores` set:

```
numbers = {1, 2, 3, 4, 5}
scores = {1, 2, 3}
```

```
print(numbers.issubset(scores))
```

Output:

```
False
```

Using subset operators

Besides using the `issubset()` method, you can use the subset operator (`<=`) to check if a set is a subset of another set:

```
set_a <= set_b
```

The subset operator (`<=`) returns `True` if `set_a` is a subset of the `set_b` . Otherwise, it returns `False` . For example:

```
numbers = {1, 2, 3, 4, 5}
scores = {1, 2, 3}

result = scores <= numbers
print(result)  # True

result = numbers <= numbers
print(result)  # True
```

The proper subset operator (`<`) check if the `set_a` is a proper subset of the `set_b` :

```
set_a < set_b
```

For example:

```python
numbers = {1, 2, 3, 4, 5}
scores = {1, 2, 3}

result = scores < numbers
print(result)  # True

result = numbers < numbers
print(result)  # False
```

In this example, the set `numbers` is not a proper subset of itself, therefore, the `<` operator returns `False`.

Summary

- Set A is a subset of set B if all elements of the set A are also elements of the set B
- Use Set `issubset()` method returns `True` if a set is a subset of another set.
- Also, use the subset operator (<=) or the proper subset operator (<) to check if a set is a subset or proper subset of another set.