Python Symmetric Difference

Summary: in this tutorial, you'll learn how to find the symmetric difference of two or more sets in Python.

Introduction to the symmetric difference of sets

The symmetric difference of two sets is a set of elements that are in either set, but not in their intersection.

Suppose that you have the following `s1` and `s2` sets:

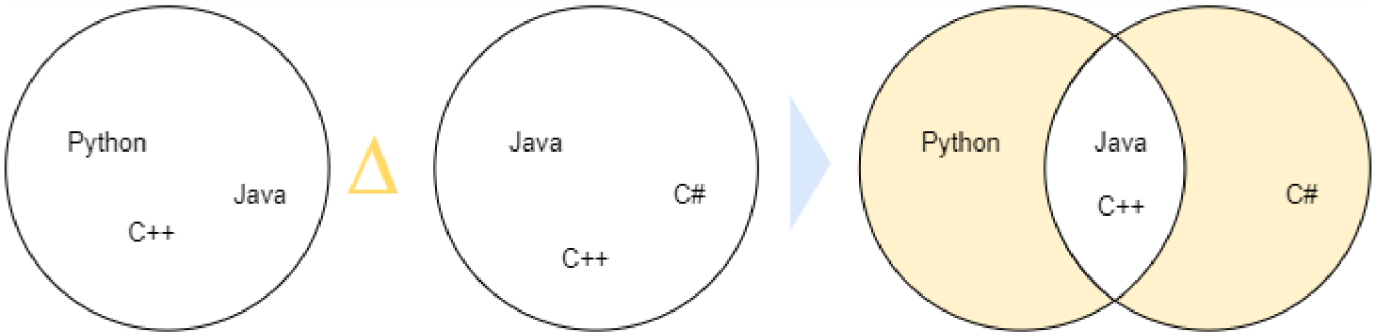
```
s1 = {'Python', 'Java', 'C++'}
s2 = {'C#', 'Java', 'C++'}
```

The symmetric difference of the `s1` and `s2` sets returns in the following set:

```
{'C#', 'Python'}
```

As you can see clearly from the output, the elements in the return set are either in `s1` or `s2` set, but not in their intersection.

The following Venn diagram illustrates the symmetric difference of the `s1` and `s2` sets:



In Python, you can find the symmetric difference of two or more sets by using the set `symmetric_difference()` method or the symmetric difference operator (`^`).

1) Using the symmetric_difference() method to find the symmetric difference of sets

The `Set` type has the `symmetric_difference()` method that returns the symmetric difference of two or more sets:

```
new_set = set1.symmetric_difference(set2, set3,...)
```

For example, the following finds the symmetric difference of the `s1` and `s2` sets:

```
s1 = {'Python', 'Java', 'C++'}
s2 = {'C#', 'Java', 'C++'}

s = s1.symmetric_difference(s2)

print(s)
```

Output:

```
{'C#', 'Python'}
```

Note that the `symmetric_difference()` method returns a new set and doesn't modify the original sets.

2) Using the symmetric difference operator(^) to find the symmetric difference of sets

Besides using the set `symmetric_difference()` method, you can use the symmetric difference operator (`^`) to find the symmetric difference between two or more sets:

```
new_set = set1 ^ set2 ^...
```

The following example shows how to apply the symmetric difference operator (`^`) to the `s1` and `s2` sets:

```
s1 = {'Python', 'Java', 'C++'}
s2 = {'C#', 'Java', 'C++'}

s = s1 ^ s2

print(s)
```

Output:

```
{'Python', 'C#'}
```

The symmetric_difference() method vs symmetric difference operator (^)

The `symmetric_difference()` method accepts one or more iterables (https://www.pythontutorial.net/python-basics/python-iterables/) that can be strings (https://www.pythontutorial.net/python-basics/python-string/) , lists (https://www.pythontutorial.net/python-basics/python-list/) , or dictionaries (https://www.pythontutorial.net/python-basics/python-dictionary/) .

If the iterables aren't sets, the method will convert them to sets before returning the symmetric difference of them.

The following example shows how to use the `symmetric_difference()` method to find the symmetric difference between a set and a list:

```
scores = {7, 8, 9}
ratings = [8, 9, 10]
new_set = scores.symmetric_difference(ratings)

print(new_set)
```

Output:

```
{10, 7}
```

However, the symmetric difference operator (`^`) only applies to sets. If you use it with the iterables which aren't sets, you'll get an error. For example:

```
scores = {7, 8, 9}
ratings = [8, 9, 10]
new_set = scores ^ ratings

print(new_set)
```

Error:

```
TypeError: unsupported operand type(s) for ^: 'set' and 'list'
```

Summary

- The symmetric difference of two or more sets is a set of elements that are in all sets, but not in their intersections.

- Use the set `symmetric_difference()` method or the symmetric difference operator (`^`) to find the symmetric difference of two or more sets.