

Python Set Union

If this Python Tutorial saves you
hours of work, please **whitelist it in**
your ad blocker 🙏 and

Donate Now

(<https://www.pythontutorial.net/donation/>)

to help us ❤️ pay for the web
hosting fee and CDN to keep the

website running.

Summary: in this tutorial, you'll learn how to union two or more sets by using the Python set `union()` or set union operator (`|`).

Introduction to the set union

The union of two [sets](https://www.pythontutorial.net/python-basics/python-set/) returns a new set that contains distinct elements from both sets.

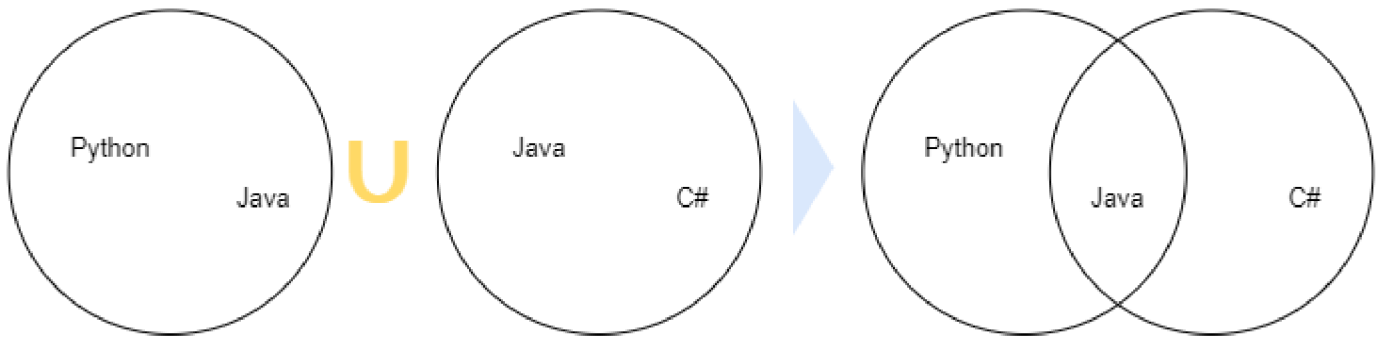
Suppose that you have the following sets:

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

The union of the s1 and s2 sets return the following set:

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

Typically, you use the Venn diagram to illustrate the union of two sets. For example:



Union sets using union() method

In Python, to union two or more sets, you use the `union()` method:

```
new_set = set.union(another_set, ...)
```

The following example shows how to union the `s1` and `s2` sets:

```
s1 = {'Python', 'Java'}
```

```
s2 = {'C#', 'Java'}
```

```
s = s1.union(s2)
```

```
print(s)
```

Output:

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

Union sets using the | operator

Python provides you with the set union operator `|` that allows you to union two sets:

```
new_set = set1 | set2
```

The set union operator (`|`) returns a new set that consists of distinct elements from both `set1` and `set2`.

The following example shows how to use the union operator (`|`) to union the `s1` and `s2` sets:

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

s = s1 | s2

print(s)
```

Output:

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

The union() method vs. set union operator

In fact, the `union()` method accepts one or more [iterables](https://www.pythontutorial.net/python-basics/python-iterables/), converts the iterables to sets, and performs the union.

The following example shows how to pass a [list](https://www.pythontutorial.net/python-basics/python-list/) to the `union()` method:

```
rates = {1, 2, 3}
ranks = [2, 3, 4]

ratings = rates.union(ranks)

print(ratings)
```

Output:

```
{1, 2, 3, 4}
```

However, the union operator (`|`) only allows sets, not iterables like the `union()` method.

The following example causes an error:

```
rates = {1, 2, 3}
ranks = [2, 3, 4]

ratings = rates | ranks
```

Error:

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

In conclusion, the `union()` method accepts the iterables while the union operator only allows sets.

Summary

- The union of two or more sets returns distinct values from both the sets.
- Use `union()` method or set union operator (`|`) to union two or more sets.
- The `union()` method accepts one or more iterables while the set union operator (`|`) only accepts sets.