

Set Comprehensions

It's time to learn about Set comprehensions in Python

WE'LL COVER THE FOLLOWING ^

- Wrapping Up

Set comprehensions are created in much the same way as dictionary comprehensions. Now a Python set is much like a mathematical set in that it doesn't have any repeated elements. You can create a normal set like this:

```
my_list = [1, 2, 2, 3, 4, 5, 5, 7, 8]
my_set = set(my_list)
print(my_set)
# {1, 2, 3, 4, 5, 7, 8}
```



As you can see from the example above, the call to set has removed the duplicates from the list. Now let's rewrite this code to use a set comprehension:

```
my_list = [1, 2, 2, 3, 4, 5, 5, 7, 8]
my_set = {x for x in my_list}
print(my_set)
# {1, 2, 3, 4, 5, 7, 8}
```



You will notice that to create a set comprehension, we basically changed the square brackets that a list comprehension uses to the curly braces that the dictionary comprehension has.

Wrapping Up

Wrapping Up

Now you know how to use the various Python comprehensions. You will probably find the list comprehension the most useful at first and also the most popular. If you start using your imagination, I am sure you will be able to find uses for all three types of comprehensions. Now we're ready to move on and learn about exception handling!