What else is for in loops

What's else in the for loops?

WE'LL COVER THE FOLLOWING

The **else** statement in loops only executes if the loop completes successfully. The primary use of the else statement is for searching for items:

```
my_list = [1, 2, 3, 4, 5]

for i in my_list:
    if i == 3:
        print("Item found!")
        break
    print(i)
else:
    print("Item not found!")
```

In this code, we break out of the loop when **i** equals 3. This causes the else statement to be skipped. If you want to experiment, you can change the conditional to look for a value that's not in the list, which will cause the else statement to execute. To be honest, I have never seen anyone use this structure in all my years as a programmer. Most of the examples I have seen are bloggers trying to explain what it is used for. I have seen several who use it to raise an error if an item is not found in the iterable that you were searching. You can read a fairly in depth article by one of the Python core developers here.

Wrapping Up

Hopefully at this point you can see the value in Python loops. They make repetition easier and pretty easy to understand. You will likely see the **for** loop much more often than the **while** loop. In fact, we are going to look at another way **for** loops are used in the next chapter when we learn about comprehensions! If you're still not quite sure how all this works, you may want to re-read this chapter before continuing.