





The if-else Statement

This lesson showcases the main properties of an 'if-else' statement.

We'll cover the following

- Structure
- Benefits of if-else
- Conditional Expressions

In the previous lesson, we were introduced to the if statement, which executes a block of code when a condition is satisfied.

What if we wanted to execute a different set of operations in case the if condition turns out to be False?

That is where the if-else statement comes into the picture.

Structure

The if-else statement looks something like this:

There's nothing too tricky going on here. If the **condition** turns out to be False, the code after the else: keyword is executed.

Hence, we can now perform two different actions based on the condition's value.

The lse keyword will be on the same indentation level as the if keyword. Its body will be indented one tab to the right just like the statement.

Here's the if-else statement in action:

```
num = 60

if num <= 50:
    print("The number is less than or equal to 50")
else:
    print("The number is greater than 50")</pre>
```

Benefits of if-else

The example above could also be written with two if conditions:

```
num = 60

if num <= 50:
    print("The number is less than or equal to 50")

if num > 50:
    print("The number is greater than 50")

\[ \bigcap \]
\[ \bigcap \]
\[ \bigcap \]
\[ \bigcap \]
```

However, for the second if, we have to specify the condition again. This can be tricky when dealing with complex conditions. The else statement automatically handles all the situations when the if fails.

Do keep in mind that the else statement cannot exist on its own. It is merely a counterpart of the if statement. It can still contain its own nested if or if-else statements. We'll leave that as an exercise for you to try on your own.

≡Cnditional Expressions



Conditional expressions use the functionality of an if-else statement in a different way.

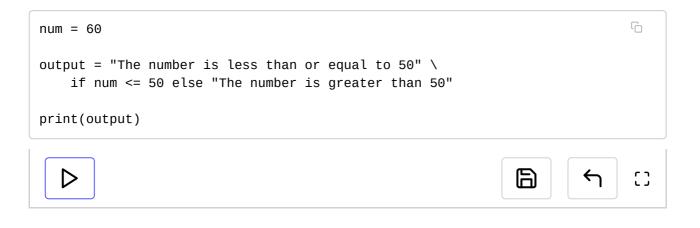
The expression returns an output based on the condition we provide. This output can be stored in a variable.

A conditional expression can be written in the following way:

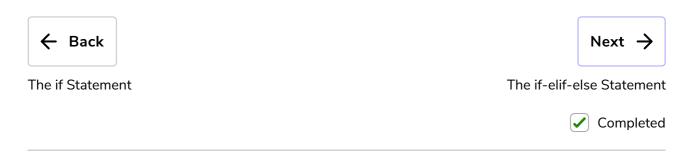
```
output_value1 if condition else output_value2
```

If the if condition is fulfilled, the output would be output_value1. Otherwise, it would be output_value2.

Let's refactor the previous if-else statement into a conditional expression:



In the next lesson, we'll build upon what we've learned in order to understand the if-elif-else statement.





Ask a Question

 $(https://discuss.educative.io/tag/the-if-else-statement_conditional-statements_learn-python-3-from-scratch)\\$



