The if-elif-else Statement

This lesson highlights the main properties of the `if-elif-else` statement.

We'll cover the following

The `if-else` statement handles two sides of the same condition: `True` and `False`. This works very well if we're working with a problem that only has two outcomes.

However, in programming, it isn't always a `True` or `False` scenario, and a problem can have multiple outcomes.

This is where the `if-elif-else` statement shines. It is the most comprehensive conditional statement because it allows us to create multiple conditions easily.

The `elif` stands for **else if**, indicating that if the previous condition fails, try this one.

Structure

The `if` and `else` blocks will remain the same. The `elif` statement comes in between the two.

Let's write an `if-elif-else` statement which checks the state of a traffic signal and generates the appropriate response:

```python
light = "Red"

if light == "Green":
    print("Go")

elif light == "Yellow":
    print("Caution")

elif light == "Red":
    print("Stop")

else:
    print("Incorrect light signal")

```

Now, our conditional statement caters to **all** possible values of `light`.

Try changing the value and see how the response changes.

Multiple `elif` Statements

This is the beauty of the `if-elif-else` statement. We can have as many `elif`s as we require, as long as they come between `if` and `else`.

Note: An `if-elif` statement can exist on its own without an `else` block at the end. However, an `elif` cannot exist without an `if` statement preceding it (which naturally makes sense).

Let's write a piece of code that checks whether the value of an integer is in the range of `0-9` and prints the word in English:

```
num = 5

if num == 0:
    print("Zero")
elif num == 1:
    print("One")
elif num == 2:
    print("Two")
elif num == 3:
    print("Three")
elif num == 4:
    print("Four")
elif num == 5:
    print("Five")
elif num == 6:
    print("Six")
elif num == 7:
    print("Seven")
elif num == 8:
    print("Eight")
elif num == 9:
    print("Nine")
```

An important thing to keep in mind is that an `if-elif-else` or `if-elif` statement is not the same as multiple `if` statements. `if` statements act **independently**.

If the conditions of two successive `if` s are `True` , both statements will be executed.

On the other hand, in `if-elif-else` , when a condition evaluates to `True` , the rest of the statement's conditions are not evaluated.

We'll understand this better through an example:

if | if-elif-else

```
num = 10

if num > 5:
    print("The number is greater than 5")

elif num % 2 == 0:
    print("The number is even")

else:
    print("The number is odd and less than or equal to 5")
```

As we can see, in the `if` tab, all the statements are computed one by one. Hence, we get multiple outputs.

In the `if-elif-else` tab, since the first condition holds true, all the others are discarded. This proves to be more efficient in terms of code performance.

At this point, we know pretty much everything about the behavior and purpose of conditional statements. Test your concepts with a quiz in the next lesson followed by some fun exercises.

After that, we'll begin our discussion on **functions**.

← Back
The if-else Statement

Next →
Quiz

✓ Completed

Report an Issue

Ask a Question
(https://discuss.educative.io/tag/the-if-elif-else-statement__conditional-statements__learn-python-3-from-scratch)