Conditionals in Python

Conditionals are an important part of programming, as they allow the program to perform different actions depending on the situation that exists when the program is run. The actions of the program are dependent on some logical expression, that likely includes arithmetic comparisons and/or boolean logic.

The if Statement

The simplest conditional is the **if statement**. If a particular condition is true, then some code is executed. Otherwise, it is not. Consider the following Python code:

if hungry:

    print("Have a snack!")

Note the structure of the if statement on line 1. It starts with the \color{red}{\verb|if|}if keyword. Then you have a condition that evaluates to either \color{red}{\verb|True|}True or \color{red}{\verb|False|}False. The condition can be any Python expression that will evaluate to \color{red}{\verb|True|}True or \color{red}{\verb|False|}False. Finally, the line ends with a colon. This colon is required and it indicates to Python that what follows is the body of the if statement. The body of the if statement can contain any Python code and can be as long as necessary. But, all lines in the body of the if statement must be indented. As with functions, the convention is that the body should be indented by 4 spaces. Once the indentation stops, the body of the if statement is over and the subsequent code will execute regardless of the value of the condition.

So, in this code, if \color{red}{\verb|hungry|}hungry is a boolean, then this program will print a message if \color{red}{\verb|hungry|}hungry is \color{red}{\verb|True|}True and will do nothing if it is \color{red}{\verb|False|}False. This allows you to write programs which will only print this message if the "user" is actually hungry.

## The else Clause

Sometimes, you would like to do one thing if the condition is true and something else if it is false. Consider the following program:

if hungry:

    print("Have a snack!")

else:

    print("Save your snack for later!")

An if statement can also include optional clauses following the body of the if statement. You can optionally include an **else clause** which will execute if the condition of the if statement evaluates to \color{red}{\verb|False|}False. The else clause starts with the keyword \color{red}{\verb|else|}else followed by a colon. After that, the body of the else clause must be indented, again with 4 spaces by convention. As with the if statement body, the body of the else clause can contain any Python code and ends when the indentation stops.

Note that either the body of the if or the body of the else will execute, but not both. If \color{red}{\verb|hungry|}hungry is \color{red}{\verb|True|}True, then the code on line 2 will execute. If it is \color{red}{\verb|False|}False, then the code on line 4 will execute.

## The elif Clause

Sometimes you want to build up more complicated conditional expressions. Consider the following code:

if hungry and thirsty:

    print("Go have lunch!")

elif hungry:

    print("Have a snack!")

elif thirsty:

    print("Have a drink!")

else:

    print("Save your food and drinks for later!")

You can also add **elif clauses** (short for "else if") to an if statement. Each elif clause includes a condition, exactly like the initial if statement. An elif clause must directly follow either the initial if statement or another elif clause and has the following structure: the \color{red}{\verb|elif|}elif keyword, an expression that evaluates to \color{red}{\verb|True|}True or \color{red}{\verb|False|}False, and a colon. The body of the elif clause is then indented by 4 spaces and ends when the indentation ends.

The way to understand the elif clause is that if none of the conditions above it have evaluated to \color{red}{\verb|True|}True, then the clause will execute. If the condition of the elif clause is \color{red}{\verb|True|}True, then the associated body will execute, and no other bodies in the \color{red}{\verb|if|}if/\color{red}{\verb|elif|}elif/\color{red}{\verb|else|}else construct will execute. If the condition of the elif clause is \color{red}{\verb|False|}False, then the associated body will not execute and any following \color{red}{\verb|elif|}elif/\color{red}{\verb|else|}else clauses will be checked.

There can be as many elif clauses as you would like, but they must follow a single if statement. They can optionally be followed by a single else clause at the end. The body of the final else clause, if it exists, executes only if none of the previous conditions were True.

## Arithmetic Comparisons in Conditionals

Arithmetic comparisons are often used within the conditions of **if statements** (and associated **elif clauses**). Consider the following code:

if (value > -10) and (value < 10):

    # do something

elif (value <= -10):

    # do something different

elif (value >= 10):

    # do another different thing

This program has three possible paths, depending on the value of the variable value. If it is between -10 and 10 (but not exactly -10 or 10), then the body on line 2 will be executed. If, instead, it is less than or equal to -10, the body on line 4 will be executed. Finally, if it is greater than or equal to 10, the body on line 6 will be executed. No else clause is necessary here, since all possibilities have been covered. Note, however, that the final elif clause could be replaced with an else clause, since if the code gets to that point, value must be greater than or equal to 10.