Conditional statements in Python

Previously, we discussed how to store different data types in variables.
Now we'll begin to move into the concept of automation, so we can create exciting actions with code.
Automation is the use of technology to reduce human and manual effort to perform common and repetitive tasks.
It allows computers to do these tasks for us so that we may get back more time in our lives to do other activities.
Conditional statements are important for automation.
A conditional statement is a statement that evaluates code to determine if it meets a specified set of conditions.
The keyword if is important in conditional statements.
if starts a conditional statement.
After this keyword, we then specify the condition that must be met and what will happen if it is.
We use if statements every day.
For example, if it's cold outside, then we'll wear a jacket.
Or if it's raining, we'll bring an umbrella.
if statements are structured with the condition we want to evaluate and the action that Python will perform if this condition is met.
Python always evaluates if the condition is True or False, and if it's True, it performs the specific action.
Let's explore an example of this.
We'll instruct Python to print an "account locked" message anytime the failed log-in attempts are greater than five.
Our keyword if tells Python to start a conditional statement.
After this, we indicate the condition we want to check for.
In this case, we're checking if the user has more than five failed log-in attempts.
Notice how we're using a variable called failed_attempts.
In our complete code, we will have assigned a value to failed_attempts prior to this if a statement.
After this condition, we always place a colon.
This signals that what follows is what we want to happen when the condition is met.
In this case, when the user has more than five failed log-in attempts, it prints a message that the account is locked.
In Python, this message should always be indented at least one space in order to execute only when the condition is true.
It's common to call this first line the "header" and to call the actions that happen when the condition is met the "body." This condition was based on a variable being greater than a specific number, but we can define our condition using a variety of operators.

For example, we can also check if something is "less than" a specified value, or we can check if it's "greater than" or "equal to" or "less than or equal to" the value.
We can also compare if something is equal to a value.
When we do this inside a conditional, we need to use a special syntax.
It's not just the equals sign, but a double equals.
The double equals sign is an important operator often used in conditional statements.
A double equals evaluates whether two objects match.
It assigns a Boolean value of True when they match and False when they don't.
There's one more operator we should discuss.
An exclamation mark followed by an equals sign represents the condition of "not equal." This operator for "not equal" evaluates whether two objects are different.
It assigns a Boolean value of True when they don't match and False when they match.
Let's more closely investigate an example that uses the double equals sign.
We'll focus on an example that prints an "updates needed" message when a particular operating system is running.
Here, we've created a condition that checks if a device's operating system matches a specific string that identifies this operating system.
To do this, we'll need to use the double equals sign in our condition.
When it matches, our program will print a message that there are updates needed.
The operating_system variable is on the left of the double equals sign.
The string "OS 2" is on the right.
If the condition evaluates to True, it performs the action that is indented in the next line of code.
Here, if the operating_system is OS 2, it will print "updates needed." If it's False, the message will not print.
Notice how this line is indented.
This tells Python that the task depends on the if statement evaluating to True.
Now let's write code that incorporates this conditional and get the results.
Before we write the conditional statement, we need to assign a value to our operating system variable.
We'll make this value the same as the operating system that we'll check for in the conditional.
Next, we'll write the condition for our if statement and use the double equals sign to check if the operating_system variable is equivalent to OS 2.
Now we'll type the action that we'll execute if the condition on the previous line evaluates to True.
We'll tell Python to print an "updates needed" message.
Since we set our operating_system variable to OS 2, the print statement will execute.
Okay, let's run this.
As expected, it printed "updates needed" because the value assigned to the operating_system variable was equal to OS 2.
Sometimes, we want our conditional statements to execute another set of instructions in the event our first condition isn't True.
In our example, not being True means that the device is running an operating system other than OS 2.
This is when we need to incorporate the else keyword into our conditional statements.
else precedes a code section that only evaluates when all conditions that precede it within the conditional statement evaluate to False.
else statements always follow an if statement and end in a colon.
Let's use our previous conditional and add an else statement to it.

We've included the same if statement, but this time, we set the operating system variable to contain a different operating system, OS 3.
Because this doesn't match the value in the condition of the if statement, the "updates needed" message won't print.
But we can add an else statement and tell it to do something else instead.
We type the else keyword followed by a colon.Then we indent the next line and tell it to print a "no updates needed" message.
When we'd run this code, it processes the else statement after the if statement.
Since our if statement will evaluate to False, it then moves on to the "else" instruction.
Let's try it.
As expected, it only prints the message "no updates needed." Great work!
Now we've covered how to use if and how to use else.
Using conditional statements allows you to incorporate logic into your code.

Revision #1
Created 9 December 2023 07:33:58 by naruzkurai
Updated 19 December 2023 03:38:35 by naruzkurai