Working with Conditionals

Lab overview

A section of code that compares two pieces of information is called a *conditional statement*. You can use conditionals to create different paths through the program. Using comparative operators, you will write a program that makes decisions.

In this lab, you will:

- Use the `if` statement
- Use the `else` statement
- Use the `elif` statement

Exercise 1: Working with the if statement

In this exercise, you will edit a Python script to ship packages.

11. From the navigation pane of the IDE, choose the **.py** file that you created in the previous *Creating your Python exercise file* section.
12. Use the `input()` function to get information from the user:

    ```
    userReply = input("Do you need to ship a package? (Enter yes or no) ")
    ```

13. Use the `if` statement to print a response.

    The statements in an `if` statement are one tab indented from the `if` statement. In other programming languages, brackets are often used to indicate the start and end of a logic block, but Python uses spacing:

    ```
    if userReply == "yes":
        print("We can help you ship that package!")
    ```

    Note: The `==` symbol is a comparative operator. It means *is equal to*.

14. Save and run the file.
15. At the prompt, enter `yes` and press ENTER.
16. Confirm that you see a response.
17. Run the file again.
18. At the prompt, enter `no` and press ENTER. Confirm that the program exits and nothing id displayed.

Exercise 2: Working with the else statement

To improve customer service, it would be nice to provide a reply even if the user doesn't want to ship a package. In this exercise, you will improve the Python script by using the `else` statement:

19. To handle the condition where the user doesn't want to ship a package, use the `else` statement:

    ```
    else:
        print("Please come back when you need to ship a package. Thank you.")
    ```

20. Save and run the file.
21. At the prompt, enter `no` and press ENTER.
22. Confirm that you see a response.
23. Run the file again.
24. At the prompt, enter `yes` and press ENTER.
25. Confirm that you see a response.

Exercise 3: Working with the elif statement

In this exercise, you will improve the Python script by offering the user additional services. When you have multiple conditions, you can use the `elif` statement, which is short for *else-if*.

> **Note:** The `elif` statement always comes after an `if` statement and before the `else` statement.

26. In the Python script, enter the following code:

```
userReply = input("Would you like to buy stamps, buy an envelope, or make a copy? (Enter stamps, envelope, or copy) ")
if userReply == "stamps":
    print("We have many stamp designs to choose from.")
elif userReply == "envelope":
    print("We have many envelope sizes to choose from.")
elif userReply == "copy":
    copies = input("How many copies would you like? (Enter a number) ")
    print("Here are {} copies.".format(copies))
else:
    print("Thank you, please come again.")
```

27. Save and run the file.
28. At the prompt, enter `no` and press ENTER.
29. Confirm that you see a response.
30. At the prompt, enter `stamps` and press ENTER.
31. Confirm that you see a response.
32. Run the file again.
33. At the prompt, enter `yes` and press ENTER.
34. Confirm that you see a response.

35. At the prompt, enter `envelope` and press ENTER.
36. Confirm that you see a response.
37. Run the file again.
38. At the prompt, enter `no` and press ENTER.
39. Confirm that you see a response.
40. At the prompt, enter `copy` and press ENTER.
41. Confirm that you see a response.
42. At the prompt, enter `2` and press ENTER.
43. Confirm that you see a response.

> **Note:** The `if`, `elif`, and `else` statements allow only one path to run at a time. The program doesn't check the other statements after it finds a condition that is true.

As you can see, each time through the program had slightly different results. These differences demonstrate the power of conditionals.

Congratulations! You have written a Python script that uses `if`, `elif`, and `else` statements.