KEY_Lesson09_Conditionals

October 23, 2019

1 Conditional Logic

In the last lesson, we learned about logic with booleans. Booleans can be used to determine if a certain line of code should be run. How would we do this? This is where we can use something called an if statement.

If statements follow the following formula:

```
if [condition]:
    [code]
```

The **condition** is a boolean, and the code under the condition only runs if the condition is true. These are called **if statements**

For example, say we had a list that we wanted to print out, but we don't want to print the list if it's too long. We could run the following:

[1]:
```
# create a list called my_list containing the numbers 0 to 7
my_list = [0, 1, 2, 3, 4, 5, 6, 7]

# write an if statement that only prints my_list if it has less than 10 items
if len(my_list) < 10:
    print(my_list)
```

```
[0, 1, 2, 3, 4, 5, 6, 7]
```

Let's test the conditional statement on its own to see what the result is:

[2]:
```
# print the boolean value of the condition in the if statement above
len(my_list) < 10
```

[2]:
```
True
```

We see that the condition is `True`, the code below runs.

What if we changed the condition above? How would this change our code?

[4]:
```
# write an if statement that only prints my_list if it has 10 or more items
if len(my_list) >= 10:
    print(my_list)
```

Nothing happened! Let's see why this is.

[5]:
```
# print the boolean value of the above if statement
len(my_list) >= 10
```

[5]:
```
False
```

In this case, the condition is `False`, so the code within the conditional did not run.

What if we wanted to run one block of code if the condition is `True`, and a different line of code if the condition is `False`? In other words, what if we wanted to run one block of code if our list has less than 10 items, and another block of code if the list has more than 10? This is where we use `if/else`.

An `if/else` looks like this:

```
if [condition]:
  [command 1]
else:
  [command 2]
```

The first two lines should look familiar; it's an if statement, which we just used! But after the if statement, we write `else:` and then some more python code. The code that comes after `else:` will run if the value of `condition` is false.

Let's see how this works! Let's write code that prints `my_list` if it has less than 10 items, and otherwise prints a message letting us know the list is too long to print.

[6]:
```
# write an if/else statement that prints my_list if my_list has less than 10 items
# and if my_list has 10 or more items, prints a message
if len(my_list) < 10:
    print(my_list)
else:
    print("The list is too long to print!")
```

```
[0, 1, 2, 3, 4, 5, 6, 7]
```

Because the list has less than 10 items, the list printed. But let's say we added more items to the list, making it longer than 10 items.

[7]:
```
# append the numbers 8, 9, 10, and 11 to my_list
my_list.append(8)
my_list.append(9)
my_list.append(10)
my_list.append(11)
```

[8]:
```
# print the length of my_list
len(my_list)
```

[8]:
```
12
```

Now, the length of my_list is above 10. How would the above line of code run now?

[9]:
```
# copy and paste the if/else statement we just wrote
if len(my_list) < 10:
    print(my_list)
else:
    print("The list is too long to print!")
```