Lesson

Conditionals

Section 3(3)

Completed

Mark as Incomplete

Why Are Conditionals Necessary?

The previous models gave us a solid foundation on how to manipulate data, store and manage different data types, and access the more flexible data structures. However, we can use this data structure to ask questions about the data and make decisions based on the answer. Conditionals are the tools for making decisions that result in the path our data takes and what is changed in the computer space.

A conditional allows us to check if certain conditions are being met before we run certain parts of the program. They are often referred to mean one possible, or "if-statement." or "if-else," and there is a very great number of ways to use conditionals. Here is a simple example of a conditional statement in the English language:

```
If the weather is sunny, I will wear a tank top. If it doesn't, I will wear a hoodie.
```

Conditionals in the code will look similar, but with the following syntax:

```
weather_sunny = True
if weather_sunny:
    print("Wear a tank top.")
else:
    print("Wear a hoodie.")
```

In the example, when we set the variable `weather_sunny` to `True`, the `if` statement checks to see if `weather_sunny` is `True`. If it is, the program prints "Wear a tank top." If it is not, the program prints "Wear a hoodie."

Notice that indent after the `if` statement. We use an indent to signal to Python that it is part of a single sequence of code. The single rule is to maintain the same indent for all code that is in the same sequence. We can think of this as "if-indent" or "if-else-indent." Or, we can think of it as "if-indent" or "if-else-indent." Or, we can think of it as "if-indent" or "if-else-indent." Or, we can think of it