Lesson 2: Conditional Logic

Recap of Last Lesson:

- *What is code?*
    - We learned that code is the way we tell computers what to do. Computers take your code and will run one line of code at a time, from top to bottom.
- *How do we write code?*
    - We can write code using any text editor. In our lessons we'll use Visual Studio Code, which makes the code look nice with highlights and lets us run it. Anytime we make changes to our code and would like to run it, we have to remember to save it first.
- *How do we run code?*
    - We run code by telling the computer what code we'd like to run. When we run python, we enter `*python <name of python file>*` in a terminal. The computer then looks at the specified file and runs it.
- *Comments in code*
    - We can write comments in code. In python that's anything (even multiple lines) between 3 ticks or 1 line starting with a # (i.e. '''COMMENT''' or # one line comment). The computer will see your comments and know it doesn't need to treat it like code, and will just skip them.
- *Variables*
    - In programming we use variables as a way to hold some information. We learned variables can have different types, such as integers, decimal point numbers (floats), and strings (words/sentences).
    - We can create variables in Python by thinking of a name, and setting a value to that name. For example, `my_variable = 0` creates a variable called my_variable and gives it the value 0.
    - We can do operations with variables, i.e. addition, subtraction, multiplication, etc.
    - We have to sometimes convert a variable to a different type. For example, if x="5" and we need to use x as a number, we might say x=float(x), to convert x to its number representation.
- *Input and Output*
    - In python we use print() to print something to the screen.
    - We use input() to get some input from a user.

Homework from Last Lesson:

Write a Python program that asks the user for the length and width of a rectangle, calculates the area and perimeter of that rectangle, and displays it 10 times to the user.

Motivation for Conditional Logic

What if I asked you to modify the program so that if the user entered the same number for length and width the program said “You entered a square!”?

Right now, we don’t know how to do that because our code always runs every line we write. If we put a print(“You entered a square!”) in our code, that will get printed every time the program runs, even if the user didn’t give us a square.

On a Boeing airplane, the software needs to lower the landing wheels only when the plane is coming in for landing and the pilot presses a certain button.

In a FIFA video game, the score only increases when one team scores a goal.

Your phone should only launch an app if you clicked on that app.

We need some way to tell the computer to run certain code sometimes.

If Statements

We use if statements as a way to control when some code should run. In every if statement we have a **condition**. The condition is something that the computer can determine is either **true** or **false.** If the computer determines the condition is true, it will start running the code inside of the if statement. If the condition is false, the computer skips over the code inside of the if statement.

We use these **operators in conditions**:

>	greater than
<	less than
<=	less than or equal to
>=	greater than or equal to
==	equal to
!=	not equal to
and	Like English. `a and b` is True if a is True and b is True.
or	Like English. `a or b` is True if a or b are True.
not	Like English. `not True` is False and `not False` is True.

There are other ways in Python that we test for conditions, but that’s enough for now!

Activity: Can you guess what the following code will do? Try creating a file called if_statements.py and running it.

```
'''
If Statements
'''
cash_balance = 99.43
price_of_headphones = 150.99
if cash_balance >= price_of_headphones:
    print("You can afford the headphones.")
```

NOTE: Python knows what is "inside of the if statement" because of tabs. If a line is tabbed-in from the if condition, Python thinks that is part of the statement.

```
if True and False:
    print("In the if statement.")
print("Not in the if.")
```

Else Clauses

What if we want the computer to tell the user "You can't afford those headphones!" if they don't have enough money. **One way to do that is write another if statement.**

```
if cash_balance >= price_of_headphones * 1.13:
    print("You can afford the headphones.")
if cash_balance < price_of_headphones:
    print("You cannot afford the headphones.")
```

This works, for now. What if we want to include the cost of tax? We have to remember to change both if statements now otherwise the code won't work.

We use an else clause/statement when we have code that should run whenever the if statement before is False.

```
if cash_balance >= price_of_headphones:
    print("You can afford the headphones.")
else:
    print("You cannot afford the headphones.")
```

Activity: Create a copy of your homework code (rectangle area and perimeter calculator). Modify the program so that **if the user enters the same number for length and width, it will print "That's a square!", and if they don't, it will print "That's a rectangle!"**. The program should still calculate the area and perimeter like it did before.

Else-If Statements

Up until now, we can only test 1 condition, do something if that's true, and if we want to do something else when it's not true. We can think of times when that might not be good enough for us….

Imagine code that picks a restaurant to order takeout from (for a family). The code might need to use information like:

- Who is home for dinner?
- Who likes what restaurant?

We want to be able to do something like:

if Dad is home for dinner:

consider ordering from restaurants [A, B, C]

if Dad isn't home for dinner, but Mom is:

consider ordering from restaurants [A, D, F]

if neither Dad or Mom is home for Dinner:

consider ordering from restaurants [X, Y, Z]

In code we do that by **testing multiple conditions** and doing different things based on which ones are true. We use an **else-if condition which will get evaluated if the conditions above it were False.** In Python else-if is **elif**.

```
if dad_is_home_for_dinner:
    result = "McDonalds"
elif mom_is_home_for_dinner:
    result = "Harveys"
else:
    result = "Baskin Robbins"
```

Activity: Write a Python script **that uses if, elif, and else** and asks the user to pick a mode (1, 2, or 3). If they selected 1, print ("MODE 1"), if 2, print ("MODE 2"), etc.

Nested If Statements

We can also put if statements within if statements!!! We use tabs like we did before to tell Python/the computer which lines of code are within what statement.

```
if mom_is_home_for_dinner:
    if cash_balance >= price_of_headphones:
        print("BLAH")
    else:
        print("WOOHOO")
elif dad_is_home_for_dinner:
    print("Something else")
else:
    print("Another thing!")
```

Homework Activity: Modify the Python script you created above that runs different "modes".

If the user enters 1, the program runs "Apple Picking" mode, **if they instead enter a 2,** the program runs a "Tax Calculator" mode**, if they enter a 3,** it runs a "Rectangle Area Calculator" mode.

In Apple Picking mode, you ask the user for a number of apples, and print how much money they can make from the apples (assume they could sell 1 apple for $1.50). If they can make more than $50, tell them "You are rich!", if they can make more than $25 but less than $50, tell them "You can make a decent amount", otherwise don't print anything besides how much money they can make.

In Tax Calculator mode, you ask the user for the price of a product, and print how much tax they will pay, and what the grand total cost would be (assume a 13% tax rate).

In Rectangle Area Calculator mode, you run the program you wrote last week - ask the user for length and width, and give them the area and perimeter.