Comp 143-302

PAL Instructor: Alex Dasneves

March 30th 2023

Recap

- What is a code block?

```
for i in range(10):
    for j in range(20):
        for k in range(200):
            print(i * j)
```

- What do we use comments for?
  To make notes, and to make the code easier to read. (#)
- What is an escape character? Give 2 examples, and list what they do.
  1. \n — Goes on another line.
  2. \t — Indents one tab stop. (Not a set number of characters)

Conditions and Booleans

Throughout this class we have been creating programs that do one, and only one thing. However, by comparing variables, we can make programs that can do many things.
An example of this is making a menu system.

```
Please choose an option:
[1] Add Student
[2] Remove Student
[3] Add Class
[4] Remove Class
[5] Add Student to Class
[6] Exit
<code>
user_menu_choice = input()
if user_menu_choice == "1":
    # Add student
</code>
```

Consider a menu above. We can make one program that will be able to do all these things (and more) using conditions.

What is a boolean?

A boolean is a True or False value.
A bool cannot be in-between these 2 states. Therefore, we can use them as a way to signal to the program whether something has happened or not.

What is a condition?

By harnessing the power of booleans, we can create 'conditions'.
Conditions are statements that are created by the programmer which evaluates actions in the program and returns a boolean.
By using these booleans, we can then have the program 'branch' into other tasks or proceedures.

For example, with the menu example above, we can take the input from the user, and compare it against known values.

When we want to do some kind of branching, we use the 'if' keyword
So, now that we have the correct control structure, lets move into making some conditions.

Condition Crafting

When we make conditions, we compare 2 things to each other. If the statement is logically correct, we get 'True', and otherwise, we get 'False'.
When we craft conditions, we use sets of symbols to tell the computer to compare.

```
==          Equal To
>           Greater Than
<           Less Than
>=          Greather Than OR Equal To
<=          Less Than OR Equal To
!=          Not Equal To
```

When we use our control structure, we use the following syntax:

```
if <condition>:
    <code goes here>
```

The code in the indent block will run iff (If and only if) the condition is true.

Indefinite Loops

Now that we know what a definite (for) loop is, we can talk about indefinite loops.
Indefinite loops, also known as condition controlled or boolean controlled loops are common in python when we want to run code until we get a certain result.
We can use the 'while' keyword to repeat code until a certain condition is met.
Here is an example:

```
user_str = input("Please choose a number")
while user_str != "q":
    print("Invalid Input!")
    user_str = input("Please choose a number")
print("Horray!")
```

This is also referred to as a 'pre-test' loop. This is because our condition is checked before our code is ever run.
We can also put 'literals' into these condition spots.
For example:

```
while True:
    <Do Something>
```

Since the value of 'True' is a literal, and by definition, never changing, we can say that this code will run forever (Or until the program is exited by the user).

We can do the same idea with an 'if' statement, however, this is fairly redundant, since an if statement with a literal 'True' will always run. Therefore, there is no need for an if statement.

```
if True:
    print("Hello!")

print("Hello!")
```

Mini-Exercise 1

Create a basic 4-function calculator (+, -, *, /)

Prompt

Create a calculator that takes in a number, then an operator, then another number.
Using if statements, change what is printed out to the user
Example test run of program:

```
Please enter a number: 5
Please enter an operator (+. -, *, /): *
Please enter a number: 6
5 * 6 is 30.
```

Please demonstrate whatever code you have before you leave.