Lesson 1: Getting Started with Python

Python is a scripting language that allows you to quickly and easily make programs.
In this lesson, you'll learn how to create and run a program to display text on the screen.

PART 1: Opening a New Program

Before we can start writing programs, we first have to open Python's *IDLE* editor.

Your instructors will show you how to open the IDLE editor. Go to File, New File to begin coding.

Once you've opened Python's IDLE editor, you should save your program by finding the File tab in the upper left corner, and **clicking Save As**.

Try to save your program often to avoid losing your changes!

PART 2: Printing Text to the Screen

First, let's learn how to make messages appear on the screen using Python.
In Python, we can tell our program to do lots of things by using different *functions*.
For example, to show messages, we can use the `print()` function.

The `print()` function tells the computer to print the text inside the parentheses to the *Python Interpreter* (also called the *shell*).

Try typing the following code into your program:

```
print("Here's my pet bird!")
```

Make sure you remember the *end quotes* " and *end parenthesis*)

PART 3: Testing your Program

Find the Run tab at the top of the IDLE editor, and click Run Module.

If your code worked, you should see your message appear in the Python shell.

Python uses the *quotation marks* " in the print function to tell where the message starts and stops, so the shell won't print any quotes when you run your program.

If an error appeared, check to make sure you typed everything correctly.

Functions in Python are case-sensitive, so make sure you use lower and upper cases correctly in your code.
Print() is different than *print()*

PART 4: Printing Multiple Lines

In Python, you can print as many lines as you want by using the `print()` function.
If you print multiple lines in a row, you can make interesting drawings with your text.

Gray lines in the lesson are ones that you have already typed.

Return to the Code Window in Python.

Type the following code into your program after the first line. Use the spacebar in the second line to create a space after the slash symbol:

```
print("Here's my pet bird!")
print("(o<")
print("/ }")
print(" L")
print("")
```

`print("")` will print a blank line, which is like hitting the Enter key

Run your program again - you should now see a bird appear in the shell as well!

PART 5: Creating a Robot

Now let's try to create a robot using the `print()` function!
This time, we'll need to be more careful about how we type the code in to our program.

Try typing the following code in at the end of your program:

```
print(" L")
print("")
print("Meet my robot, Max!")
print(" [@@] ")
print("/|__|\\")
print(" d  b")
print("")
```

Make sure to type in the second *backslash* `\` before the end quotes!

When using the `print()` function in Python, you usually have to give it a *string*.

In Python, a string is anything between two quotation marks, "like the text here" .
Strings represent messages, which can contain either letters, numbers, or symbols.

A *backslash* `\` is a special symbol in strings, so in order to print one, you should type two in a row, like this: `print("This will print one backslash here -> \\")`

`print("\")` will usually cause an error, because \" tells the print() function to show a single " so the string can't find an actual end quote.

PART 6: BONUS: Printing More Drawings

If you've finished making Python print your bird and robot drawings, try either coding some of these drawings, or create your own drawing! If you're not sure what to make, try making a pine tree!

```
  Fish
o<
    o<
 o<
```

```
 Owl
{o,o}
/)__)
 ' '
```

```
  Rabbit
 _     _
( \___/ )
(= '.' =)
('')_('')
```

Creating drawings from keyboard characters is called *ASCII art*. In the early days of the computer, it was easier to make images using text rather than graphics.

PART 7: BONUS: Create an Emoticon Program

If you're done making extra drawings, try making a program that displays emoticons!

An emoticon expresses emotion by using text, like this one, which looks happy :-)
Some other emoticons can looks like this (-_-) zzz or this (*_*)

Can you create your own emoticons using Python?