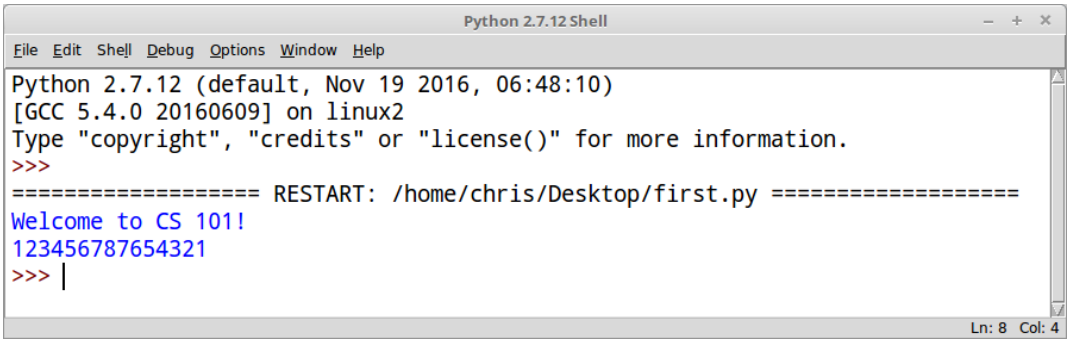
Activity 7: Hello, Python!

"By the way, the language is named after the BBC show 'Monty Python's Flying Circus' and has nothing to do with reptiles. Making references to Monty Python skits in documentation is not only allowed, it is encouraged!" (Source: https://docs.python.org/2/tutorial/appetite.html)

Model 1 Using IDLE

"IDLE is Python's Integrated Development and Learning Environment. It has two main window types: the Shell window and the Editor window. It is possible to have multiple editor windows simultaneously." (Source: https://docs.python.org/2/library/idle.html)



Questions (15 min) **Start time:** _____

1. Which of the two screenshots in Model 1 is the Shell window? Which is the Editor window?

The top window is the editor, and the bottom window is the shell. You can tell based on the window titles.

2. Explain the terms "Editor" and "Shell" based on what you learned previously in the course.

An editor is a simple program for writing plain text files. A shell (or terminal) is an interface for running commands.

3. What is the name of the file in the editor? What directory is it saved in?

The name is first.py, and the directory is Desktop.

4. Explain the Python code in the editor window. What does each line do?

The first line is a comment, the second line displays a message, the third line is blank, the fourth line is another comment, the fifth line creates a variable named x, and the sixth line displays the value of x squared.

5. What is the output of the program? Where should you look for output?

The output is in the shell window (in blue):

```
Welcome to CS 101!
123456787654321
```

6. Predict the output of each line below. Then type each line into the Shell window (one at a time) and check your answers.

a) `print 2 * 5` 10

b) `print 2 + 5` 7

c) `print "2 + 5"` 2 + 5 (without quotes)

d) `print CS rocks!` SyntaxError: invalid syntax

e) `print 2 # 5` 2 (because # makes a comment)

7. Explain the difference between b) and c) in the last question. Why are the results different?

`2 + 5` computes the number 7, whereas `"2 + 5"` is literal text.

8. What is wrong with the code in d)? Explain the error message. How do you fix the error?

It's interpreting the code `CS rocks!` as an arithmetic expression, but it's not valid (e.g, there's no operator). Syntax error means the code is not correctly structured. Add quote marks around `"CS rocks!"` to fix that line.

Model 2 Guessing Game

Create a new file named `guess.py` and enter the following code. Replace the name in Line 2 with your own name. Be careful to type the code ***exactly*** as shown.

```
1 name = raw_input("What is your name? ")
2 if name == "Taylor":
3     print name, "is a great name!"
4 else:
5     print name, "is an okay name."
```

Note: `raw_input` is a **function** that displays a **prompt** on the screen and reads a line from the keyboard. In this program, the result of `raw_input` is stored in the **variable** `name`.

Questions (15 min) Start time: _____

9. What is the prompt? Why is there a space at the end of it?

The prompt is `"What is your name? "`. The space at the end makes it so that the input isn't "touching" the question mark when the user types.

10. Run the program a few times, entering a different name each time. Feel free to modify the messages as you see fit.

11. Enter each of these lines into the IDLE shell, and explain where the syntax error occurs.

a) `name? = raw_input("What is your name?")` question mark

b) `your name = raw_input("What is your name?")` space between your and name

c) `1st_name = raw_input("What is your name?")` the word 1st

d) `from = raw_input("Where were you born?")` the equals sign (from is a keyword)

12. Based on the errors in the previous question and the following correct examples, describe three rules that need to be followed when naming a variable.

```
name2 = raw_input("What is your name?")
your_name = raw_input("What is your name?")
firstName = raw_input("What is your name?")
```

Answers may include: it can't have punctuation or other symbols, it has to be one word, it can't start with a number, and it can't be a keyword.

13. At the end of your `guess.py` program, create two new variables named `number` and `guess`. Set the value of `number` to be an integer between 1 and 100 (of your choice). Ask the user to guess your number, and store the result in `guess`. When asking for numbers, use `input` instead of `raw_input`. Write your two statements in the space below.

```
number = 74  # or some other value
guess = input("Guess my number: ")
```

14. Add the following logic to your program: If the guess is too high, display the message `"Too high!"`; if the guess is too low, display the message `"Too low!"`; if the user guessed the number, display `"You got it!"`. Write your statements in the space below.

```
    if guess < number:
        print "Too low!"
    if guess > number:
        print "Too high!"
    if guess == number:
        print "You got it!"
```

15. What is the difference between = and == in the programs you have written today?

The = operator is used to ***assign*** values to variables, whereas the == operator is used to ***compare*** values for equality.

16. At this point, you should have a program that allows the user to make only one guess. Rather than run this program over and over again, you can use a `while` loop to make it repeat the guessing part. Insert the following two lines before the `input` line you wrote in #13.

```
guess = -1
while guess != number:
```

17. What did you have to do after inserting the `while` loop to make it work? In other words, how did you make the `input` and `if` statements part of the `while` loop?

You need to indent the corresponding lines of code underneath it. Select the rest of the program lines, and then press the Tab key.

18. Rather than guess the same number every time, you can have the computer select a random number for you:

- At the top of your program, add the line "`import random`" (without the quotes).
- Then change the line where you set value of `number` to use this example instead:

```
number = random.randint(1, 100)
```