# *Homework: Introduction to Python and Thonny*

## Learning Outcomes

- *Explain the basics of Python code*
- *Demonstrate editing  Python code in the Thonny IDE (Integrated Development Environment).*

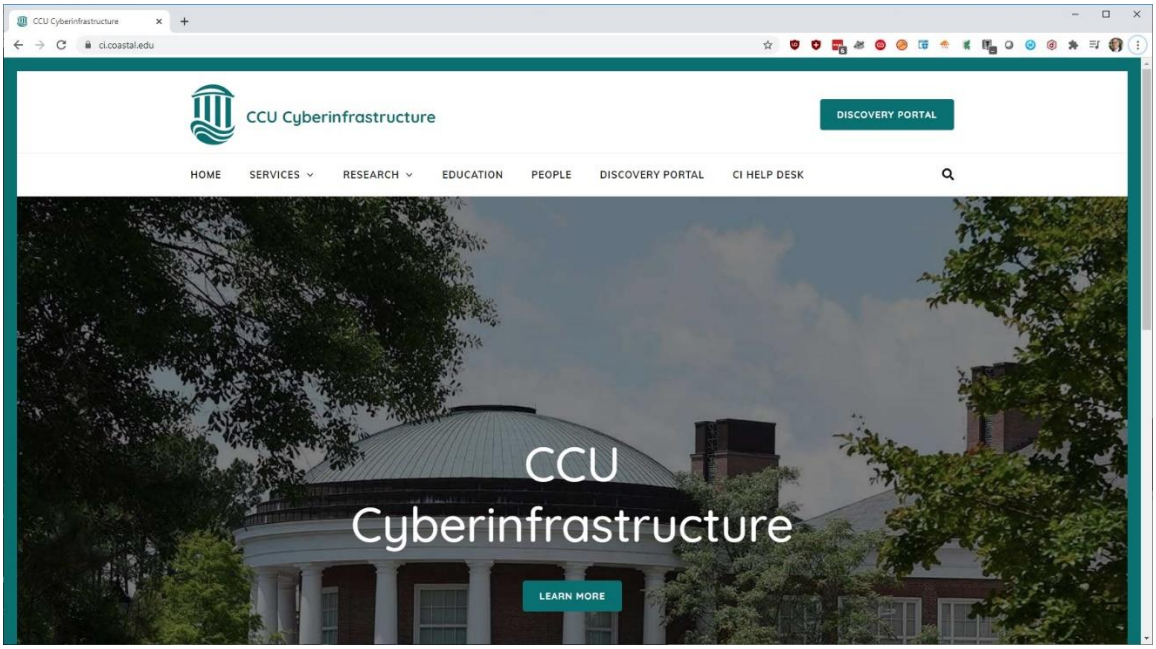
## Rubric (One Hundred Points)

One hundred Points:  Submission of a simple working program

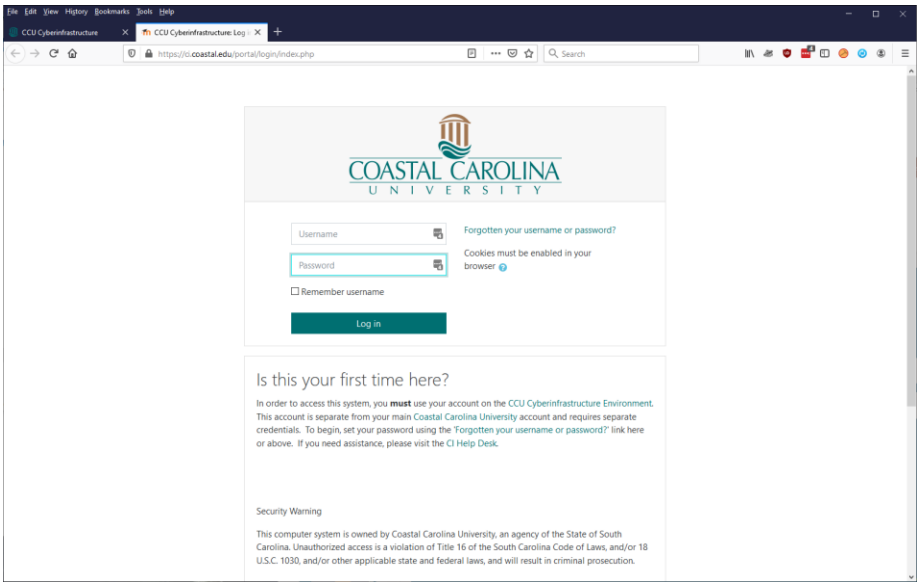
***NOTE: You must have completed the setup of the Coastal CI system or downloaded and installed Thonny to your computer prior to doing this lab. If you have not done that, please try the instructions in the installation document(s).***

## A. Getting Signed In (If Using the COASTAL CI Only. If you are using the app on your machine, skip to part B)

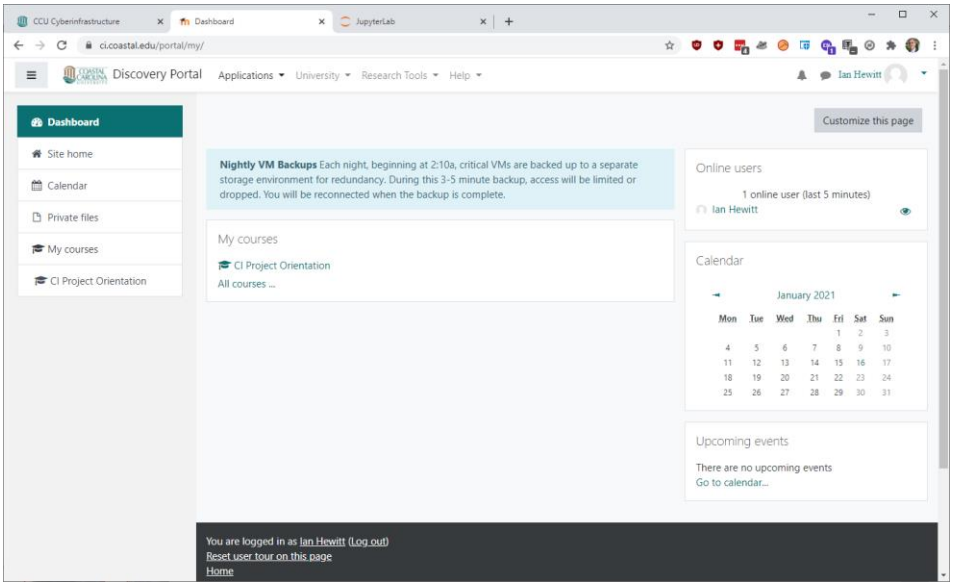
1. To get started go to https://ci.coastal.edu



2. Click the **DISCOVERY PORTAL** button and enter your Coastal CI Username and password.

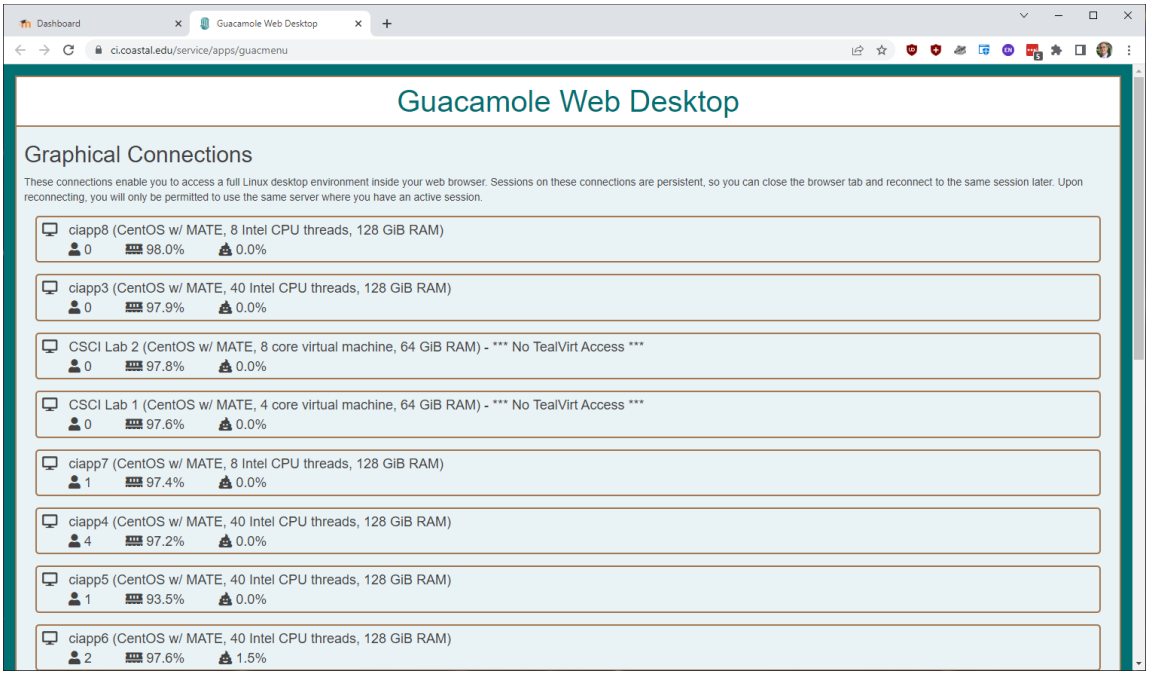



3. You should then see the *Dashboard* page that looks something like this:

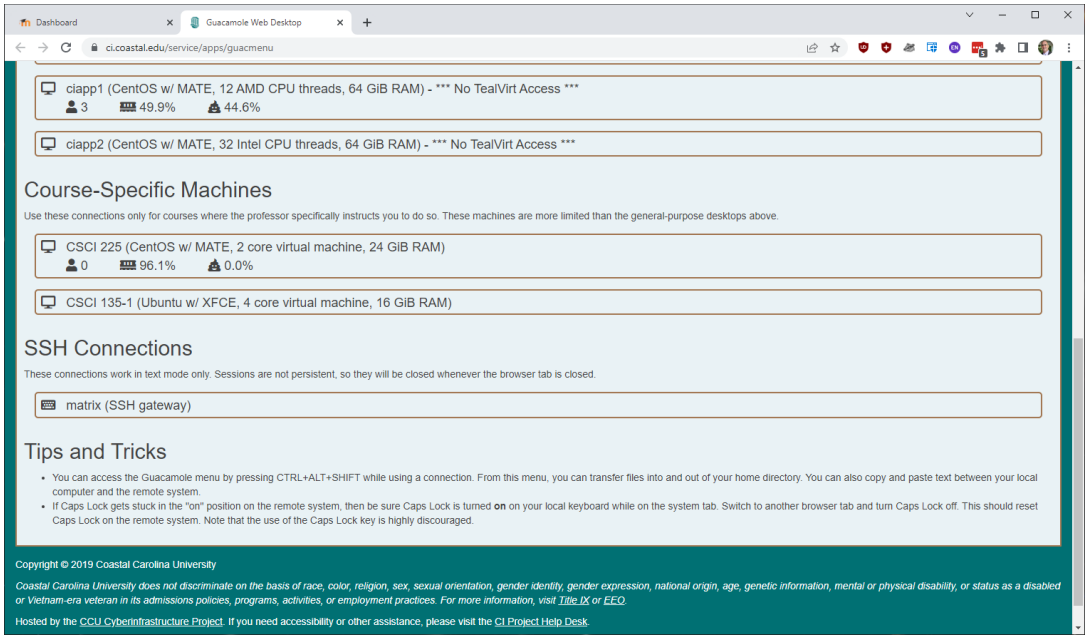



4. Once you are on the *Dashboard* page it may look suspiciously like a Moodle shell, and you would be correct (but it is a different system than the one you use for classes).

5. Click the Applications→Guacamole Web Desktop to get a list of virtual machines.

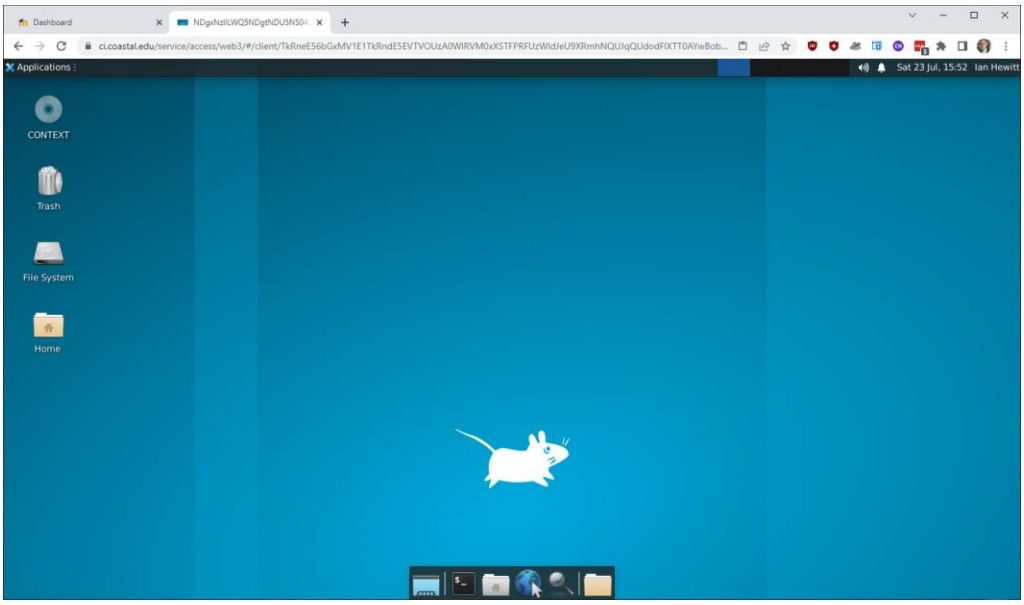



6. If you are already logged into one of the machines, you may see a shorter list. you will see a list of virtual machines. Scroll down until you see one of the CSCI-135 machines in the ***Course-Specific Machines*** section.



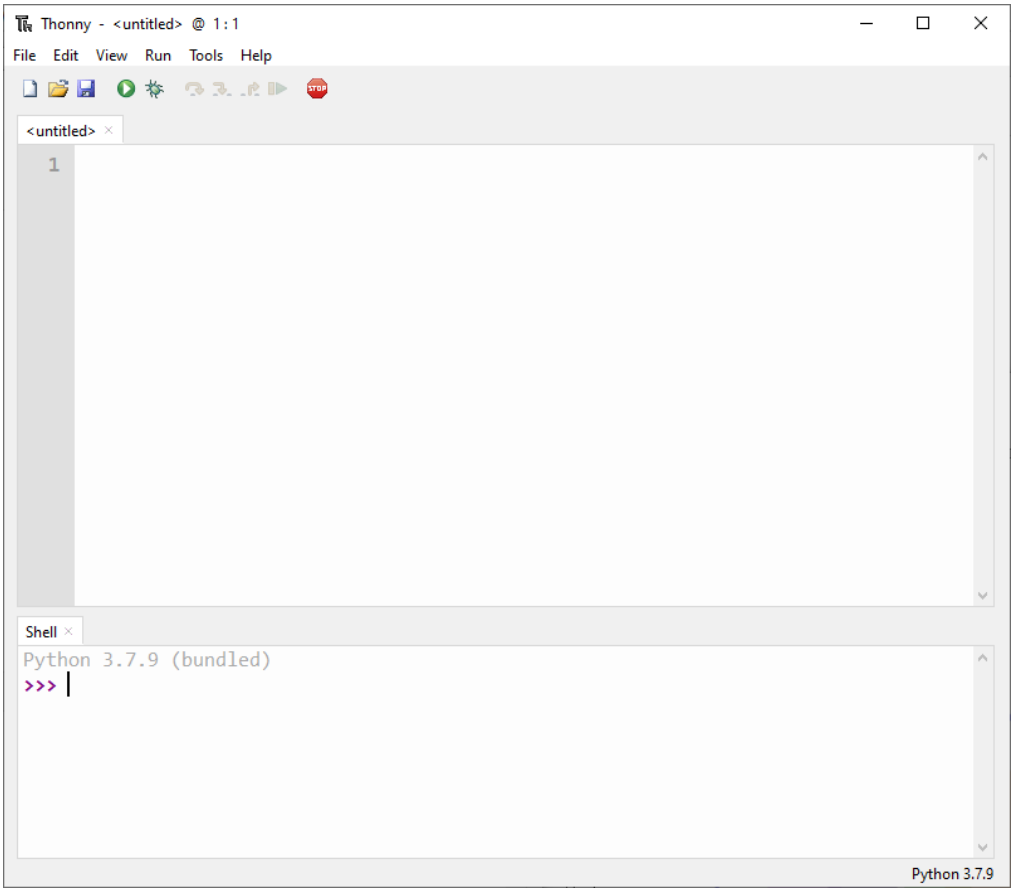
7. Select one of the machines named CSCI-135 to be logged into that machine. It does not matter which machine you use and once you pick one, you can logout and go to another at any time. Your work will still be there. If it shows you just one machine and says, *"Logged In,"* you can just click that machine.

8. Once you see the virtual desktop, you are ready to start (yours may look slightly different, that is OK – if you did not login to a non CSCI-135 machine)!




## B. Starting Thonny and Creating Your First Program

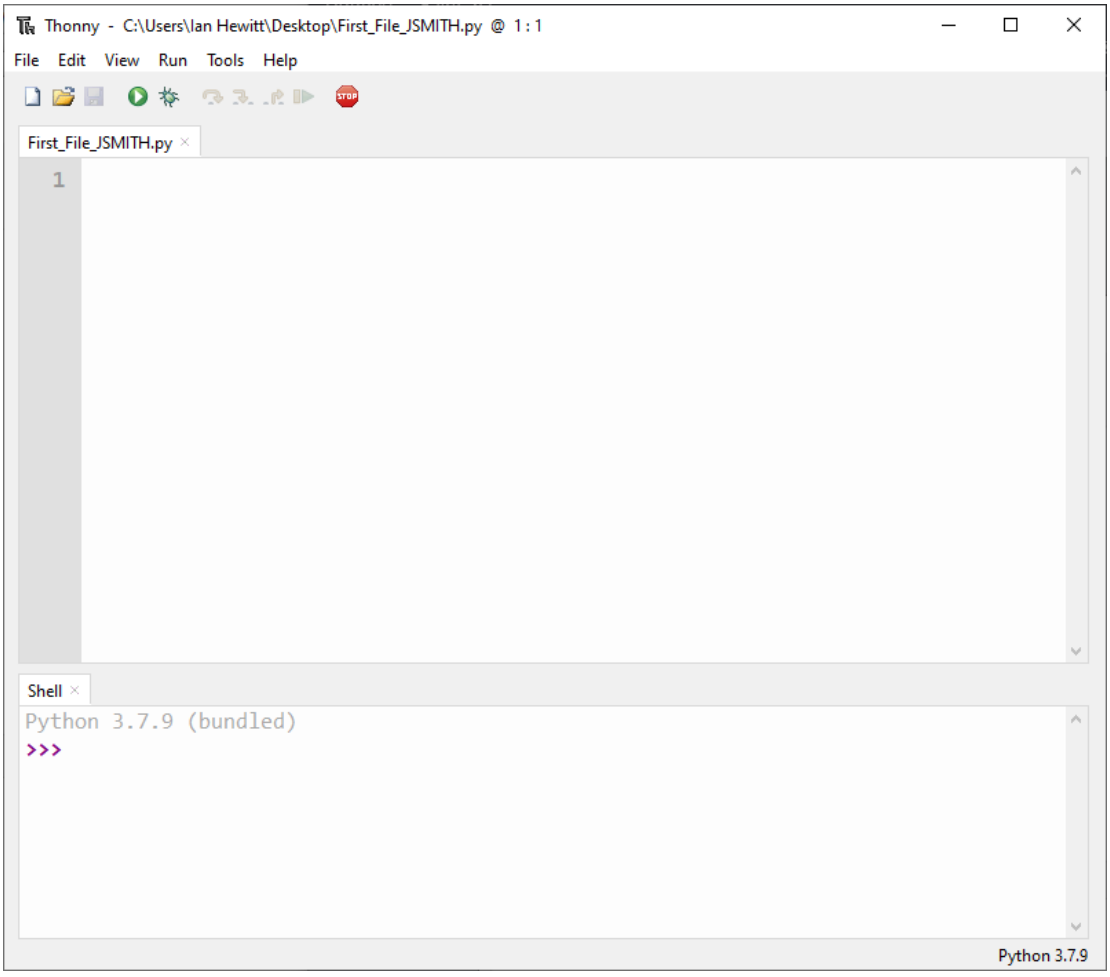
9. Now we need to start Thonny. If you are using the Coastal CI, click the ***Applications→Development*** menu item in the and select the ***Thonny***. If you are using your own system, go ahead and start the Thonny application. This should open a basic IDE window:




10. One good thing about Thonny is that it has all the key features you need, but not a lot of extras. This means the interface is simple and easy to learn. The basic layout has two key windows. The top and largest window is you program window. This is where you will put your Python code. Once you put code in this window, you can save it to a file. You can also load Python code from any file you have created. Right now, the window says *Untitled*. That means this a new program that has NOT been given a name and saved.

11. The lower window should have a title that says *Shell*. This window shows the Python interpreter. This is where Python will show the input and output from your code. For example, if your program prints anything out, it will appear here. If your program asks for input, you will enter it here and finally, any errors will appear here. There is one other thing you can do in the shell window. You can type Python statements one at a time and they will run right after you enter them. That sounds convenient, so why would we not do everything in the shell window? Well, the problem is that you are not able to save the code in the shell window, so there is no way to go back and work on it (and no way to submit the work). So, we will do all our work in the main program window for this course.

12. Let us go ahead and create our first program. First, let us give the program a name and save it. This is good habit to get into, so you do not lose your work if you forget to save. Click on ***File→Save As***. This should bring up a window that will let you choose a directory and a filename to save your work (something like *First_File_xxxxxx*, where *xxxxxx* is your name). Notice that Thonny will open a new window that will have the name you chose int the title bar.

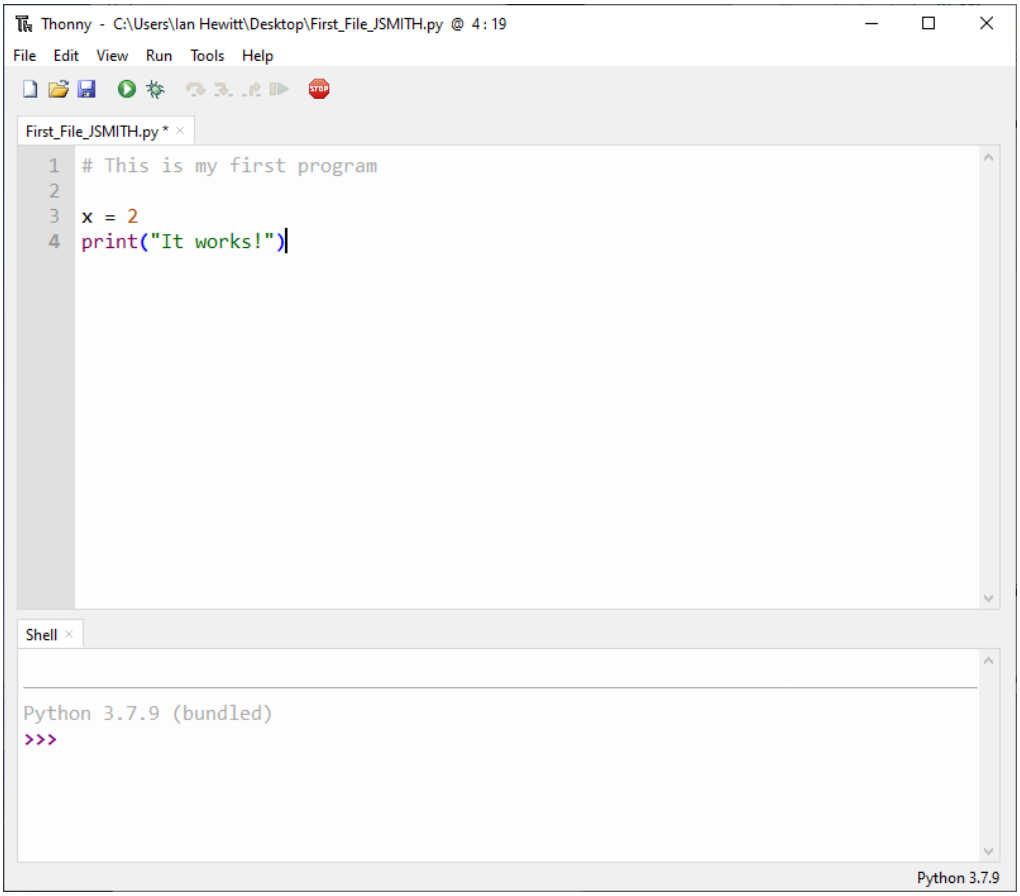



13. Let get started with some basic Python code. Enter the following in the programming window:

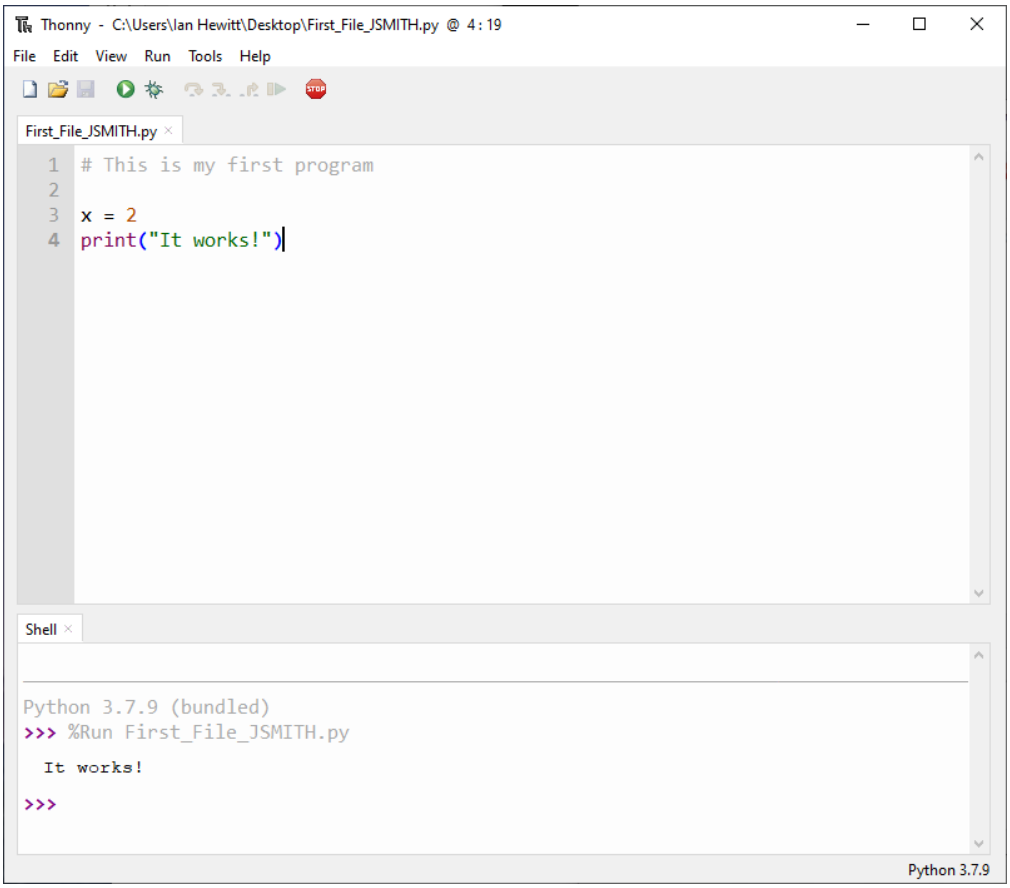
```
# This is my first program

x = 2
print("It works!")
```

14. Let us to look at the program we just created. Here are the important things in this simple program:
    - Line numbers: If you look in the left-hand column, you will see line numbers (in the gray area). This will help with any Python error messages, as any message will reference a line number.
    - Line 1: This line starts with an *octothorpe (#)*. This symbol tells Python that everything else on this line is comment and to ignore it. Comments are great ways to document your program so that anyone who may use the code in the future will understand why the program is structured the way it is. In this class, we recommend that you put comments in the code, so we know what you were thinking or trying to do.
    - Line 2: Line 2 is blank with nothing on it. Python will ignore any blank lines like this. Using lines like this is simply to make your code look more readable. It is important to remember that Python ignores these lines and having blank lines does not put blank lines in any output of the program!
    - Line 3: This is a simple assignment statement. We will learn more about these, but for now you can imagine the variable, ***x***, like a box for storing things, in this case the integer 2 is being stored in the box called ***x***.
    - Line 4: In order to have the program display something in the shell window when you run it, you need to use one or more ***print()*** statements. ***Print()*** is a *function*, which means it accepts *arguments*. In this case, we are telling it what to display (the text “It works!”). Notice we put the **argument** inside parentheses. We also surround the argument with quote marks. This tells Python where this argument begins and ends.



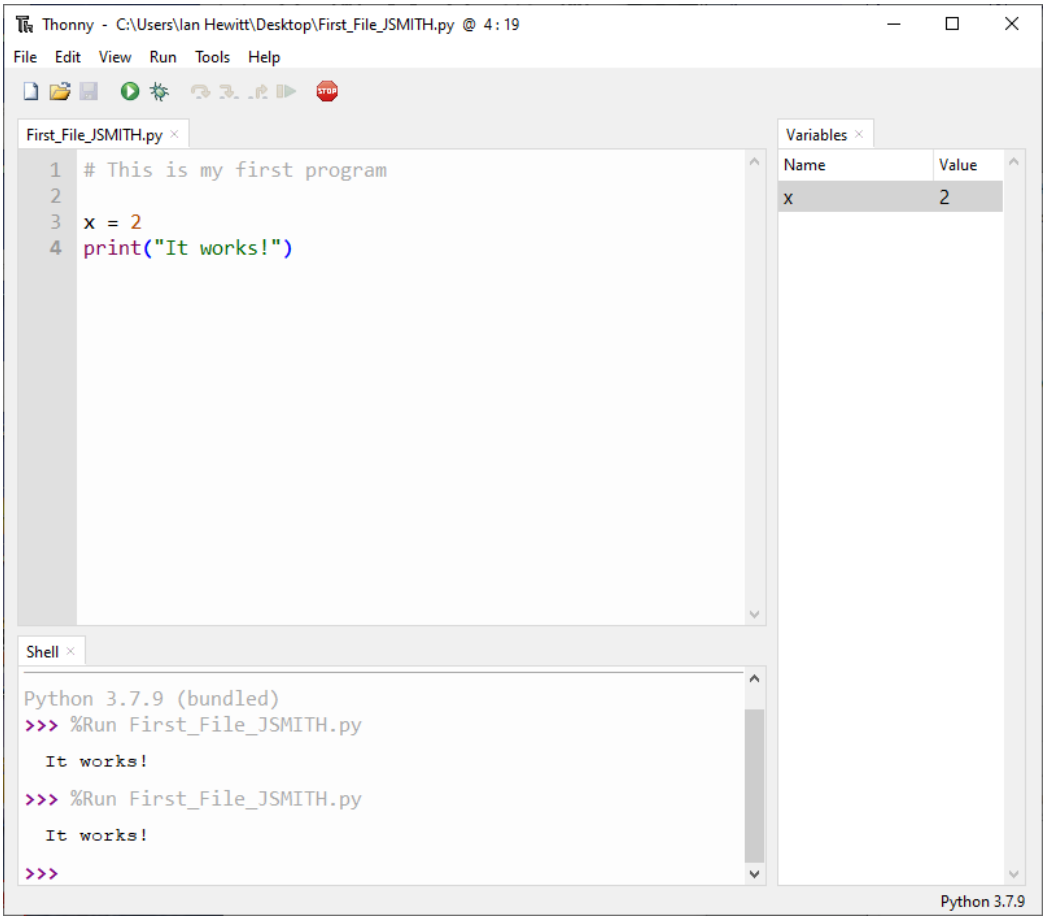

15. You may have noticed that after you typed in your code nothing happens. That is because you need to run the program before the Python code you have written does anything. To run the program, you can either push the  button, or go to the menu and select ***Run→Run current script***.




16. You will see in the shell, the output from the Python code. (If you got an error, then you need to go back and correct your code). You should also go the menu and select ***File→Save*** to save your updated code to the file name that you gave it. This should be saved as a file with a ***.py*** extension. This is the file you should submit to Moodle to get credit for this assignment. You only need to submit the ***.py*** file for this assignment.

## C. Watching Variables

17. Thonny has useful features for working with programs. One that is particularly good is a way to watch variables as they change in your program. You can try this with the program that you just made. Got the menu and select View→Variables. Then go ahead and rerun your program. You should see something that looks like this:




18. The new right-hand window will show variables as they change. This will be convenient when you start doing more complex programs.