

Computer Programming Startup for BIOL 480/680

1) Download and install the Python language developer tools from anaconda.

<https://www.anaconda.com/download> Select the download for your computer's operating system.

Run the downloaded program to install the software.

The Anaconda download includes the basic Python system, a large set of packages for numerical and scientific computing, and an IDE (Integrated Development Environment) called Spyder that is useful for developing and running your programs.

2) Using the Python system

Start by opening the Spyder program.

You might notice an entry in the upper right, referred to as the working directory. It can be useful to create a directory on your computer where you will keep all your Python program files. You can set a default working directory by finding it using the little icon just to the right for "Browse a working directory".

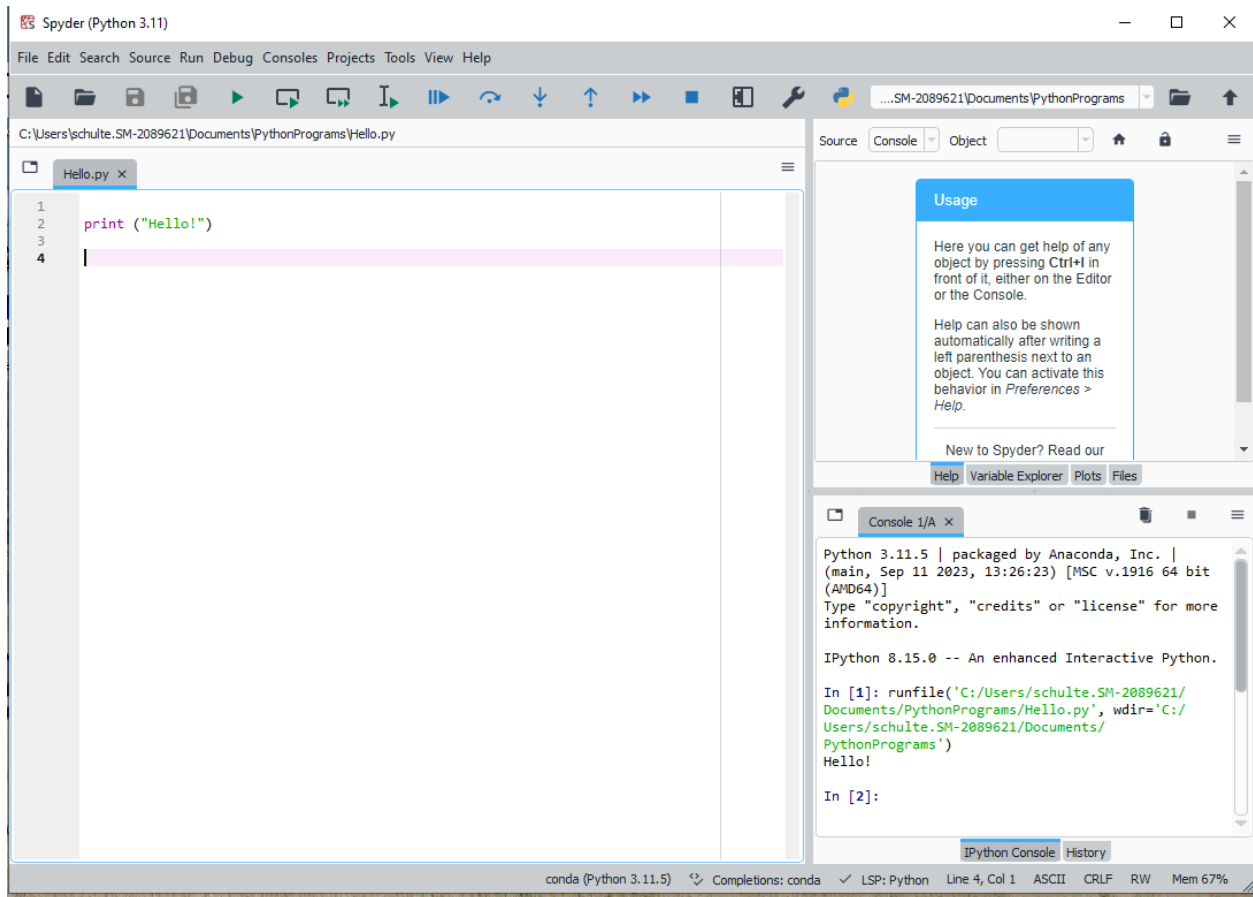
From the Spyder display, select "New file" from the File menu or pick the corresponding icon from the toolbar.

Now create the classic hello program by entering this single line in the editor panel on the left:

```
print ("Hello!")
```

Next, you can run the program by clicking on the little green triangle in the toolbar (or select the menu item "Run" and pick "Run").

At this point, your Spyder window should look something like the next figure. Note that the Spyder display preferences shown here have been changed from what you may see the first time the program is run. Personally, I like the bright background better than the dark background. You can make these selections by looking under the menu item Tools > Preferences. Under Appearance, I chose the main interface theme of "Light", the Icon theme of "Spyder 3", and the Syntax highlighting theme of "Spyder". You of course can make other choices, this is only mentioned because your view of the program may differ from the ones shown below.



The large region (sometimes called a window pane) on the left is the editor where you entered the program text. The upper right area can be used for viewing variables (this program so far doesn't use any variables) or other results like plots. The lower right area shows the Python console where the program was run and shows the output the program produced.

That's it! You have written and tested the classic Hello program.

On the other hand, if you mistyped some of the text, you may get error messages. These are sometimes obscure. If you typed "Print" instead of "print", you'll see that programming languages are generally case-sensitive. You might see a message saying that "Print" is not defined. Fixing the spelling should allow things to work. Variable names are also case-sensitive: the names a and A are not related objects.

Try something a little more interesting. Change the program to read:

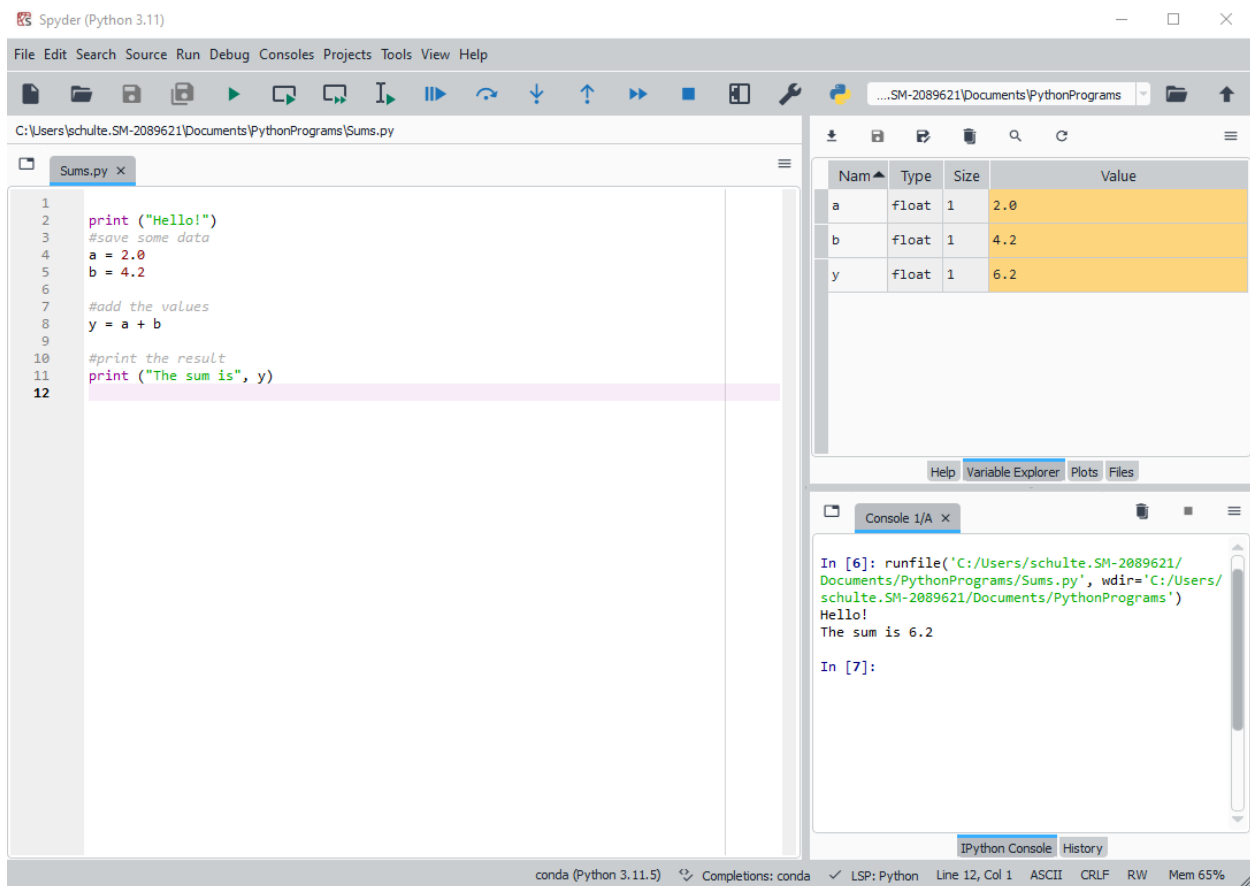
```
print ("Hello!")
#save some data
a = 2.0
b = 4.2

#add the values
y = a + b

#print the result
print ("The sum is", y)
```

The lines starting with # are comments that are not processed when the program runs but can be useful as notes to the programmer.

After running the program, the Spyder window should look like this:



Notice the result in the lower right pane. Also in the upper right pane, the “Variable Explorer” tab has been selected and you can view the values of the three variables that were set by the program.

That’s the Python start. We will build on this by further exploring the use of the Python programming language. Later we will introduce a tool called a Debugger that can be helpful for finding errors in programs.

Here are a few references on Python programming that might be useful. All three of these can be obtained from the UNLV library web site by searching on the book title. In each case, you should be able to view the book online or even download a full pdf version. You might have to be on campus or enter your UNLV ID on the library web site to get free access. The books shown below are listed in order of increasing level of detail as reflected in their page lengths.

Sundnes J. 2020. Introduction to Scientific Programming with Python. Springer. 148 pp.

Linge S, HP Langtangen. 2020. Programming for Computations – Python. A Gentle Introduction to Numerical Simulations with Python 3.6. Springer. 323 pp.

Langtangen HP. 2016. A Primer on Scientific Programming with Python, 5th ed. Springer. 914 pp.

