

## **Introduction to Programming with Python 3, Ami Gates**

### **Chapter 1: Creating a Programming Environment**

#### **1.1: Python, IDEs, Libraries, Packages, and Platforms**

A first step to learning and using any new programming language is to set up a development environment. This process can take some time and may depend on development goals. For Python 3, the topic of this book, there are several options. While it is certainly feasible to download Python 3 and program directly on the command line, the utilization of an Integrated Development Environment (IDE) and a Platform, offer improved support, organization, and management.

An IDE is an optional environment that can ease and organize the development process. There are many free IDEs available for Python 3, such as Spyder, PyDev, and PyCharm. A platform is a greater environment that can bring together and organize packages, Python, and IDEs.

This book will use the IDE called Spyder and the platform called Anaconda. The download and installation of Anaconda will include Python3 and Spyder, as well as hundreds of Python packages. That being said, all examples in this book can be applied to any IDE.

A package is a collection of modules (functions and code that perform tasks). For example, numpy is a package for scientific computing with Python. The numpy package contains tools that improve the use and manipulation of mathematical and numerical functions, as well as offering the use of a data structure called an array. Numpy is excellent in combination with graphing packages, such as matplotlib, and can be used in conjunction with Matlab. Later chapters will explore numpy and matplotlib. Considerable documentation exists for numpy (<https://docs.scipy.org/doc/numpy-dev/user/quickstart.html>). Similarly, matplotlib offers a plotting and graphing library (many packages and modules) that can be used to create quality figures and graphics (<http://matplotlib.org/>). There are hundreds of other packages and libraries that work with Python. The selection of packages and libraries often depends on programming project goals.

As noted, a platform is a larger environment that combines Python, IDE options, libraries, and packages. For example, Anaconda is a platform (package and environment manager) that contains over 720 open source packages, as well as a Python download, and the Spyder IDE. Over 150 packages are automatically installed with the Anaconda install and hundreds of others are available to add on. Anaconda is owned by Continuum Analytics and also works well with R (a data science programming language). Anaconda offers download options for Windows PCs, MAC OSX, and Linux/Unix (<https://www.continuum.io/downloads>). The Anaconda download and installation will automatically install Python 3, as well as the Spyder (the IDE).

Publication Pending  
Author: Dr. Ami Gates  
Date: 8/11/2016

All material is subject to copy write laws. Do not post, print, or reproduce without written permission.

Python is highly versatile. For example, IronPython is an open source implementation of Python that can utilize the .NET Framework. Jython is an implementation of Python that runs on the Java Virtual Machine (JVM), and works in conjunction with Java. PythonAnywhere is Python for the Cloud and more easily enables the development and hosting of websites (without the installation of special server software.) In addition, Django (an open source web application framework) can be used on PythonAnywhere for advanced web development. While all of these options are excellent and dependent on goals and intended project specifications, this book will focus on the introducing programming concepts using Python 3.

This book is intended for readers who are new to programming and wish to learn to program while also learning Python 3 basics. This book is also useful for those who have programming experience and would like to learn Python 3. This is an introduction to programming with Python 3.

## **1.2. Steps for Installing Python 3, Spyder, and Anaconda**

At this time, the latest Python 3 release is 3.5.1. This is likely to change over time. The direct download location for Python 3 itself can be located on the Python Site:

<https://www.python.org/downloads/>

However, it is recommended that rather than downloading Python 3 directly, the platform called Anaconda (containing Python 3) is downloaded and installed. All examples in this book may be used with any installation of Python 3. However, all examples in this book will be created using Anaconda and Spyder (an IDE that downloads with Anaconda).

Common Python installations (including that from Anaconda) will automatically include the Standard Python Library. Once Python/Anaconda is installed, other modules, packages, and libraries can be added it.

Often, the words library, module, and package seem interchangeable. However, a Python Module is a Python source code (with a .py extension) that can be imported (using *import*) for extra functionality. A module is a collection of code that can be reused. Once programs get large, separating them into individual and reusable modules improves organization, reduces debugging time, and allows for sharing and reusability of functionality. A Python Library is a collection or directory of modules, files, and packages. A package is a collection of modules or code, and must contain *init.py* (don't worry about that for now). As noted above, Anaconda is a Python Platform for Python programming, as well as for data management, analysis, and the visualization of large datasets.

### 1.2.1: The Steps for Downloading and Installing Anaconda

To download and install Anaconda, first go to the site called, <https://www.continuum.io/downloads>. Next, determine if your computer is a 64 or 32 bit machine. To do this, go to the Control Panel and choose System and Security. Then choose System. See Figure 1 for an illustration.

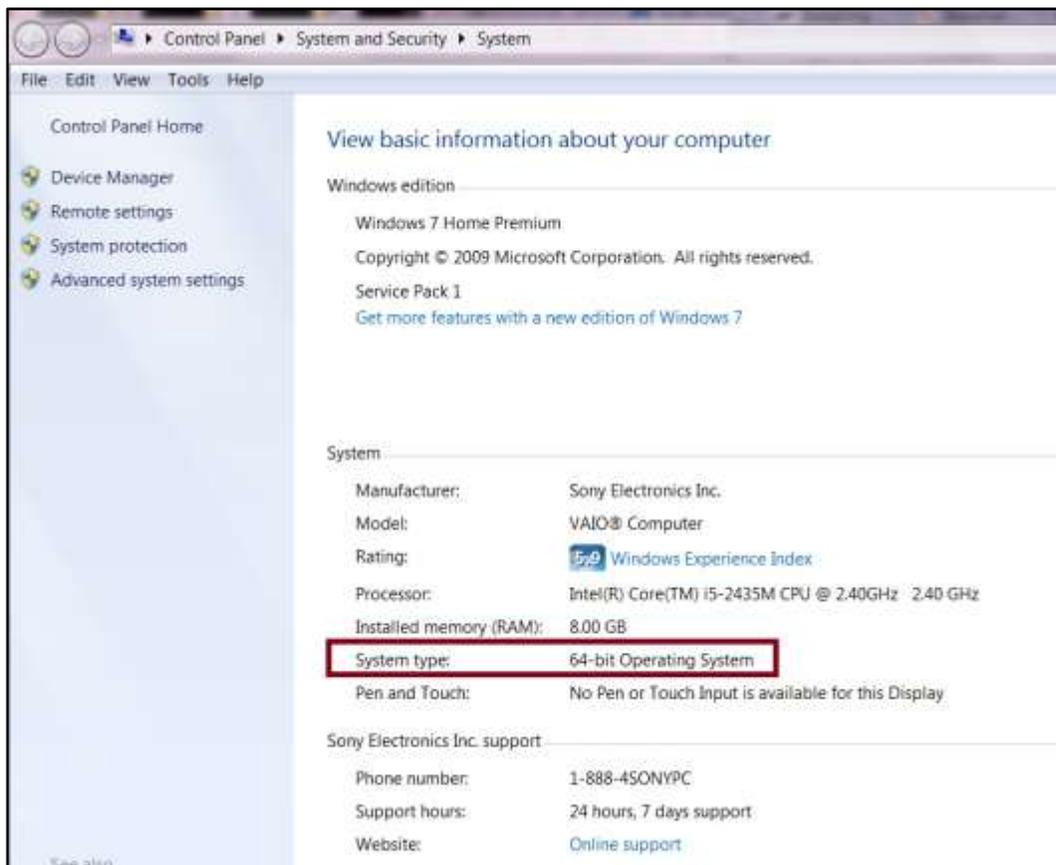


Figure 1: Illustration of the determination of a computer system type (32 or 64 bit) on a PC.

From the website, <https://www.continuum.io/downloads>, select and download the appropriate version of Python 3, namely the Windows version for PC machines using a Windows operating system, the OS X version for MAC machines, or the Linux/Unix version for Unix machines.

Follow the download process carefully. Keep in mind **where** files are being saved. Complete the installation. If installing for Windows, it is likely that a step will include downloading, or saving and running an executable file (a file that ends in .exe). This executable file will manage the

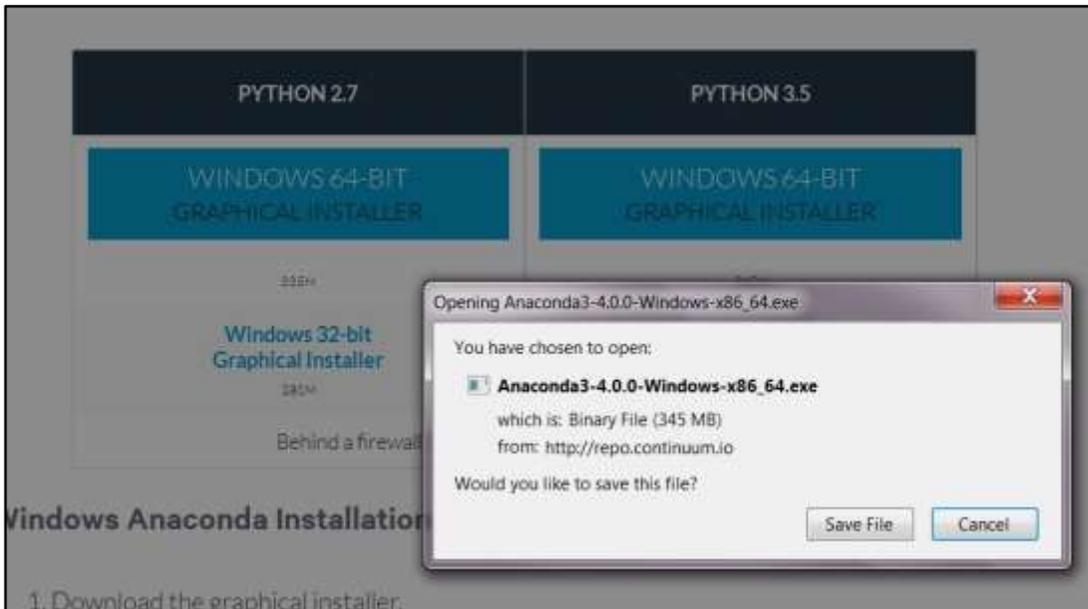
Publication Pending

Author: Dr. Ami Gates

Date: 8/11/2016

All material is subject to copy write laws. Do not post, print, or reproduce without written permission.

remainder of the download process. Figure 2 offers an illustration of saving the Anaconda executable file. In this case, save the .exe file and then double click it to run it.



**Figure 2:** Illustration of executable file download request for Anaconda on Windows.

There will be a very similar option for MAC or Linux. In any case, save the .exe file. Note the location where the file is being saved. Then double click the file to initiate the installation. Specific methods for MAC and Linux installation will not be covered in this book, but can be located on the Anaconda web site, <https://www.continuum.io/downloads>.

Once the Anaconda (.exe) installation file is saved and double clicked, it is likely that a security warning will appear. Figure 3 illustrates a possible example of a security warning. Choose to run the installation.

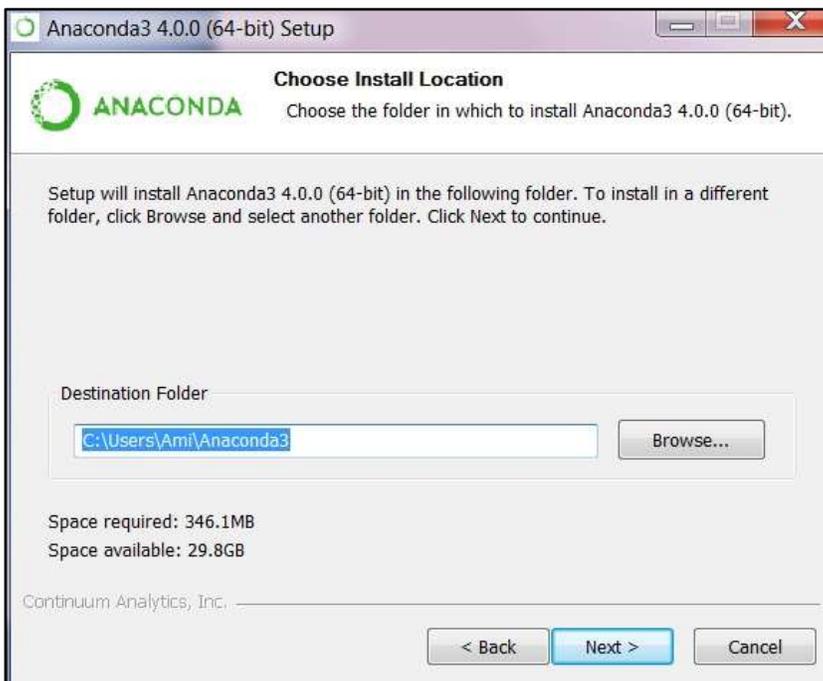
Publication Pending  
Author: Dr. Ami Gates  
Date: 8/11/2016

All material is subject to copy write laws. Do not post, print, or reproduce without written permission.



**Figure 3:** Illustration of a security warning.

Follow the steps for installation. When choosing a location for Anaconda, review the destination folder and make a note of it. In addition, choose the option to add Anaconda to your path and to register Anaconda as default for Python. Figures 4 and 5 illustrate these steps.



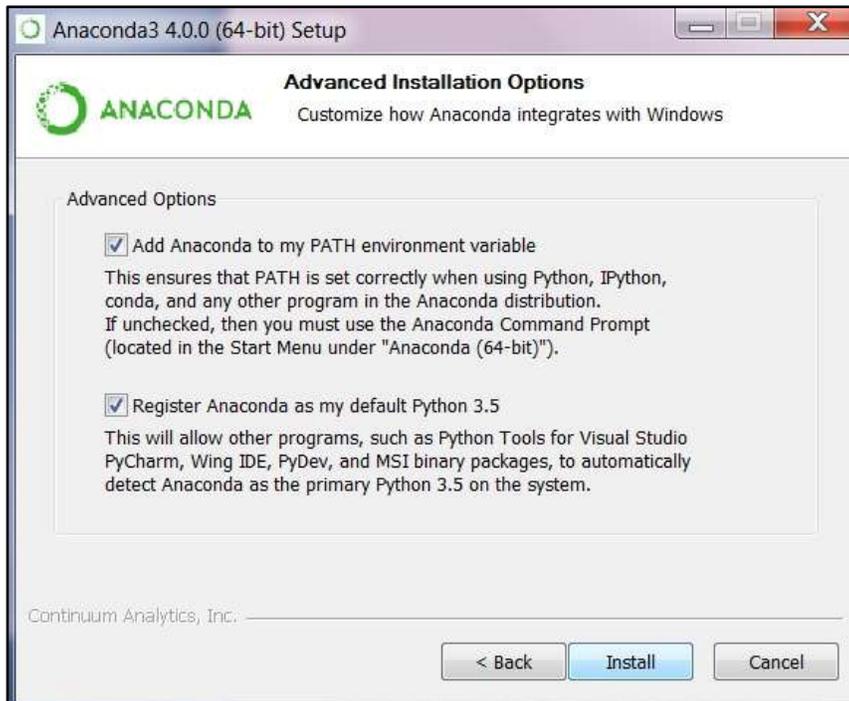
**Figure 4:** Illustration of the destination folder for the installation.

Publication Pending

Author: Dr. Ami Gates

Date: 8/11/2016

All material is subject to copy write laws. Do not post, print, or reproduce without written permission.



**Figure 5:** Illustration of the advanced options selections.

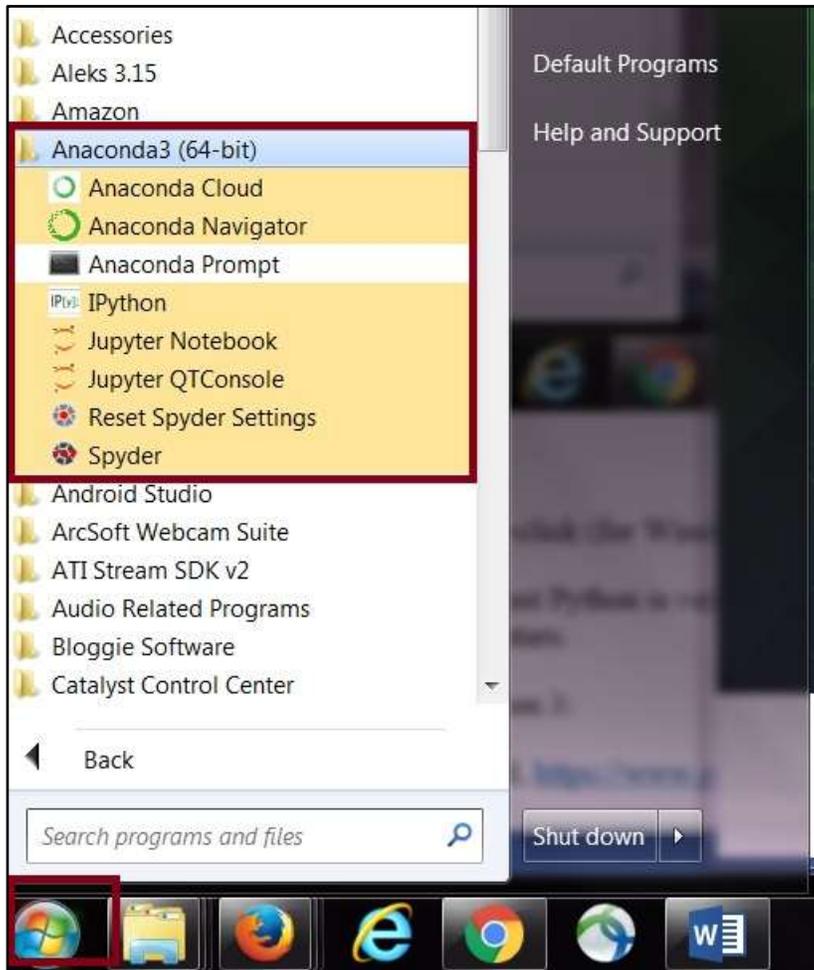
Once the installation process is completed, locate Anaconda and create a Desktop shortcut if desired. Figure 6 illustrates where to locate Anaconda and Spyder on a PC.

Publication Pending

Author: Dr. Ami Gates

Date: 8/11/2016

All material is subject to copy write laws. Do not post, print, or reproduce without written permission.



**Figure 6:** Locating Anaconda and Spyder.

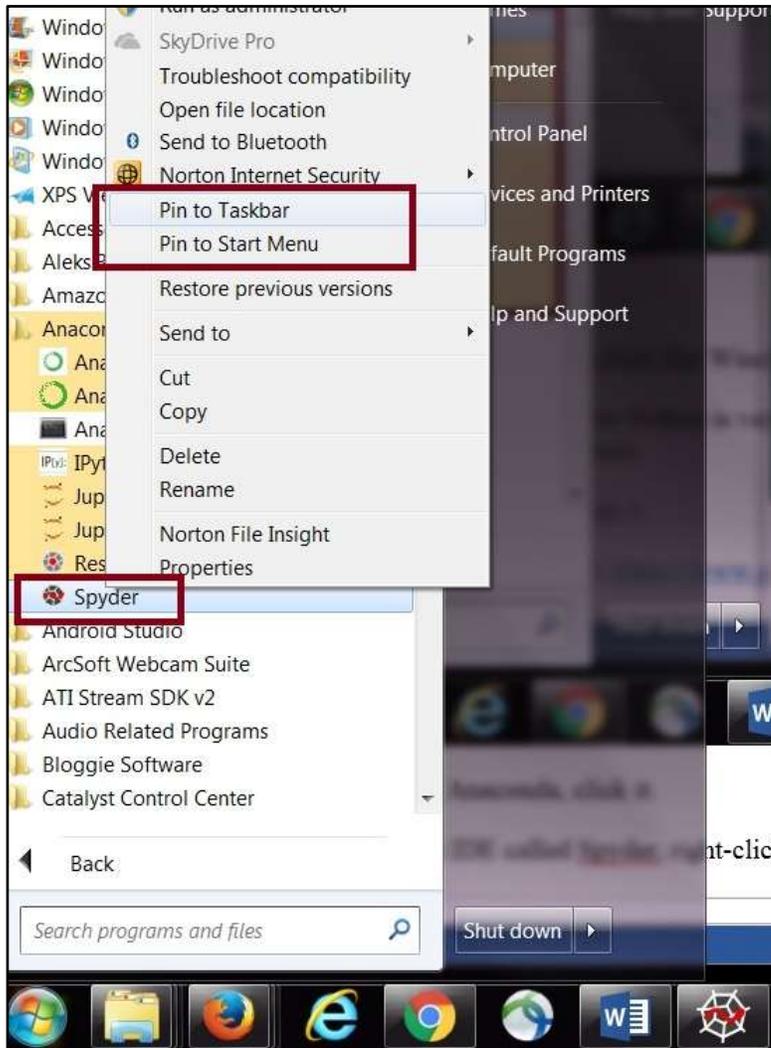
To create a shortcut to the Spyder IDE, right-click (for Windows) and pin to the Start Menu, or a preferred location. Figure 7 illustrates the option to pin Spyder.

Publication Pending

Author: Dr. Ami Gates

Date: 8/11/2016

All material is subject to copy write laws. Do not post, print, or reproduce without written permission.



**Figure 7:** Illustration of pinning Spyder to the taskbar or start menu.

At this point, Python 3 via Anaconda, as well as the Spyder IDE, should be installed. The remainder of this book will assume the use of Spyder for all programming examples.

### 1.2.2: “Hello World” in Python 3

No matter whether a person is new to programming, or is a seasoned coder, the first step when learning any new programming language is to create a program that prints, “Hello World.” By doing this, one can assure that the programming environment is set up and functional.

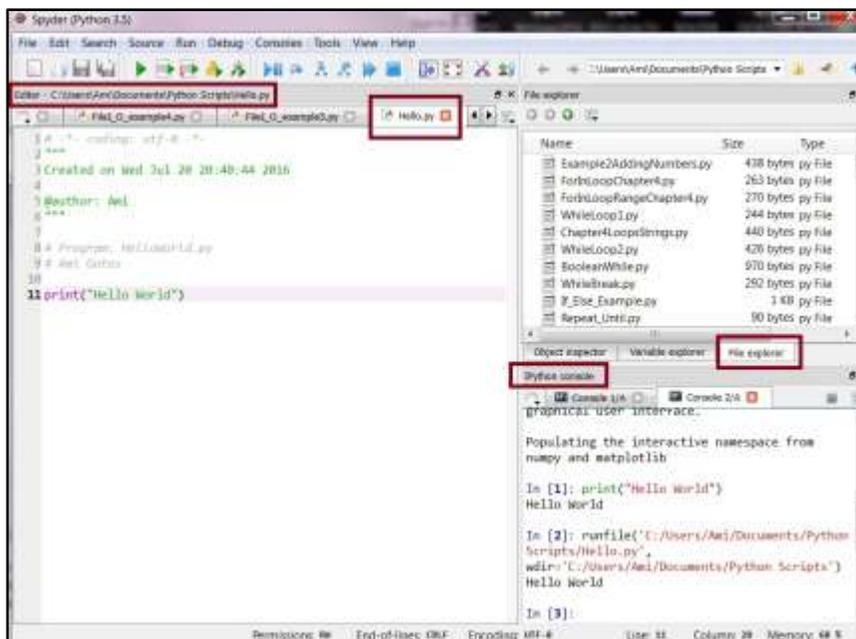
All material is subject to copy write laws. Do not post, print, or reproduce without written permission.

In this section, the “Hello World” program will first be completed on the IPython command line console in Spyder and will then be completed and run as a python (.py) file. To begin the process, open the Spyder IDE (or any other Python 3 IDE.)

When the Spyder IDE is open, three main areas are visible. The left side is the editor. It is the area where new files are created and where Python code can be typed and saved. The editor can be used to create .py Python code files, as well as other types of files.

The right side of Spyder, on the top, is the file explorer area. This same area can be toggled using the tabs at its base, between File explorer, Variable explorer, and Object inspector. The file explorer tab allows a view of all files created and saved in the editor, as well as any files saved to that folder. The file explorer is a folder that can also be accessed from outside of Spyder. For example, on my computer, the file explorer is actually the folder on my computer in location, C:\Users\Ami\Documents\Python Scripts. As such, placing a file in that location on my computer will cause the file to appear in the file explorer in Spyder.

The right side of Spyder, on the bottom, is the IPython Console (the command line). When code is typed into the left editor, saved as .py, and run, its output appears on the IPython Console on the lower right. In addition, it is possible to type code directly onto the IPython Console and run it by clicking “enter.” Figure 8 illustrates the Spyder environment, with the editor on the left, the file explorer on the upper right, and the IPython Console command line on the lower right. It is an option to rearrange the environment as desired.



**Figure 8:** An illustration of the Spyder IDE

## Exercise 1.1: Hello World and the IPython Command Console

Inside of Spyder, locate the IPython Console (normally on the lower right). Type the statement, `print("Hello World")`, into the Console, and press enter. The result should be `Hello World` printed out in the Console. As a side note, this will be true for any Python command line or IDE console. Figure 8 illustrates these steps.

The IPython Console is a command line that can interpret any Python code. For example, it is possible to type any of the following statements into the console, press enter, and receive a result. Try this for each of the following statements below. Figure 9 illustrates the results.

```
print("Hello World")
```

The output on the console will be *Hello World*.

```
print("3 + 5 = ", 3+5)
```

The output on the console will be `3 + 5 = 8`.

```
(3 + 5)**2 - (4 * 1.45) / 2
```

The output on the console will be *61.1*.

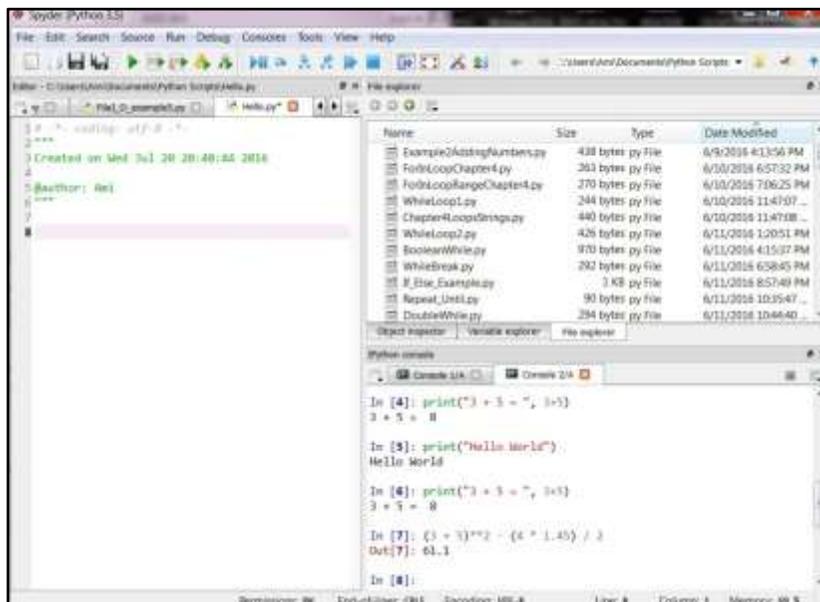


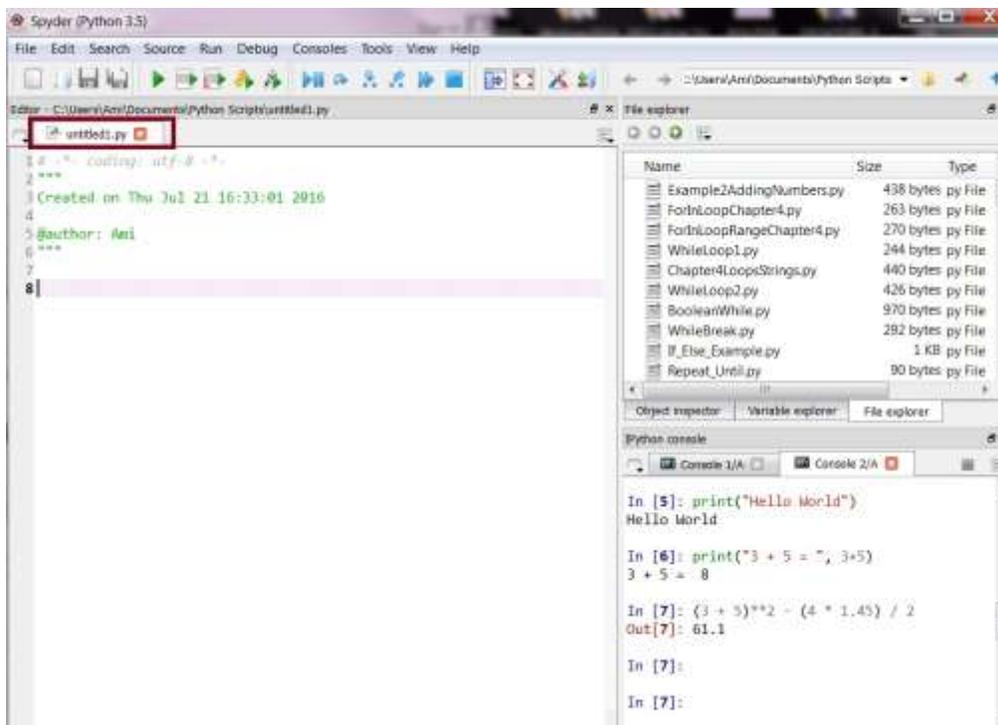
Figure 9: Illustration of Exercise 1.1 results.

However, in general, programs are not written on the command line console. The console does not offer organization or ease, and it does not save code once closed. Instead, programs are written and saved into python files (with the extension .py).

## Exercise 1.2: Creating, Running, and Saving a “Hello World” Program File

In Spyder, and other IDEs, a new file can be created, code can be added to the file, and that file can be executed (run), saved, edited, and referenced later. The executed results of a file will appear in the IPython command console in Spyder, unless they are expressly written to another location.

To create a new Python .py program file in Spyder, choose the “File” option on the upper left. Then choose “New file.” This causes the editor to create a new and untitled python file. Figure 10 illustrates the new file.



**Figure 10:** An example of a new and untitled python (.py) file.

To give this new python file a name, choose “File” and then “Save as...”. Then type in a name for the file. Avoid using spaces or special characters in file names. Save this file as “HelloWorldExample.py”. Once saved, the name of the file will update as Figure 11 illustrates.

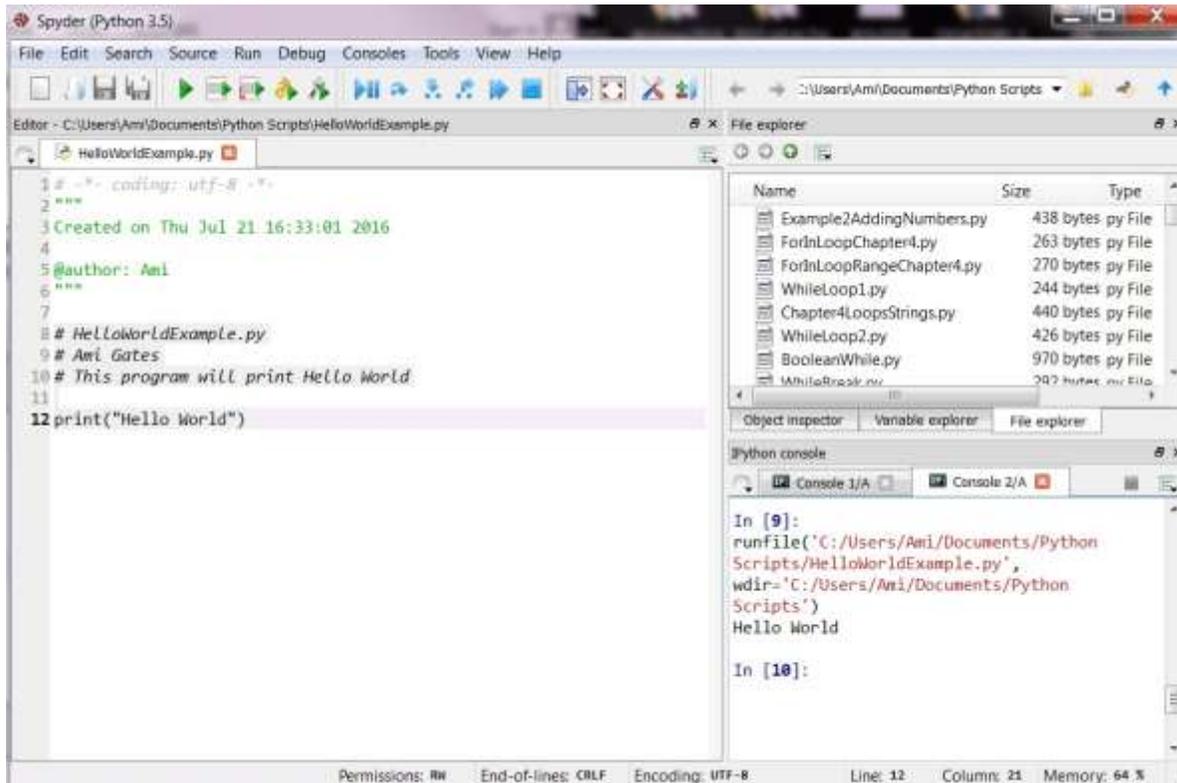
Publication Pending  
Author: Dr. Ami Gates  
Date: 8/11/2016

All material is subject to copy write laws. Do not post, print, or reproduce without written permission.

Next, type the following few lines of code into the file:

```
# HelloWorldExample.py  
# Your Name  
# This program will print Hello World  
print("Hello World")
```

Then, choose “File” and then “Save” or click the save icon. To run the program, choose “Run” and then choose “Run” from the options, or click the green run arrow (which represents the Run shortcut). Figure 11 will illustrate the program and the output. Notice that the output, the printing of Hello World, appears on the Console.



**Figure 11:** An illustration of a new Python program file called HelloWorldExample.py, the code, and the output.

Note that Spyder may pop up a “Run settings...” box to determine the location of the of the program execution (where to run the program). While settings can be altered and updated over time, it is best for now to select “Execute in current Python or IPython Console.” Also take note of the working director path (for reference). Figure 12 illustrates this pop up settings box.

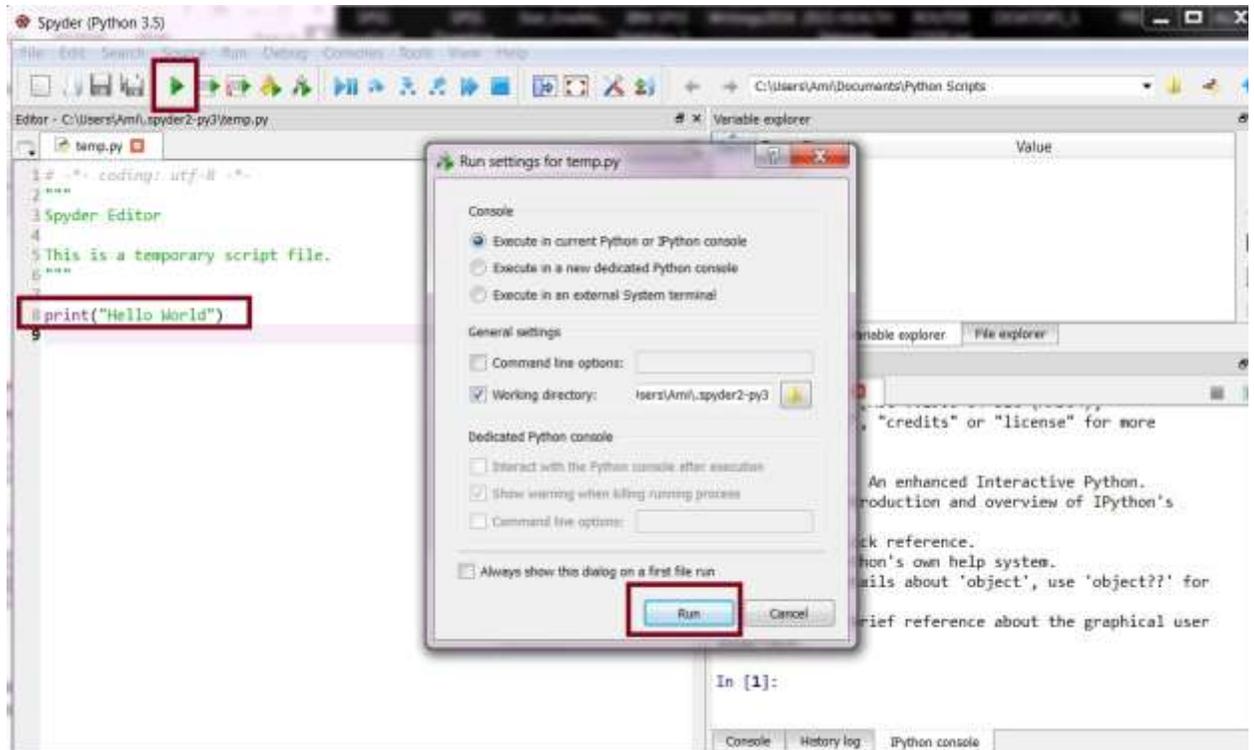


Figure 12: The Run setting pop up box that may present itself upon the running of the first .py file.

To complete the exercise, add the following statements to the HelloWorldExample.py file, save it, and run it again.

```
print("3 + 5 = ", 3+5)
(3 + 5)**2 - (4 * 1.45) / 2
```

Figure 13 illustrates these steps and the resulting output. It is important and interesting to notice that the output does not include the result of expression statement,  $(3 + 5)**2 - (4 * 1.45) / 2$ . A Python program file is not the same as the IPython console. The program file will not automatically output the results of mathematical calculations as the console will. However, it possible to create a statement in the file that will output this calculation. To do this, change the expression,  $(3 + 5)**2 - (4 * 1.45) / 2$ , to the statement, `print((3 + 5)**2 - (4 * 1.45) / 2)`. Placing the print function around the calculation causes the Python interpreter to print the result of the mathematical expression (which is 61.1).

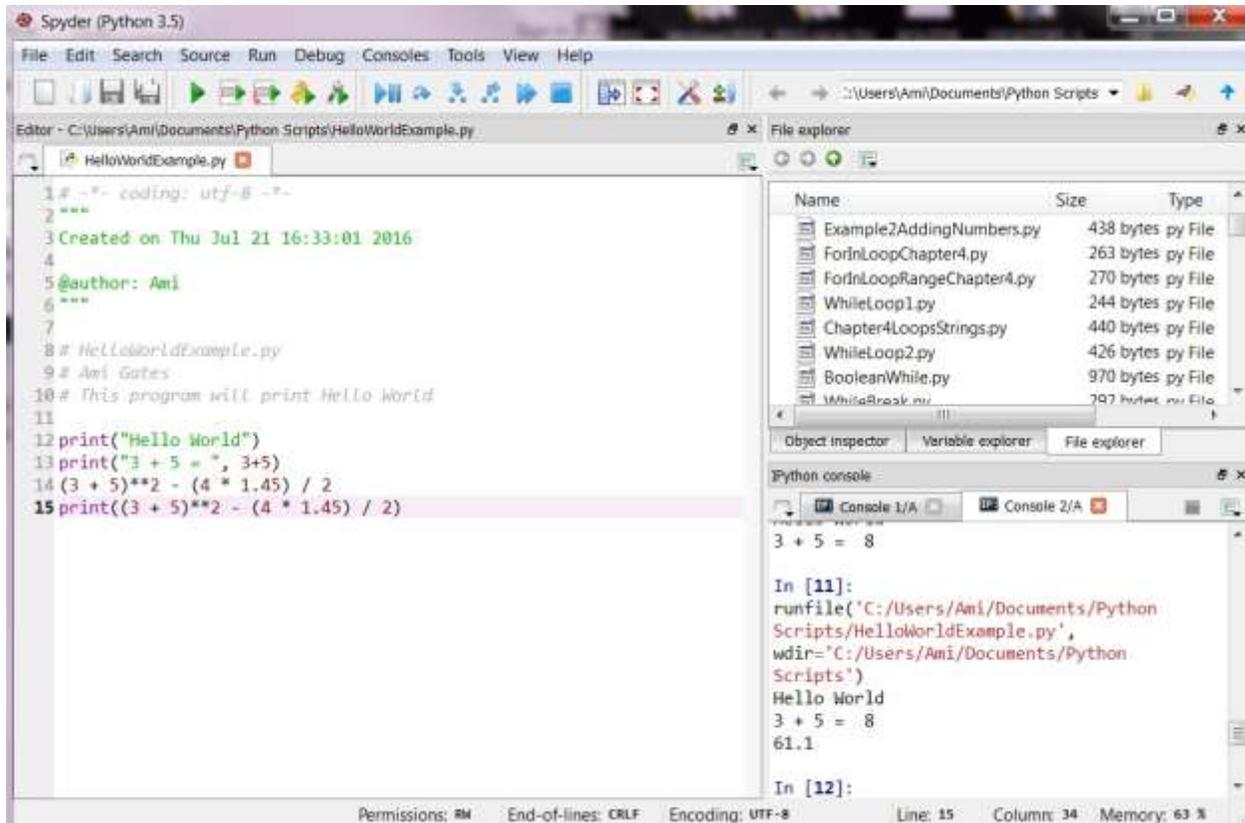


Figure 13: The output of running the updated `HelloWorldExample.py` file.

### Exercise 1.3: Locating and opening Python files

Notice, in Figure 13, that when the Python file called `HelloWorldExample.py` was run, the IPython Console showing the output also displayed the path or location to the file. In this case, the exact location on my computer of the file called `HelloWorldExample.py` is `C:\Users\Ami\Documents\Python Scripts`.

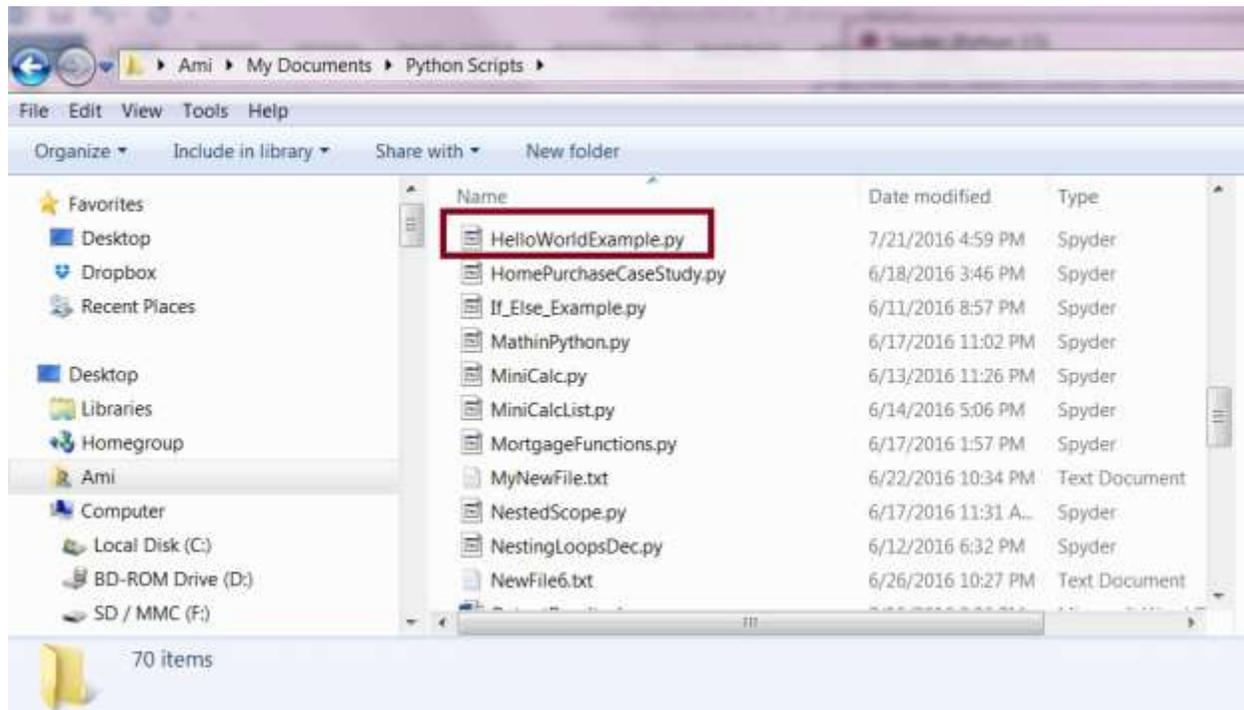
Explore your computer and locate the `HelloWorldExample.py`. Note that the folder (path) on your computer will be different from mine. Figure 14 will illustrate the location of `HelloWorldExample.py` on my computer as an example. It is important to be able to locate your Python programs (also called scripts), as well as the folder on your computer where they reside. Later in the book, it may be necessary to create files within that folder so that Python can access them (such as data files).

Publication Pending

Author: Dr. Ami Gates

Date: 8/11/2016

All material is subject to copy write laws. Do not post, print, or reproduce without written permission.



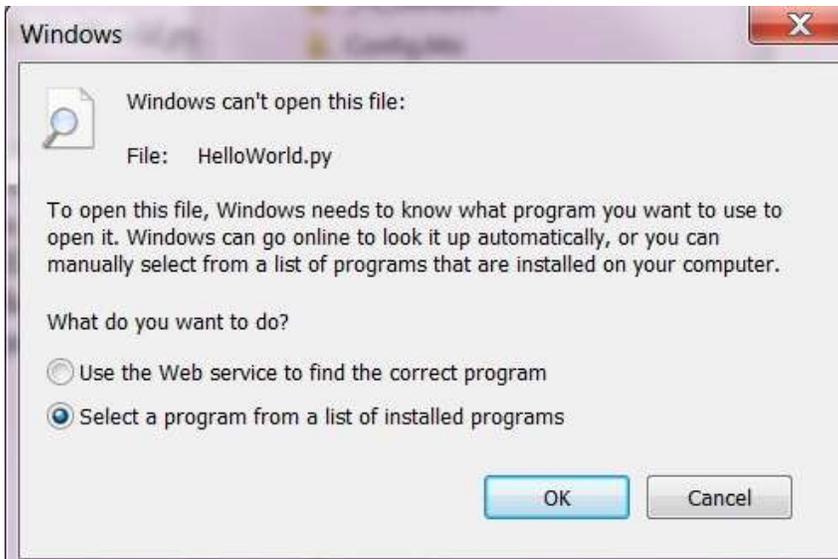
**Figure 14:** An example of the actual location on Ami's computer where the Python programs and the Hello World Example.py program are stored.

### Exercise 1.4: Associate Spyder with .py files

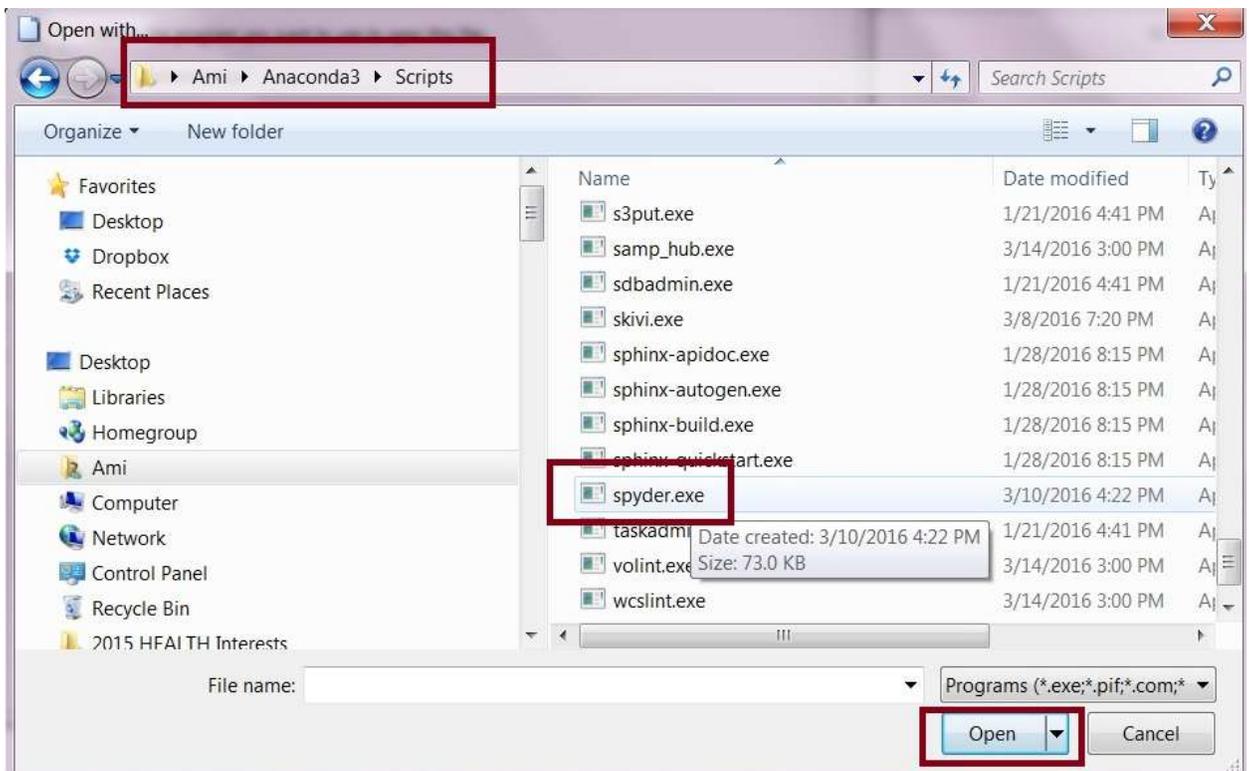
For this Exercise, first gain access to the location of the file, HelloWorldExample.py, on your computer (not through Spyder) as discussed in Exercise 1.3. Next, close the Spyder IDE if it is open. Then, from your computer, double click the file called HelloWorldExample.py.

If this is the first time doing this, a box may pop up asking you to associate a program on your computer with this type of file extension. In other words, your computer is asking you what you want it to do with files that end in .py (Python files).

To answer, first choose, "Select a program from a list of installed programs." Then choose OK. Your computer will give you a list of options. The likely probability is that Spyder will not be one of the options. From here, you must find **Spyder.exe** and associate it with this type of file. Figures 15 and 16 will illustrate these steps.



**Figure 15:** The pop up window that will occur the first time a .py file is double-clicked



**Figure 16:** An illustration of how to locate the spyder.exe file to choose to associate with .py files.

As a note, it is also possible to accomplish the same file association between Python .py files and the Spyder IDE, by right-clicking (in Windows) the file called HelloWorldExample.py, choosing “Open with...”, and finding and choosing Spyder.exe. This process may take a few tries.

To test that the association between .py files and Spyder has been successful, follow these steps:

- 1) Close Spyder (if it is open)
- 2) Go the file called HelloWorldExample.py via the folder on your computer where you located it (see Exercise 1.3).
- 3) Double click the file. If the association is working properly, Spyder will automatically open and allow the running of the file, HelloWorldExample.py.

### Exercise 1.5: Utilizing the File explorer in Spyder

In general, to create or edit programming projects, first open Spyder. Then code, save, edit, and run files within the Spyder IDE. Within Spyder (generally on the upper right) is an area that offers a view of all files located in the Python Scripts folder (directory). Figure 17 illustrates the location and my current personal contents of this area.

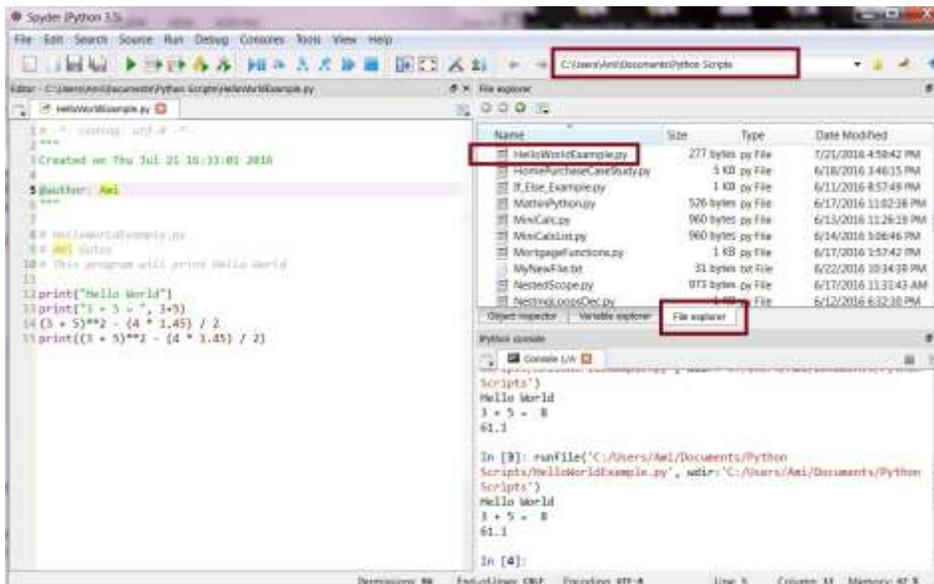


Figure 17: An illustration of the File explorer area in Spyder

Publication Pending

Author: Dr. Ami Gates

Date: 8/11/2016

All material is subject to copy write laws. Do not post, print, or reproduce without written permission.

Each time a new Python file is created in Spyder, it will appear in the File explorer area. In addition, if a file is placed in this location on your computer, it will also appear in the File explorer. The File explorer is a convenient view of the folder that contains all current Python programs and any other files created or placed into this location.

As an exercise, create a new file in Spyder called, MyNewFile.py. Add a few statements to the file. Save the file and locate the new file within the File explorer.

### **Chapter 1 Summary**

The key to this chapter is to take all the necessary steps to create a Python 3 programming environment. Once these steps and practice exercises are completed, you will be ready to move forward with learning Python. While all examples in this book will utilize Spyder, these same (or similar) steps can be taken for any IDE or on a Python command line.