



CS-1004, Introduction to Programming
for Non-Majors, A-Term 2018

Setting up Python 3.6.5, matplotlib, and numpy on your own Windows PC or laptop

Hugh C. Lauer[©]
Teaching Professor
Worcester Polytechnic Institute

Programming assignments in CS-1004 will be carried out in the programming language *Python* — specifically, version 3.6.5 of *Python*. In addition, you will need several *Python* packages, including one called *matplotlib*, a *Python* version of the popular *Matlab* system, and one called *numpy* (meaning “Numerical Python”). This document provides instructions for installing *Python 3.6.5* on *Windows 7*, *Windows 8*, and *Windows 10* laptop and desktop computers. This document also includes instructions for installing *Python* packages such as *matplotlib* and *numpy*.¹

Public laboratory computers at WPI will have *Python 3.6.5*, *matplotlib*, and *numpy* installed on them for the academic year 2018-2019.

In general, it is expected that assignments will be compatible among Windows, Macintosh, and Linux systems, assuming they all use compatible versions of *Python*, *matplotlib*, and *numpy*.

Note: Two different, incompatible versions of *Python* are in general use around the world — *Python 2.7* and variations of *Python 3*. Significant changes were made to the *Python* language between *Python 2.x* and *Python 3*. The *Python 3* language is cleaner, more self-consistent, and more user-friendly. Programs written for versions of *Python 2* will not necessarily run on *Python 3* installations; if they do run, they may get **different answers** to the same problem.

Note that a lot of legacy *Python 2* code is still in use, and new *Python 2.7* code is still being written and distributed by organizations that have not yet upgraded to *Python 3*. Not all *Python 2* packages have been ported to *Python 3*.

Note 2: There are many integrated environments for supporting Python programming, including *PyCharm* and *Enthought*. Most of these are more advanced than what is needed for this course. If you choose to use one of them, you are on your own for installation.

[©] Copyright 2018, Hugh C. Lauer. All rights reserved. Permission is given for use in courses at Worcester Polytechnic Institute, Worcester, Massachusetts.

¹ If you have a Macintosh or Linux computer or laptop, please refer to these documents instead:— [docx](#), [pdf](#)

Installing Python 3.6.5 on Windows Systems^{2, 3}

Python 3.6.5 for Windows exists in two variations — a 32-bit version and a 64-bit version. Almost all Windows PCs sold over the past few years are 64-bit systems. Therefore, these instructions focus primarily on installing the 64-bit versions. If you have a 32-bit version of Windows, please seek assistance from the Professor, the TAs, or the Helpdesk.

To obtain the correct 64-bit version of *Python*, click on this link — [python-3.6.5-amd64.exe](#)⁴ — and download the resulting file to a convenient folder or directory. Alternatively, you may browse to

<http://www.cs.wpi.edu/~cs1004/a18/Resources-A18>

and download it from there.

Right-click on your downloaded copy of the file **python-3.6.5-amd64.exe** and select *Run as Administrator* to start the installation. You should be greeted by a dialog box resembling the following:–

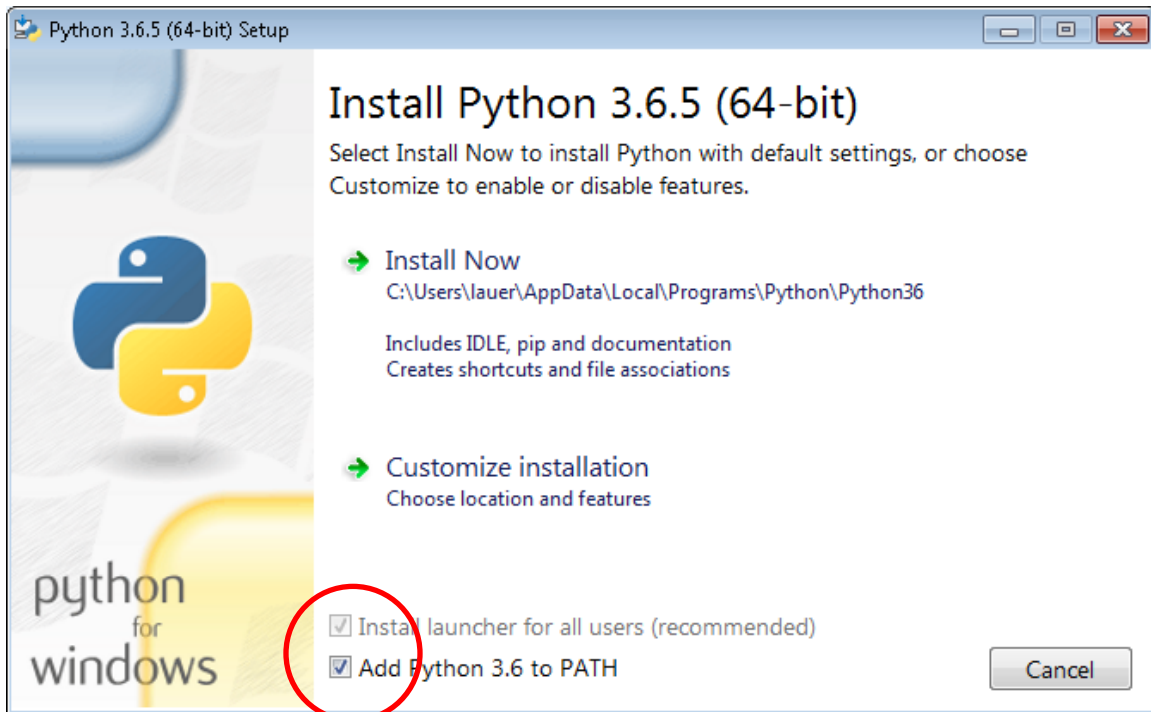


Figure 1

Be sure to check the box at the bottom labeled “Add Python 3.6 to PATH” (shown in a red circle).

² It is useful to print out the relevant sections of this document. If you try read them on-screen, the dialog boxes of the installation tend to obscure the instructions, just when you need them the most!

³ These instructions have been tested on both *Windows 7* and *Windows 10*. The professor no longer has access to an installation of *Windows 8* for testing.

⁴ The processor designation “**amd64**” applies to microprocessors made by *both* Intel and AMD.

Setting up Python 3.6.5 on Windows

Note1: The “Add Python 3.6 to PATH” checkbox may seem trivial but forgetting to check it has led to numerous problems for students in previous terms!

Note 2: It is *essential* that you run the installation as *Administrator*. If you forget to so, the installation will appear to proceed successfully, but you are likely to get weird errors at runtime and even in the installations of *numpy* and *matplotlib*.

If you forgot to select *Run as Administrator*, uninstall *Python* and start over.

If all goes well, you should see a window resembling the following:–

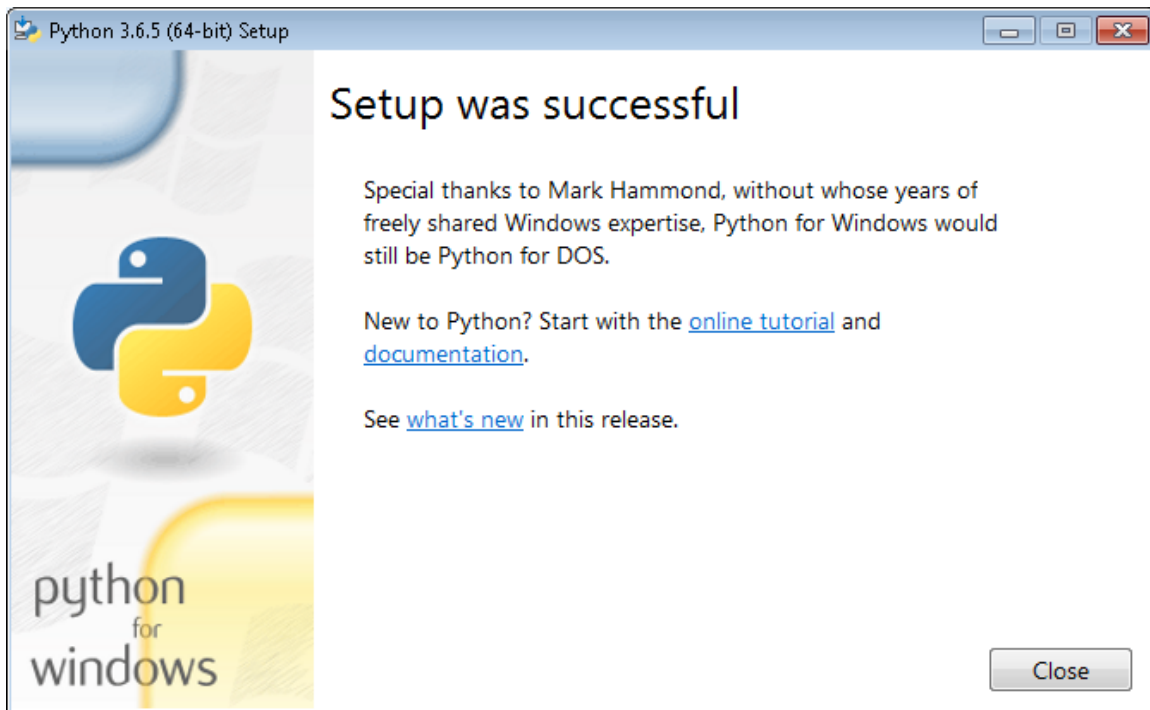


Figure 2

Three big “ifs”:–

- If an earlier version of *Python 3.x* is already installed on your computer, you may see a window resembling Figure 3 (below) instead of Figure 1. In this case, you may proceed with the installation of *Python 3.6.5* by clicking *Upgrade Now*. If successful, you will see a window resembling Figure 2 above.
- If you have a 32-bit version of Windows (an unlike event nowadays), you should cancel this installation and download and install [python-3.6.5.exe](#) instead. You are strongly encouraged to seek help for 32-bit issues.
- If something else goes wrong, please contact the Professor, the TAs, and/or the Helpdesk to get a working version of *Python 3.6.5* installed on your computer.

You should do this right away. *There won't be enough time in the days before the first assignment is due.*

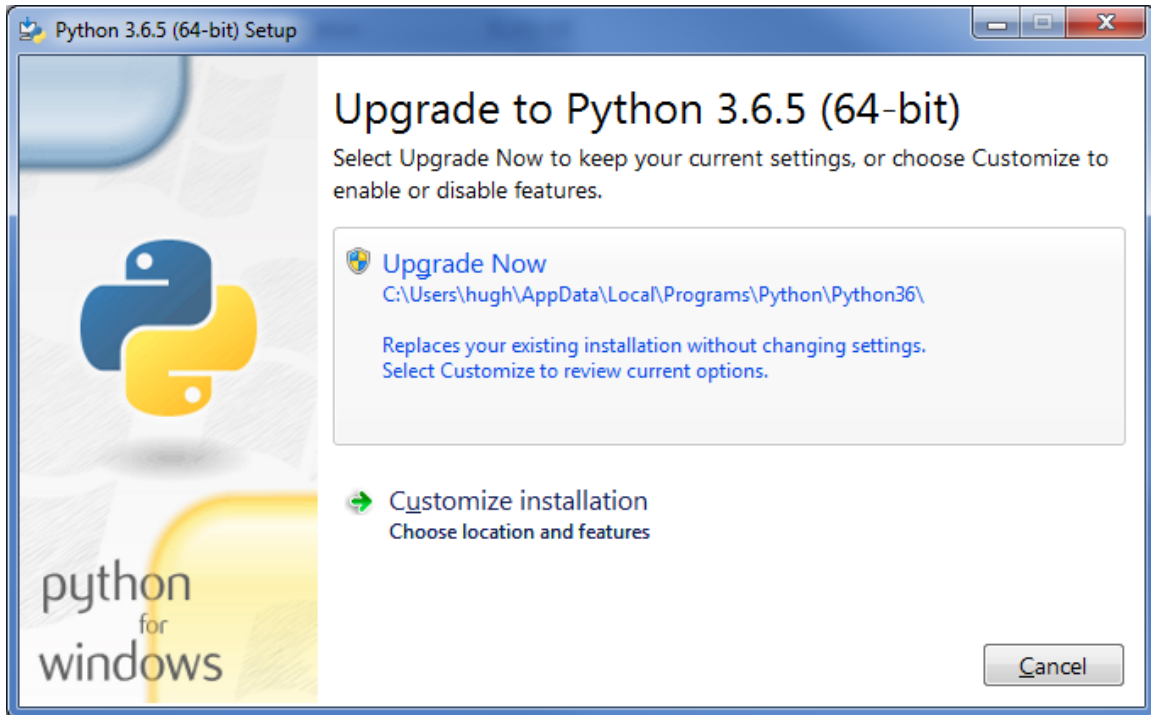


Figure 3

Customized installation

If you share a computer with someone else (unlikely) or have a non-standard installation of Windows, you *may* have to invoke the *Customize Installation* button in Figure 1 or Figure 3. This will bring up a dialog box similar to that of Figure 4. Be sure all the boxes are checked, and then click *Next* to bring up a dialog similar to Figure 5.

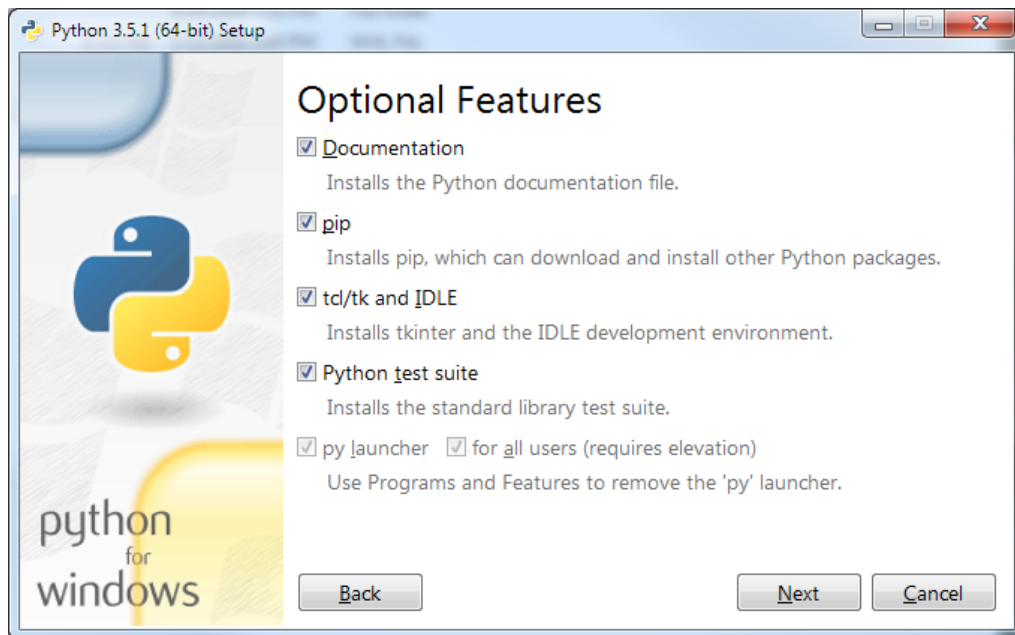


Figure 4

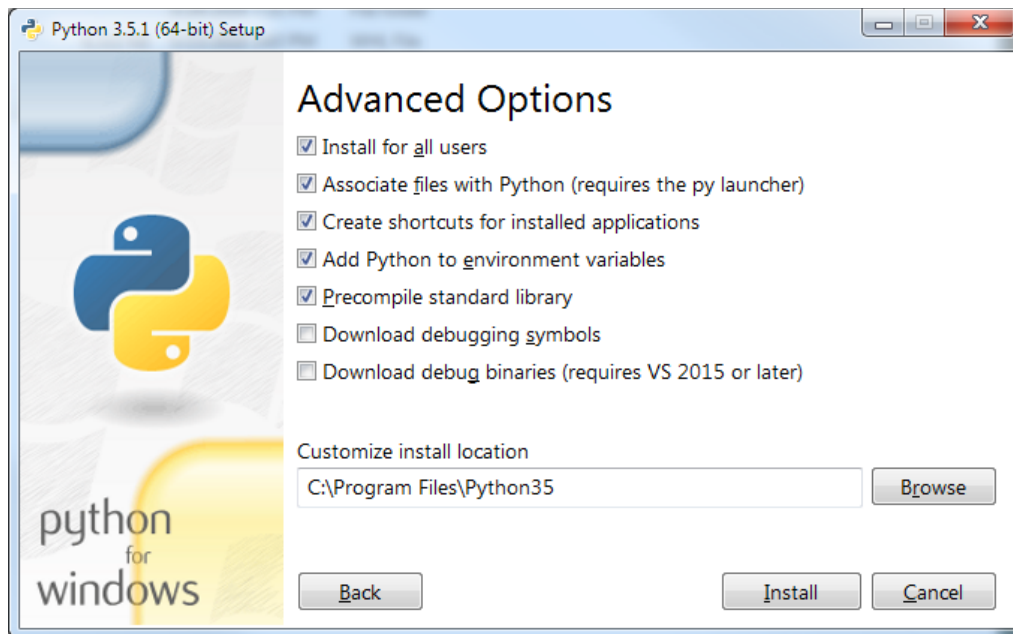


Figure 5

Click *Install for all users* to force the installer to choose a commonly accessible directory for all the users of this computer. Be sure that the check boxes in your installation match those in Figure 5.⁵

Click *Install* to begin the installation. If the installer tells you the directory already exists and asks if you are sure that you want to overwrite existing files, click *Yes*. The progress of the installation will be shown in the dialog box.

The installation will take several minutes and will finish with the dialog box of Figure 3 on page 4 of this document. Click *Close* to complete the installation of *Python 3.6.5*.

Testing your installation

The following two sections pertain to testing on Windows 7 and Windows 10, the two most common Windows versions in common use. In the (unlikely) event that you have Windows 8, please refer to the appendix of this document for testing.

Testing on Windows 7

To confirm and test your installation, we will start *IDLE*, the *Python Integrated Development Environment* window. If you are running *Windows 7*, click the *Start* button to bring up the Windows *Start* menu. Select *All Programs* and scroll down to *Python 3.6.5*. This is a folder shown circled in the left side of Figure 6.

When you open this folder, you will see four options, shown in the right side of Figure 6. Select *IDLE (Python 3.6 64-bit)* to start the *IDLE* program shown in Figure 7.

⁵ The “Customize install location” near the bottom of Figure 5 will refer to an appropriate folder for your computer.

Setting up Python 3.6.5 on Windows

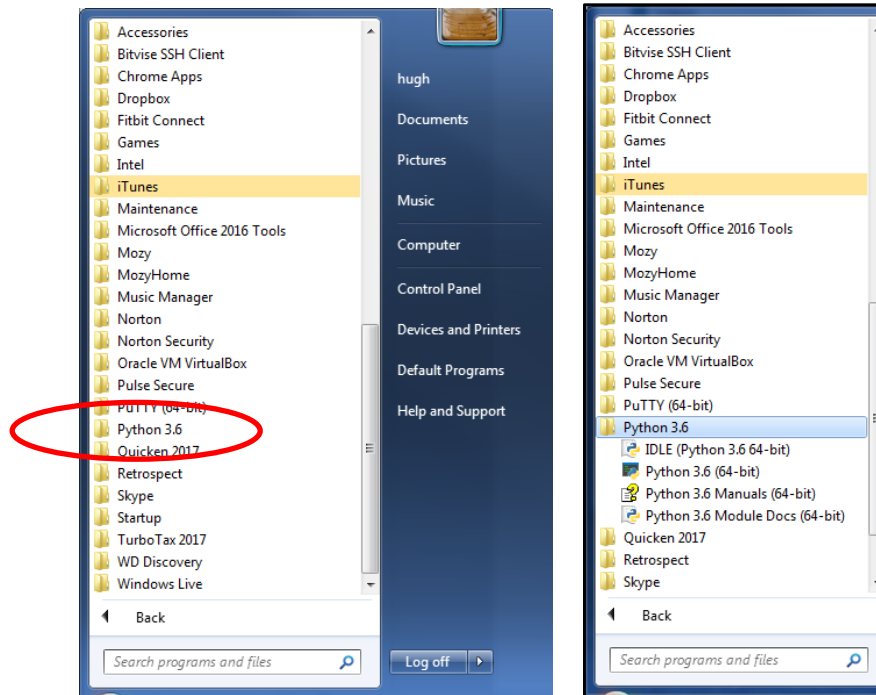


Figure 6

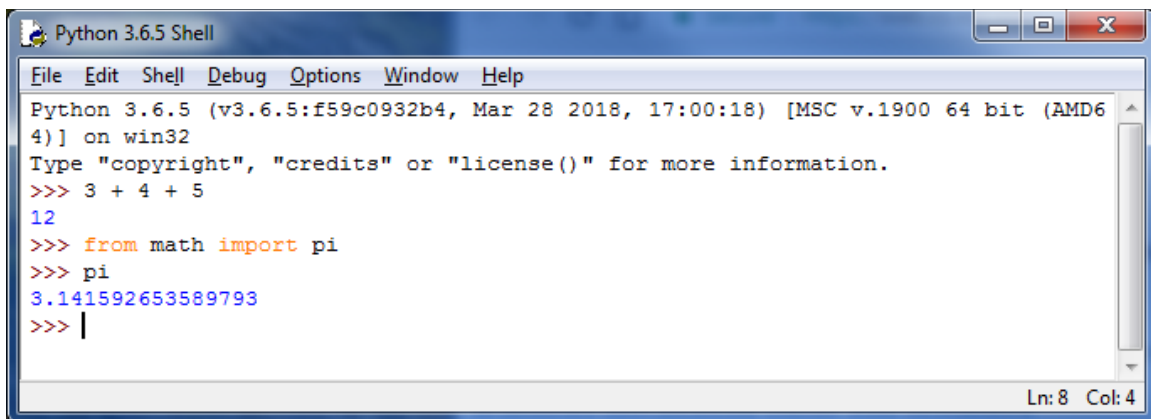


Figure 7

This is *IDLE*, the *Python* command prompt and graphical user interface. This is where we will start all programs and projects in this course. For now, simply type any *Python* statement or expression after the “>>>” prompt. For example, Figure 7 shows the expression

$$3 + 4 + 5$$

as typed, followed by *Python*'s response of the value 12.

After the next two “>>>” prompts, type the command

```
from math import pi
```

and the expression

```
pi
```

Python responds by printing the value of *pi* to 15 decimal places.

Setting up Python 3.6.5 on Windows

Continue testing by typing out the code on pages 10-11 of the textbook, just to make sure that your installation works as expected.

Note that if you mistype something, you will probably get some unexpected behavior. Simply try again. If you get really lost, type CTRL-C to get back to the IDLE prompt.

Testing under Windows 10

Windows 10 has something like the *Start* button, but it looks a little different — the symbol



at the left end of the taskbar. Click on this to bring up a menu something like Figure 8 below:—



Figure 8

In most cases, *IDLE (Python 3.6.5 64-bit)* should be shown under “Recently added” (five lines up from the bottom of this Start Menu). If you don’t see it there, click on *All apps*, at the bottom of the *Start Menu*. This will provide an alphabetical list of all applications. *Python 3.6.5* appears under “P” and shows the same four options as the right side of Figure 6. Since you will be using this a lot, it is suggested that you right click on *IDLE (Python 3.6.5 64-bit)* and select either *Pin to Start* or *More > Pin to Taskbar*.

To test your installation, double-click on this icon and carry out the same tests as shown above under Figure 7.

Congratulations! You now have a usable *Python 3.6.5* running on your *Windows* computer or laptop.

Installing `graphics.py`

Graphics.py is a simple drawing package that we will use a lot in this course. It was written in *Python 3* and created by the textbook author for making simple drawings. To install it, click on this link — [graphics.py](#) — and download the file to the folder where you keep your *Python programs*. Follow the instructions in sections 4.3-4.8 of the textbook (3rd edition).

Installing *matplotlib*, *numpy*, and other packages

These steps require you to have a working internet connection.

One of the many benefits of *Python* is the vast number of third-party packages⁶ that can be downloaded and used by your *Python* programs. Many of these are open-source and free. For this course, we will use at least the following:–

- *matplotlib* (a package for creating 2D plots and graphs similar to *Matlab*),
- *numpy* (meaning “Numerical Python,” a package for efficient handling of large arrays of numerical data, also needed by *matplotlib*), and

The installation of these packages has been simplified from that of previous years. There is no need to download these packages separately. Instead, you will use a *Windows Command Prompt* to run **pip**, the *Python Installation Program*.

A *command prompt* is a window into which you type “commands” to tell the computer and operating system what to do. An example *command prompt* window is shown in Figure 10.

In this window, the system prints a prompt starting with the “path” of the current folder and ending with a “>” character. After the prompt, you type a *command*, consisting of a command name followed by zero or more operands that control what the command does. When you terminate the command with the *Enter* key, the system performs the command.

A *command* may work silently and then print out its results in the same window, or it may engage in a textual conversation with you, requiring you to respond by typing, or it may open its own window with its own graphical user interface. When the command has completed, the system prints a new prompt for the next command. The command consisting solely of the word “**exit**” causes the system to close the command prompt window.

To install *matplotlib*, you must run the Command Prompt in *Administrator* mode.

- In *Windows 7*, click on the *Start* button, select *All Programs*, scroll to the *Accessories* folder and open it. This folder will show many useful accessories, such as shown in Figure 9.

⁶ Recent reports indicate that there are at least 45,000 different packages covering many fields of science, engineering, and mathematics.

Setting up Python 3.6.5 on Windows

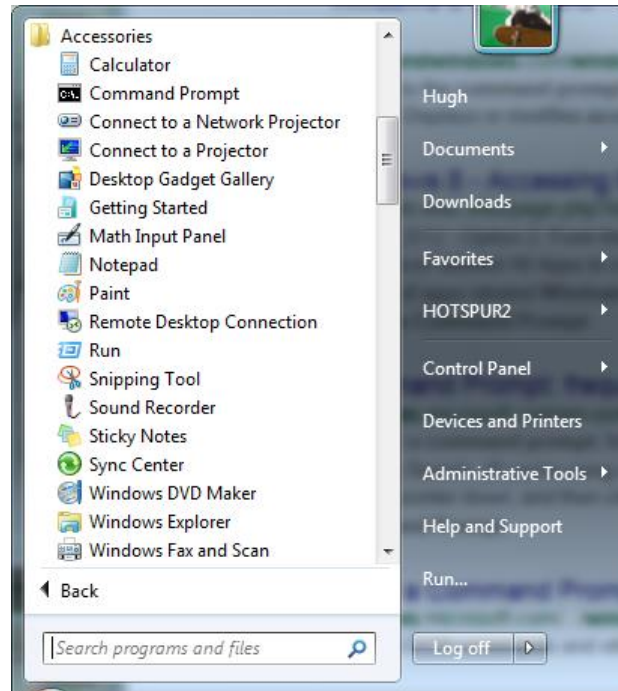


Figure 9

Right click on “Command Prompt” on the second line and select *Run as Administrator*. Windows will ask you if you really want to do this. Click yes. This will open a window into which you can type textual commands.

- In *Windows 8*, search for “Command Prompt” in the pallet of “charms,” the same way you searched in Figure 19. Drag an icon for the *Command Prompt* to your desktop, right-click on it, and select *Run as Administrator*.
- In *Windows 10*, click on the *Windows* icon at the left end of the taskbar, select *All apps*, and scroll down to *Windows system* (listed alphabetically). Open the *Windows system* item to expose a list of system applications, including *Command Prompt*. Right-click on *Command Prompt* and select *More > Run as administrator*.

If you are unable to find the *Command Prompt* or to run it as *Administrator*, seek help from the Professor, one of the TAs, or the Helpdesk. If you are successful, you should be presented with a window resembling Figure 10 below. The background is likely to be black or some other dark color. Note that the header of this window labels it as an *Administrator Command Prompt*.⁷

⁷ This means that you have a lot of power to do good or evil to your computer!

Setting up Python 3.6.5 on Windows

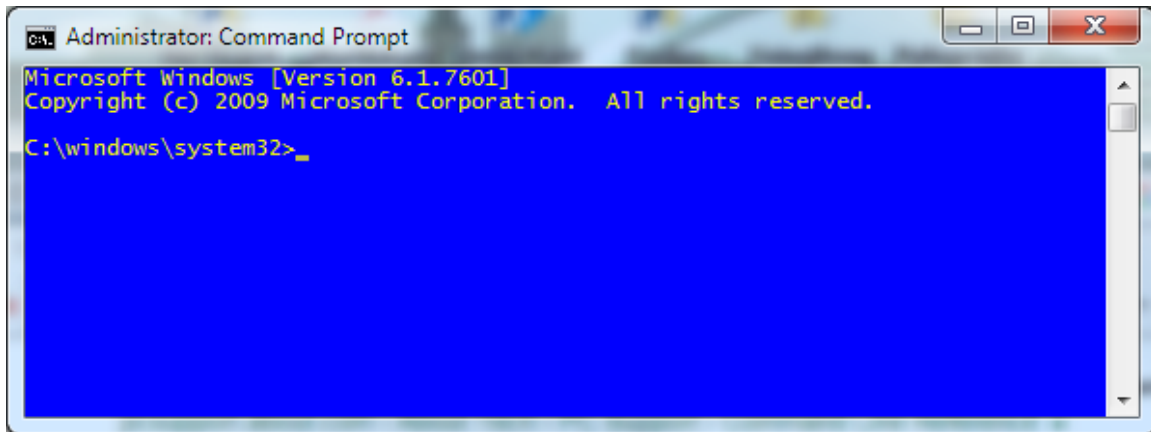


Figure 10

Next, in the *Command Prompt* window of Figure 10, type the following three commands, exactly as written here:–

```
python -mpip install -U pip
python -mpip install -U matplotlib
python -mpip install -U nose
```

In these commands, **pip** is the *Python Installation Program*. It knows how to find packages on the web and to keep versions consistent with each other. The **matplotlib** package is one that we will use extensively in this course. Installing it automatically causes the correct supporting version of **numpy** to be installed. Finally, **nose** is a comprehensive test package for the **numpy** package; we will use it to verify our installation.

Note: These commands will access the internet to retrieve files from an online *Python* distribution center.

Any of these commands may report errors, including that a package is already installed. To check what is installed, type the command

```
pip list
```

This will list the programs that **pip** knows about — shown, for example, Figure 11.

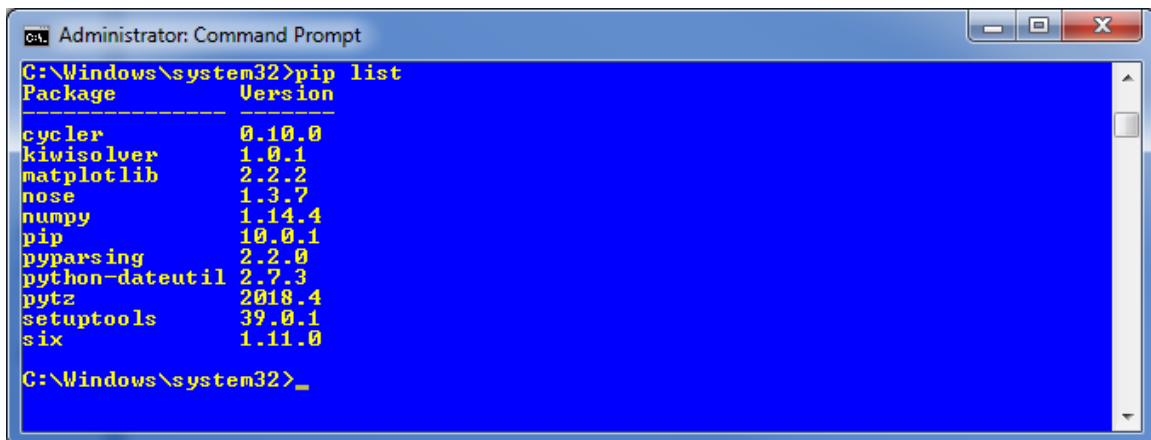


Figure 11

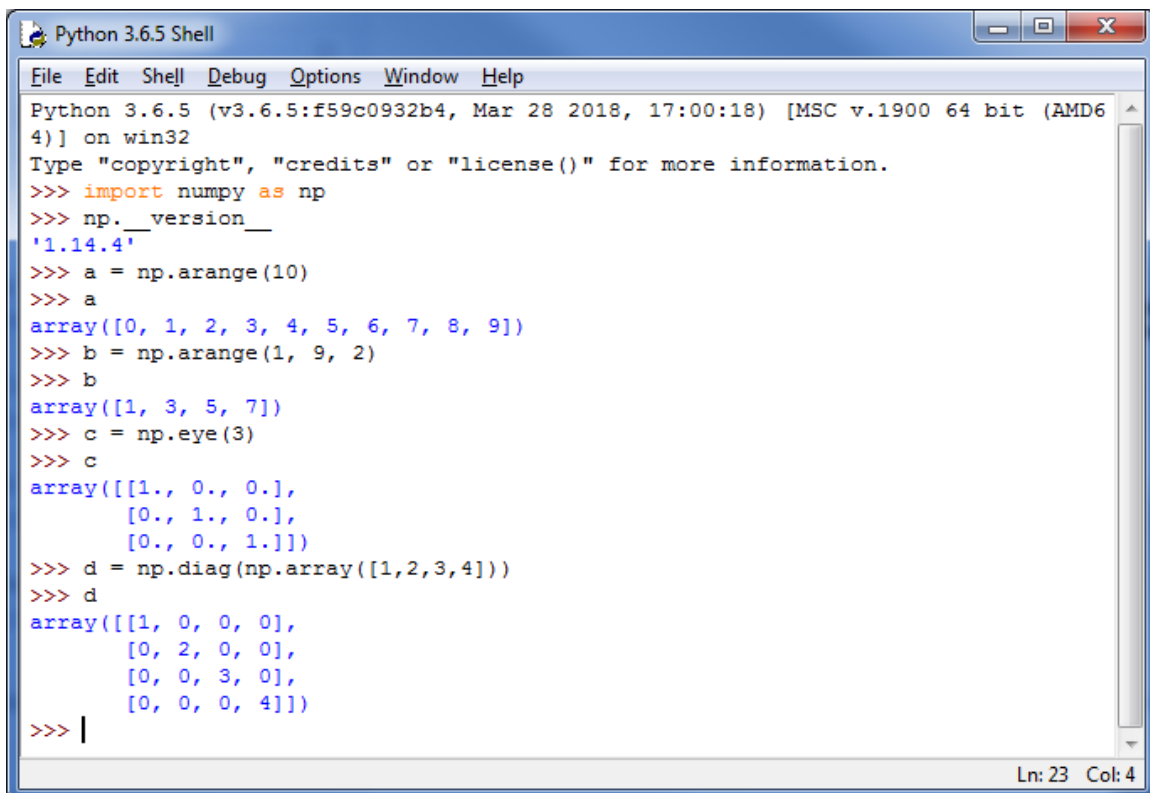
Setting up Python 3.6.5 on Windows

When this installation is complete, make a quick test *numpy* by opening an *IDLE* window, as in Figure 7. Type or paste the following commands into *IDLE*, *one line at a time, exactly* as written. Note that the word **version** is preceded by *two* underscore characters and followed by two more. Also note that the word **arange** is spelled with *one* “r”.⁸

Note: Even if you had help installing *Python* and *matplotlib*, you should type out the test lines below *yourself*. This will help you get used to *Python*.

```
import numpy as np
np.__version__
a = np.arange(10)
a
b = np.arange(1, 9, 2)
b
c = np.eye(3)
c
d = np.diag(np.array([1, 2, 3, 4]))
d
```

The result should resemble Figure 12:–



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import numpy as np
>>> np.__version__
'1.14.4'
>>> a = np.arange(10)
>>> a
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> b = np.arange(1, 9, 2)
>>> b
array([1, 3, 5, 7])
>>> c = np.eye(3)
>>> c
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
>>> d = np.diag(np.array([1,2,3,4]))
>>> d
array([[1, 0, 0, 0],
       [0, 2, 0, 0],
       [0, 0, 3, 0],
       [0, 0, 0, 4]])
>>> |
```

Figure 12

Finally, type the **exit** command to close the Administrator Command Prompt of Figure 10.

⁸ This is a variation of the “**range**” function that we will learn about in the first week of class.

Testing Your Installation

Testing *numpy*

To carry out the comprehensive test of *numpy*, open a new *IDLE* window and type the following two commands:–

```
import numpy
numpy.test()
```

This uses **nose** to run the standard package of *numpy* tests for several minutes. It prints a bunch of stuff in the *IDLE* window. Although some of the output may look like error messages, these are known issues with the tests. The test should start with something resembling the following:–

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import numpy
>>> numpy.test()
Running unit tests for numpy
NumPy version 1.14.4
NumPy relaxed strides checking option: True
NumPy is installed in C:\Users\hugh\AppData\Local\Programs\Python\Python36\lib\site-packages\numpy
Python version 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)]
nose version 1.3.7
.....K.....
Ln: 47 Col: 96
```

Figure 13

After a few minutes, the tests should end with something resembling the following:–

```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Ran 6419 tests in 48.844s
FAILED (KNOWNFAIL=19, SKIP=15, errors=2)
<nose.result.TextTestResult run=6419 errors=2 failures=0>
>>> |
Ln: 80 Col: 4
```

Figure 14

Congratulations! You have now installed a working versions *numpy 1.14.4* and *matplotlib 2.2.2*.

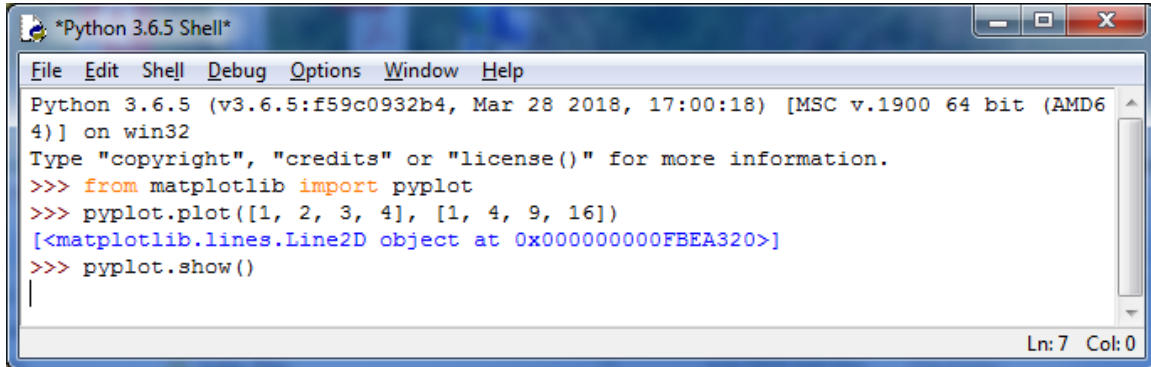
Testing *matplotlib*

You can carry out a simple test of your *matplotlib* installation by typing or pasting the following commands into an *IDLE* window, one line at a time, *exactly* as written:–

Setting up Python 3.6.5 on Windows

```
from matplotlib import pyplot
pyplot.plot([1, 2, 3, 4], [1, 4, 9, 16])
pyplot.show()
```

The IDLE window should look something like the following:–



```
*Python 3.6.5 Shell*
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> from matplotlib import pyplot
>>> pyplot.plot([1, 2, 3, 4], [1, 4, 9, 16])
[<matplotlib.lines.Line2D object at 0x000000000FBEA320>]
>>> pyplot.show()
|
Ln: 7 Col: 0
```

Figure 15

After you type the **ENTER** key following the last line, the following window should appear:–

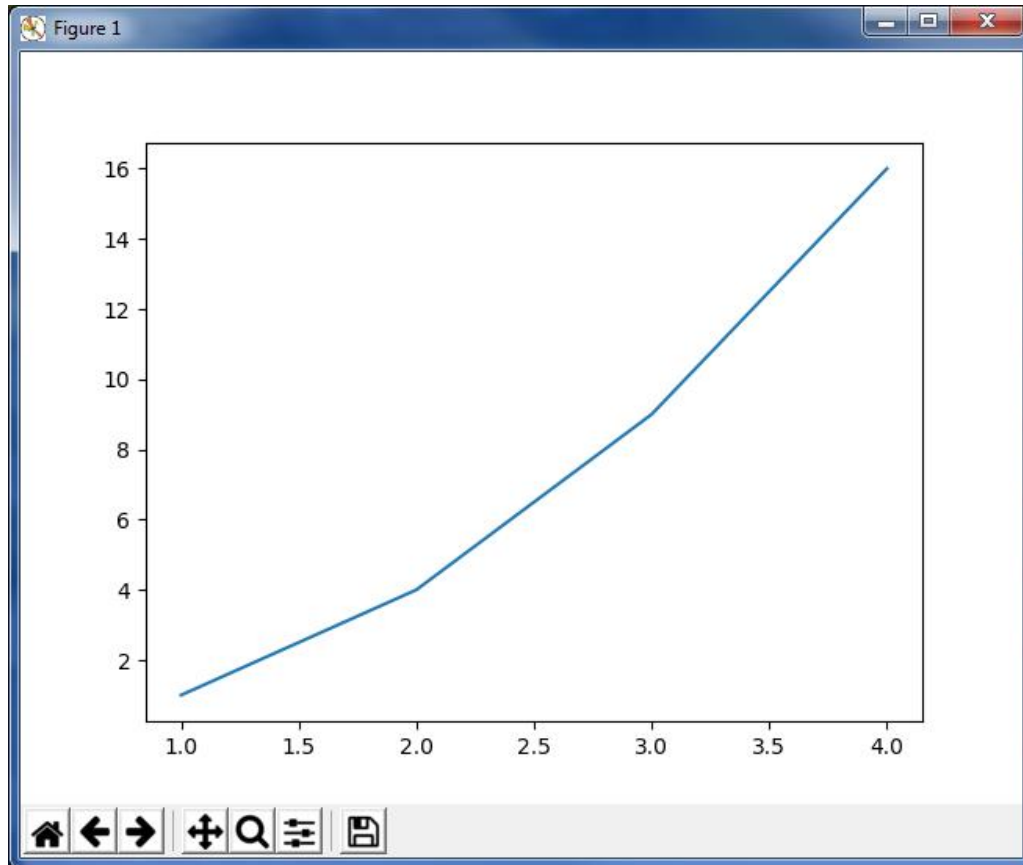
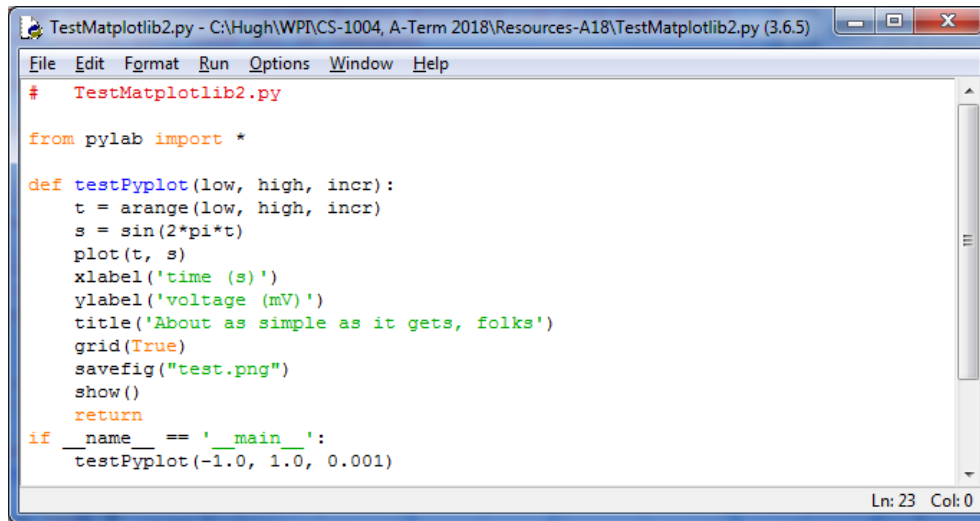


Figure 16

To close this window, click on the “close” button in the upper right.

For a more interesting test, download the file [TestMatplotlib2.py](#), and then use the *File* menu in the *IDLE* window to open this file. The file should look like the following:–

Setting up Python 3.6.5 on Windows



```
TestMatplotlib2.py - C:\Hugh\WPI\CS-1004, A-Term 2018\Resources-A18\TestMatplotlib2.py (3.6.5)
File Edit Format Run Options Window Help
# TestMatplotlib2.py

from pylab import *

def testPyplot(low, high, incr):
    t = arange(low, high, incr)
    s = sin(2*pi*t)
    plot(t, s)
    xlabel('time (s)')
    ylabel('voltage (mV)')
    title('About as simple as it gets, folks')
    grid(True)
    savefig("test.png")
    show()
    return
if __name__ == '__main__':
    testPyplot(-1.0, 1.0, 0.001)

Ln: 23 Col: 0
```

Figure 17

Click the *Run > Run Module* command in the menu at the top of the window to produce the following window:–

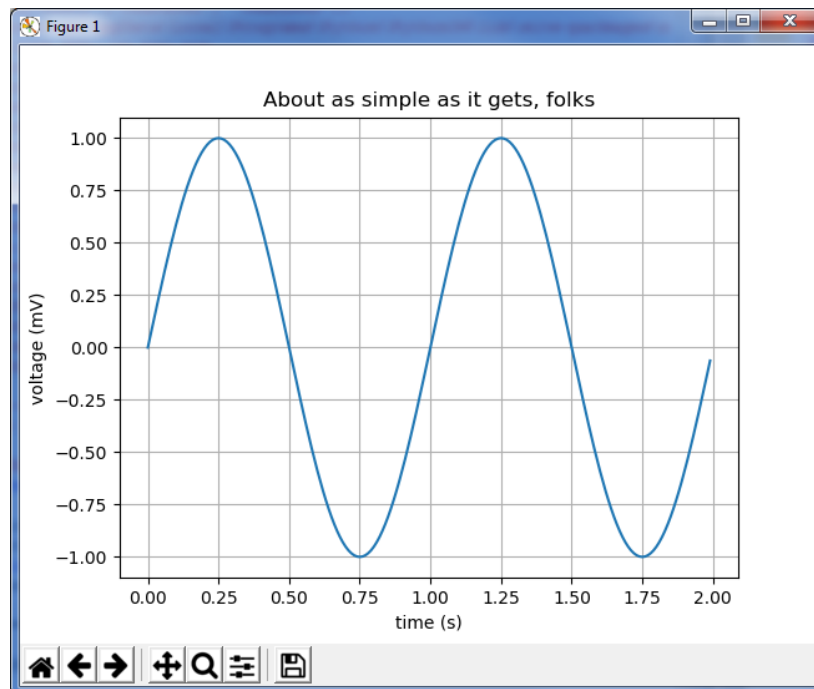


Figure 18

Congratulations! You now have a working version of *matplotlib* installed.

Appendix

Testing in Windows 8

Windows 8 does not have a *Start* button but rather a *Start* screen that is intended to make the user experience more like the smartphone experience. Unfortunately, when *Python* is installed as instructed above, its icon does not automatically appear on the *Start* screen. It also does not appear in the list of apps.

To find it, move the cursor to the upper-right or lower-right corner of the screen to expose the *Windows 8* pallet of “charms”. Select the *Search* charm to bring up a *Search* box. Type the word “Python.” This will bring up a list of matching items, like that shown in Figure 19 below but referring to the 64-bit version of *Python 3.6.5*.⁹

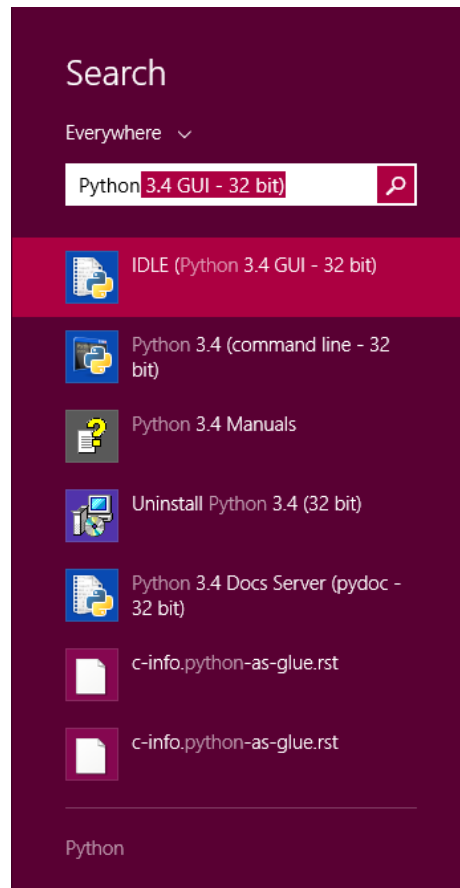


Figure 19

Note that this list is like the *Python 3.6.5* folder in the Start Menu in Figure 6. *Right-click* on the item labeled *IDLE (Python 3.6.5 GUI)*. From the resulting menu, select “Pin to Start” to cause an icon to be added to the *Start* screen. You may also want to pin the item to the *Task bar* (i.e., the bar of tiny icons at the bottom of the screen). You may also select “Open file location,” which will bring up the following window:–

⁹ Note that Figure 19 and Figure 20 refer to an older version of Python because they were copied directly from an older version of this document. If you follow these instructions for Windows 8, you should see the current version of Python that you downloaded on page 2.

Setting up Python 3.6.5 on Windows

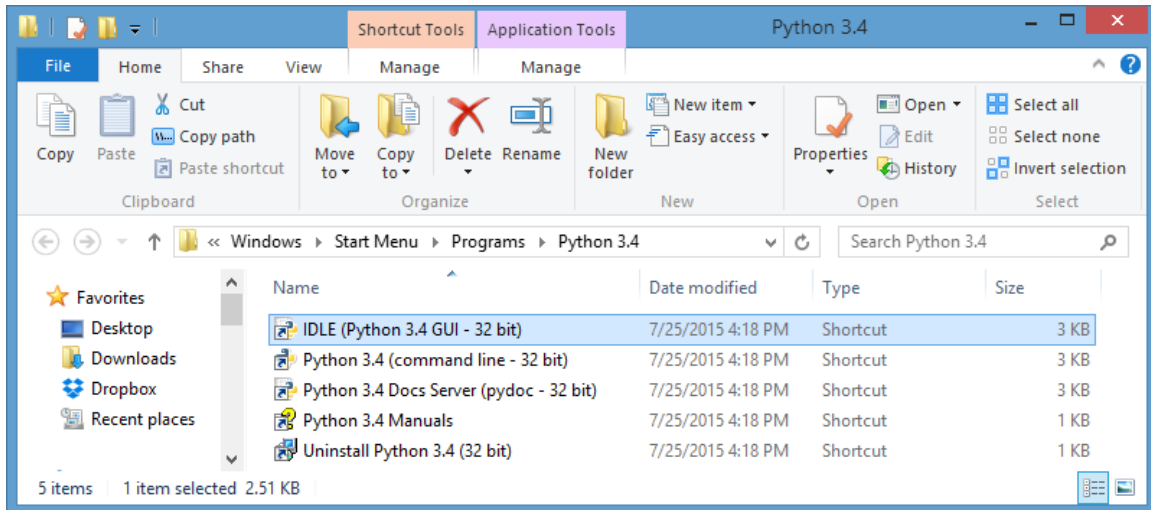


Figure 20

From this window, you can copy any or all the *Python* links to the desktop.

To test your installation, double-click on the *IDLE (Python GUI)* icon and carry out the same tests as shown above under Figure 7.