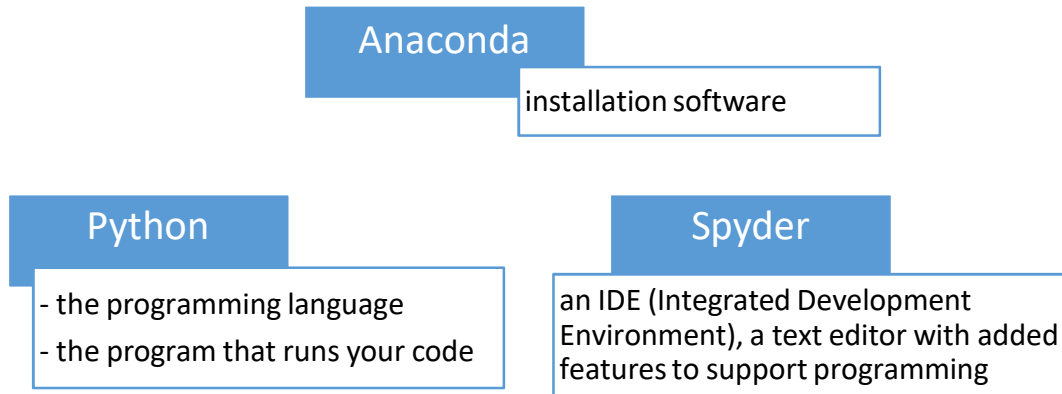


GETTING STARTED WITH PYTHON USING WINDOWS

March 3, 2017

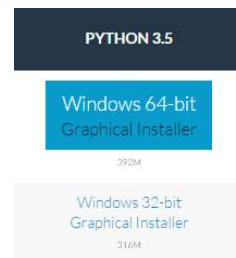
Python is a programming language, and you'll be writing your Python code in the programming environment called Spyder. The Anaconda distribution simplifies the installation process by including Python, Spyder, and other packages and tools in one installation file.



Installing Anaconda

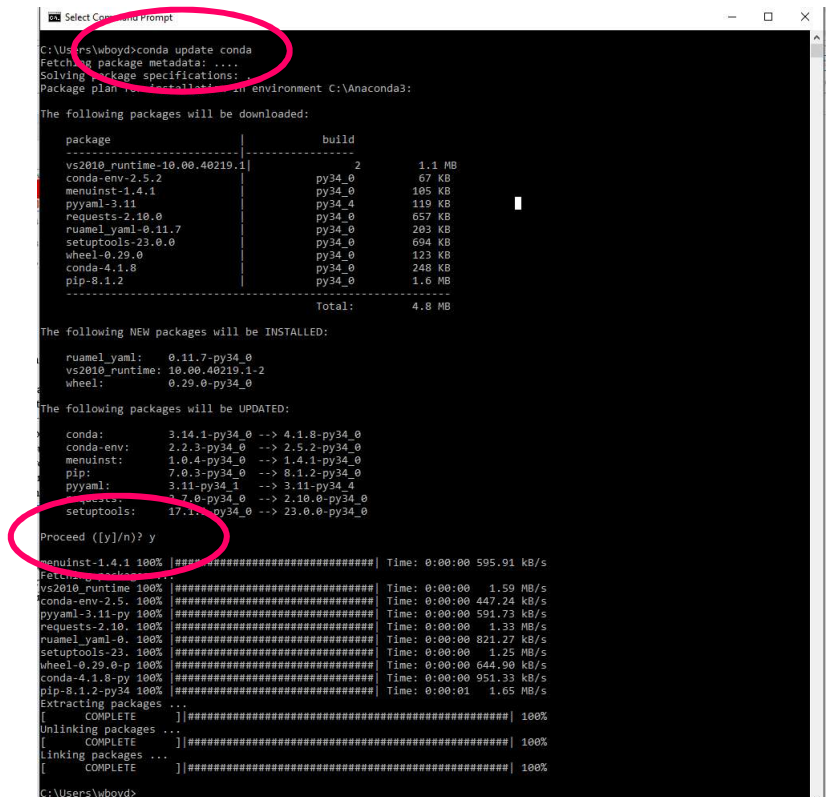
1. Go to <http://continuum.io/downloads>
2. Scroll down to find your operating system and click on Python 3.5 (or the latest version) to download the graphical installer.

For windows, see <http://windows.microsoft.com/en-us/windows/32-bit-and-64-bit-windows> to find out whether your computer has a 32-bit or 64-bit version of windows. Click on the FAQ “*How can I tell if my computer is running a 32-bit or 64-bit version of windows?*”, and follow the instructions to find out. If your computer is running a 32-bit version of Windows, click on “Windows 32-bit Python 3.5 Graphical Installer” under OTHER INSTALLERS, instead of the Windows 64-bit installer, and follow the same directions to install it. Actually, any Python 3.x version from Anaconda should work for this course. Python 2.x will not work for any version x.



3. Save the file to your computer.
4. Double click on the downloaded file to open it.
5. Follow the on-screen installation instructions, leaving options as they are currently set. This finishes the installation process.
6. **(OPTIONAL)** Next, check for any updates using Conda. Conda is one of the extras that is installed through the distribution Anaconda. It handles things like updates, set-up, and package installation through a command line interface. If there are many updates this can take 10 minutes or more. It is rare that you will have to use it

- Open Anaconda Prompt. Start typing **Anaconda Prompt** into the search box in the start menu, and it will show up.
- Type **conda update conda** at the command prompt, typing **y** for Yes and then pressing enter when it asks if you want to proceed. Your installation may identify different packages that need updating.
- After that completes, type **conda update anaconda** at the command prompt. If it prompts you to proceed with installation or updating, type **y** for Yes and press enter.
- After that completes, you can then close the command prompt window.



```

C:\Users\wboyd>conda update conda
Fetching package metadata: ....
Solving package specifications:
Package plan for specifying in environment C:\Anaconda3:

The following packages will be downloaded:

package                                     build                                size
-----
vs2010_runtime-10.00.40219.1               py34_0                             1.1 MB
conda-env-2.5.2                             py34_0                             67 KB
menuinst-1.4.1                             py34_0                             105 KB
pyyaml-3.11                                py34_4                             119 KB
requests-2.19.0                             py34_0                             657 KB
ruamel.yaml-0.11.7                         py34_0                             203 KB
setuptools-23.0.0                          py34_0                             694 KB
wheel-0.29.0                               py34_0                             123 KB
conda-4.1.8                                py34_0                             248 KB
pip-8.1.2                                   py34_0                             1.6 MB
Total: 4.8 MB

The following NEW packages will be INSTALLED:

ruamel.yaml: 0.11.7-py34_0
vs2010_runtime: 10.00.40219.1-2
wheel: 0.29.0-py34_0

The following packages will be UPDATED:

conda: 3.14.1-py34_0 --> 4.1.8-py34_0
conda-env: 2.2.3-py34_0 --> 2.5.2-py34_0
menuinst: 1.0.4-py34_0 --> 1.4.1-py34_0
pip: 7.0.3-py34_0 --> 8.1.2-py34_0
pyyaml: 3.11-py34_1 --> 3.11-py34_4
requests: 2.7.0-py34_0 --> 2.10.0-py34_0
setuptools: 17.1.0-py34_0 --> 23.0.0-py34_0

Proceed ([y]/n)? y
menuinst-1.4.1 100% |#####| Time: 0:00:00 595.01 kB/s
vs2010_runtime 100% |#####| Time: 0:00:00 1.59 MB/s
conda-env-2.5. 100% |#####| Time: 0:00:00 447.24 kB/s
pyyaml-3.11-py 100% |#####| Time: 0:00:00 501.73 kB/s
requests-2.10. 100% |#####| Time: 0:00:00 1.33 MB/s
ruamel.yaml-0. 100% |#####| Time: 0:00:00 821.27 kB/s
setuptools-23. 100% |#####| Time: 0:00:00 1.25 MB/s
wheel-0.29.0-p 100% |#####| Time: 0:00:00 644.90 kB/s
conda-4.1.8-py 100% |#####| Time: 0:00:00 951.33 kB/s
pip-8.1.2-py34 100% |#####| Time: 0:00:01 1.65 MB/s
Extracting packages ...
[ COMPLETE ]|#####| 100%
Unlinking packages ...
[ COMPLETE ]|#####| 100%
Linking packages ...
[ COMPLETE ]|#####| 100%
C:\Users\wboyd>

```

Getting to Know Spyder

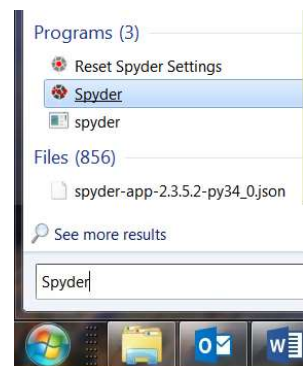
You will write your programs and run them inside the Spyder IDE (Scientific Python Development Environment):

Spyder

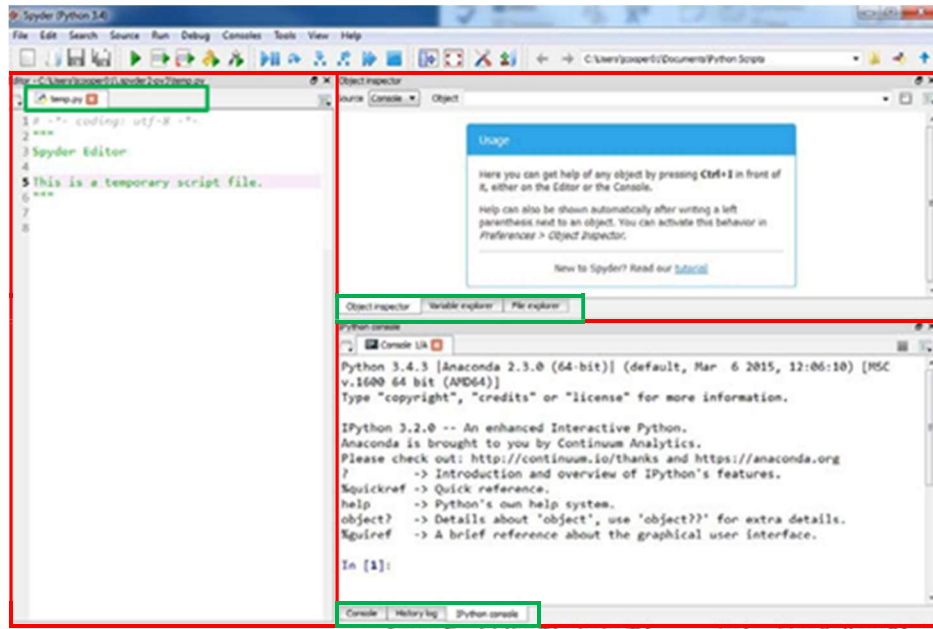


There are various ways to find and open this program – we only describe one way below.

In the search box associated with the start menu, type **Spyder**. Click on the program with its logo. Right clicking on it will give you the option to pin it to your start menu or taskbar for easier access in the future.



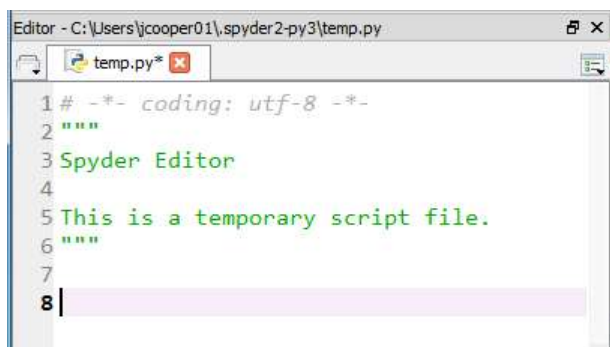
After Spyder has launched, you should see an interface that is organized into multiple windows (marked with red rectangles below) each of which has or could have multiple tabs (marked with green rectangles below).



The default starting screen has three windows visible: Editor, Object Inspector (now called Help), and Console.

If you accidentally close a window or change the layout, you can revert to the default by clicking "View" on the menu bar and then clicking "Reset Window Layout".

- **Editor** – where you can write and save sequences of commands – essentially, where you will write your full program.



In the sample script that shows up in the editor, anything after # is a comment, meaning that Python will ignore that text. Three double quotation marks are a special type of comment that spans multiple lines.

Here, you would actually begin writing your syntax on line 8 after the comments that say what your file does.

- **IPython Console or IP Console** – where Python runs your code.

Spyder offers two choices for a console – a basic one and an IPython (Interactive Python) console. It is the IPython console we want to use. We will often refer to it, however, as just the console. The prompt (where you can type in a command) for an IPython console is

In [1]:

Caution: If your console looks like the one shown to the right, where the prompt is `>>>` then you are in the *wrong* type of console and need to click on the tab for the IPython console.

- **Object Inspector (aka Help)** – On starting Spyder, the object inspector (as shown at right) just gives you a link to the tutorial (this can be useful if you want to know more details about using Spyder). Going to **Help** menu and choosing **Spyder Tutorial** will also bring up the tutorial. The rest of the time, this window will continue to give you information.

The **Object Inspector** gives you more information about a function – it gives you the documentation from the help files and tells you what parameters (or input) the function takes.

Note: In the latest version of Spyder this is called **Help**.

The **Variable Explorer** shows you the value and type of any variables you have created. In the image to the right, we created three variables in the console, and those three variables showed up in the Variable Explorer. This is useful when you are trying to debug your code.

Name	Type	Size	Value
x	int	1	4
y	int	1	3
z	str	1	abc

Frequently Asked Questions about Installation

- I already use Python 2.7 or I would rather use Python 2.7. Will that work for this course or do I need to install Python 3.x?
 - No. Python 2.7 will not work. The programs are auto-graded using a Python 3 system and it will not be able to compile Python 2.7 programs, because Python 2.7 is not compatible with Python 3.x. Yes, this is unusual. But the developers of Python decided to change certain features when going to Python 3 that made it impossible for them to maintain complete compatibility with Python 2. However, should you wish to shift to Python 2, later, you will be able to adapt at the expense of very little time and effort. Various websites, including <https://wiki.python.org/moin/Python2orPython3>, outline some of the syntactic differences.
- I have a different IDE that I like to use. Will that work for this course?
 - You could, but it will take more effort than it is worth. One big benefit of the Anaconda installation is that it includes IPython and IPython notebooks. These allow us to keep a whole lesson in one file, separated into cells. We can easily execute the code in a single cell, so that we don't have to write complete programs to run and test our code. After you finish the course, you can change to a different IDE with little effort, however. Most IDEs are similar to Spyder, but don't include the teaching convenience of the cells.