

# Getting Started with Python

This handout covers how to set up **Python** and introduces **Spyder**, the Python development environment we will be using throughout this course. It also shows how to check that the Python library packages **matplotlib** and **numpy** were successfully installed. They will be used later in this class to work with charts.




If you have problems with installation, be sure to go to **office hours** or **post on the forum** for help.

## Downloading the Installer

NOTE: the installer we'll be downloading and installing is larger than the average file, because it contains Python, associated packages, a code editor, and some other toys. It may take 15–20 minutes in total to download and install when executing the commands.

1. Visit the [download page](#) for **Anaconda Distribution**. Scroll to the bottom to see the following options for download. The current version is Anaconda3 2022.05, and it comes with Python 3.9. **The class is taught using Python 3.6 or higher. Do not download any version with an older Python.**

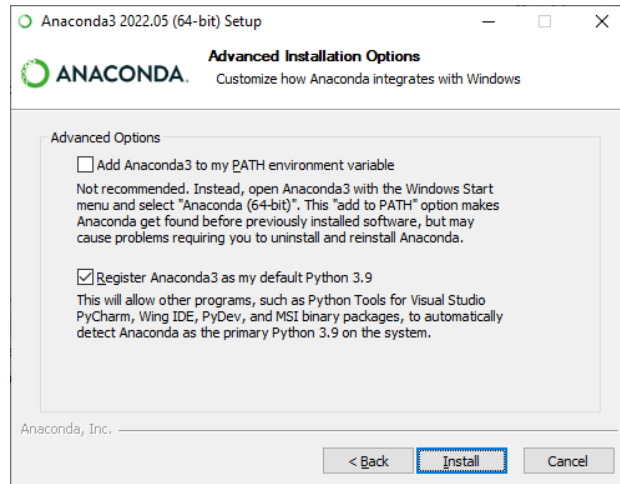
### Anaconda Installers

Windows 	MacOS 	Linux 
Python 3.9	Python 3.9	Python 3.9
64-Bit Graphical Installer (594 MB)	64-Bit Graphical Installer (591 MB)	64-Bit (x86) Installer (659 MB)
32-Bit Graphical Installer (488 MB)	64-Bit Command Line Installer (584 MB)	64-Bit (Power8 and Power9) Installer (367 MB)
	64-Bit (M1) Graphical Installer (316 MB)	64-Bit (AWS Graviton2 / ARM64) Installer (568 MB)
	64-Bit (M1) Command Line Installer (305 MB)	64-bit (Linux on IBM Z & LinuxONE) Installer (280 MB)

- a) Windows Operating System: There are two versions of the installer, one for 32-bit Windows, and one for 64-bit Windows. Microsoft provides instructions [here](#) to determine (relatively easily) if you have a 32- or 64-bit setup. If you are running 32-bit Windows, be sure that you select the 32-bit version, and if you are running 64-bit Windows, you can download either version.
- b) Macintosh Operating System: Macs post-2011 are [64-bit only](#) (and post-2007 are 64-bit compatible), and most Macs post-2020 use Apple's M1 processor. For most users, we recommend the graphical installer over the command line option.

## **Installing Python and Spyder**

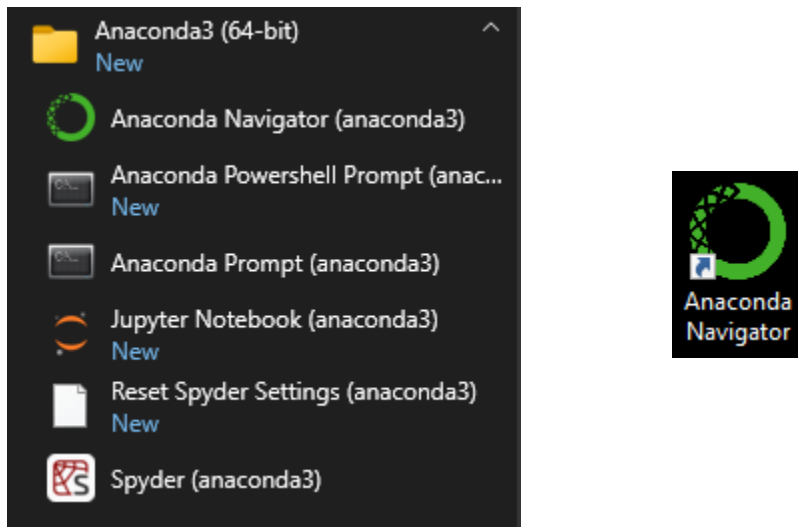
1. To install Python after downloading the graphical installer, double click the .exe (Windows) or .pkg (Mac) file and follow the instructions on the screen. There are some additional notes and comments on installing/uninstalling Python on Anaconda's documentation website [here](#).
2. Note that Anaconda's Python distribution should install cleanly into a single directory, should not require Administrator or root privileges, and should not affect other Python installs on your system (if you have any), or interfere with macOS Frameworks. Let us know via the forum if you're having any issues, so that others can learn along with you.
3. When installing Anaconda, the installer may ask whether or not to "Add Anaconda3 to my PATH environment variable" and "Register Anaconda3 as my default Python". We suggest accepting the default options of **not** adding Anaconda to your PATH, but registering it as your default Python.



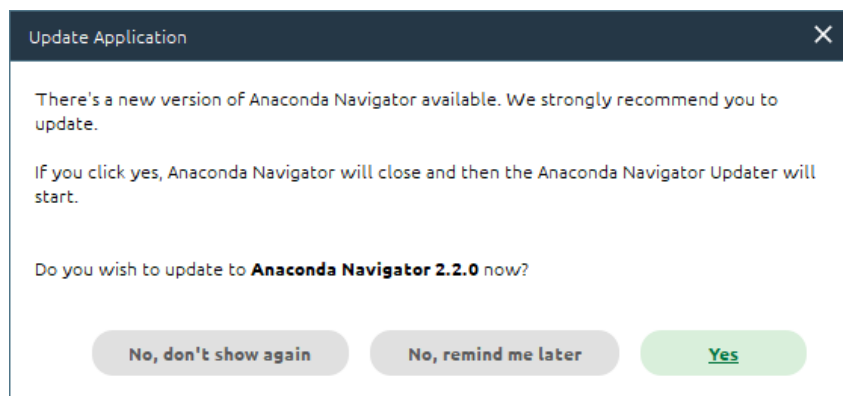
## **Starting and Updating Spyder**

When you download and install the Anaconda distribution of Python, you are getting several tools related to Python development, accessible via **Anaconda Navigator**. One of the tools is **Spyder** (Scientific Python Development Environment), an [integrated development environment](#) useful for writing, running, and debugging code. If either Navigator and Spyder have been updated since the installer's release, you may be prompted to update them on first launch. Below we walk through the steps to do so.

1. First, open Anaconda Navigator from the Windows start menu or macOS launchpad. There may also be a desktop launcher.

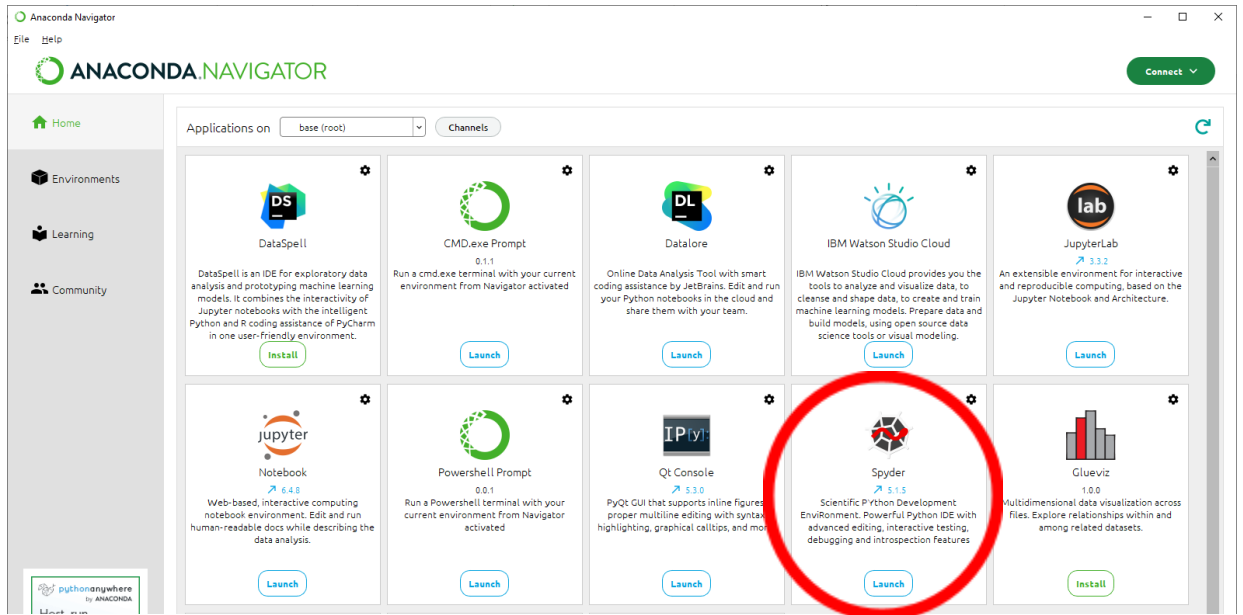


After a bit of initialization, the Anaconda Navigator window will display. If it asks you to update the application, go ahead and click "Yes".

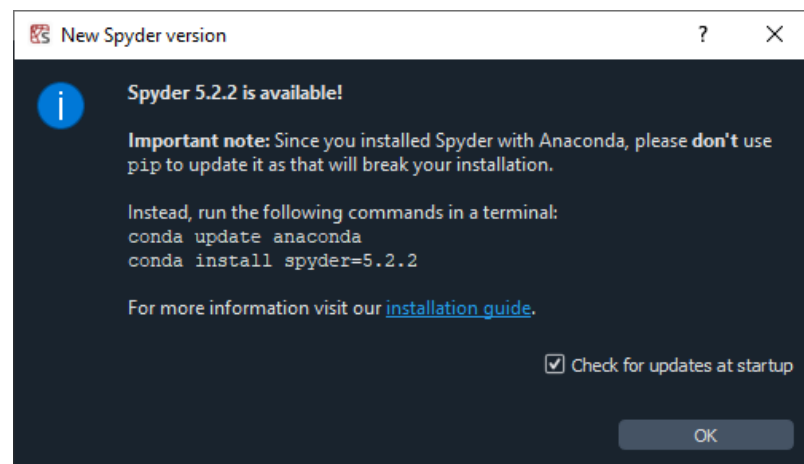


Click the follow-up "Yes" to quit Anaconda Navigator, and then "Update now". When the update completes, click "Launch Navigator". Click the follow-up "Yes" to quit Anaconda Navigator, and then "Update now". When the update completes, click "Launch Navigator".

- Once you've restarted Navigator, find the "Spyder" application, and click its blue "Launch" button. The Spyder window will appear after a brief initialization.



- Like Navigator, Spyder will likely need updating. **If you installed Anaconda3 2022.05, you must update Spyder**, as there is a [bug](#) in how it interacts with Python's input() function.



**Do not** update Spyder through Anaconda Navigator. Instead, follow the instructions from Spyder's auto-update popup. Close Spyder, then open **Anaconda Prompt** (from the start menu or launchpad), and run the following commands: (see next two pages for screenshots)

- > conda update anaconda

It will calculate dependencies for a little bit. Type "y" when prompted to proceed. Then it will spend some time downloading and updating. It will finish only when the prompt returns.

- > conda install spyder=5.2.2

This will take several minutes to calculate dependencies. Do not be alarmed when it fails. It will automatically retry and eventually succeed. Type "y" when prompted to proceed, and wait for it to finish downloading and updating.

- You can now close Anaconda Prompt and relaunch Spyder. You may want to relaunch Anaconda Navigator first and make sure you are now using Spyder 5.2.2.

```
Anaconda Prompt (anaconda3) - conda update anaconda

(base) C:\Users\bob>conda update anaconda
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\bob\anaconda3

  added / updated specs:
    - anaconda

The following packages will be downloaded:

package | build | size
-----|-----|-----
_anaconda_depends-2022.05 | py39_0 | 7 KB
anaconda-custom | py39_1 | 4 KB
anaconda-client-1.10.0 | py39h2bbff1b_0 | 166 KB
anaconda-project-0.11.1 | py39h2bbff1b_0 | 511 KB
astropy-5.1 | py39h2bbff1b_0 | 6.4 MB
bcrypt-3.2.0 | py39h2bbff1b_1 | 37 KB
bitarray-2.5.1 | py39h2bbff1b_0 | 157 KB
bkcharts-0.2 | py39h2bbff1b_0 | 132 KB
black-22.6.0 | py39h2bbff1b_0 | 244 KB
bokeh-2.4.3 | py39h2bbff1b_0 | 7.6 MB
boto3-1.24.28 | py39h2bbff1b_0 | 114 KB
botocore-1.27.28 | py39h2bbff1b_0 | 4.6 MB
bottleneck-1.3.5 | py39h2bbff1b_0 | 106 KB
```

```
Anaconda Prompt (anaconda3) - conda update anaconda

sqlite 3.38.2-h2bbff1b_0 --> 3.39.2-h2bbff1b_0
tabulate 0.8.9-py39h2bbff1b_0 --> 0.8.10-py39h2bbff1b_0
tenacity 8.0.1-py39h2bbff1b_0 --> 8.0.1-py39h2bbff1b_1
testpath pkgs/main/noarch::testpath-0.5.0-pyhd3eb1~ --> pkgs/main/win-64::testpath-0.6.0-py39h2bbff1b_0
tk 8.6.11-h2bbff1b_0 --> 8.6.12-h2bbff1b_0
tomli pkgs/main/noarch::tomli-1.2.2-pyhd3eb1~ --> pkgs/main/win-64::tomli-2.0.1-py39h2bbff1b_0
twisted 22.2.0-py39h2bbff1b_0 --> 22.2.0-py39h2bbff1b_1
typing-extensions pkgs/main/noarch::typing-extensions-4~ --> pkgs/main/win-64::typing-extensions-4.3.0-py39h2bbff1b_0
typing_extensions pkgs/main/noarch::typing_extensions-4~ --> pkgs/main/win-64::typing_extensions-4.3.0-py39h2bbff1b_0
ujson 5.1.0-py39h2bbff1b_0 --> 5.4.0-py39h2bbff1b_0
urllib3 1.26.9-py39h2bbff1b_0 --> 1.26.11-py39h2bbff1b_0
xz 5.2.5-h62dc97_0 --> 5.2.5-h8cc25b3_1
yarl 1.6.3-py39h2bbff1b_0 --> 1.8.1-py39h2bbff1b_0
zict pkgs/main/noarch::zict-2.0.0-pyhd3eb1~ --> pkgs/main/win-64::zict-2.1.0-py39h2bbff1b_0
zipp pkgs/main/noarch::zipp-3.7.0-pyhd3eb1~ --> pkgs/main/win-64::zipp-3.8.0-py39h2bbff1b_0

The following packages will be DOWNGRADED:

anaconda 2022.05-py39_0 --> custom-py39_1

Proceed ([y]/n)?
```

```
Anaconda Prompt (anaconda3)

setuptools-63.4.1 | 1.0 MB | ##### | 100%
libbrotlenc-1.0.9 | 233 KB | ##### | 100%
numpy-base-1.21.5 | 4.4 MB | ##### | 100%
tk-8.6.12 | 3.1 MB | ##### | 100%
brotli-1.0.9 | 18 KB | ##### | 100%
imageio-2.19.3 | 3.3 MB | ##### | 100%
qtpy-2.2.0 | 106 KB | ##### | 100%
bokeh-2.4.3 | 7.6 MB | ##### | 100%
holoviews-1.15.0 | 4.0 MB | ##### | 100%
anaconda-custom | 4 KB | ##### | 100%
msgpack-python-1.0.3 | 75 KB | ##### | 100%
_anaconda_depends-20 | 7 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: |

Windows 64-bit packages of scikit-learn can be accelerated using scikit-learn-intelex.
More details are available here: https://intel.github.io/scikit-learn-intelex

For example:

$ conda install scikit-learn-intelex
$ python -m sklearn my_application.py

done
Retrieving notices: ...working... done

(base) C:\Users\bob>
(base) C:\Users\bob>
```

```
Anaconda Prompt (anaconda3) - conda install spyder=5.2.2

done
Retrieving notices: ...working... done

(base) C:\Users\bob>
(base) C:\Users\bob>conda install spyder=5.2.2
Collecting package metadata (current_repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
Solving environment: done

## Package Plan ##

  environment location: C:\Users\bob\anaconda3

added / updated specs:
- spyder=5.2.2

The following packages will be downloaded:

package | build | size
----- | -
conda-build-3.22.0 | py39haa95532_0 | 541 KB
flake8-4.0.1 | pyhd3eb1b0_1 | 118 KB
ipython-7.31.1 | py39haa95532_1 | 1024 KB
jellyfish-0.9.0 | py39h2bbff1b_0 | 46 KB
jupyter_client-7.3.4 | py39haa95532_0 | 216 KB
```

```
Anaconda Prompt (anaconda3) - conda install spyder=5.2.2

jellyfish      pkgs/main/win-64::jellyfish-0.9.0-py39h2bbff1b_0
m2-msys2-runtime pkgs/msys2/win-64::m2-msys2-runtime-2.5.0.17080.65c939c-3
m2-patch       pkgs/msys2/win-64::m2-patch-2.7.5-2

The following packages will be UPDATED:

conda-build      3.21.8-py39haa95532_2 --> 3.22.0-py39haa95532_0
flake8           3.9.2-pyhd3eb1b0_0 --> 4.0.1-pyhd3eb1b0_1
jupyter_client   pkgs/main/noarch::jupyter_client-6.1.~ --> pkgs/main/win-64::jupyter_client-7.3.4-py39haa95532_0
pycodestyle      2.7.0-pyhd3eb1b0_0 --> 2.8.0-pyhd3eb1b0_0
pyflakes         2.3.1-pyhd3eb1b0_0 --> 2.4.0-pyhd3eb1b0_0
python-lsp-server 1.2.4-pyhd3eb1b0_0 --> 1.3.3-pyhd3eb1b0_0
spyder           5.1.5-py39haa95532_1 --> 5.2.2-py39haa95532_1
spyder-kernels   2.1.3-py39haa95532_0 --> 2.2.1-py39haa95532_0

The following packages will be SUPERSEDED by a higher-priority channel:

qtconsole        pkgs/main/win-64::qtconsole-5.3.2-py3~ --> pkgs/main/noarch::qtconsole-5.2.2-pyhd3eb1b0_0

The following packages will be DOWNGRADED:

ipython          8.4.0-py39haa95532_0 --> 7.31.1-py39haa95532_1

Proceed ([y]/n)?
```

```
Anaconda Prompt (anaconda3)

ipython          8.4.0-py39haa95532_0 --> 7.31.1-py39haa95532_1

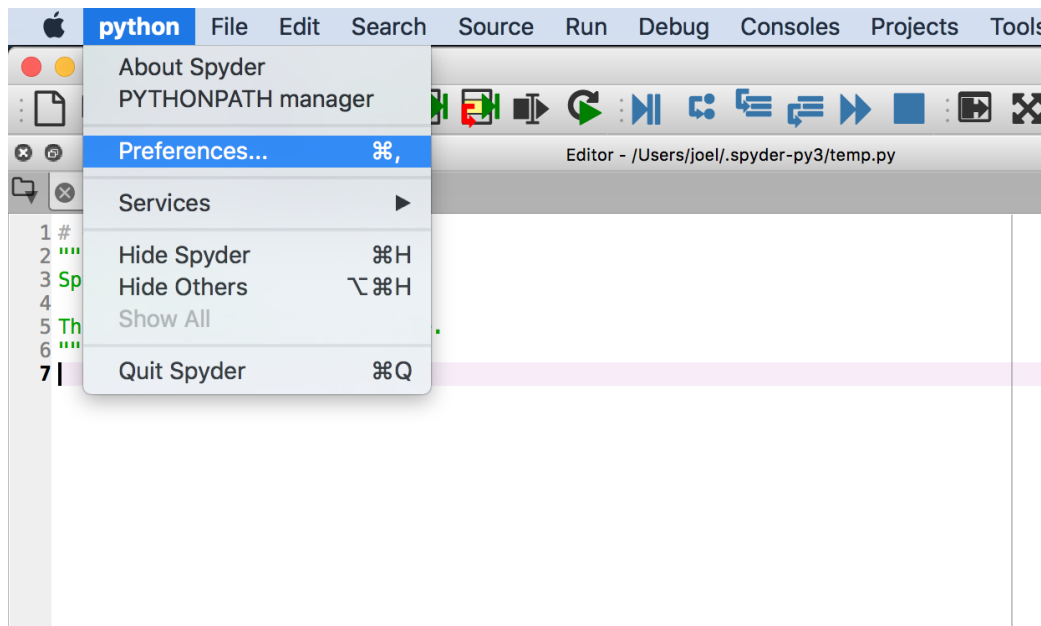
Proceed ([y]/n)? y

Downloading and Extracting Packages
spyder-kernels-2.2.1 | 115 KB | ##### 100%
pycodestyle-2.8.0 | 41 KB | ##### 100%
flake8-4.0.1 | 118 KB | ##### 100%
spyder-5.2.2 | 9.4 MB | ##### 100%
pyflakes-2.4.0 | 60 KB | ##### 100%
python-lsp-server-1. | 43 KB | ##### 100%
m2-patch-2.7.5 | 89 KB | ##### 100%
conda-build-3.22.0 | 541 KB | ##### 100%
ipython-7.31.1 | 1024 KB | ##### 100%
qtconsole-5.2.2 | 92 KB | ##### 100%
jellyfish-0.9.0 | 46 KB | ##### 100%
jupyter_client-7.3.4 | 216 KB | ##### 100%
m2-msys2-runtime-2.5 | 3.0 MB | ##### 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Retrieving notices: ...working... done

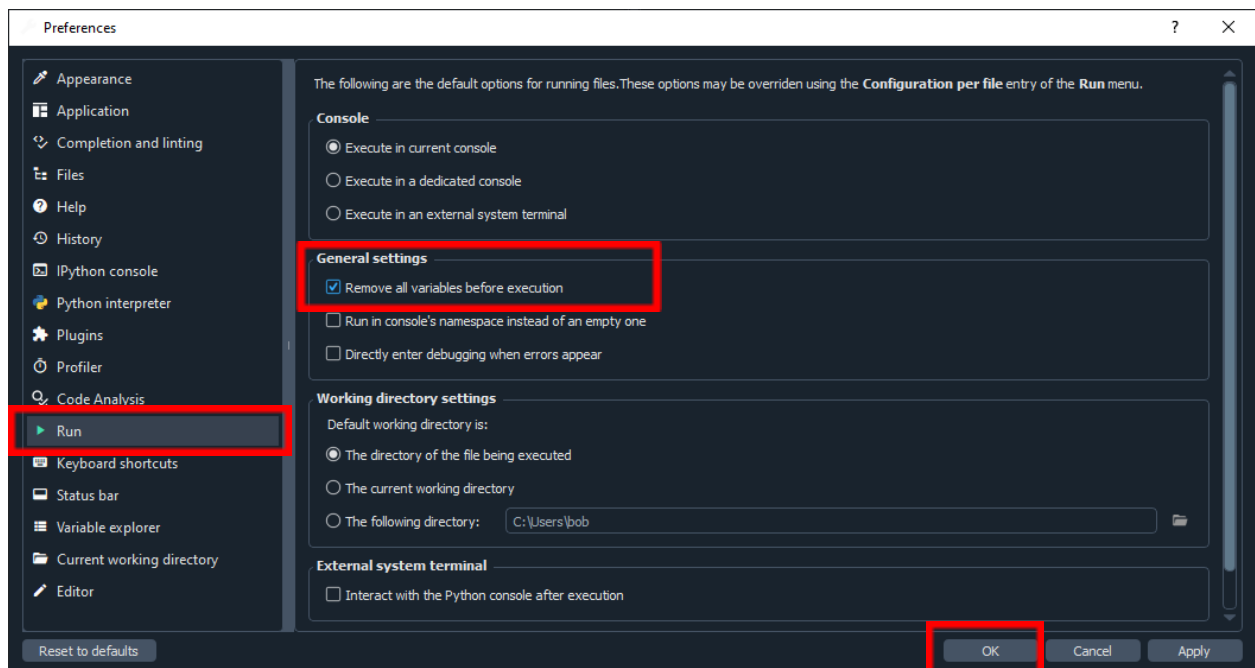
(base) C:\Users\bob>
```

## Configuring Spyder

Before using Spyder, we recommend configuring it to use the “Clear variables before execution” option. This will help prevent future debugging issues due to retaining old variables between executions of your code. This feature can be found in the preferences, accessed by Python → Preferences on Mac, or Tools → Preferences on Windows and Linux.



Once in Preferences, choose the “Run” category, find the “General setting” group, and check “Remove all variables before execution”. Click “OK” when finished.

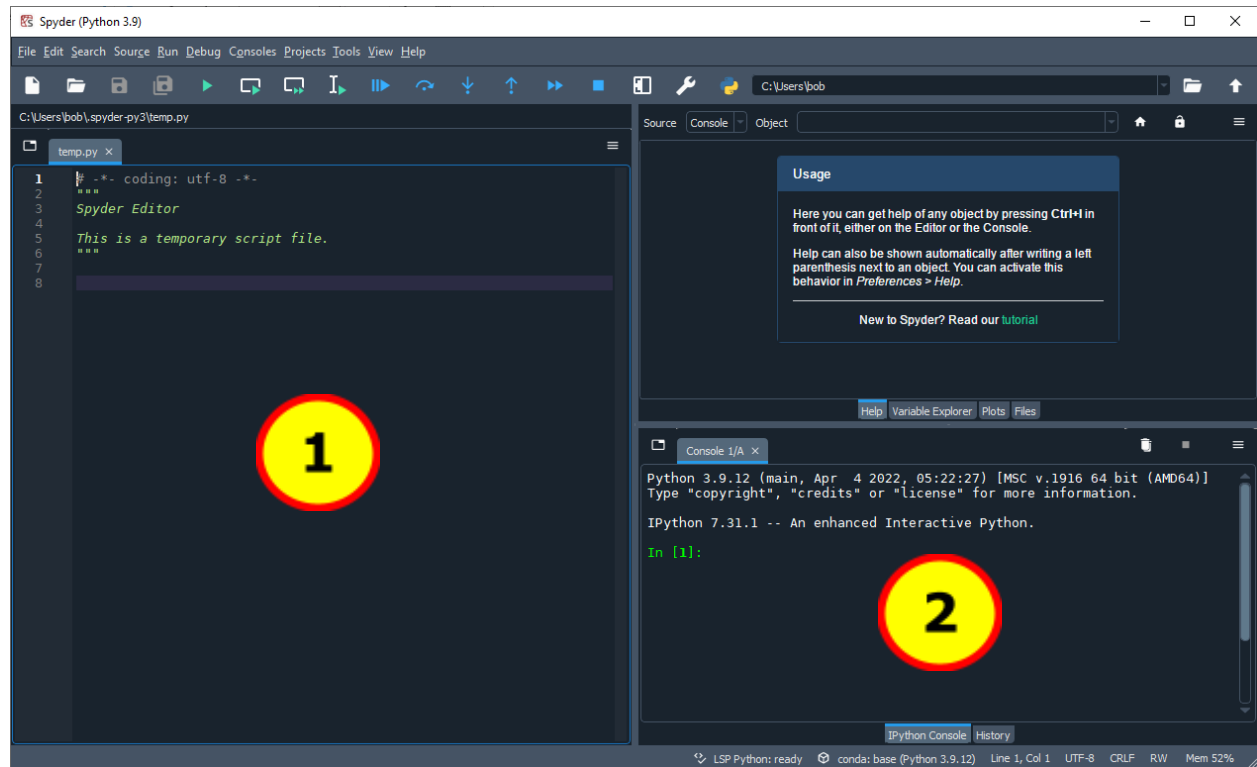


## Using Spyder for Editing and Running Code

Opening the Spyder application should present you with the following window. This window contains two commonly used parts, among others:

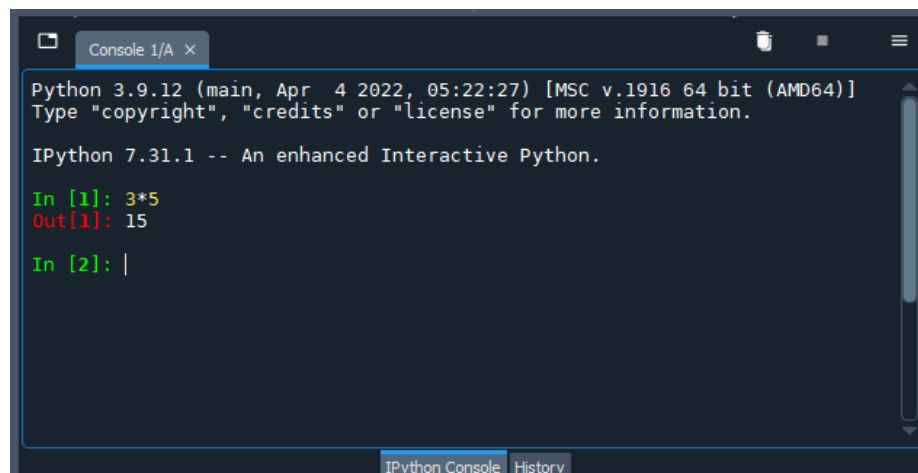
- (1) a Code Editor that lets you create and edit existing Python source files
- (2) the IPython interpreter pane, which gives you access to the Python interactive mode

During the following discussion of Spyder's features, you should start Spyder and try to replicate the screenshots.



### Using the IPython prompt:

1. The IPython prompt looks something like this: "In [1]:", and can be used to run Python code. You can type Python code directly into this prompt, and pressing Enter executes the code fragment.
2. Try typing the following after the prompt and pressing the enter key: `3*5`





## Using the IPython prompt – examples and exercises:

Addition (+), subtraction (-), multiplication (\*), division (/), modulo (%) and power (\*\*) operators are built into the Python language. This means you can use them right away. If you want to use a square root in your calculation, you can either raise something to the power of 0.5 or you can *import* the *math* module. Do not worry about what it means right now, we will cover this later during the course. Below are two examples of square root calculation:

```
In [1]: 16**0.5
```

```
Out[1]: 4.0
```

```
In [2]: import math
```

```
In [3]: math.sqrt(16)
```

```
Out[3]: 4.0
```

The math module allows you to do a number of useful operations:

```
In [1]: math.log(16, 2)
```

```
Out[1]: 4.0
```

```
In [2]: math.cos(0)
```

```
Out[2]: 1.0
```

**Exercises:** Use the IPython prompt to calculate:

*(These are just for practice, solutions will not be graded or collected in class.)*

1.  $6 + 4*10$

2.  $(6+4) * 10$

a. (Compare this to #1, and note that Python uses parentheses just like you would in normal math to determine the order of operations!)

3. 23.0 to the 5th power

4. Positive root of the following equation:

$$34x^2 + 68x - 510$$

a. Recall, given  $ax^2 + bx + c$ , then  $x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ .

5. import math

math.cos(3.4)\*\*2 + math.sin(3.4)\*\*2

## Testing the Spyder update:

Recall on Page 4 that we needed to update Spyder to fix an issue with Python's input() function. To verify that it works now, enter the following line at the IPython prompt. It should repeat the question back at you, and once you type in an answer (and hit Enter), it will print that again as the output.

```
In [1]: input("What do you want for your birthday? ")
```

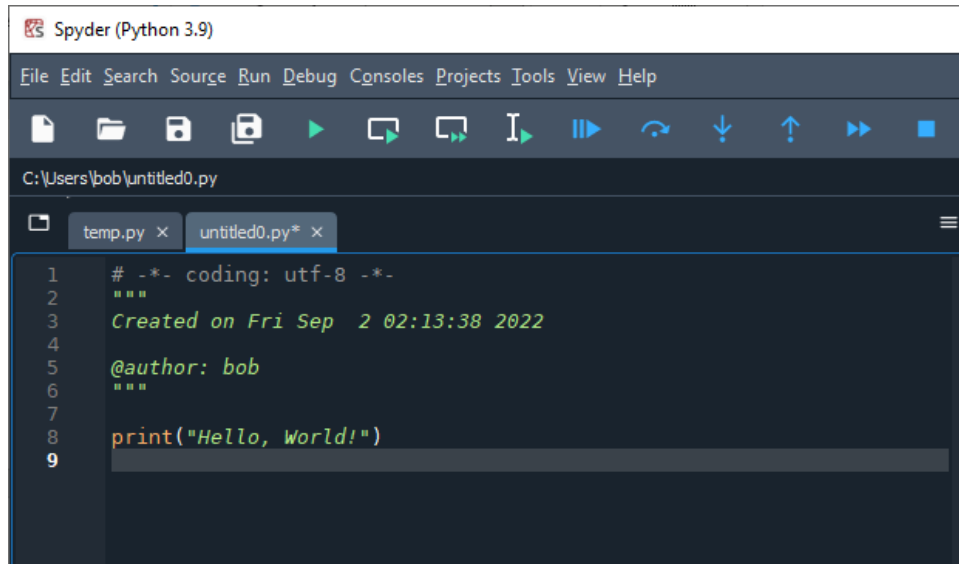
What do you want for your birthday? I would like a hat.

```
Out[1]: 'I would like a hat.'
```

## To create, save, and run a file:

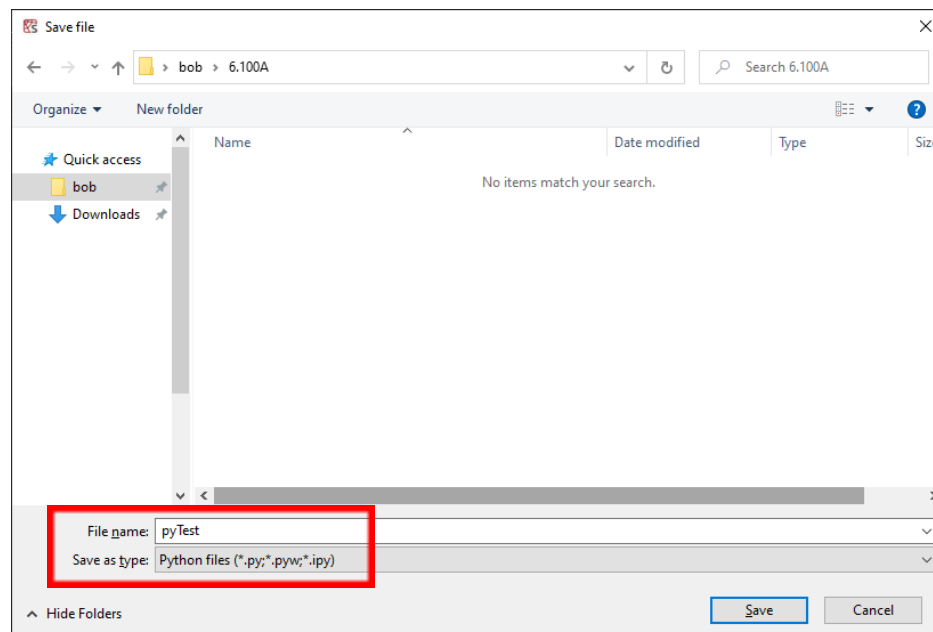
### Creating the file

1. In Spyder's File menu, select "New file".
2. In the new file, type the following: `print("Hello, World!")`



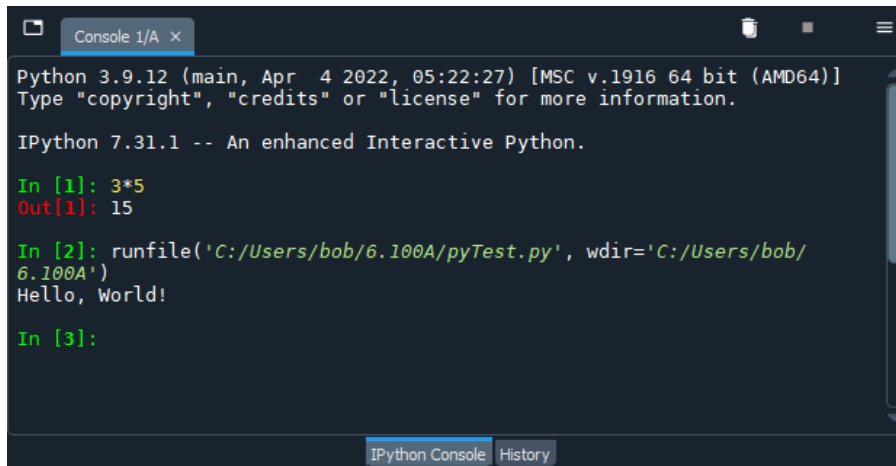
### Saving the file

1. To aid in organization, save your files in a location specific for this course. First create a directory (i.e., folder) for your 6.100A material, with an appropriate name.
2. From Spyder's File menu, click "Save as..." and then navigate to your course folder before typing a name for this file, e.g., "pyTest". Make sure that you're saving the file as type Python before clicking "Save".



## Running the file

1. Go to Spyder's "Run" menu, click on "Run". (A popup may appear asking to confirm run settings. Go ahead and click "Run".) You should see two parts to the output in the IPython console. First, the `runfile(...)` line shows you the path for the file you just ran. Second, the output of the file, "Hello, world!", appears.



```
Python 3.9.12 (main, Apr  4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

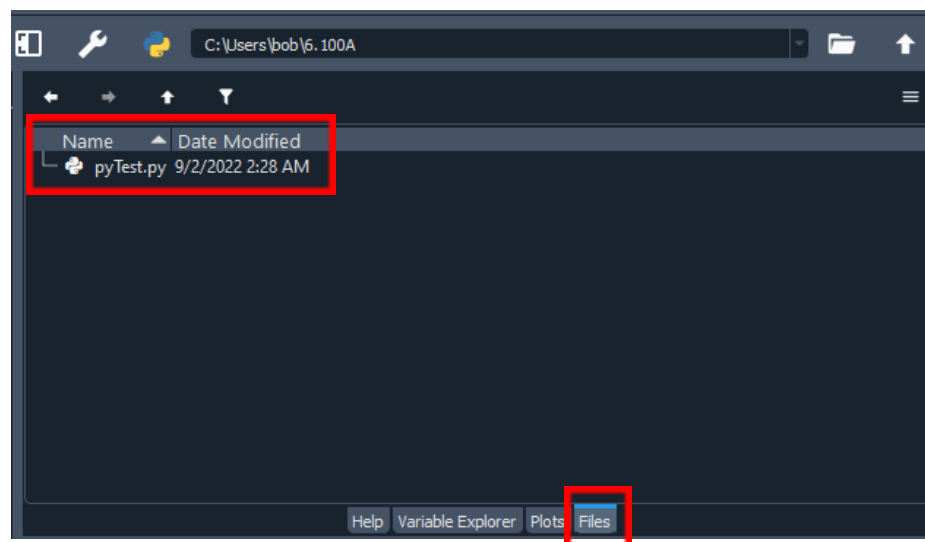
IPython 7.31.1 -- An enhanced Interactive Python.

In [1]: 3*5
Out[1]: 15

In [2]: runfile('C:/Users/bob/6.100A/pyTest.py', wdir='C:/Users/bob/
6.100A')
Hello, World!

In [3]:
```

2. Go back to the Code Editor, and add the following line: `print("I like 6.100A!")`
3. Select "Run" again (you can also use the green triangular Run button on the toolbar or the shortcut F5), and observe the change in result.
4. Close your test file by clicking the X in its filename tab. Then reopen it by double-clicking its name in the Files pane towards the top right.

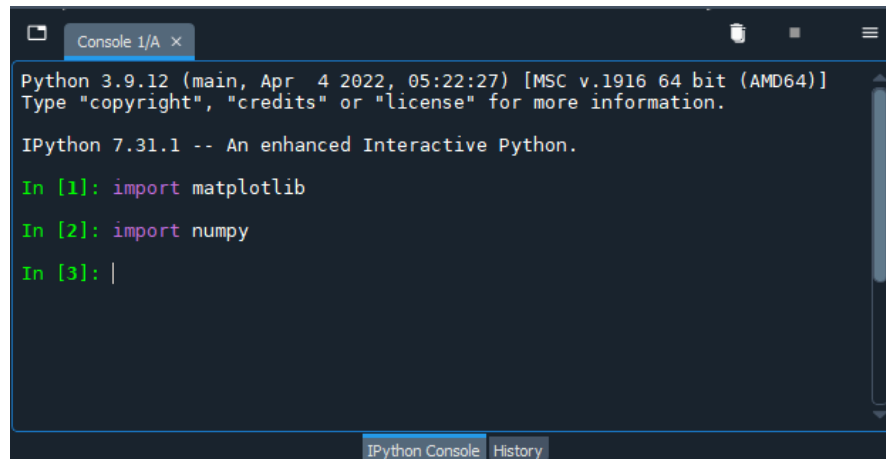


5. Congratulations – your Python environment has been installed, and you now know how to enter expressions in Python, and how to save, run, and open files!
6. For additional help on Spyder, go to the Help menu within Spyder, and select the "Spyder tutorial" option.

## Installing and Testing Matplotlib and Numpy

To work with charts, you will need these Python library packages: [matplotlib](#) and [numpy](#). You will not need these packages until later assignments, but they will be used in class demos during lecture so we encourage you to download them now.

If you're using Anaconda Distribution, your system should already have matplotlib and numpy installed. Start Spyder and type "import matplotlib" and "import numpy" into the prompts. If no errors show up, you already have them installed. If errors show up, reach out via the forum to the instructors!

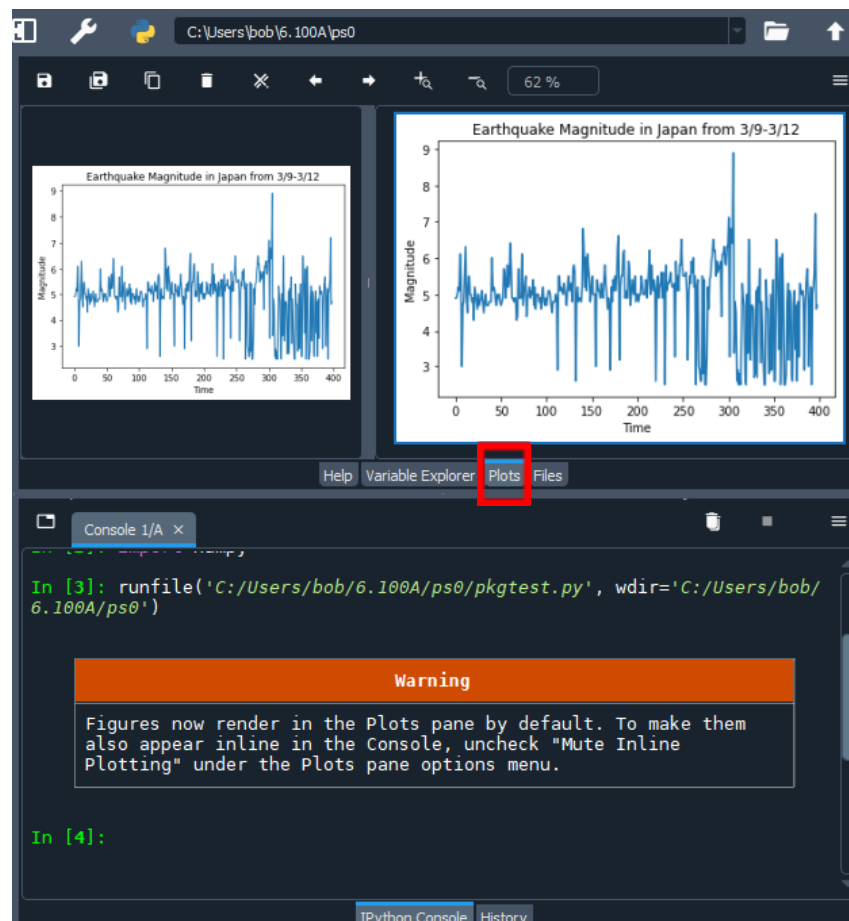


```
Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]: import matplotlib
In [2]: import numpy
In [3]: |
```

To test via a script that you have successfully installed matplotlib and numpy, run the code provided in pkgtest.py. It's okay if a warning like the one below appears, and you can find the generated chart in the "Plots" pane.



-----

*Asfandiyar Qureshi, Feb 2006*

*Edited by Vladimir Bychkovsky, Sept 2006*

*Edited by Calvin On, Feb 2007*

*Edited by Yang Zhang, Sep 2008*

*Edited by Chih-yu Chao, Feb 2009*

*Edited by Sari Canelake, Dec 2009*

*Edited by Anjali Muralidhar, Feb 2013*

*Edited by Niki Castle, Feb 2013*

*Edited by Prashan Wanigasekara, Feb 2014*

*Edited by Zachary Gil Freeman, Feb 2015*

*Edited by Ana Bell, August 2016*

*Edited by Andrew Wang, September 2022*