PHYS 4332 – Computational Physics – W2025

Python Tutorial

1.0 – Introduction and Installation Instructions

Python is a high-level, interpreted programming language that has now become the language of choice in many fields of science. This is largely due to its ease of programming and readability, the large number of libraries and community support, and the variety of programming environments (e.g. command line interpreters, scripts, web-based interpreters, notebooks). This tutorial, which is largely based on Corey Schafer's excellent YouTube series, is a crash course in Python designed to give you an overview of the programming essentials, as well as some useful tools for scientific computing, data processing, and visualization.

Although a variety of web browser-based Python development environments exist, there are significant advantages to installing Python locally on your computer. This tutorial provides step-by-step instructions on how to install and configure Python from source, as well as how to install third-party libraries that are essential for data science.

The majority of the modules in this tutorial are written in Jupyter Notebooks (formally IPython Notebooks). This is a web-based interactive development environment (IDE) for writing code (e.g., Python, Markdown, LaTeX, etc.) and displaying graphics. It uses a notebook format similar to Wolfram Mathematica or Windows OneNote, where interactive cells of code can be executed one at a time by a kernel (a centralized "computational engine"). While the kernel is running, all the data generated by your code is stored temporarily in your computer's random access memory—allowing you to access data generated in previous cells.

Jupyter Notebooks can be run in a web browser if you have Python and Jupyter installed locally. However, it is highly recommended to install and use Visual Studio Code (VS Code) to work through this tutorial. This IDE integrates the three most common Python environments (command line, scripts, and notebooks), allowing you to choose the most convenient option. It also has built-in features for debugging and streamlines the installation of third-party extensions for additional features like code linting.

The latest stable (and trustworthy) version of Python is **3.12.7** (released Oct. 1, 2024). We recommend selecting this option when installing it on your system using the instructions below.

Installing Python (3.12.7) for Windows

1. Go to https://www.python.org/downloads/release/python-3127/
2. Click on "Python 3.12.7" under "Stable Releases"
3. Scroll down to "Files" and select the "Windows Installer" corresponding to your operating system type (32-bit) or (64-bit). (To check your operating system type, click the Start menu, select Settings → System → About. It should be listed under "System type".)
4. Once the Installer has downloaded, run the `.exe` file (usually located in your Downloads folder).
5. In the Python Installer Setup window, click on the checkbox "Add python.exe to PATH" at the bottom. This will add the location of the Python.exe to your system's PATH environment variable so you can run python from the command prompt.

6. Next, click "Install Now" to install Python. Choose the default setting for any remaining setup options.
7. Once Python has finished installing, click on the Start Menu, type `cmd`, and open a command prompt by selecting `cmd.exe`.
8. To verify Python installed correctly, type `python --version` and hit Enter. "Python 3.12.7" should appear.

Installing Python (3.12.7) for MAC

MACs usually come pre-installed with Python version 2. To check which version you have, open a terminal and type `python --version` + Enter.

1. Go to https://www.python.org/downloads/release/python-3127/
2. Click on "Python 3.12.7" under "Stable Releases"
3. Scroll down to "Files" and select the "macOS 64-bit universal2 installer" (for macOS 10.13 and later).
4. Once the Installer has downloaded, run the `.pkg` file. Choose the default setup options.
5. Upon completion, the installer will create a "Python 3.12" directory in your applications folder.
6. To verify Python installed correctly, open a new terminal and type `python3 --version` + Enter. "Python 3.12.7" should appear. (Note that you must use the "python3" command to access Python 3 from the command prompt. The command "python" will default to Python 2 unless you create an alias.)

Running Python from a Terminal

- From a command prompt or terminal, type `python` and hit Enter (`python3` on a MAC). You should get something like the following:

  ```
  Python 3.12.7 (Oct  1 2024, 03:06:41) [MSC v.1929 64 bit (AMD64)] on win32
  Type "help", "copyright", "credits" or "license" for more information.
  >>>
  ```

  Note for Windows users: if the command `python` is not recognized from the terminal, you will need to add the location of your python installation (containing the executable file `python.exe`) to your PATH environment variable.

- This is an interactive prompt that allows us to write one line of Python at a time.
- To create a "Hello World" application, we can simply write `print(`hello world!")" + Enter:

  ```
  >>> print("hello world!")
  hello world!
  ```

- We can also do simple operations with integers, floats, and booleans:

```
>>> 1+2 - 2*3
-3
>>> 4/3
1.3333333333333333
>>> True & False
False
>>> True | False
True
```

- We can also define variable and print the output of calculations:

```
>>> x=10
>>> y=2
>>> print(x,"**",y,"=",x**y)
10 ** 2 = 100
>>> x=20
>>> print(x,"**",y,"=",x**y)
20 ** 2 = 400
```

- To exit out of Python, simply type `exit()` or Ctrl-Z + Enter.

Installing Python Packages

The base installation of Python includes the Python standard library. This is an extensive library of build-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers (e.g., the `os()` module), as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming (e.g., the `datetime()` module).

In addition to the standard library, the Python Package Index (PyPI) is an active collection of more than 100,000 third-party modules, packages, and entire application development frameworks. This is a repository of Python software developed, maintained, and shared by the Python community. The Package Installer for Python (`pip`, part of the standard library) provides a simple way of installing third-party packages from PyPI and other repositories.

- Open a terminal and test if you can use `pip` by typing:

```
>>> pip --version

pip 24.3.1 from C:\Python 3.12\Lib\site-packages\pip (python 3.12)
```

  Note for Windows users: if the command `pip` is not recognized from the terminal, you will need to add the location of the `Scripts` folder in your python installation to your PATH environment variable.

- To list the packages currently installed, use `list`:

```
>>> pip list
```

- To install a new package, use `install`. For example, to install the `numpy` package:

```
>>> pip install numpy

Collecting numpy
  Downloading numpy-2.1.3-cp312-cp312-win_amd64.whl.metadata (59 kB)
Downloading numpy-2.1.3-cp312-cp312-win_amd64.whl (12.6 MB)
---------------------------------------- 12.6/12.6 MB 14.9 MB/s eta 0:00:00
Installing collected packages: numpy
Successfully installed numpy-2.1.3
```

- Now go to Python and use the numpy package by importing it numpy as a module called 'np':

```
>>> import numpy as np
>>> pi = np.pi
>>> print(pi)
3.141592653589793

>>> print(np.cos(pi))
-1.0

>>> print(np.exp(1))
2.718281828459045
```

- Before continuing further with this Python tutorial, we recommend installing the following packages:

```
>>> pip install ipykernel
>>> pip install matplotlib
>>> pip install scipy
```

  Note that any packages with dependencies that are not already installed will automatically be installed/upgraded by `pip`.

Installing VS Code

- Go to https://code.visualstudio.com/
- Click on the down arrow next to the "Download for Windows" button. This will display options for all available operating systems. Download the installer appropriate for your system.
- Follow the default installation instructions.
- Next, open VS Code and install the Python extension:
  - Click on the "Extensions" icon on the left sidebar (or type `Ctrl + Shift + X`).
  - Type `python` into the search bar and select the first result "Python". This extension includes Intellisense (PyLance), Linting, Debugging, Jupyter Notebooks, and more.
- Next, install the Code Runner extension:
  - Click on the "Extensions" icon on the left sidebar (or type `Ctrl + Shift + X`).
  - Type `Code Runner` into the search bar and select the first result. This extension includes additional options, controls, and shortcuts for running code.

- You are now ready to open Python files (`.py`) or Jupyter Notebook files (`.ipynb`) and start writing code.
- To run a Python file, open it and click on the "Run Code" button in the top right corner (or type `Ctrl + Alt + N` if you have Code Runner installed).
- When running for the first time, you may be asked to select a Python interpreter:
  - Open the command palette by typing `Ctrl + Shift + P` and type `Python: Select Interpreter`. You can also select the "Python Environment Option" on the bottom right corner of the status bar.
  - A list of available interpreters should appear (VS Code finds these automatically on your system). Select the version of Python that you just installed.

Additional resources

- Corey Schafer's Python tutorial and related playlists:
  https://www.youtube.com/playlist?list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU
- To get your barrings with VS Code, check out their Getting Started videos:
  https://code.visualstudio.com/docs/getstarted/introvideos
- To get started with Python in VS Code:
  https://code.visualstudio.com/docs/python/python-tutorial
- Additional documentation on the VS Code interface can be found here:
  https://code.visualstudio.com/docs/getstarted/userinterface
- To further customize your VS code environment, check out Corey Schafer's video:
  https://www.youtube.com/watch?v=-nh9rCzPJ20