

EN.540.635 “Software Carpentry”

How to setup a programming environment

Introduction

In this handout we will outline the different components that you can install that will help you get through the course. There is no fixed software that all python programmers use to write code. We list some options that you can choose between so that you set up a programming environment that best suits you. At the same time, the options listed here are not exhaustive, and we only list the ones that we have encountered ourselves and seen students in previous iterations of this class use. However, we do recommend some specific ones because those are the ones that the instructor will use and grading will also be based off of the errors that show up in the formatting of code.

Installing a python distribution

While it is possible to just install Python from the source itself (from the [Python Software Foundation](#)) for any Operating System, this usually is a very bare-bones installation. We recommend installing Anaconda which is “a free and open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS.”. The distribution can be downloaded from the Anaconda [website](#). Installation instructions can be found [here](#), depending on the build of your machine. **Windows users should make sure that they ‘Add to path’ when prompted during the installation process.**

Installing an IDE

If you chose to install python directly from the source, you improve you programming experience by installing [PyCharm](#) which is full-fledged Integrated Development Environment(IDE) for python. For users who installed Ananconda, it provides 2 programs that yo can use to write and execute python scripts: Jupyter and Spyder. The former is a free, open-source web-tool that amny researchers use Spyder is also open-source and is an IDE. For this course, we recommend [Sublime Text](#) which is technically a text/source-code editor. However, it can further be customized and enhanced through plug-ins and provide you with a IDE-like experience.

Customizing Sublime with Packages

We recommend the [Anaconda plug-in for Sublime](#) as it makes the use of Sublime as a programming enviroment very convenient. Once installed, please make sure that the Anaconda linter for PEP8 is working and indicates PEP8 styling errors in your script.

Setup a ‘Shell’

For some sections of the course, you will be required to use a ‘Bash’ shell. For Mac users, no additional installations are required and the default terminal can be used. Windows users should make sure that you have either the [Git Bash terminal](#) or the [Windows Subsystem for Linux](#). For the former, enabling ‘experimental support for pseudo consoles’ could be useful as well for runnig python from the GitBash terminal when your scripts require user input. For the latter, you would not be able to do the same unless you install Anaconda again through the WSL terminal, this time following the instructions for Linux. You also have access to the Anaconda prompt through Anaconda which gives you access to python through the terminal and limited Bash functionality.

Additional Resources:

- A History of Unix and Linux ([link](#)).
- Vim Cheatsheet ([link](#)).
- Vim “Masterclass” ([link](#)).