

Python BootCamp

–lecture notes–

February 7, 2024

Title: Python BootCamp

Instructor: Saeed Varasteh Yazdi

Date (Version 1.): February, 2024.



Python BootCamp - Lecture 00

Setting up your working environment

Note

Since this section is about the installation, please try to go step by step on your own local machine. If you encountered any issues, We will help you figure out a solution in our next session.

0.1 Installing Python

To get started, we of course need to download and install the Python language. There are many, many flavours of Python, but for our purposes it mostly will not matter which one we choose. We will provide instructions on how to install Anaconda, which has some additional features that will come in handy.

Many operating systems, including macOS and Linux, come with Python preinstalled. The version of Python that comes with your operating system is called the system Python. The system Python is used by your operating system and is usually out of date. **Do not attempt to uninstall the system Python!** You can have multiple versions of Python installed on your computer. In this section, you'll install the latest version of Python 3 alongside any system Python that may already exist on your machine.

0.1.1 Downloading Anaconda

First, open your favourite browser, go to the Anaconda website (<https://www.anaconda.com/products/individual>) and download the appropriate installer for your operating system (you probably have a 64-bit system). Save the file someplace where you can find it again.

0.1.2 Installing Anaconda

For Windows Users: Find the installer you downloaded and double click it. You can accept the default options except for the section "Advanced Options". Click on Install and let the installer do its thing.

For MacOS Users: Navigate to the installer you downloaded, double click it. Follow the installer steps.

Make sure both the box about the PATH variable and the box about the system's default Python are checked.

Note on PATH

Programs and other executable files can be in many directories, so operating systems provide a search path that lists the directories that the OS searches for executables. The path is stored in an environment variable, which is a named string maintained by the operating system. This variable contains information available to the command shell and other programs. The path variable is named as **PATH** in Unix or **Path** in Windows (Unix is case sensitive; Windows is not). In Mac OS, the installer handles the path details. **To invoke the Python interpreter from any particular directory, you must add the Python directory to your path.**

For Linux Users: Open a terminal, navigate to the installer you downloaded type `bash Anaconda3-latest-Linux-x86 64.sh` and press return. Once again press the enter key and hold it until the User License Agreement has finished. If you agree, type `yes` and press enter. Once again press enter to confirm the location of the installation. After the installation has finished, once more type `yes` and press enter to confirm necessary additions to your `.bashrc` file.

0.1.3 Testing your Installation

To check whether everything is set up correctly, open a terminal. In the terminal, type python and press return. You should now see something like this:

```
(py3) saeed@Saeeds-MacBook-Pro:~$ python
Python 3.7.6 (default, Jan 8 2020, 13:42:34)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

This is the Python shell. It says your Python version (in my case 3.7.x).

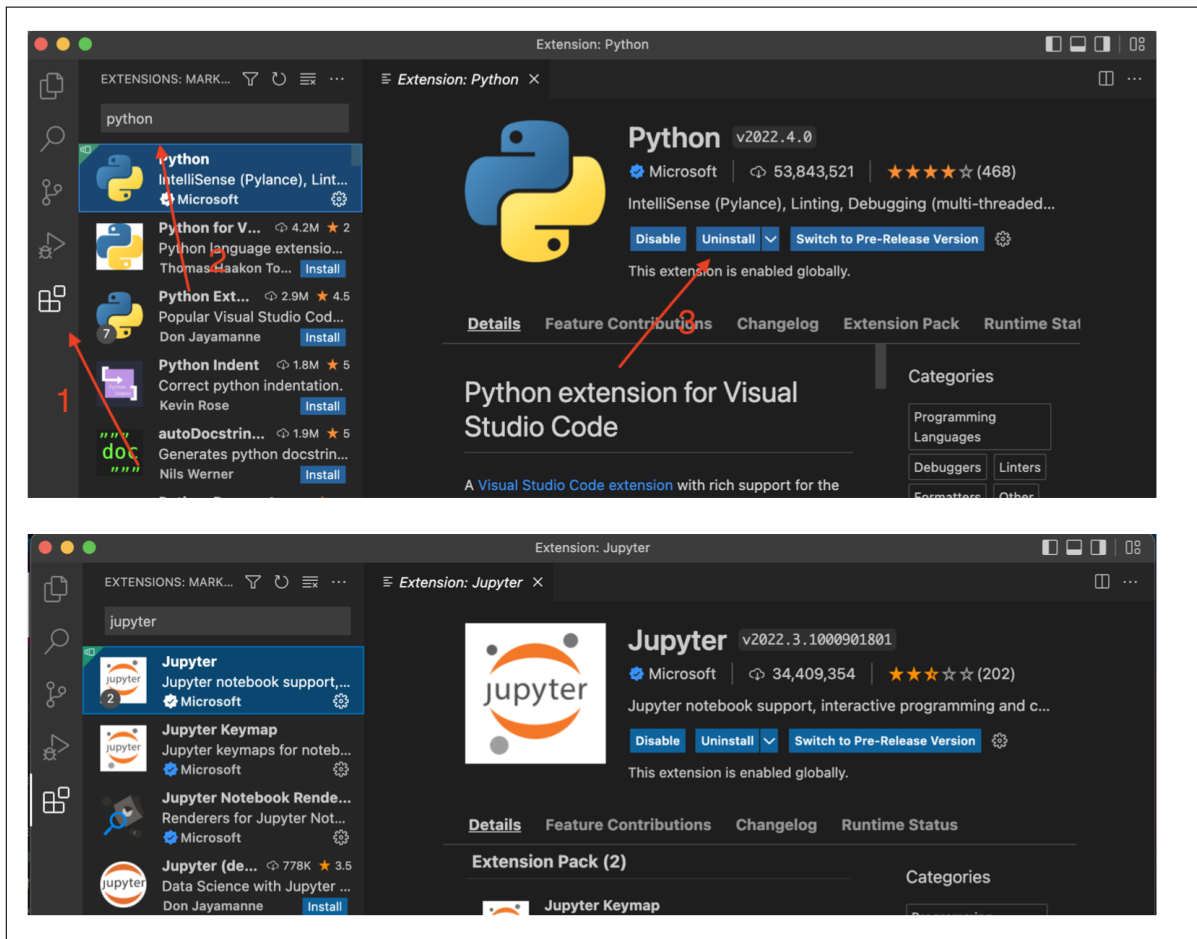
0.1.4 Installing Visual Studio Code

Python scripts are no more than simple text files. As such, you can use many different programs to edit them (but perhaps not all the ones you think - Microsoft Word for instance is not a pure text editor!). We suggest you use VSCode because it is simple to use and can be adapted to more complex needs as you progress in your programming life.

You can download VSCode from (<https://code.visualstudio.com/download>). The installation process should be straightforward. After the process is done, you just open visual studio code like any other program.

0.1.5 Installing Extensions for VSCode

We would like to have two extensions in our visual studio code, "Python" and "Jupyter". Follow the below screenshots to see how to install them.



0.1.6 One Last Step

On Visual studio code press `command+shift+p` on Mac or `ctrl+shift+p` on windows. This is the "Command Palette" Type "Python". Choose "Select interpreter". From here you should choose the `/opt/anaconda3/` (depending on your installation it might be a little different).

0.1.7 Python Shell

Usually, if you intend to write a program, you will just write into a file and go from there (we will cover this in the lecture). Python additionally offers a **prompt** for your terminal, the **python shell**. Simply put, this just executes every line of python code you give it immediately, and prints a result.

You can try this out by using the command `python`. If python is installed properly, you will now see three "greater than" signs (`>>>`) and a trailing cursor at the start of the line. Try typing in a calculation, like `6*7`. Once you hit enter, the result should display in the next line.

Executing Python in the interactive window can be described as a loop with three steps (**read, evaluate, print**).

The python shell has all the functionality of python, but will mostly come in useful for smaller things (like quickly trying something out).

Don't be afraid of breaking anything either. If you want to stop whatever you told python to do, you can use the keyboard interrupt: just press `Ctrl+C`.

You can end the python prompt by typing `quit()`, `exit()` or `Ctrl+D`.

```
(py3) saeed@Saeeds-MacBook-Pro:~$ python
Python 3.7.6 (default, Jan 8 2020, 13:42:34)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 6*7
42
>>> exit()
(py3) saeed@Saeeds-MacBook-Pro:~$ █
```

0.1.8 Python Script

With algorithms in mind, we often want to execute many lines of code in immediate succession. This is when we need a **python script**. We can in theory write python commands in most modern text editors (save the file with `.py` extension) and can execute that script with the terminal.

`python my_script.py` will execute your python script called `my_script.py` via the Terminal.

Using this method we can run multiple lines of code and not just a single line of code. But instead of using a simple text editor, we are using Visual Studio Code.

VSCoDe helps the user while implementing code and could also be used for other programming languages than Python. Smart code completion with IntelliSense will also help you throughout coding. We can also use VSCoDe to run our script directly in the application, instead of using the Terminal to run our code.

Tip: After opening the terminal, run command `conda activate` that will activate your conda environment.

Python BootCamp - Lecture 00

Tips on using the terminal

0.2 The Terminal

Throughout this course, we may make use of your system's terminal (or command prompt). If you don't yet know what it does, we will explain in this section (if you're already experienced with it, feel free to skip it). By the end, you should be able to navigate your files, create or remove files and folders, and have an intuition on what commands are. Of course, you can do much more via your terminal, but this is most of what we will need throughout the semester.

All popular operating systems give you the option to navigate and manipulate your files and folders through a visual interface, like Microsoft's Explorer or Apple's Finder. While the concepts are the same across different operating systems, we will need to separate between Unix systems (like Linux and MacOS), and Windows. The reason for this is that Windows uses different commands.

Please also keep in mind that you will not have to memorize all of these commands right away, you may always look them up and there are many cheat sheets available on the internet. If you use your terminal regularly, you will look things up less often over time.

0.2.1 Open Terminal

You can find the command prompt by typing `cmd` into the Windows search and hitting enter. If you are a Mac user, open Terminal app from Applications - Utilities. Note that if you are using Powershell instead of `cmd` (or something like the Anaconda Powershell prompt), you can also use many Unix commands, if that's your thing.

0.2.2 Commands List

Linux / Mac users (Terminal)

- List directory contents with *ls*
- Change directory with *cd* directory
- Copy with *cp* file directory (file can be renamed)
- Delete with *rm* file
- Move or rename with *mv* file directory (like cp, but deletes original)
- Create folder with *mkdir* directory_name
- Delete folder with *rmdir* directory_name

Windows users (Command prompt)

- List with *dir*
- Change directory with *cd*
- Copy with *copy*
- Delete with *del* file
- Move with *move* file directory
- Rename with *rename* old_name new_name
- Create folder with *mkdir* directory_name
- Delete folder with *rmdir* directory_name

Python BootCamp - Lecture 00

Getting started with python and the terminal

0.3 First exercise

Have a look at programming with Python using the terminal. You might skip this section if you prefer visual IDEs (Integrated development environment) like Visual Studio Code environment but if you want to taste command line programming feel free to do this small exercise.

0.3.1 Step One

First, open a terminal. Try to navigate to your desktop (the desktop is just another directory).

0.3.2 Step Two

It is probably a good idea to have a dedicated directory for this exercise. On your desktop, create a new directory called `TestPython` using the terminal. What goes wrong if you try to name it `Test Python` (with a space)?

Inside the `TestPython` directory, run the command `code script1.py`. This will create a new file `script1.py` and open it with VSCode. Save the file and go back to the terminal.

Create a copy of `script1.py` using the `cp` or `copy` (Windows users) command. Use `ls` or `dir` to show both files.

Now use `rm script1.py` to delete the file `script1.py`. Did you notice there wasn't any warning or confirmation? The file also did not go to the trash folder, but was deleted permanently.

You can do a lot of damage to your system with the terminal, so always double check what you type in..

Lastly, Use `mv` command to rename the copy you created before to `first_script.py`.

0.3.3 Step Three

For this task you have to open a Python shell by opening a terminal. and then executing the command `python`.

In the lecture you learned that the Python shell can be used like a text-based calculator. In this task we want you to make use of it. Using the Python shell and the operators `+`, `-`, `*`, `/` and `**`, calculate the results of the following formulas:

- $5 + 2 - 7$
- $\frac{7}{5}$
- $12 * 274.3$
- $2^{\frac{24}{7}}$
- $\frac{3674+28998}{(3648+2321)*43}$

Finally, which commands would you use to get the following outputs in the terminal?

- How are you?
- Hello Hello Hello