Learning_Resources

April 8, 2021

---

1 Requirements

Due to the large number of Python versions, for this course we decided to use Python version 3.5 or above. However,

1.1 Python installation

The easiest way to install python is via Anaconda: https://www.anaconda.com/products/individual

Simply put, Anaconda is an environment manager for Python/R, which simplifies the installation of different versions of Python. Furthermore, it allows to create separate and independent python environments, each with different versions for both python and its libraries.

For this course we will assume you have installed python using Anaconda

→ Linux

The most widely used Linux distributions already come with a pre-installed version of Python. However we advise against using the operating system's python distributions as some modifications/customizations can mess-up the system. To install anaconda on a linux system you just need to download the correct installer and run it on the command line:

```
$ cd path_where_installer_was_downloaded
$ sh ./Anaconda3-2020.11-Linux-x86_64.sh
```

→ Windows

The installer for windows is graphical, so you just need to download the correct version and follow the on-screen instructions. Note that the installer will also install *Anaconda Navigator*, a graphical tool used to manage your virtual environments.

→ MacOS

In MacOS, anaconda can either be installed via a graphical installer (like Windows) or command line (like Linux). Like with Windows, the graphical installer will install *Anaconda Navigator* (see above).

1.2 Optional tools

1.2.1 IDE's (Integrated development environment)

Python does not require a specific IDE as python scripts are just text files, and this can be edited by any text editor. However, using a more advanced IDE has benefits, namely in terms of code completion, verification, etc. The majority of paid/professional IDE's have a "student" pack, that gives, without charging, all of the paid features. To obtain it, a university email is needed.

Here are a few of the best known IDEs for Python:

→ Visual Studio Code (https://code.visualstudio.com/)

Albeit not a dedicated python IDE, VS Code is probably one of the most powerful IDE'ss around. Out of the box, it is little more than a text editor. However its strenght comes out of its Extension Marketplace. There you can find literally any Extension you will require to expand its capabilities to turn it into the only IDE you will require for all your projects (LaTeX, Python, Java, C, Bash, …). Furthermore, it has an extensive and clear documentation. To turn it into a Python powerhouse you can folow the instructions at https://code.visualstudio.com/docs/languages/python

→ PyCharm (https://www.jetbrains.com/pycharm/)

PyCharm is a fully dedicated python IDE, and probably the most complete. Out of the box, it does code completion, re-factoring, project management, source control, you name it. On the downside, it uses quite a lot of resources to run, namely RAM.

→ Spyder (https://www.spyder-ide.org/)

Also a fully dedicated python IDE, Spyder is probably one that people know better due to its simplicity and lightweight.

1.2.2 For Windows Users

Although Anaconda offers a bash terminal to run python code, it is quite basic and limited. However, as of late, Microsoft has been converging towards the linux architecture, introducing a linux subsystem that works as a linux virtual machine totally integrated with the Windows OS, giving Windows users a *Linux-like* experience. To make use ot its potencialities, users only have to install

→ the Windows Subsystem for Linux (WSL, https://docs.microsoft.com/en-us/windows/wsl/install-win10)

→ a Linux distribution (you just need to search for "Linux" on the Microsoft Markeplace and several will show up, such as Debian, Ubuntu, Fedora, Suze, …)

→ the Windows terminal (optional, https://github.com/microsoft/terminal)

These tools, combined with VS Code, turn any Windows machine into a powerful python (and not only) developing platform.

2 Running Python

2.1 2.1 Managing anaconda environments

→ create a conda environment

“‘ $ conda create -n env_name python=3.5

```
&#8594; activate a conda environment
```

>```
$ conda activate -n env_name python=3.5
```

→ deactivate a conda environment

Exit terminal or

“‘ $ conda deactivate -n env_name python=3.5

```
→ adding libraries to your environment
```

>```
$ conda install -n env_name lib_1 lib_2 lib_3
```

or, from inside your environment, >`$ pip install lib_1`

2.2 Running Python

Once python is installed, and the environment is active, to run it you just need to open a terminal and type: >`$ python`

to open the python shell and start scripting.

Another alternative is to write the script in a text editor/IDE and run it from the command line:

```
$ python name_of_the_script.py
```

3 Some useful resources

→ **Stackoverflow** (https://stackoverflow.com/)

Stackoverflow is one of the largest websites for the programming community. Most of the problems that one might find should have some answer in it! and if it's not there, you can submit your own question and expect for an answer in a short period of time.

→ **Hackerrank** (https://www.hackerrank.com/domains/python)

It provides a wide array of problems, with different difficulties. It also allows one to test their own code against some tests.

→ **Euler Project** (https://projecteuler.net/archives)

This site has a large amount of math-related challenges. These are generally are more math-heavy.

→ **Coursera** (https://www.coursera.org/)

Coursera is known worldwide for its extremely large database of online course, spanning over themost diverse subjects. And of course, they also offer courses on python.

→ **some youtube channels:**
https://www.youtube.com/results?search_query=corey+schaffer

https://www.youtube.com/user/sentdex
https://www.youtube.com/channel/UCI0vQvr9aFn27yR6Ej6n5UA