Python Short-course Study Guide

Last updated: Franco Pretorius (2021-08-15)

1 Introduction

The aim of this course is to teach the very basics of Python programming for engineering purposes. This will make it easier for you to navigate the online documentation and tutorials in future. These instructions assume you have a Windows operating system.

- The most important instructions in every section below are given in a blue box.

Contents

2 Installing Python

We will use the Anaconda distribution of Python. Anaconda 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.

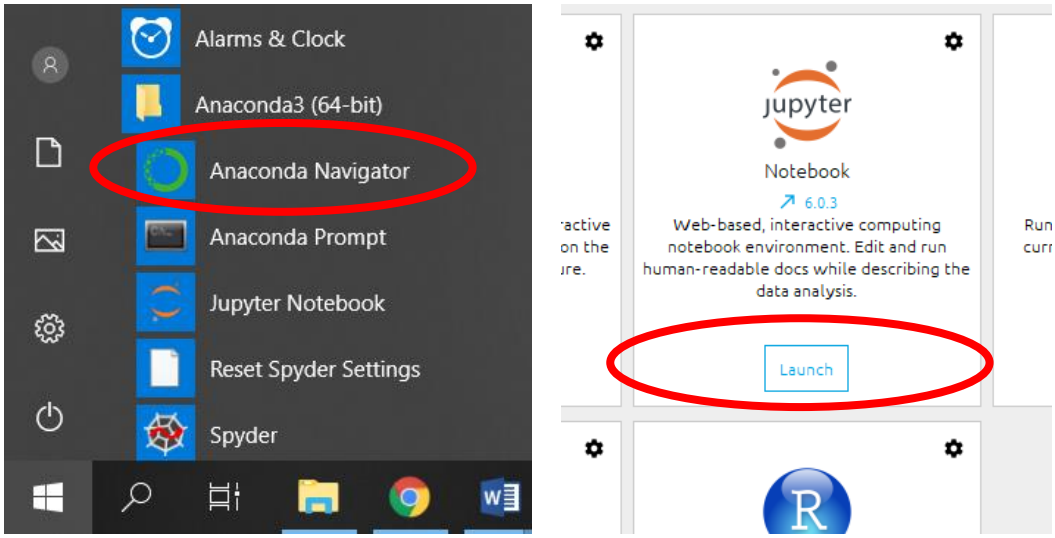
- Download Anaconda3 here (Windows 64-bit)

Further information is available on the UP Chem Eng wiki. Note (if you see old code on the internet) that Python 3 came out in 2008 with some new syntax. Python 2 will have no support past 2020.

After installation, the getting started guide should open. You may skip the section on Spyder and read *Run Python in a Jupyter Notebook*, or just close the website and carry on with the instructions here.

3 Opening Jupyter

- In your start menu, under the Anaconda3 folder, open the Anaconda Navigator (see Figure 1 below)
- In the Anaconda Navigator, launch the Jupyter notebook (alternatively, you can click the Jupyter notebook button in your start menu as a shortcut)
- This will open the Jupyter dashboard in your internet browser



(a) (b)

Figure 1: (a) Anaconda in the Start menu; (b) Jupyter notebook launch button

The dashboard will have opened in your web browser (note that an internet connection is not required). As shown in Figure 2, it should show the files in C:\Users*Your Name*.

After you have become accustomed to using Python, you may wish to change the start-up folder. To do this, consult the documentation and this Stack Overflow post for help.

3.1 Creating a new notebook

Check if everything is working by creating a blank new notebook. In your File Explorer, create a folder where you want to keep your Python material (*e.g. My Python notebooks*). Navigate to this folder in the Jupyter dashboard. Click the "New" button (see Figure 2). and select Python 3. The new notebook will open automatically as *Untitled*. Click on the name (see Figure 3)and type a new name for your notebook (*e.g. My first notebook*). Back in the dashboard menu, your notebook should appear with a green book showing that the kernel is active. Now select the notebook and delete it (Figure 4).

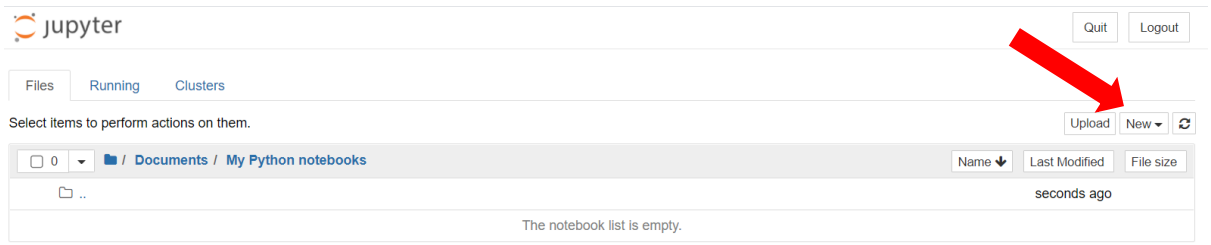


Figure 2: Jupyter dashboard in Google Chrome

Figure 3: A new notebook. Click on the word "Untitled" to rename the notebook.

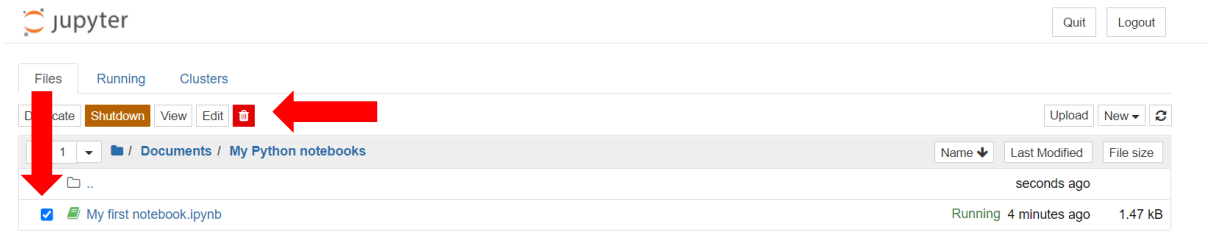


Figure 4: Checking left of a notebook brings up the red (delete) button.

4 Version control

Code is often developed simultaneously, thus a version control system (VCM) is necessary. One example of a VCM is Git, which works both online and offline. It shows who made changes at which times and allows you to revert back to old versions. GitHub is a website that hosts open-source coding projects using Git.

Many useful code repositories are available publicly on GitHub. To transfer this code to your computer, apply one of the three different options presented in the following subsections.

1. **Download the ZIP file every time:** This is simplest method and is meant for people who are not contributing to the repository. However, the downsides are that you will have to download the whole thing each time that there is an update to the repository, and you will write over your own changes.
2. **GitHub Desktop app:** This is an application that enables you to interact with GitHub using a GUI (graphical user interface) instead of the command line. It is likely to be suited for nonprogrammers who still want to pull the code.
3. **Git Bash:** This uses the command line interface to interact with GitHub.

If you are planning to push code to GitHub (*i.e.* making your code available to public or private groups, instead of just pulling repositories to your computer), you should create a GitHub account.

- Find the Python Tutorial notes on GitHub here. (https://github.com/Franco-Pretorius/Python-Tutorial.git)
- Click the green Code button (see Figure 5).

4.1 Download ZIP folder

Click the Code button, then the Download ZIP button at the bottom of the menu.

Be aware that when updates are applied to the repository you will have to download the entire new repository again. It is recommended to then delete the old folder on your computer and keep your own code/changes in a separate folder.

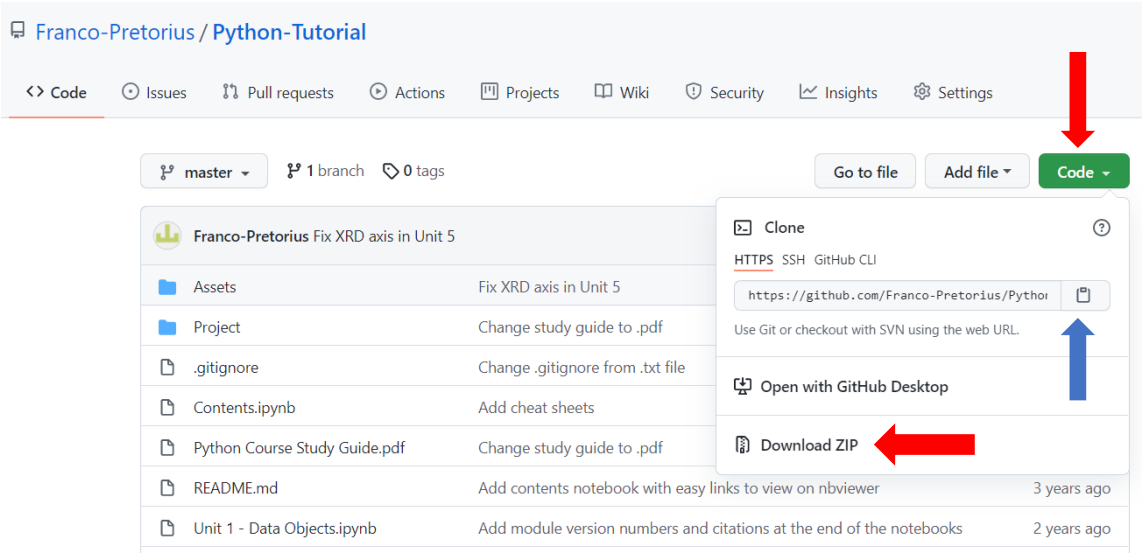


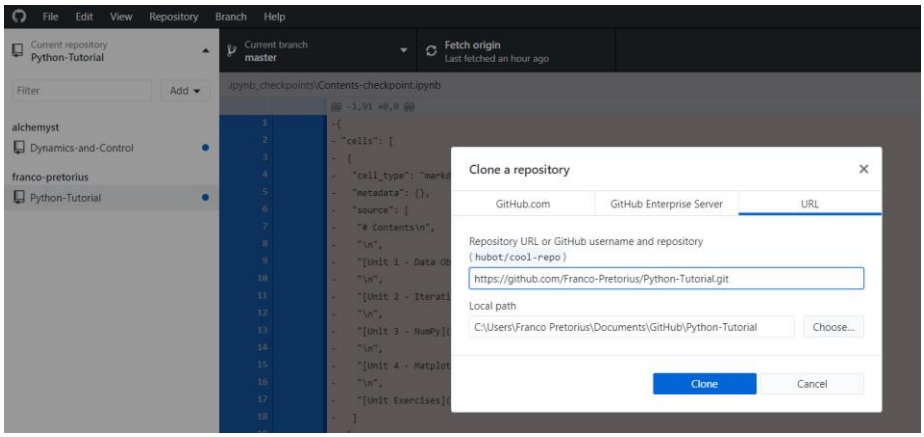
Figure 5: A GitHub repository. If you wish to clone a repository, press the clipboard button (blue arrow) which will copy the URL.

4.2 GitHub Desktop

Download the application here.

4.2.1 Cloning a repository

Once you have it opened, click: Current repository at the top left corner, then Add, Clone repository after which the following menu should open. Paste the URL from Section 4 under the URL tab. Change the local path to where you would like the repository. Press **clone**. This will clone a local repository on your computer. You can now navigate to your notes in the Jupyter dashboard and open them.



4.2.2 Updating a repository

If the owner of the repository updates the code, you can **pull** the new version by clicking the Repository tab, then Pull.

If you want to create your own repository, read this.

4.3 Git Bash

4.3.1 Cloning a repository

To clone the files (*i.e.* to create a local repository for the first time) do the following. If you get stuck, look at the example here.

1. Download Git Bash here (click the picture of a computer screen with a blue button inside it).
2. Complete the default installation.
3. Find the folder on your computer where you'd like to clone the Python-Tutorial notes and right-click to select Git Bash (a black terminal should open). Alternatively, you can open the Start menu and click on Git Bash, then change the working directory by following the advice here.
   In short, type cd "c/Users/Your name/Document/My Python notebooks"
4. Type **pwd** and press enter to print the working directory (to make sure you're in the right folder where you'd like to clone notes).
5. Copy the repository URL as mentioned in Section 4.
6. Go back to the black terminal and type **git clone** *followed by the URL* you copied, and press Enter.
7. After it has been cloned, you may close the terminal window.

4.3.2 Updating a repository

To get the latest version of a repository from GitHub (after the author has made updates):

1. Open the local repository (Python-Tutorial file) you have cloned on your computer, right-click and select Git Bash here.
2. Type **git reset --hard** and press Enter. This restores the local repository to where it was when you had it originally by undoing all changes you have made (like running, adding or removing cells).
3. Type **git pull** and press Enter. This downloads the new files and the ones which have changed.

More information on making a repository is available here.

5 Customisation

Extra packages can be installed from the Python Package Index (PyPI) using pip, which is included with Anaconda. pip is a recursive acronym for "Pip Installs Packages". Note that another package management system, **conda**, is also often used to install packages which may contain software written in any language.

Currently, there are some technical difficulties with these options, and you may have to restart your computer for it to work.

Using pip, you can add a menu of extensions to your Jupyter dashboard.

- Open the Anaconda Prompt from the Start menu, and execute the following: **pip install jupyter_contrib_nbextensions**

The extensions include a spell-checker, variable inspector, table of contents, and even a button to format code according to PEP8 automatically.

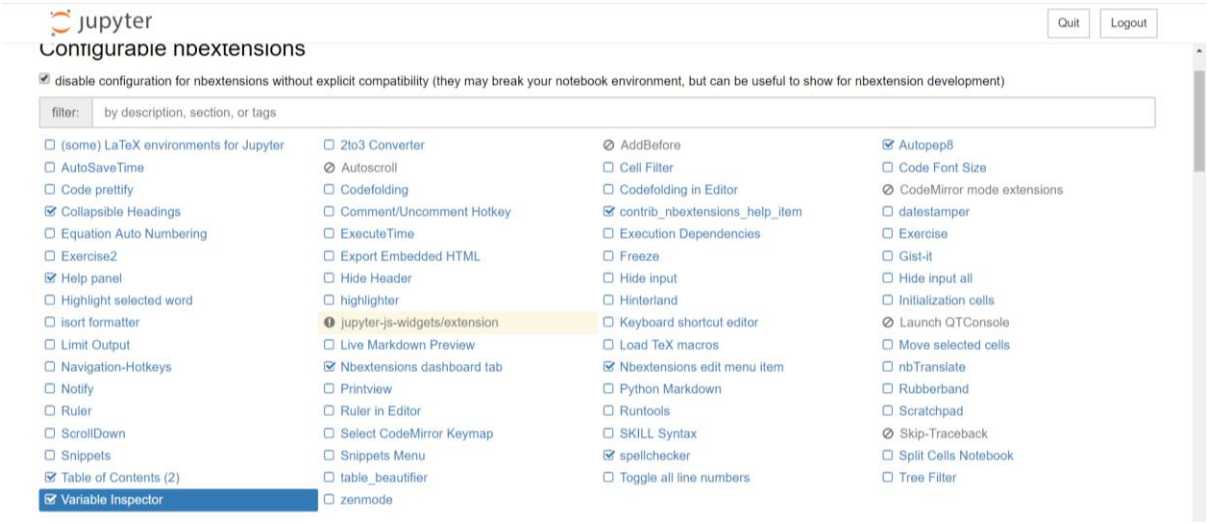


Figure 6: Some handy notebook extensions

You can also change the appearance of notebooks (e.g. use dark mode)

- Open the Anaconda Prompt from the Start menu, and execute the following: **pip install jupyterthemes**

Within a notebook, run the following in a cell to see the list of themes: **!jt -l**

They include:

- onedork
- chesterish
- grade3
- oceans16
- monokai
- solarizedl
- solarizedd

To change the theme run the following (**jt -t <name of the theme>**) and refresh the page.

The following is the creator's preferred style:

!jt -t onedork -fs 95 -tfs 11 -nfs 115 -cellw 88% -lineh 130 -T -N -kl

To reset the theme, run: **!jt -r**

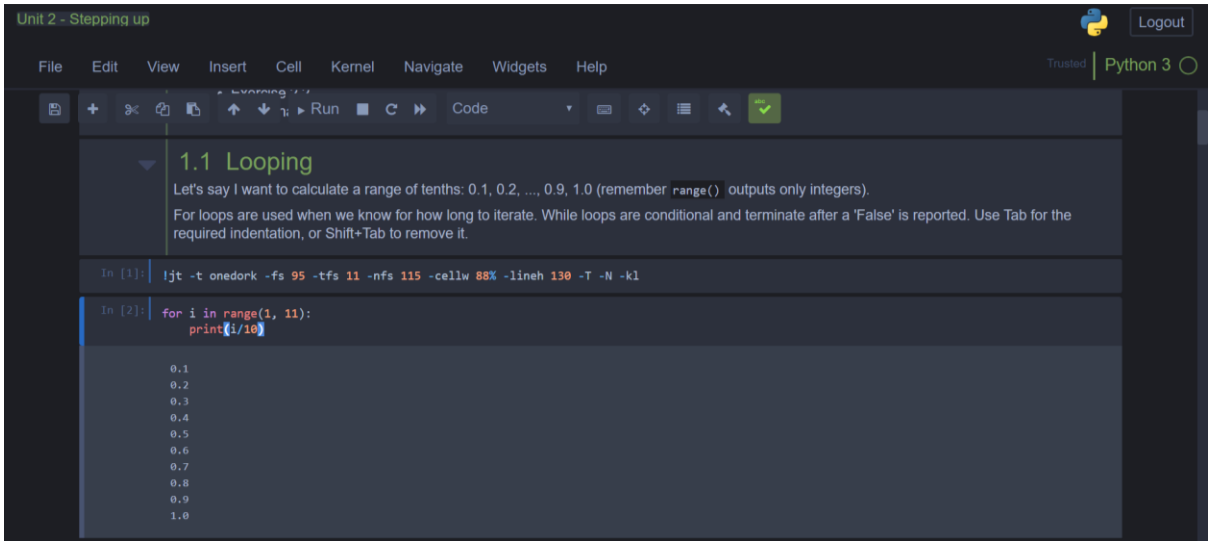


Figure 7: Custom dark mode for a notebook

6 Course outline

The course notes introduce a new module per unit. It is recommended to do the Unit exercises before continuing with the next unit.

Extra help and additional examples:

- Python Tutor (to visualise code, especially looping)
- NumPy for MATLAB users
- Markdown syntax
- Interesting Jupyter Notebooks
- Python for Chemical Engineers by CAChemE
- Chemical Engineering Analysis and Control by JC Kantor
- Chemical Engineering Python Cookbook by C Sandrock
- Dynamics and Control by C Sandrock