

# 1.041/1.200/11.544: Introduction to Python

Our course will make use of the python programming language, which is an easy to learn and powerful programming language. Python is often used for Data Science applications and has a large ecosystem of libraries for machine learning, optimization, etc.

In order to get started with python, some general previous (first) programming experience in high level language (such as, e.g., MATLAB) should be enough to pick up the necessary tools and knowledge. A nice feature of python is that there is an enormous amount of high-quality free resources available online for learning, ranging from online tutorials to video lectures, books and interactive notebooks.

Below, we have provided some information and a few links as a starting point. In addition, we provide some recommendation that may be useful, in particular at the beginning. You do not have to stick to those recommendations however.

## 1 Python – Installation and environment

Before you can start you will need to have a working python installation on your computer. Installing python is not difficult, and you should be able to set it up on your computer without major problems.

The official python homepage – where you can download python and find a lot of useful documentation.

- <https://www.python.org/>

However, you may find it simpler to install a python environment via a 'package' such as Anaconda, or Canopy.

- <https://www.anaconda.com/download>
- <https://www.enthought.com/product/canopy/>

This will install not only python but also a graphical user interface to control your installation and a host of useful packages/toolboxes, some of which we make use of within the course (this includes in particular the matplotlib package for plotting, numpy and scipy for numerical and scientific computing). We would thus recommend using one of those two installation options though you are free to use any other variant. There might be a default version of python installed on your computer as well. If so make sure that you have an up-to-date version.

Note that there are two versions of python available. Python 2.X and Python 3.X. Both will be fine for the purposes of the course. However, for the computation problems which we will be providing you with the initial framework, we will use python 3.X. It should be noted that there are some small differences in the syntax and conventions (be aware of the differences when it comes to division by an integer!) in the two versions.

If you feel uncomfortable with installing python etc., simply use the Anaconda or Canopy python distributions.

## 2 Introduction to python

### 2.1 Python 3.X

Since most of the computational problems of this course will have an initial framework written in python 3.X, we recommend using python 3.X for this course. To get yourself familiarized with python 3.X, we recommend going through the first eight notebooks (0 to 7) provided in the link below. Notebook 8 and 9 are optional but may be helpful for the course.

- <https://gitlab.erc.monash.edu.au/andrease/Python4Maths/tree/mastercontents>

### 2.2 Python 2.X

If you are interested in learning python 2.X, we recommend going through the eight notebooks (0-7) as provided in the link below.

- <https://github.com/rajathkmp/Python-Lecturestable-of-contents>

### 2.3 Matplotlib - Python 2D plotting library

Matplotlib is a Python 2D plotting library which is used in this course. We recommend going through few of the official tutorials to get yourself familiarized with 2D plotting in python.

- <https://matplotlib.org/3.1.1/tutorials/index.html>

## 3 Some further recommendations

The official python homepage has a complete tutorial on python programming. This documentation is in general a great resource and contain a lot of detailed information.

- <https://docs.python.org/3.6/tutorial/>

While there are many ways to write and run programs in python, many people find it easy to work within a more visual environment. If you prefer such a working environment, you may find that Pycharm and Spyder as convenient environments for you.

- <https://www.jetbrains.com/pycharm/>
- <https://github.com/spyder-ide/spyder>

Spyder is also included in the Anaconda distribution.

Another useful environment to use is called jupyter (or sometimes called ipython) notebooks. These notebooks allow you conveniently to combine text, LaTeX, and code-blocks in a single document, which can be displayed and executed from your browser and be exported to pdf/html.

- <https://jupyter.org/>