Y1 Computing Primer

Dr. Brian D. Appelbe & Dr. Aidan J. Crilly

October 2022

1. Introduction

Welcome to Physics at Imperial College London. An essential tool for all Physicists is the art of computer programming. At Imperial we teach a programming language called Python. For some of you this will be the first time you have ever had to program. Don't worry, the Y1 Computing course assumes that you have no prior programming experience. By the end of the course you will be able to write your own programs in Python and use it for analysing experimental data.

This introductory guide contains information on Python, the software that you will use for writing Python code and some tips for setting this up.

2. What is Python?

Python is a programming language used universally in academia and industry. The official website for Python describes it as "an interpreted, object-oriented, high-level programming language with dynamic semantics". Let's briefly examine what this statement means:

- Python is *interpreted*: Once you have written your code, you can run it immediately. In programming languages that are not interpreted you first need to compile your code (translate it into a set of instructions that your computer operate on) before you can run it.
- Python is *Object-Orientated*: Everything you encounter in Python is based on an object, which is a data structure that contains data and functions associated with it. We will use this aspect of Python passively in this course and you will learn more about it later in your degree.
- Python is as a *high-level* programming language: It is written in a way that is very human friendly to read and execute.
- Python has *dynamic semantics*: Variables can be declared and changed within the middle of code execution, and do not need to be declared ahead of time.

As your programming knowledge develops, the significance of these features will become clear to you.

2.1 Why do scientists use Python?

There are a variety of reasons why Python is popular with scientists:

- Python is open source. This means anyone can download it and start developing with it.
- Because it is a high-level programming language it is easy to quickly develop code and implement ideas.

- Python has a wide variety of tools, known as packages, that have been developed to carry out many scientific tasks. These packages include plotting tools that allow you to create line plots, histograms, box plots, etc. and numerical tools that allow you to calculate important statistics such as the mean and variance of your data.
- It is very general and is not just for scientists, making it appeal to a wide selection of people.

3. Software that we will use

We will use a software package called Anaconda. Anaconda is the distribution of Python. It contains applications for writing Python code and also many packages (pre-written Python code that carries out specific tasks) that are useful for scientific computing and data science. We will use two applications of Anaconda for writing Python code. These are Jupyter Notebooks (for Computing Lab sessions 1 & 2) and Spyder (from Computing Lab session 3 onwards).

For Y1 Computing Lab sessions you will use computers in the Blackett Computing Suite, on which Anaconda is pre-installed. We also strongly encourage you to install Anaconda on your own computer so that you can practise your Python programming at any time.

3.1 How to install Anaconda on your own computer

Anaconda is free to install on your own computer. It can be downloaded here . Follow this link to a video describing the installation process for Windows. The procedure for installing on other operating systems will be similar.

3.2 Remotely accessing a computer in the Blackett Computing Suite

It is also possible to log onto a computer in the Blackett Computing Suite from your own computer. This is called remote access. Instructions can be found on the Imperial ICT webpages here .

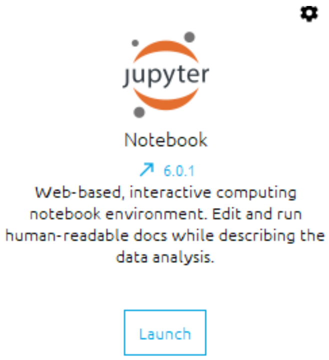
4. Getting started with Jupyter Notebooks

For the first two Computing Lab sessions you will use an application called Jupyter Notebooks for writing your Python code. Here are the basic steps for launching a Jupyter Notebook on a computer in the Blackett Computing Suite:

1. Go to the Imperial software hub. This should open automatically after you log on, but if you don’t see it then here is the link .
2. Launch the Anaconda Navigator. The Anaconda Navigator logo looks like this

When you hover your cursor over the logo then a launch button will appear. Click on the launch button.

3. Launch Jupyter Notebook from the Anaconda Navigator menu. The Jupyter Notebook logo looks like this



Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

4. When you launch Jupyter Notebook this will open a browser page which shows the contents of your Jupyter Notebook start up folder. When you are using a computer in the Blackett Computing Suite then the Jupyter Notebook start up folder will be in your H: drive (link). You should make a new folder here for your computing work so that you can easily find it. Give it a sensible name such as “My Computing Lab session 1”.
5. Go to the course folder on Blackboard, download the files for Computing Lab session 1 and put them in your new computing folder.
6. Go back to your Jupyter Notebook browser page and navigate to your new computing folder. Click on the file called “Worksheet Core Session 1.ipynb”. This will open a new browser page showing the Jupyter Notebook for your first Computing Lab session.

The process of launching and navigating between folders with Jupyter is shown briefly in this video (link).

If required, follow this link to find out how to change your Jupyter Notebook start up folder.

5. Some useful resources for learning Python

The Computing Lectures and Lab sessions will cover the basics for learning to program in Python. However, in order to become a skilled programmer you will need to be able to learn independently. An essential part of this is to effectively use online resources in order to find solutions to problems and to learn new tools and techniques. Useful online resources include:

- Python.org - this is the official website for the Python programming language. It contains extensive documentation on Python, including a beginner’s guide.
- Web Search Engines - If you have a question or problem with your Python code then it is likely that other programmers have had similar issues. An online search can often provide answers.
- Stackoverflow.com - this is an online community where programmers can ask and answer programming questions.
- NumPy.org and SciPy.org - NumPy and SciPy are two Python packages that are very useful for scientific computing. You will be using them throughout the course and the websites have useful documentation.

- Matplotlib.org - this is the Python package that we will be using for making plots.