

Python Programming and Data Visualisation for Beginners

Congratulations on deciding to learn a new skill!

Python is a popular, general purpose programming language with applications in web development, software development, mathematics, data science, and many other areas. It has a thriving community and a long-term outlook, so the time and effort that you invest in learning Python now is likely to pay dividends in the future.

This 8-week course focuses on data visualisation. By the end of it you will be able to load data into Python and make basic charts to describe the data, and you should also see the benefits of continuing to learn Python and using it instead of, or alongside, other familiar software such as Microsoft Excel.

Although our focus will be on working with data, we will start from the very beginning and much of what you learn will prepare you for any future Python endeavors. For example, you may end up using Python as part of your everyday workflow, or you may be inspired to apply it to existing hobbies or to undertake fun and rewarding projects, such as using a [raspberry pi microcomputer](#) to make an [infrared bird box that live streams to YouTube](#), an [automatic cat feeder](#), an [environmental sensor for monitoring the health of plants](#), or a [face recognition door lock](#).

Course overview

This is a broad outline of what we will cover in the 2-hour sessions over the next 8 weeks. It may be subject to change at short notice! Please note that I will also provide practice worksheets that should keep you busy for 1-2 hours between our weekly sessions.

- **Week 0:** *An introduction to the Python programming language.* This will cover the history and community of the Python programming language and offer a broad overview of how it can be used. We will also take the opportunity in this first session to introduce the software that we will be using throughout the course ([Anaconda](#), [Jupyter](#), [Spyder](#)) and to make sure that everyone has downloaded and installed them successfully.
- **Week 1:** *Syntax, data types and built-in functions.* This will be an introduction to the very basics of Python. We will talk about syntax, the rules that govern how Python code should be written, and data types, the "objects" that are used to store bits of information, such as lists, tuples, booleans, dictionaries, integers, strings, and floating points. We will also consider functions that are already "built in" to the language.
- **Week 2:** *Logical operators, control statements and defining functions.* This will be a light introduction to logical operators (e.g., **and**, **not**, **or**) and

control statements (e.g., `if`, `while`, `for`), which are combined to determine how a program is executed. For example, you may want to only execute some code if a certain condition is true, in which case you can use a logical `if` statement. Or you may want to execute some code until a certain condition is met, in which case you would use a `while` statement. We will also cover how to define functions to handle repetitive tasks.

- **Week 3:** *Imports, the standard library, and third party libraries.* In Python, most functionality is buried away in modules that must be imported at the beginning of a script. This session will cover how to locate and import relevant modules, and it will explore what is already available in the [Python standard library](#). Examples include `os`, `time`, `datetime`, `math`, `random`, `csv`, etc. We will also talk about third party libraries. Owing to its popularity and widespread adoption, Python has a thriving development community. There are literally hundreds of thousands of Python packages, all designed to solve particular problems. This session will introduce the [Python Package Index](#) (where different python packages are stored) and how to search for and install packages that may be of interest. Finally, I will introduce three key packages that we will use in the next four sessions: `numpy`, the core numerical computing package for python; `matplotlib`, a powerful plotting library for making graphs; and `pandas`, an advanced package for handling and manipulating large datasets.
- **Week 4:** *An introduction to data visualisation.* Today we will begin by exploring basic principles of data visualisation and considering different methods of presenting data, such as line graphs, scatter plots and bar charts. We will look at examples of what are widely considered to be some of the best and worst data visualisations ever made, which is a great way to refresh one's chart reading abilities. I will then introduce three separate data sets that we will be using for the rest of the course and demonstrate how to load the data and make basic charts.
- **Week 5:** *numpy, matplotlib, and pandas.* This session will explore these three packages in depth. We will begin by exploring the key functionality of `numpy` and work through examples of how to create and operate on numerical arrays. Then we will import `matplotlib` and visualise the data from `numpy`. Finally, we will import `pandas` and use it to load and describe the example datasets. By the end of the session, you will be able to import these packages, load example data, and make basic plots.
- **Week 6:** *Locating and visualising interesting data.* This session will focus on locating data and describing the data with visualisations using `numpy`, `matplotlib` and `pandas`. I will then set an assignment for nexts week's presentations, which will involve locating some data, loading it into python, and making a plot that describes the data. You may choose from some set examples that I will provide, or come up with your own idea entirely.
- **Week 7:** *2-5 min presentations and inspiration.* In this session you will

give a short presentation on the assignment set in the previous week. I will then describe some of my own Python projects and aim to inspire you with ideas for continuing your Python development.

Getting help

As a beginner learning Python, the internet will almost certainly have answers to any questions that may arise, but knowing how to find them can be tricky. A carefully worded google search is usually the best place to start for specific issues, but you may wish to bookmark the following resources.

General

- [What is Python?](#) - the executive summary.
- [The Python Language Reference](#) - Technical, informative, dry. If all of Python suddenly disappeared and had to be re-implemented from a single document, this is the one.
- [The Python Standard Library](#) - Documentation for the 200+ modules in the standard library.

Self-guided tutorials

For supplementary learning during the course, or continued learning afterwards.

- [The official Python Tutorial](#) - Comprehensive and at times entertaining.
- [W3schools' Python tutorial](#) - Comprehensive, interactive and easy to navigate.
- [Codecademy Python tutorials](#) - Free and paid courses covering a wide range of topics and for all levels of ability.

Websites

- [RealPython](#) - A great website with tutorials and articles for all abilities.
- [learnpython.org](#) - Another great websites with trivia and tutorials for all abilities.
- [StackOverflow](#) - Probably the best place to search for answers to Python-related questions, or to ask questions if you can't find the answer you are looking for.
- [PythonCheatSheet](#) - A very useful language reference for all levels

Social media

If you use Twitter and want to fill up your feed with Python trivia, consider following these accounts:

- [@driscollis](#) - Mike Driscoll writes Python tutorials and posts about Python every day. His "Python pop quizzes" often focus on strange and unexpected features of Python.

- [@randal_olson](#) - Randy Olson is a data scientist who frequently tweets about Python and data visualisation.
- [@TalkPython](#) - The #1 Python-focused podcast covering the people and ideas in Python. Created and hosted by [@mkennedy](#).
- [@PythonHub](#) - News and links about Python programming.
- [@PythonStack](#) - StackOverflow feed of Python questions matching certain criteria.

Finally...

It is often said that Python is one of the easiest programming languages to learn. This may be true in a relative sense, but the learning curve can be steep for beginners. If at times you find yourself feeling frustrated, overwhelmed, or out of your depth, remind yourself that this is completely normal, and believe me when I say that there is nothing quite like the feeling of satisfaction that accompanies those *Ah ha* moments, when your perseverance pays off and you finally *get it*!

I hope you enjoy learning Python. Please don't hesitate to contact me directly if you need a hand at any point throughout the course.

Joel Martin

joel.martin@york.ac.uk