

Programming with Python

General information

Lecturer: [Dr Stefan Güttel](#). Office hour: Wednesday 14–15 in ATB 2.114.

Email: stefan.guettel@manchester.ac.uk **(Not for computer code!)**

The teaching times are as follows:

- Thursday 11–12, lecture in ATB, G.205
- Thursday 15–16, lecture in ATB, G.209
- Friday 9–11, lab classes in the ATB computing cluster G.105

The lectures present the technical knowledge and discuss some of the exercises.

The lab classes are designed for you to solve and discuss exercises and assignments.

Feedback on code

If you are new to programming, you will make a lot of mistakes (bugs) in the beginning and you will require feedback on your code writing.

For obvious reasons (viruses, uniformity of help, etc.) your lecturer will not give feedback on code being sent via email.

However, the lab classes provide a great opportunity for getting feedback. There will be 3-4 student helpers and myself for this purpose, so make sure to make use of this!

Feedback on code will be given in the lab classes and only there!

Assessment

There will be four lab tests (8 problems in total), namely in weeks 3, 5, 7, and 8 (or 9 depending on speed) of the semester.

The results of the 6 best problems count 30% towards the final mark.

The remaining 70% are coursework, which will be a large programming project specified around Easter time.

Every week (except week 1) there will be exercises to be completed before the lab classes. These exercises are usually posted on Wednesdays but may be available earlier.

Exercises are very important for you to actually learn Python and to prepare for the lab tests. Make sure to solve them!

Discuss: What *is* programming actually?



Discuss: Why should we learn about programming?

<https://www.weforum.org/agenda/2016/01/8-jobs-every-company-will-be-hiring-for-by-2020/>

https://www.monster.co.uk/jobs/search/?q=python&where=&cy=uk&intcid=HP_HeroSearch

<https://www.techworm.net/2016/06/15-highest-paying-programming-languages-2016.html>

Can you name 5 devices/places/industries that rely on computer programs?

We often don't even notice running computer programs until they crash...



Can you name 5 devices/places/industries that **do not** rely on computer programs?

So, how does one learn how to program?



Install Anaconda on your computer. (First assignment.)

Learn Python (or any language) interactively! Trial and error!

Use Google. (Example: “How to sort a list in Python?”)

Study the lecture notes carefully. Solve exercises. Discuss problems.

Read online resources. Watch online tutorials. For example:

<https://www.codecademy.com/learn/python>

<https://www.learnpython.org/en/>

<https://mva.microsoft.com/en-us/training-courses/introduction-to-programming-with-python-8360>

Will I ever be done with learning how to program?

No.

In fact, this course will only get you started. If you want to become *good* at programming, you have to actively work on this even after the course is finished.

If you want to become *really good* and catch a programming job, you ideally have to start your own projects, maybe your own software company, get involved with the community, keep up-to-date with latest developments, etc.

<https://madlab.org.uk/groups/python-north-west/>

What is Python?

Python is a computer **programming language**. This means, it is a language that you can use to give instructions to your computer.

You can think of your computer as a good friend who follows your instructions, and Python would be the language that you communicate in with him/her.

There are various ways of communication (just as in the “real world”: phone, email, chatting, etc.), some of which are more interactive than others (chatting is more interactive than writing a letter, for example).

Similarly, there are various ways to use the Python language.

A little bit of terminology

Python: One (of many other) **programming languages** we will be using. It is the language we will write computer programs in.

IPython: A Python **interpreter**. A computer application that provides a convenient and interactive mode for executing Python commands and programs.

Jupyter: A **web application** that allows to run IPython in the browser.

Spyder: An **integrated development environment (IDE)**. A computer application that includes IPython, a text editor for writing and debugging programs, and more.

PyPlot: A **module** that provides visualization tools.

NumPy: A standard **library** (collection of modules, data types, etc.) that provides numerical arrays and mathematical functions.

Anaconda: A Python **distribution**. A single download that conveniently packages all of the above and installs it on your computer.

Important to understand

Python is “just” a programming *language*!

We can execute Python code via different *interpreters*.

Python codes are just text files (**.py**) which can be written in any (text) editor.

In the lab classes we will use Spyder for convenience.

For the lectures I will use Jupyter and some online interpreters.

➔ Switching IPython Notebook 01c-python_intro.ipynb