Introduction to (Scientific) Programming with Python

Jelka Stojanov, Timo Flesch

MT 2021

University of Oxford

Outline

- Why Python?

- Practical
  - Part 1: Intro to Python
  - Part 2: Python for scientific programming
  - Part 3: Python for statistical data analysis

- Additional resources

Why Python?

It’s a general-purpose programming language, used to power many of the apps you use every day!

Why Python?

There are free (!) add-ons for everything:

Create your own experiments!

PsychoPy3

Organise and analyse numerical data!

Pandas

NumPy

SciPy

Make beautiful figures!

matplotlib

seaborn

Perform sophisticated statistical analyses!

scikit learn

PyMVPA

PyMC3

SM StatsModels Statistics in Python

Hop on the machine learning hype train!

TensorFlow

PyTorch

Why Python?

It's just better than its competitors!

> MATLAB

Matlab is very expensive! Python is **free**. When using the right packages/add-ons, Python has the same functionality and syntax as Matlab.

> C++

Python is much easier to learn, as the **syntax is less complex**.
At least for science, there is a **better community/ecosystem**.
Some Python packages run C(++) code in the background, which makes them almost as fast as their C++ counterparts.

> R

Python is a **general-purpose language**. It can be used for almost everything. R was developed for statistical analyses.

> Java

Again, much **simpler syntax**. You can achieve the same results with fewer lines of code.

Python Practical

1. Go to http://www.github.com/timoflesch/intro2python

A brief (2h) introduction to programming in Python for absolute beginners, 2021 Edition

v1.0 (2019): The course was initially developed and delivered by Timo Flesch and Mirta Stantić.

v2.0 (2020): The course was refined and delivered by Timo Flesch and Jelka Stojanov.

v3.0 (2020): The course was refined and delivered by Timo Flesch and Jelka Stojanov.

A brief intro to Python, a powerful general-purpose programming language, designed for second-year undergraduate students in Experimental Psychology and Biomedical Sciences at the University of Oxford.

In this course, you will learn:

- how to assign values to variables and perform different mathematical operations;
- how to manipulate data formats and data sets;
- how to use loops, functions and methods (you will not only use existing functions in Python, but also create your own).

We will also introduce you to some of the most important libraries in the scientist's toolkit.

Finally, you will apply the knowledge you have gained to run a few simple statistical analyses (repeated-measures ANOVA, linear regression) and to visualize the results.

This course does **NOT** assume any previous knowledge of Python or any other programming language. Resources for those who are interested in learning more will be provided in the interactive worksheet in class.

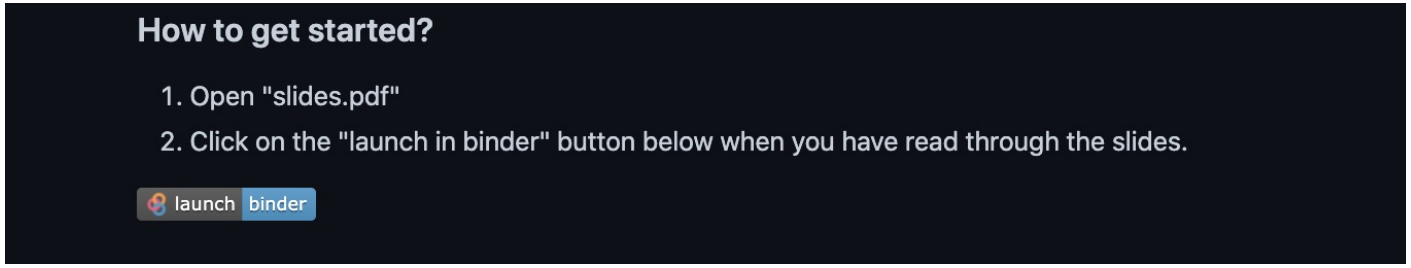
How to get started?

1. Open "slides.pdf"
2. Click on the "launch in binder" button below when you have read through the slides.

launch binder

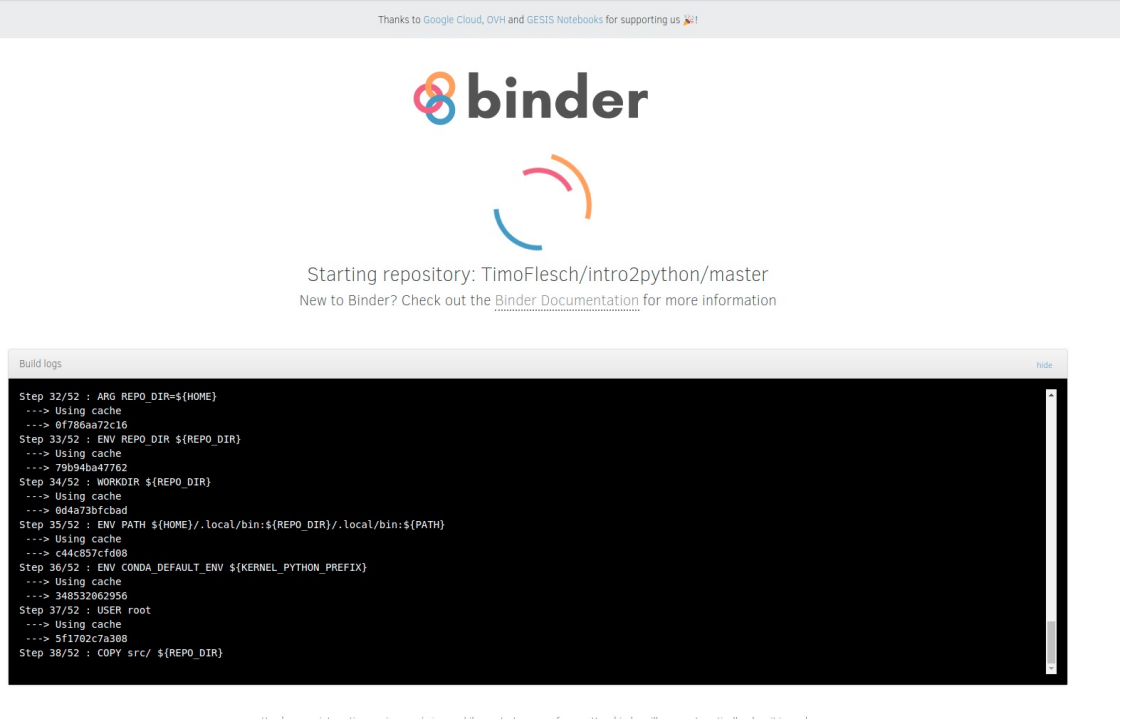
Python Practical

2. Click on the “launch binder” button:



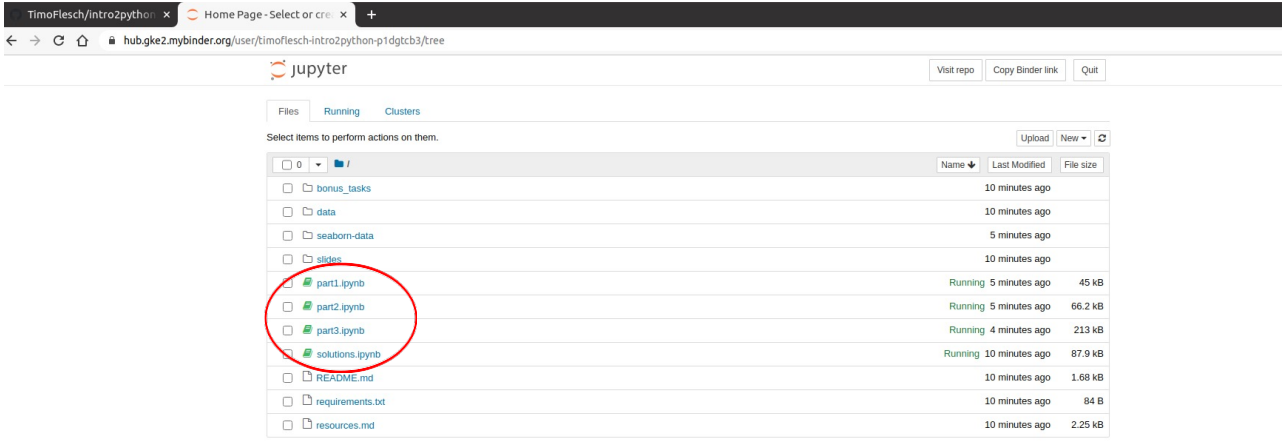
Python Practical

3. Stare at this screen for a while:



Python Practical

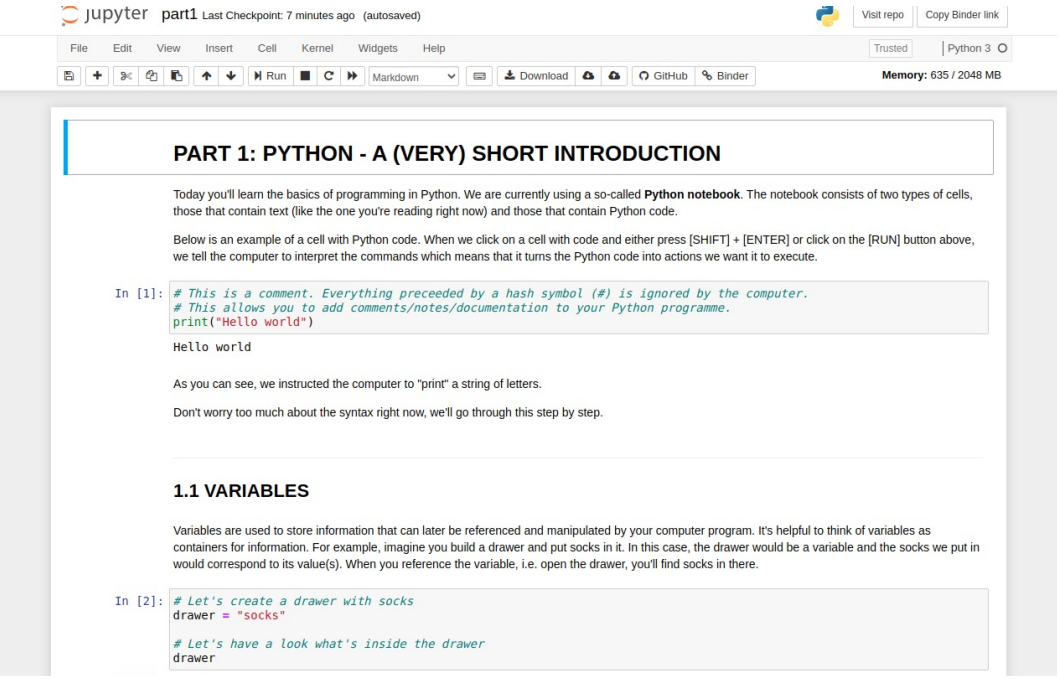
4. Now this screen (or similar) should show up:



From here you can open the worksheets that we have provided for you. This will run an interactive Python session in your browser, where you can write and execute your own Python programs.

Python Practical

5. You are now running an interactive worksheet with Python code in your web-browser. **Have fun** ☺



Outlook

Learn to code!

1. A complete course https://www.learnpython.org
2. Ditto https://www.w3schools.com/python/
3. Advanced course https://automatetheboringstuff.com

How to install Python?

1. Just Python https://www.python.org/downloads/
2. Jupyter Notebook (that you have used/will use today) https://jupyter.org/install
3. Anaconda (a collection of useful packages and other software for data scientists)

https://www.anaconda.com/distribution/

Outlook

Text editors

1. Atom https://atom.io/
2. Visual Studio Code https://code.visualstudio.com/
3. Sublime https://www.sublimetext.com/

All-in-one solutions (similar to the Matlab interface or R-Studio)

1. Spyder (free) https://www.spyder-ide.org/
2. Pycharm (free basic and commercial pro version) https://www.jetbrains.com/pycharm/

Outlook

Coding challenges

1. Hackerrank https://www.hackerrank.com/
2. Leetcode https://leetcode.com

Python for psychologists

https://www.marsja.se/best-python-libraries-psychology/