

Introduction to (Scientific) Programming with Python

Mirta Stantic, Timo Flesch

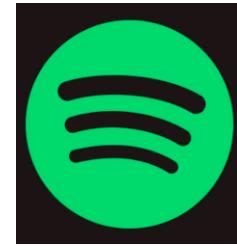
MT 2019

Outline

- Why Python?
- Practical
 - Part1: A quick intro to programming (in python)
 - Part2: Python for scientific programming
- Additional Resources

Why Python?

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



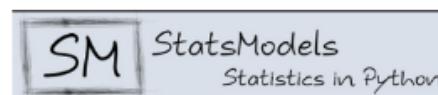
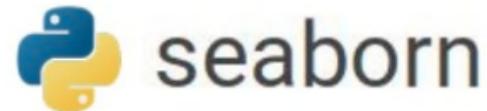
Why Python?

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

Create your own experiments!



Make beautiful figures!

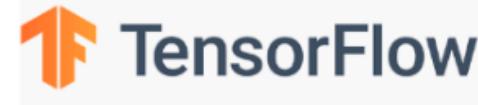


Organise and analyse numerical data



Perform sophisticated statistical analyses

Hop on the machine learning hype train!



Why Python?

It's just better than its competitors!



>



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



>



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



>



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



>



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>

The screenshot shows the GitHub repository page for 'TimoFlesch / intro2python'. The repository has 27 commits, 1 branch, 0 packages, 0 releases, and 1 contributor. The latest commit was 32 seconds ago by TimoFlesch. The repository contains files like README.md, .wakatime-project, and various Python scripts and notebooks. The README.md file is open, showing a brief introduction to Python programming for absolute beginners.

A brief intro to python, designed for 2nd year students in Experimental Psychology

27 commits 1 branch 0 packages 0 releases 1 contributor

Branch: master New pull request Find file Close or download

TimeFlesch tidied up Latest commit db1f2c1 32 seconds ago

File	Description	Time Ago
html	finalised material	16 hours ago
.wakatime-project	tidied up	32 seconds ago
README.md	tidied up	32 seconds ago
intro_to_python_students.ipynb	finalised material	16 hours ago
intro_to_python_students_SOLUTIONS.ipynb	finalised material	16 hours ago
slides.odp	tidied up	32 seconds ago

README.md

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

The course was developed and delivered by Mirta Stantic and Timo Flesch.

A brief intro to Python, designed for 2nd year undergraduate students in Experimental Psychology and Biomedical Sciences at the University of Oxford.

This course will introduce you to Python programming and will assume no previous knowledge of other programming languages. We'll skip the basics of setting up the programming environment to teach you how to assign values to variables, manipulate data formats and data sets, use basic scientific visualizations and basic statistics. We'll introduce you to loops, conditionals and functions and discuss some of the most important libraries in the scientist's toolkit.

This class might be of particular interest to those who are interested in doing computational work in the future – whether academic or in industry. Python is the most widely used programming language amongst data scientists in the industry but also an incredibly powerful tool to have at your disposal should you choose to continue your academic career.

No prerequisites expected. Resources for those who are interested in learning more will be provided in the interactive worksheet in class.

How to get started

Click on the "launch in binder" button below. This will open a new website where you'll find further instructions.

Search code

Python Practical

2. Click on “launch binder”

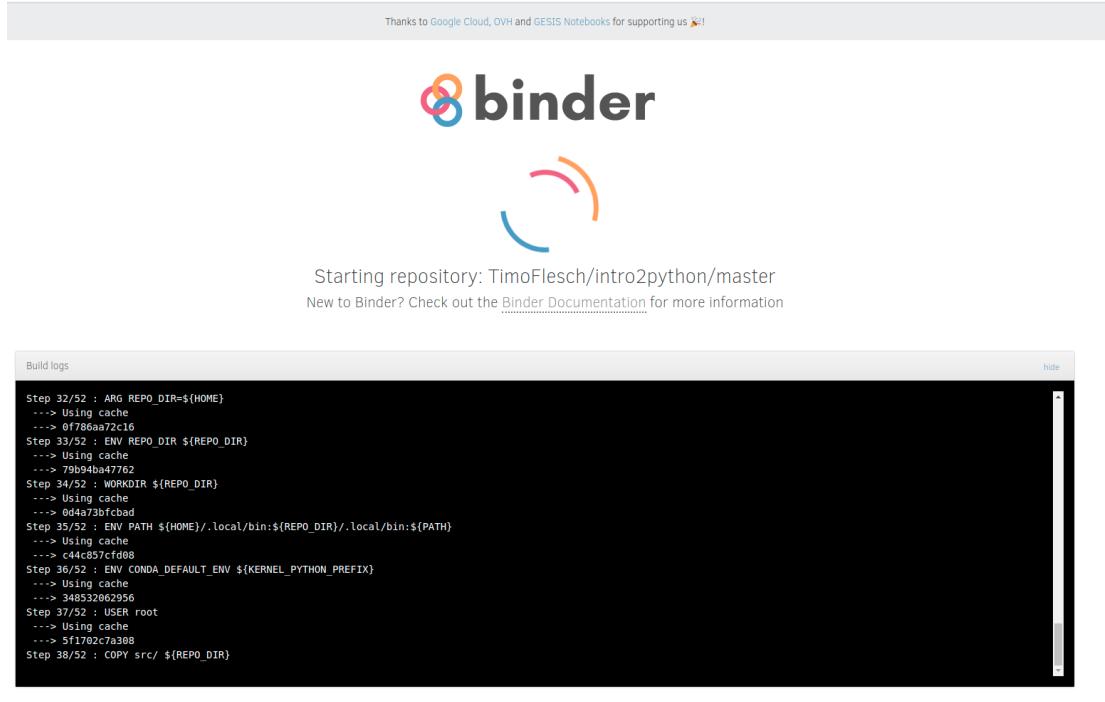
How to get started

Click on the "launch in binder" button below. This will open a new website where you'll find further information.



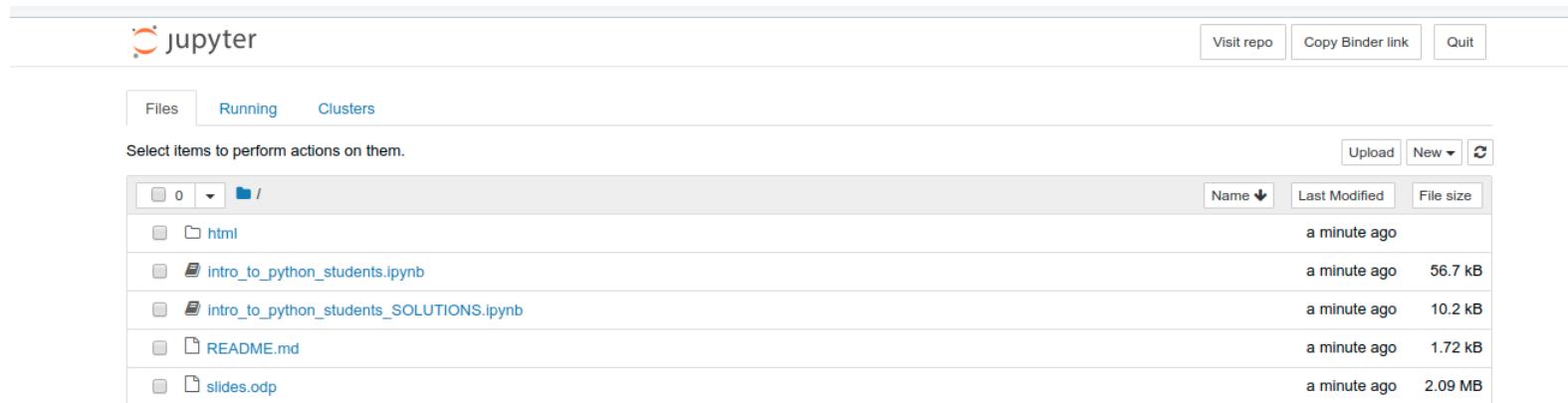
Python Practical

3. Stare for a while at this screen:



Python Practical

4. Now this screen should show up

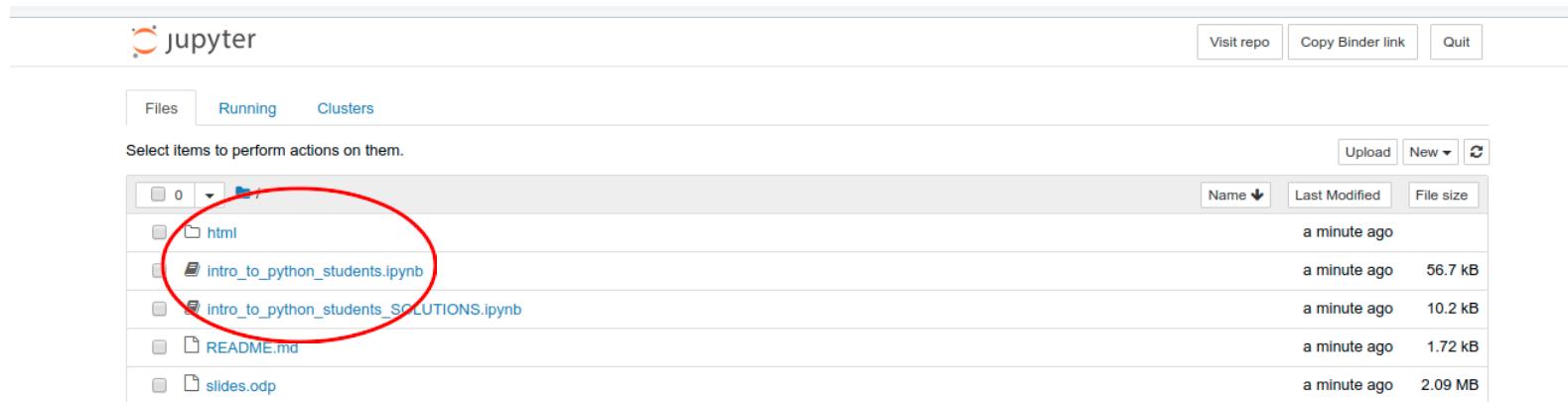


From here you can open the worksheet that we have provided for you.

This will run an interactive python session in your webbrowser, where you can write and execute your own python programs

Python Practical

4. Now this screen should show up



From here you can open the worksheet that we have provided for you.

This will run an interactive python session in your webbrowser, where you can write and execute your own python programs

Click on “intro_to_python_students.ipynb”

Python Practical

5. You're now running an interactive worksheet with python code in your web-browser.
Have fun :-)

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** imoflesch-intro2python-u9uixe02/notebooks/intro_to_python_students.ipynb
- Toolbar:** jupyter intro_to_python_students (unsaved changes) with icons for File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and various execution and cell manipulation buttons.
- Header Buttons:** Visit repo, Copy Binder link, Not Trusted, Python 3
- Section Header:** PART 1: PYTHON - A (VERY) SHORT INTRODUCTION
- Text Content:** Today you'll learn the basics of programming in Python. We use a so-called *Python notebook*. The notebook consists of two types of cells, those that contain text, like the one you're reading right now, and those that contain python code. Below is an example of a cell with Python code. When we click on a cell with code and either press [SHIFT] + [ENTER] or click on the (RUN) button above, we tell the computer to interpret the commands which means that it turns the Python code into actions we want it to execute.
- Code Cell:** In []:

```
# this is a comment. Everything preceded by a hash symbol (#) is ignored by the
# computer. This allows you to add comments/notes/documentation to your python programme.
print("hello world")
```
- Text Content:** As you can see, we instructed the computer to "print" a string of letters. Don't worry too much about the syntax right now, later we'll go through this step by step!
- Section Header:** 1.1 VARIABLES
- Text Content:** Variables are used to store information that can later be referenced and manipulated by your computer program. It's helpful to think of variables as containers for information. For example, imagine you build a drawer and put socks in it. In this case, the drawer would be a variable and the socks we put in would correspond to its value(s). When you reference the variable, i.e. open the drawer, you'll find socks in there.
- Code Cell:** In []:

```
# let's create a drawer with socks
drawer = "socks"

# let's have a look what's inside the drawer
```
- Section Header:** 1.1.1 Assign Values to Variables
- Text Content:** Variables would be useless if you couldn't assign values to them. Python is able to handle all sorts of different datatypes, such as numbers, strings (=sequences of letters and symbols) and lists of numbers/letters. Below are a few examples
- Code Cell:** In []:

```
# this is an integer (a number without any decimal points)
variable1 = 5
```

Outlook

Learn coding!

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.codecademy.com/articles/install-python>
2. The Jupyter notebook (the thing you've been working with 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 editor <https://atom.io/>
2. vscode <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/>