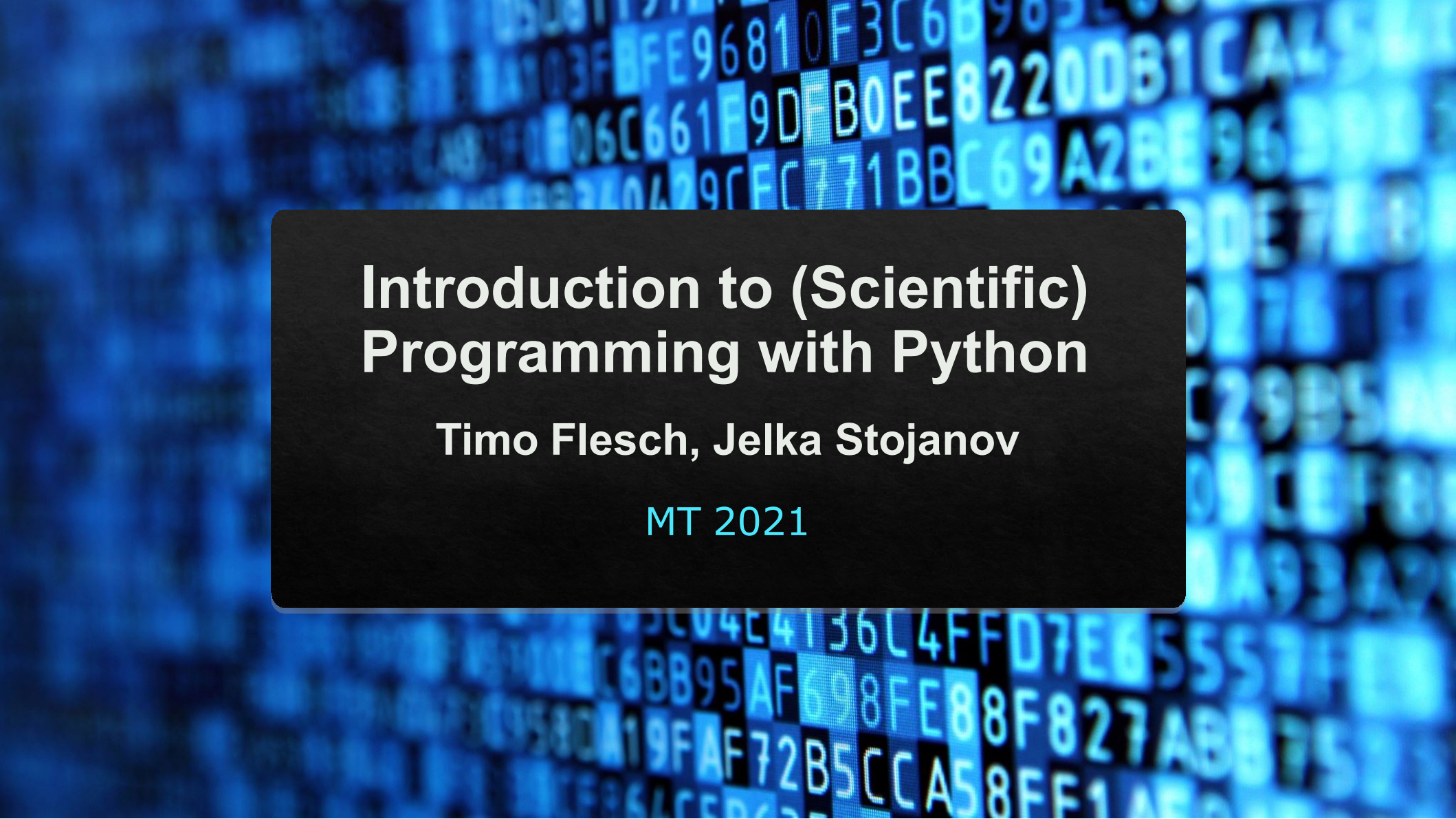


Introduction to (Scientific) Programming with Python

Timo Flesch, Jelka Stojanov

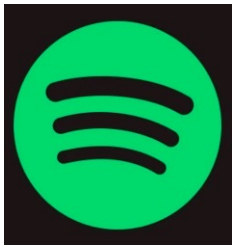
MT 2021

Outline

- Why Python?
- Practical
  - Part 1: A quick intro to programming (in Python)
  - Part 2: Python for scientific programming
  - Part 3: Hands-on data visualisation and analysis
- Additional resources

Why Python?

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



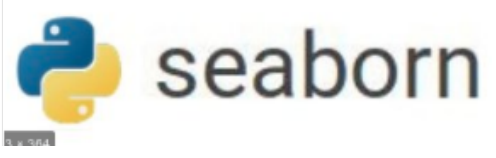
Why Python?

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

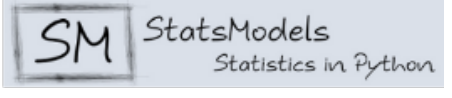
Create your own experiments!

Organise and analyse numerical data!

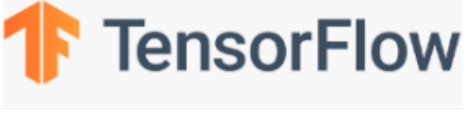
Make beautiful figures!



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/add-ons, Python has the same functionality and syntax as Matlab.

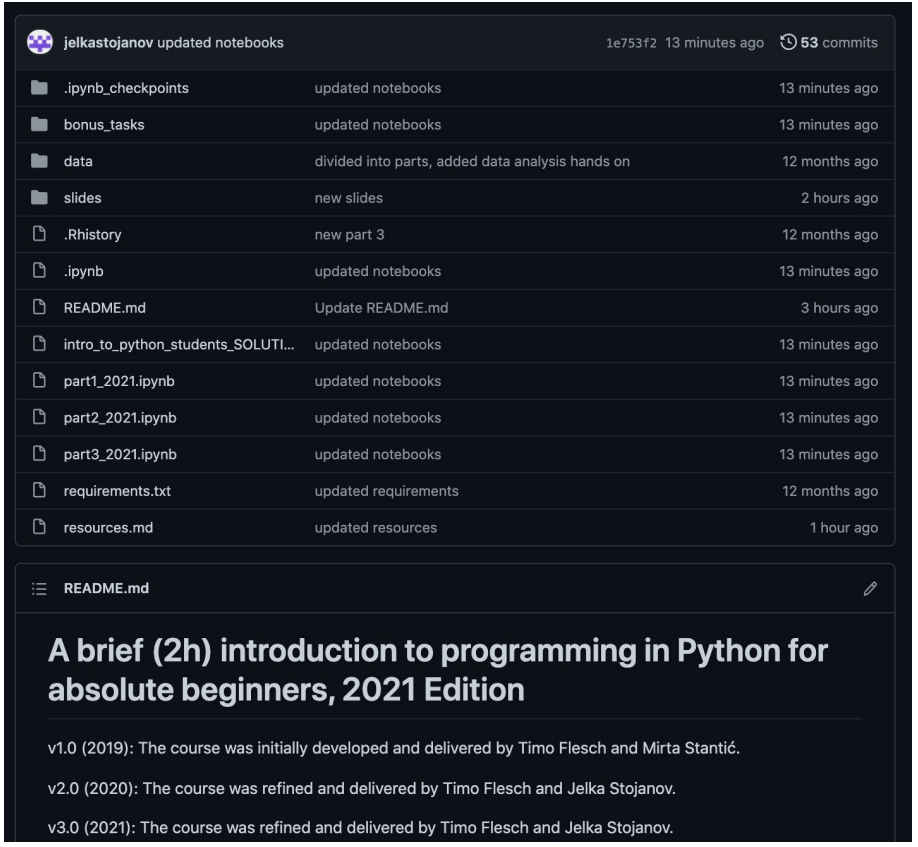
Python is much easier to learn, as the syntax is less complex.
At least for science, there is a better community/ecosystem.
Some Python packages run C(++) code in the background, which makes them almost as fast as their C++ counterparts.

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.

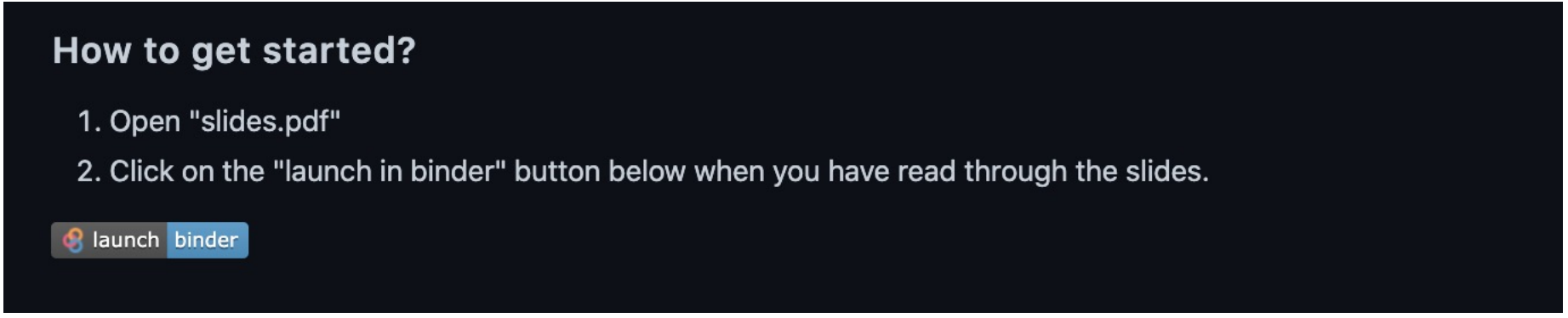
Python Practical

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



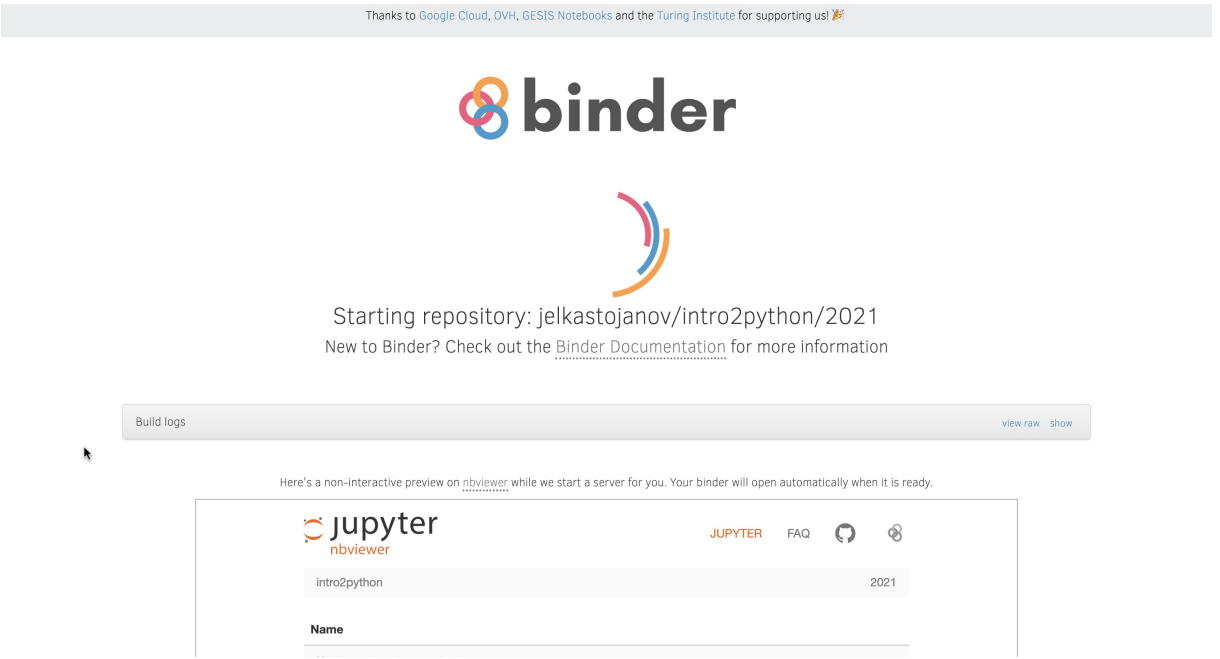
Python Practical

2. Click on the “launch binder” button:



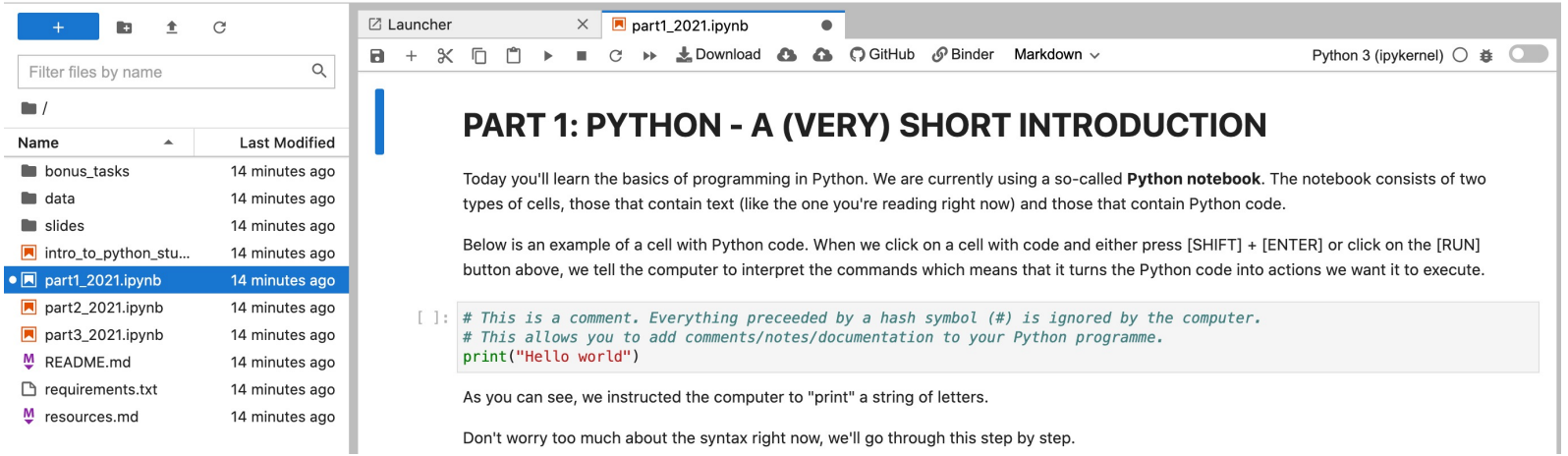
Python Practical

3. Stare at this screen for a while:



Python Practical

4. This screen should show up:



From here you can open the worksheets 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.

Click on “part1_2021.ipynb”

Python Practical

5. You are now running an interactive worksheet with Python code in your browser. Have fun :)

PART 1: PYTHON - A (VERY) SHORT INTRODUCTION

Today you'll learn the basics of programming in Python. We are currently using 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.

[ ]:
```
# This is a comment. Everything preceeded by a hash symbol (#) is ignored by the computer.
# This allows you to add comments/notes/documentation to your Python programme.
print("Hello world")
```

As you can see, we instructed the computer to "print" a string of letters.

Don't worry too much about the syntax right now, we'll go through this step by step.

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. The Jupyter notebook (the thing you are 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. 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/