

# PYTHON FOR GEOSCIENTISTS

(short intro to python)



**NAWI Graz**  
Natural Sciences

Lecturers:

Raoul Collenteur  
Matevž Vremec



# Lecture overview

- Introducing the lecturers
- Why learn to code?
- Why Python?
- Course overview
  - Motivation
  - Structure
  - Materials & Environment
  - Deadlines & Grading
  - Communication

# THE LECTURERS



## Raoul Collenteur

PhD Candidate in Hydrogeology

Previously consulting hydrologist with lots of Python programming to solve real-life problems.

Core developer of **Pastas**, a Python package to analyze hydrogeological time series

<https://doi.org/10.1111/gwat.12925>



## Matevž Vremec

PhD Candidate in Hydrogeology

Python Projects:

- **phydrus** – implementation of HYDRUS-1D
- **pyet** – estimating evaporation

Using Python for:

- data analysis
- hydrological modeling
- uncertainty analysis



# Course learning objective

The goal of this course is to learn to solve real-life geoscience problems through (Python) Programming

# WHY LEARN PYTHON PROGRAMMING?

- The students will separate into 3 or 4 breakout groups.
- Discuss among each other the following questions:
  1. Do you have any experience in coding?
  2. Why do think you need to learn to code?

Report back to the entire group after your discussion

# WHY LEARN PYTHON PROGRAMMING?

The goal of this course is to learn to solve real-life geoscience problems through Python Programming

- Automate repetitive tasks through scripts (save time)
- Analyse and visualize (large) datasets (be smarter)
- Use scripts that ensure your work is reproducible (be scientific)
- Get ready for the 21st Century Job market! (get hired)

# Why Python?

- Widely used in science and engineering
- Simple to read and easy to learn
- Free and open source
- Powerful: useful for large projects, big data, high performance computing, etc.
- Check job adverts and see what employers are looking for...
- Cross-platform (Windows, Mac or Linux)
- (BUT many good alternatives: R, Matlab, Julia ...)

Rank	Change	Language	Share	Trend
1		Python	29.5 %	-1.0 %
2		Java	17.51 %	-0.6 %
3		JavaScript	8.19 %	+0.2 %
4		C#	7.05 %	-0.2 %
5	↑	C/C++	6.73 %	+1.0 %
6	↓	PHP	6.23 %	+0.0 %
7		R	3.86 %	+0.0 %
8		Objective-C	2.77 %	+0.3 %
9	↑	TypeScript	1.87 %	-0.0 %
10	↓	Swift	1.85 %	-0.3 %
11	↑	Kotlin	1.78 %	+0.3 %
12	↓	Matlab	1.77 %	-0.1 %
13	↑	Go	1.37 %	+0.1 %
14	↓	VBA	1.33 %	-0.0 %
15		Ruby	1.21 %	-0.1 %

The PYPL Popularity of Programming Language Index is created by analyzing how often language tutorials are searched on Google.

# HISTORY OF PYTHON

"Invented" by Guido van Rossum at the Centre for Mathematics and Informatics in Amsterdam, The Netherlands, in 1991.



```
# Python 3: Simple output (with Unicode)
>>> print("Hello, I'm Python!")
Hello, I'm Python!

# Input, assignment
>>> name = input('What is your name?\n')
>>> print('Hi, %s.' % name)
What is your name?
Python
Hi, Python.
```



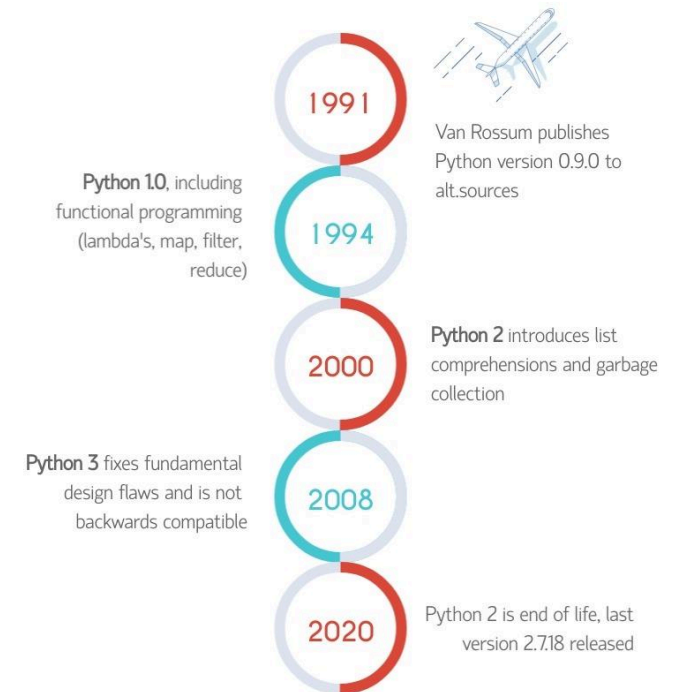
## Quick & Easy to Learn

Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. [Whet your appetite](#) with our Python 3 overview.



<https://www.python.org>

## Python History





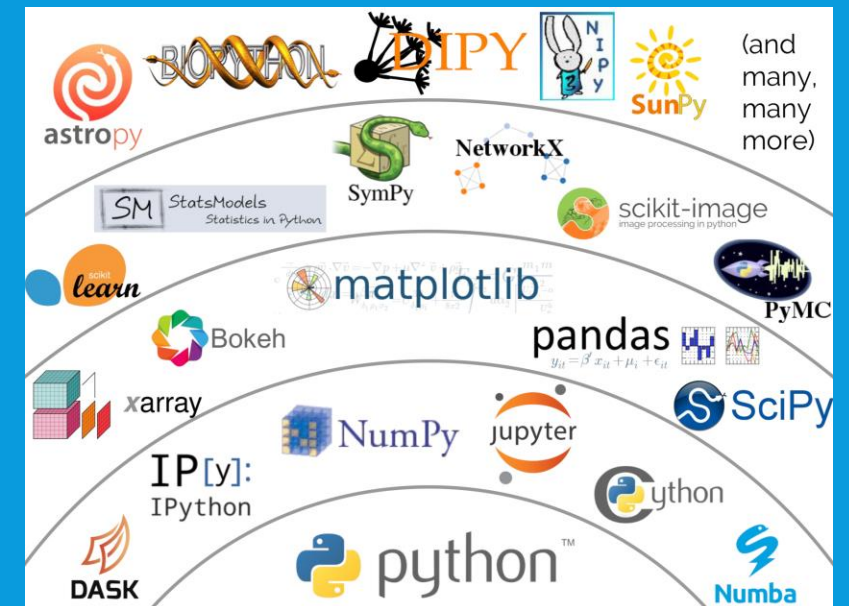
# THE PYTHON ECOSYSTEM

A big advantage of Python is the massive ecosystem, ready-made packages exist for many tasks.

We will mostly learn about packages from the Scipy Stack:

- Numpy: Deals with large N-dimensional arrays
- Pandas: Data structure and analysis tools
- Matplotlib: Produces plots and figure
- Scipy: Python tools for scientists

More info: <https://scipy.org/index.html>



# PYTHON PACKAGES FOR GEOSCIENTISTS

- Python resources for geoscientists

<https://github.com/softwareunderground/awesome-open-geoscience>

- Python resources for hydrologists

[http://abouthydrology.blogspot.com/2016/11/python-resources-for-hydrologists.html? sm au =iVV0kRWqq5FFWV5r](http://abouthydrology.blogspot.com/2016/11/python-resources-for-hydrologists.html?sm_aui=ivV0kRWqq5FFWV5r)

<https://github.com/raoulcollenteur/Python-Hydrology-Tools>

# WHAT IS A SCRIPT?

- A script is a collection of written commands to the computer to perform a certain task. In Python the text-files with computer instructions have the ".py" extension.
- Scripts make it easy to perform repetitive tasks.
- Scripts provide a documentation of the entire workflow / analysis.

## script1.py

```
1  """
2  $UWHPSC/codes/python/script1.py
3
4  Sample script to print values of a function at a few points.
5  """
6  import numpy as np
7
8  def f(x):
9      """
10     A quadratic function.
11     """
12     y = x**2 + 1.
13     return y
14
15  print "      x      f(x)"
16  for x in np.linspace(0,4,3):
17      print "%8.3f  %8.3f" % (x, f(x))
```

# THE REPRODUCIBILITY CRISIS

**"non-reproducible single occurrences are of no significance to science.** Thus a few stray basic statements contradicting a theory will hardly induce us to reject it as falsified. We shall take it as falsified only if we discover a reproducible effect which refutes the theory"

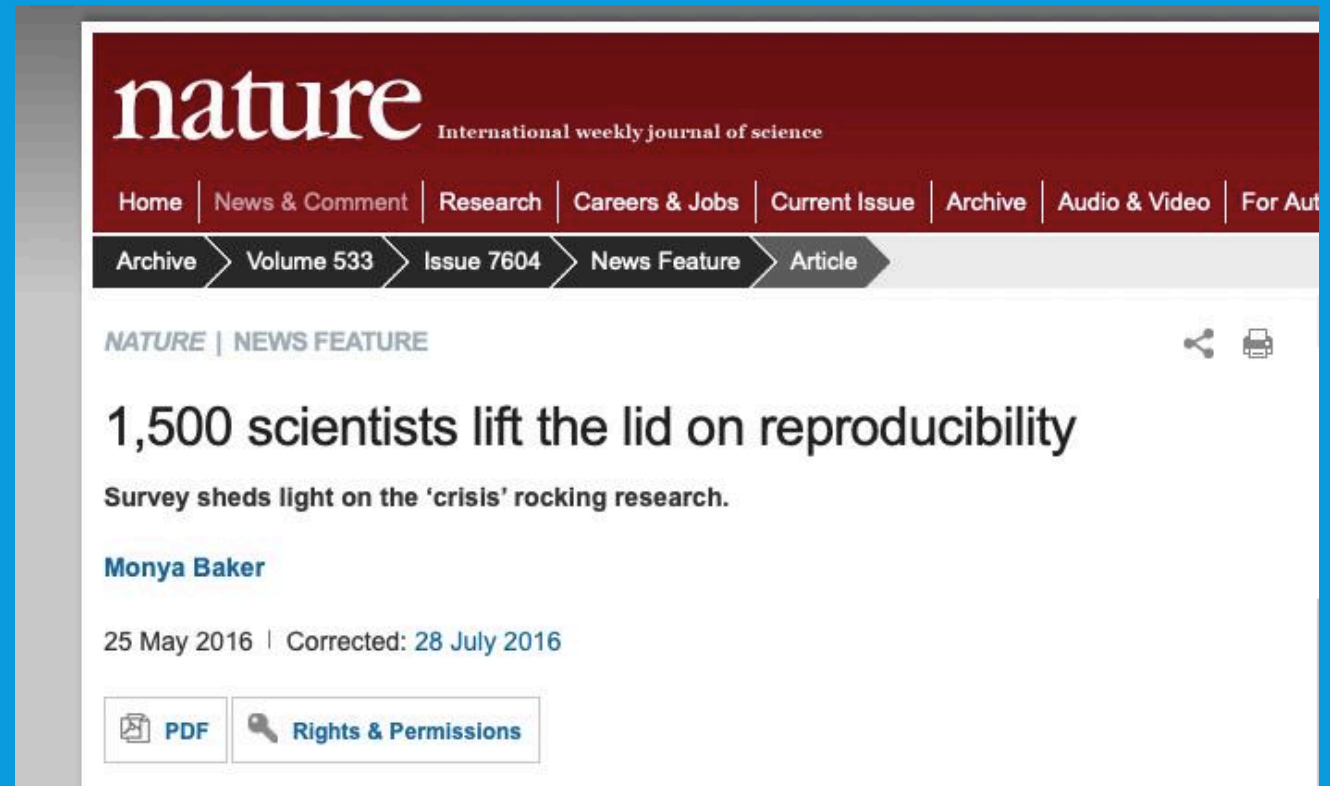
Popper, K. (1959) The Logic of Scientific Discovery

# IS THERE A REPRODUCIBILITY PROBLEM?

Probably yes. A recent Nature study found that:

“More than 70% of researchers have tried and failed to reproduce another scientist's experiments, and more than half have failed to reproduce their own experiments.”

It is up to scientists (you!) to make their work reproducible. Scripts can be really useful for this purpose!



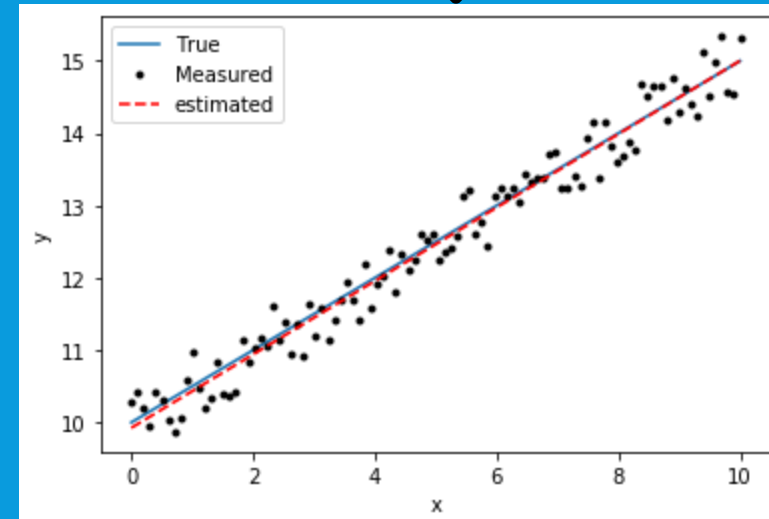
# Course overview

1. Programming basics
2. Math and arrays
3. For-loops
4. Python functions
5. Data Analysis
6. Curve Fitting
7. Recap

```
x = 2  
y = x / -2 + x ** 2  
print(y)
```



```
x = np.arange(0, 11, 1)  
for n in x:  
    print(np.sqrt(n))
```



# Online course format

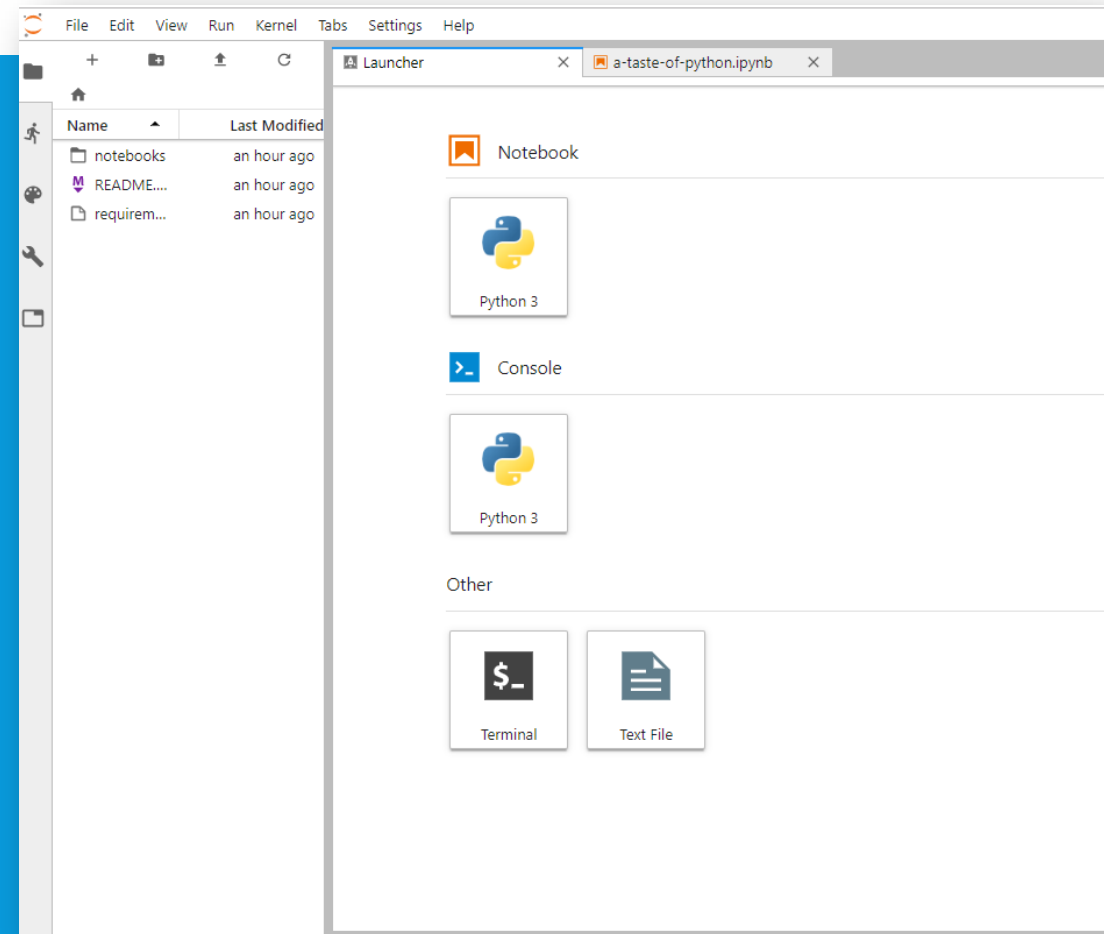
- **For each of the six lectures we provide:**
  - A 15-30 minute video explanation of the topic
  - A jupyter notebook with text and exercises
- It is recommended to first watch the lecture and then go through the notebook.
- Bottomline: if you want to learn to code, you need to code!

## Communication:

- Moodle forum
- Unimeet chat: Tuesdays from 10:45 to 11:15
- Additional resources (background reading) will be provided on Moodle

# Materials & Environment

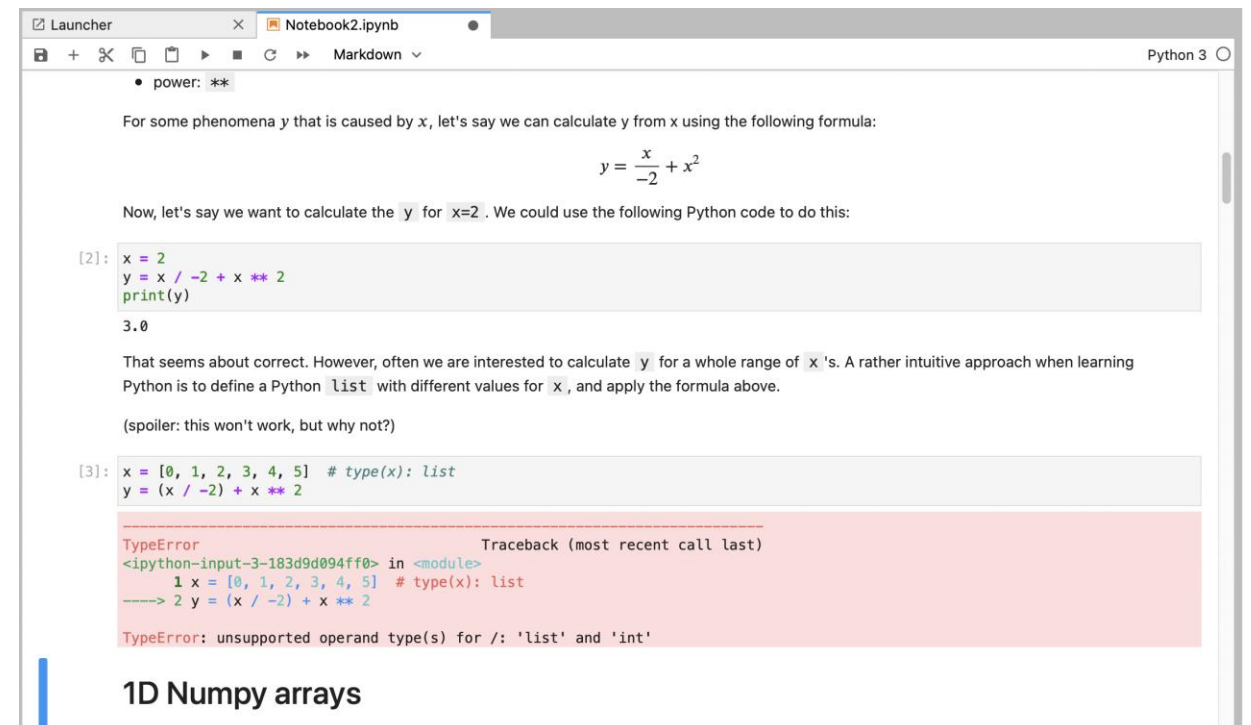
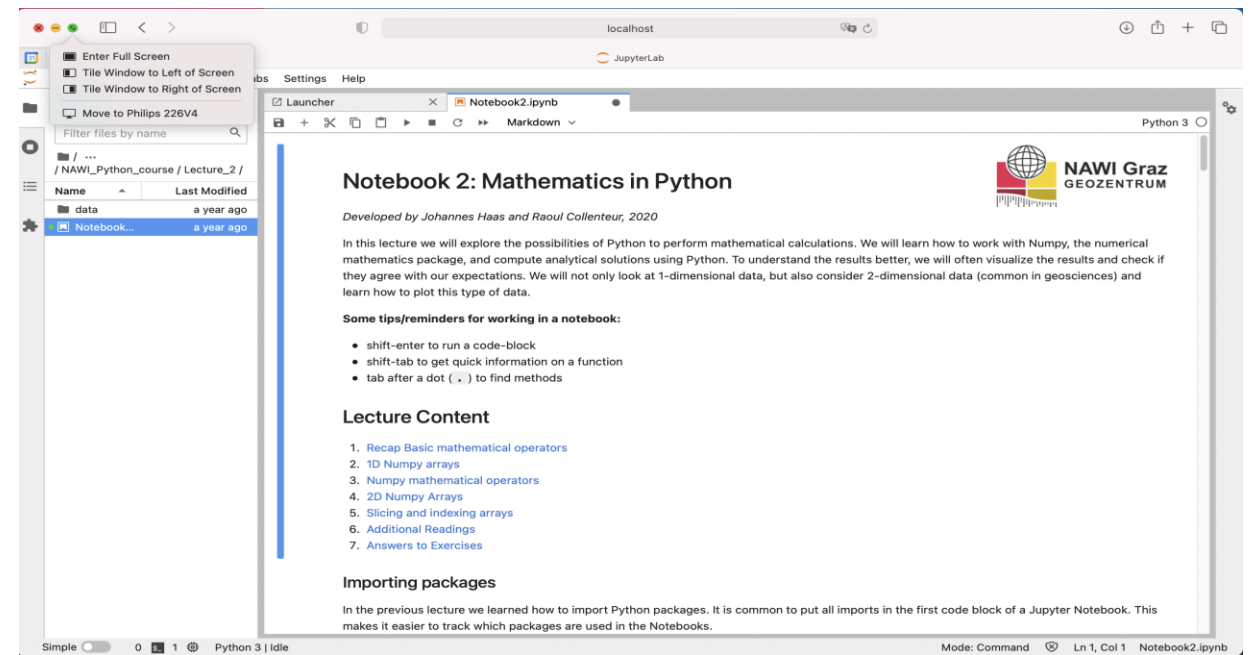
- Anaconda Python Distribution
  - Installation guide accessible on Moodle or:
    - <https://www.anaconda.com/products/individual>
- Jupyter Notebooks
  - documents inside the JupyterLab environment which contain computer code, and rich text elements (figures, links, etc.)





# JUPYTER NOTEBOOKS

- Runs in your browser
  - Combines Code and Text
  - Interactive
  - Shareable
- 
- Unfortunately showing this through screen-share won't work well in BBB..
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- <https://jupyterlab.readthedocs.io/en/stable/>



# Grading

1. Grading is based on three mandatory notebooks that need to be submitted to Moodle.
2. All of the above notebooks contain 4 to 5 exercises that are graded. Each exercise has an equal weight.
3. There is no final exam for this course!
4. Assignments will be made available after each lecture.
5. Grading will occur within 2 weeks. If incorrect, you get the possibility to improve within one week.

# Deadlines

- There will be no extensions of these deadlines. The materials are provided 2 weeks in advance and there is enough time to work on the exercise.

Group 1 (15:45-17:30)	Group 2 (10:00-11:45)	Type	Content	Attendance
Mon. 26.04.2021	Tue. 27.04.2021	BBB live.	Introduction	Obligatory*
Mon. 03.05.2021	Tue. 04.05.2021	NB1	Programming basics	
Mon. 10.05.2021	Tue. 11.05.2021	NB2	Math and arrays	
Mon. 17.05.2021	Tue. 18.05.2021	NB3	For-loops	
<b>Sun. 23.05.2021</b>				<b>Deadline Notebook 2</b>
Mon. 31.05.2021	Tue. 01.06.2021	NB4	Python functions	
Mon. 07.06.2021	Tue. 08.06.2021	NB5	Data Analysis	
<b>Sun. 13.06.2021</b>				<b>Deadline Notebook 4</b>
Mon. 14.06.2021	Tue. 15.06.2021	NB6	Curve Fitting	
Mon. 21.06.2021	Tue. 22.06.2021	BBB live.	Recap	
<b>Sun. 04.07.2021</b>				<b>Deadline Notebook 4</b>

# FINAL QUESTIONS?

- Did you manage to install Anaconda and start Jupyter Lab?

Group 1 (15:45-17:30)	Group 2 (10:00-11:45)	Type	Content	Attendance
Mon. 26.04.2021	Tue. 27.04.2021	BBB live.	Introduction	Obligatory*
Mon. 03.05.2021	Tue. 04.05.2021	NB1	Programming basics	
Mon. 10.05.2021	Tue. 11.05.2021	NB2	Math and arrays	
Mon. 17.05.2021	Tue. 18.05.2021	NB3	For-loops	
<b>Sun. 23.05.2021</b>				<b>Deadline Notebook 2</b>
Mon. 31.05.2021	Tue. 01.06.2021	NB4	Python functions	
Mon. 07.06.2021	Tue. 08.06.2021	NB5	Data Analysis	
<b>Sun. 13.06.2021</b>				<b>Deadline Notebook 4</b>
Mon. 14.06.2021	Tue. 15.06.2021	NB6	Curve Fitting	Obligatory*
Mon. 21.06.2021	Tue. 22.06.2021	BBB live.	Recap	
<b>Sun. 04.07.2021</b>				<b>Deadline Notebook 4</b>