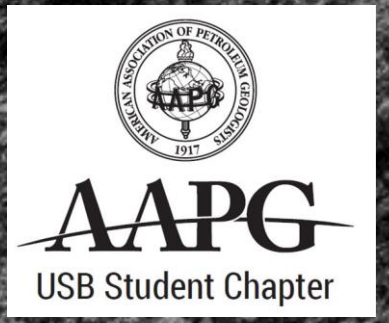


Introduction to Python for geosciences

M.Sc. Manuel David Soto

Instructor

Manuel **David** Soto. Geological Engineer (UCV, 1997). Master in Geological Sciences (UT. Austin, 2007). Marter in Big Data and Business Analytics (U. Complutense of Madrid, 2023).He worked as support and training engineer (Schlumberger-Geoquest, Caracas), as production geologist (Marathon, Houston), as operations and exploration geologist and petrophysicist (Repsol, Bogotá and Madrid). He has used Matlab and Python, since 2005 and 2019 respectively, for teaching and develop custom programs for petrophysic.

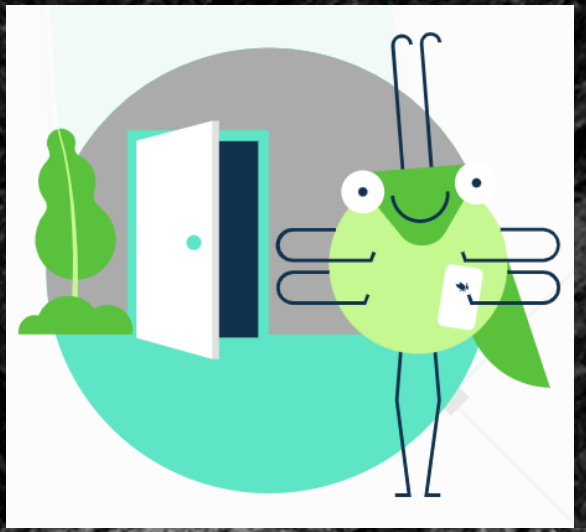
Agenda

- Coding or programming
- What is Python and why?
- Python library structure
- Python for geosciences
- Execution option
- Basic libraries
- Jupyter

Coding or programming

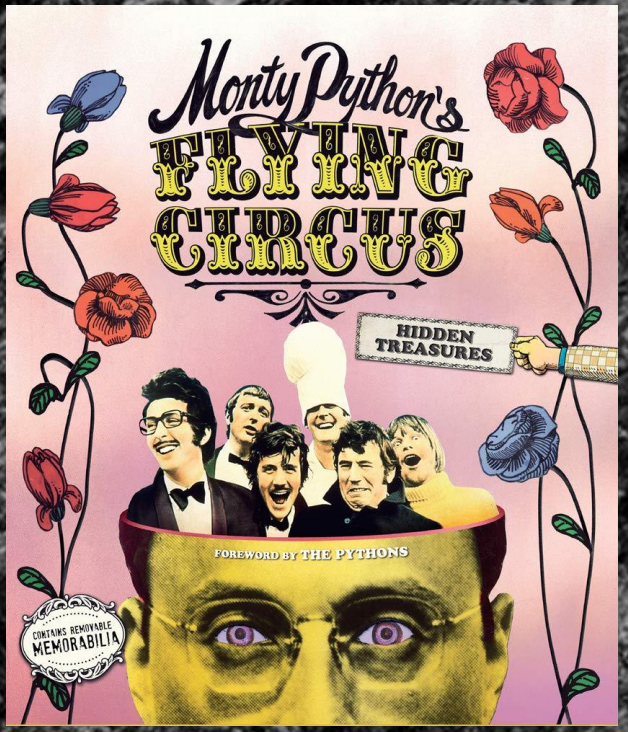
“Coding, sometimes called computer programming, is how we communicate with computers. Code tells a computer what actions to take, and writing code is like creating a set of instructions. By learning to write code, you can tell computers what to do or how to behave in a much faster way. You can use this skill to make websites and apps, process data, and do lots of other cool things.”

https://grasshopper.app/why-coding/



The set of instructions are established in what are known as programming languages which can be, Based on their use, specific-purpose (Java, C, …) or general-purpose languages (Python, Matlab, Basic, …).

What is Python and why?



"In December 1989, Guido Van Rossum had been looking for a hobby programming project that would keep [him] occupied during the week around Christmas as his office was closed when he decided to write an interpreter for a new scripting language [he] had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers. He attributes choosing the name Python to being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus)"

Photo: https://www.facesofopensource.com/guido-van-rossum/
Text: https://en.wikipedia.org/wiki/Guido_van_Rossum
Poster: https://www.amazon.ca/Monty-Pythons-Flying-Circus-Treasures/dp/1853759740

What is Python and why?

after 3:25 is advertising

- Python (Py) is an interpreted (no compilation), high-level (user friendly) programming language with dynamic semantics (typing and binding) , and with multiple programming language paradigm (imperative, functional, object-oriented).
- Py is very attractive for rapid development, as well as glue language to connect existing components together.
- Py is easy to learn and read therefore reduces the cost of maintenance.
- The Py interpreter (skull) and the extensive standard library (skeleton) are available for free for all major platforms (Windows, Mac, Linux, …).
- Py supports external libraries, modules or packages (muscles), which encourages program modularity and code reuse.

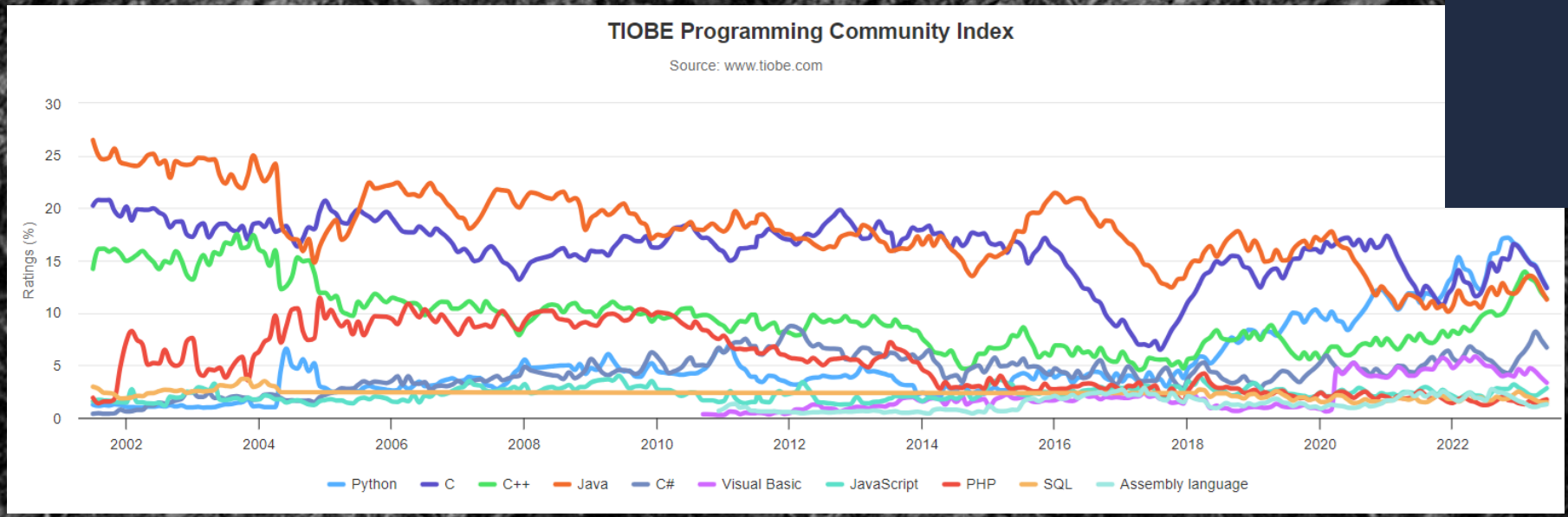
More on Py: https://www.python.org/doc/essays/blurb/

What is Python and why?

C, Java, JavaScript, HTML/CSS, and SQL are specific-purpose programming languages. Py is a general-purpose programming language

↓TIOBE is a company specialized in assessing and tracking the use and quality of software: https://www.tiobe.com/tiobe-index/

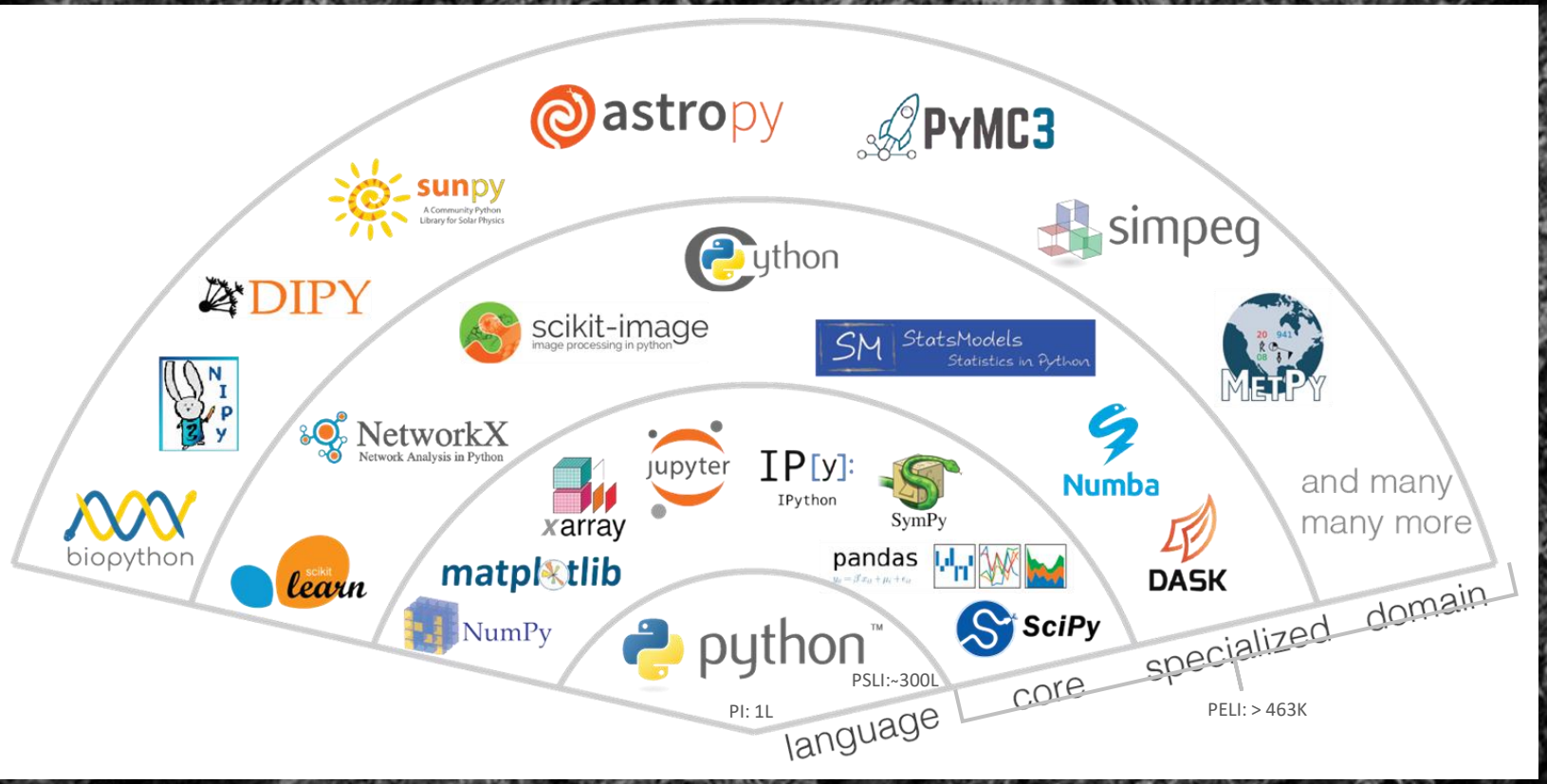
→ Stack Overflow is a question-and-answer website for professional and enthusiast programmers: https://survey.stackoverflow.co/2022/#most-popular-technologies-language-prof



1st

What is Python and why?

Ecosystem of open-source scientific libraries based on Py



Key aspects of the ecosystem:

- community
- a common environment
- interaction and interoperability
- reliance and interdependence

PI: Py Interpreter
PSLI: Py Standard Libraries
PELI: Py External Libraries

More on Jupyter: https://jupytearth.org/jupyter-resources/introduction/ecosystem.html

Python library structure I

- The Py Interpreter (PI) is the central library of Py (skull) which has basic functions (~70) always available:

  https://docs.python.org/3/library/functions.html

- Just as the bones support the skull, there are about 300 already-installed Py Standard Libraries (PSLI), available only when called them (import), and provide special functions in many areas support as systems, math, statistics, and other:

  https://docs.python.org/3/library/

  PI and PSLI are build and maintained by the Python Software Foundation

- In addition to these basic libraries (PI and PSLI) you can create your own libraries (YOL), composed of several functions (ideas or thoughts).

https://www.cgtrader.com

Python library structure II

- Finally, there is a huge group (> 463K) of Py External libraries (PELI). Like specialized muscles or organs, they cover many areas, from data visualization, machine learning, astronomy, geosciences … , each one with its own set of functions and sometime data type.

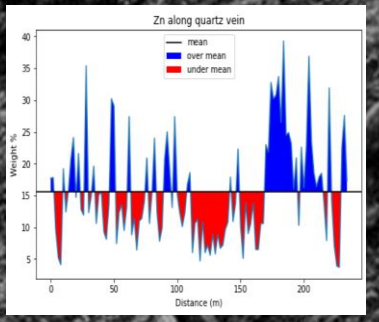
  As is indicated by the name PELI, they have to be installed (only once) and then called (imported) each time they are going to be used. They have been developed by individuals, groups, public and/or private institutions. The Python Software Foundation is in charge of organize and distribute them. The complete list of PELIs is at:

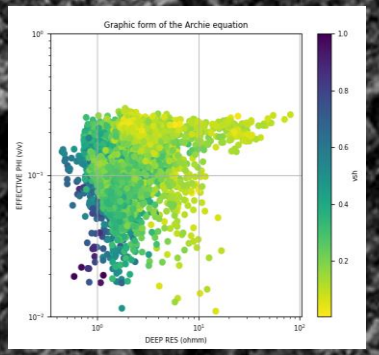
  https://pypi.org/

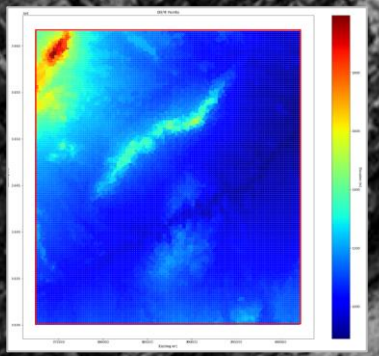
https://www.cgtrader.com

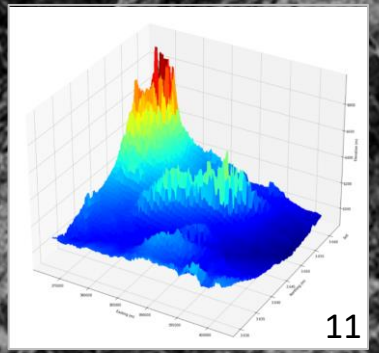
Python for geosciences

- Py provides an excellent platform for teaching, developing and testing new procedures, equations, and algorithms.
- Py has tools to deals with the increasing use of structured (spreadsheets, tables, matrixes) and unstructured data (text, image).
- Py allows loading, QC, manipulation, visualization, and analysis of data of a very diverse nature (1-3D), from simple compass measurements to seismic data.
- Py allows to automate repetitive tasks.
- Most important programs for geosciences incorporate utilities to code your own Py programs.
- Advance statistical and machine learning (ML) methods are gaining importance in geosciences. Py is the main gate for such advance applications.



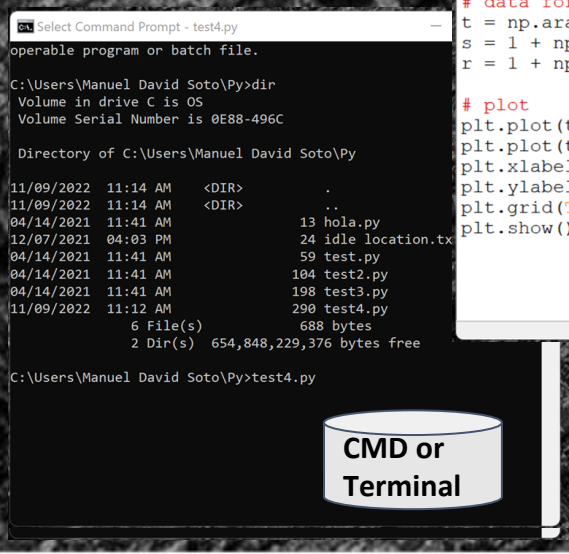


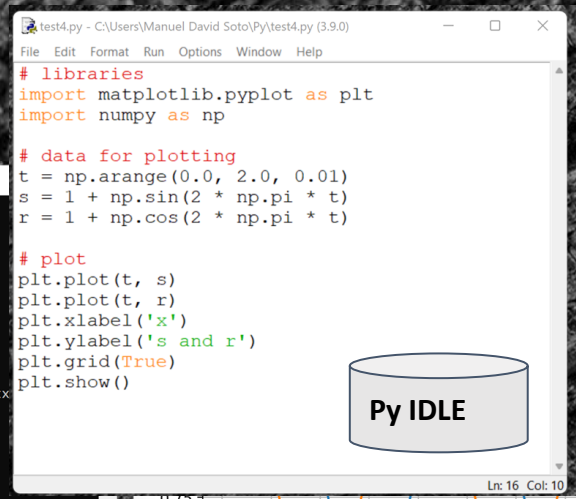


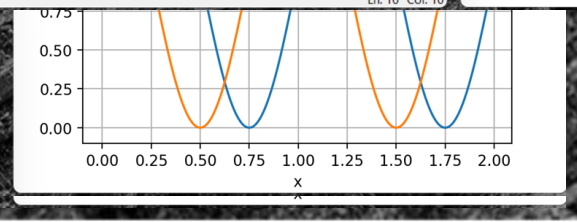


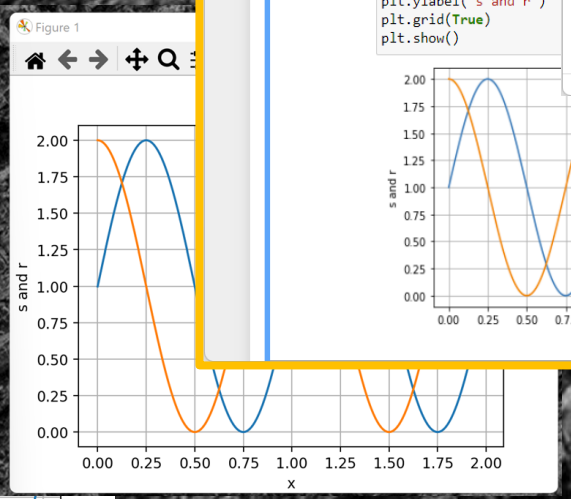
Python execution options

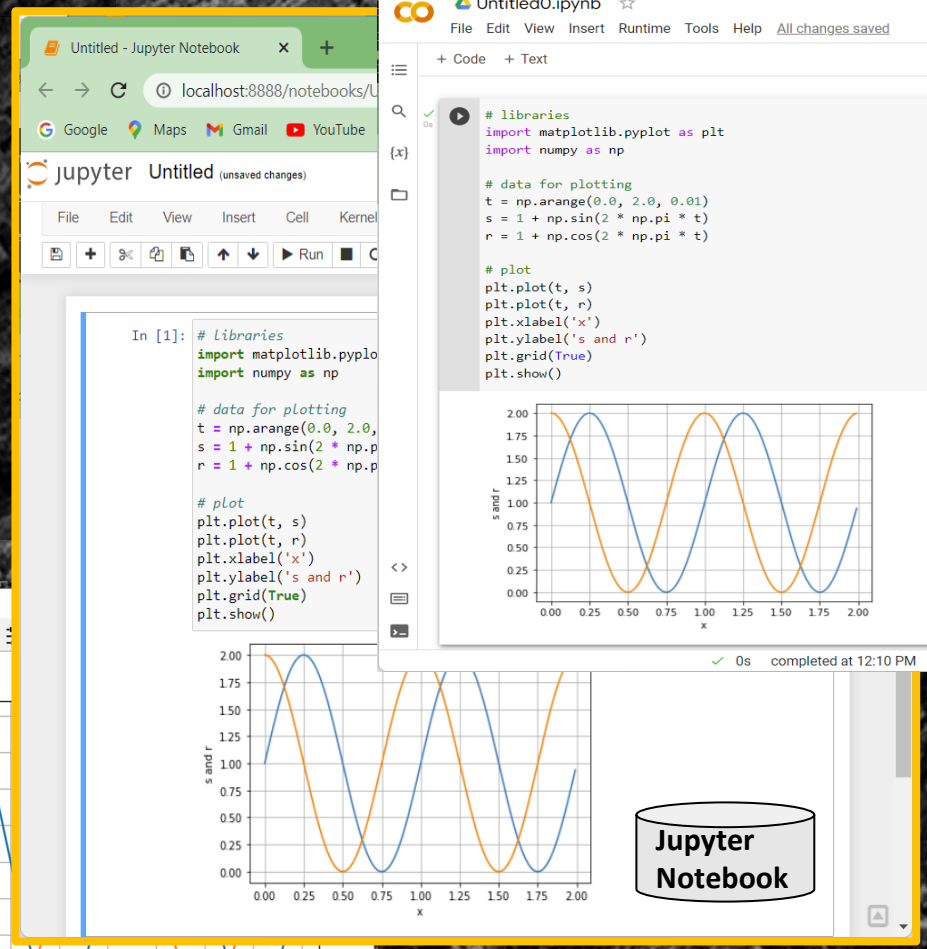
Different IDE (Integrated Development Environments) to work with Py, on different platforms, from the most simple and modest to the most complex and powerful. Here are four options, three running in your pc and on the cloud:

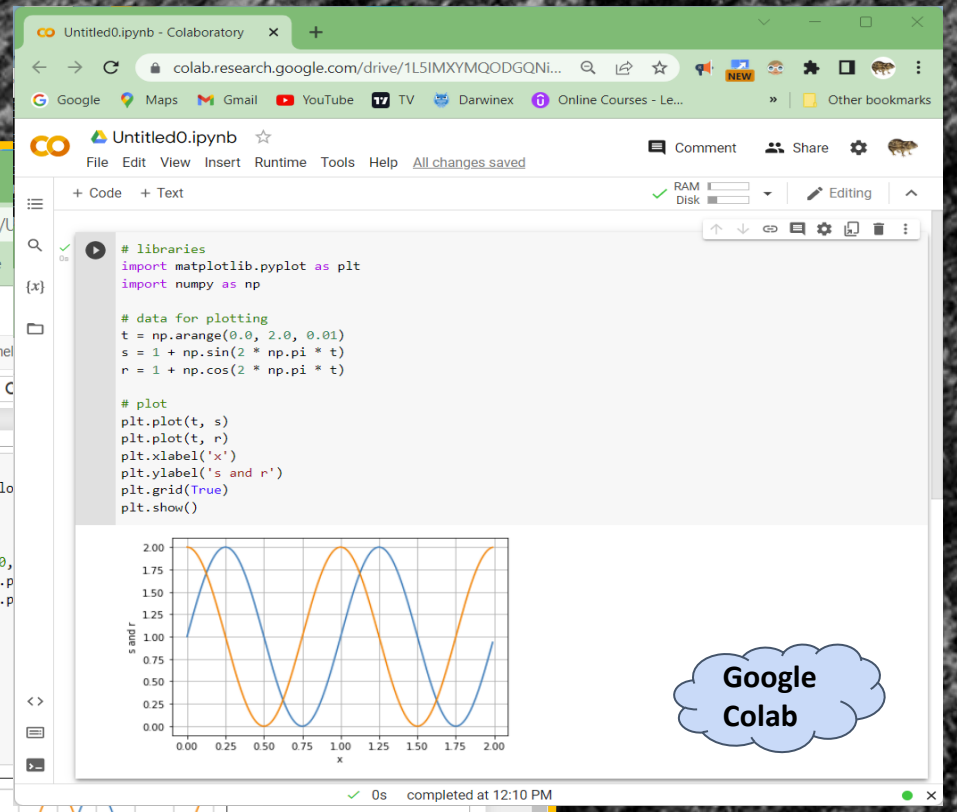


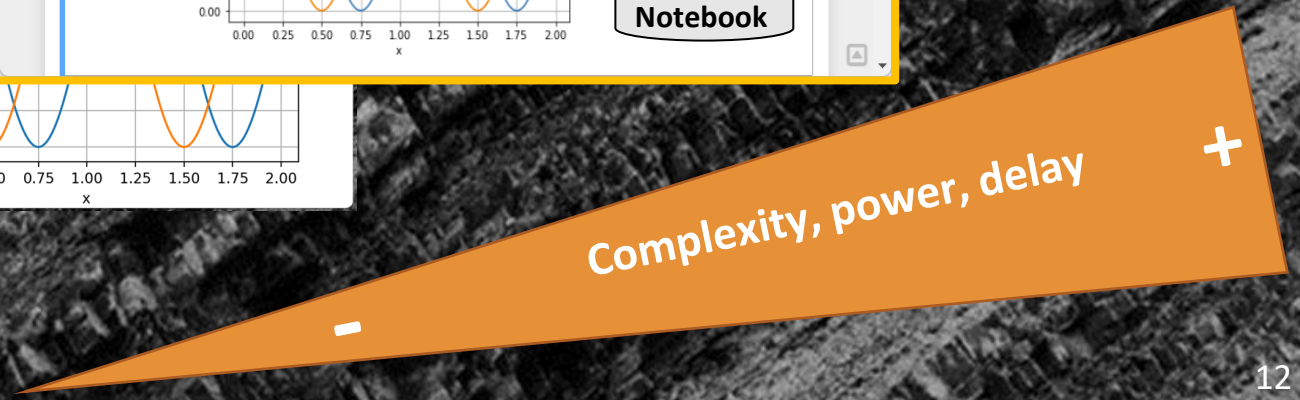












Python installation

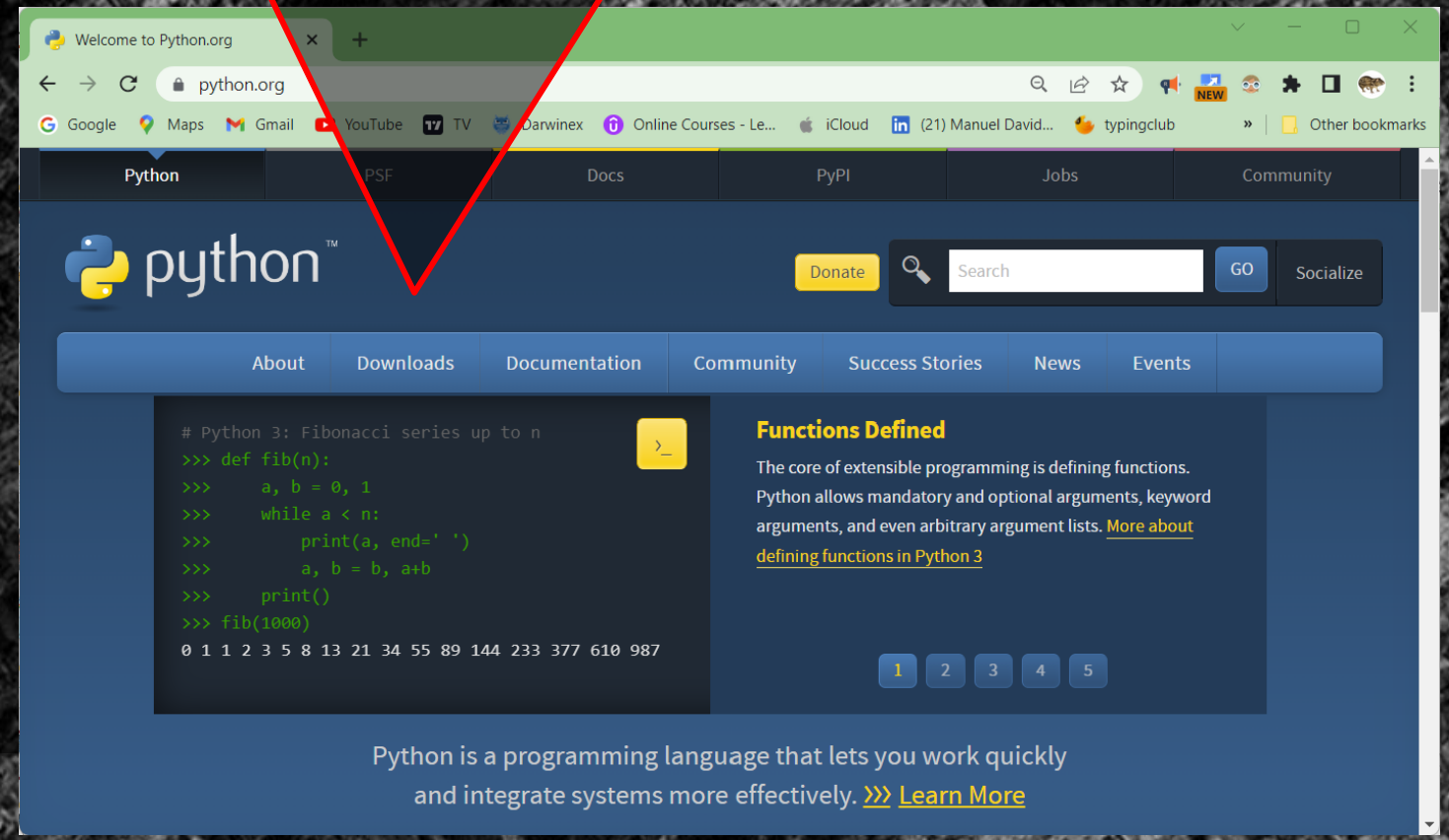
We are going to work with pure Py, running on Jupyter Notebook, so first, let's go and install Py from the official web site:

www.Python.org

For the installation and the course material you will need about 1 GB of free disk space. Anaconda is another common option to have Py and associated libraries, but is 10 time heavier, therefore we do not recommend such option in this course.

Step 1 Select your OS and the installer according to your PC. We recommend version 3.10.0:

https://www.python.org/ftp/python/3.10.0/python-3.10.0-amd64.exe

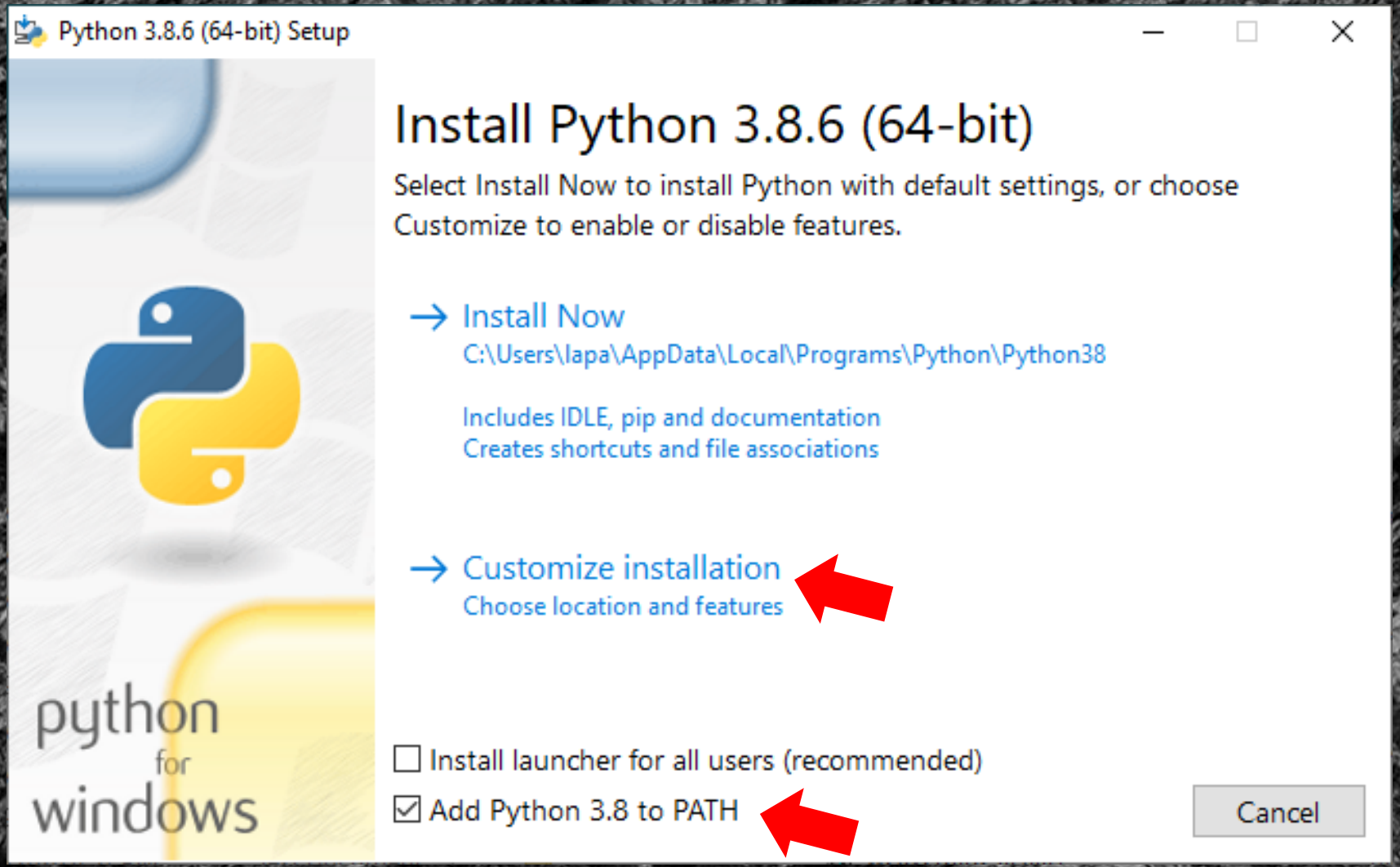


Python installation

Step 2

Double click on the executable file on your Downloads and this window should appear.

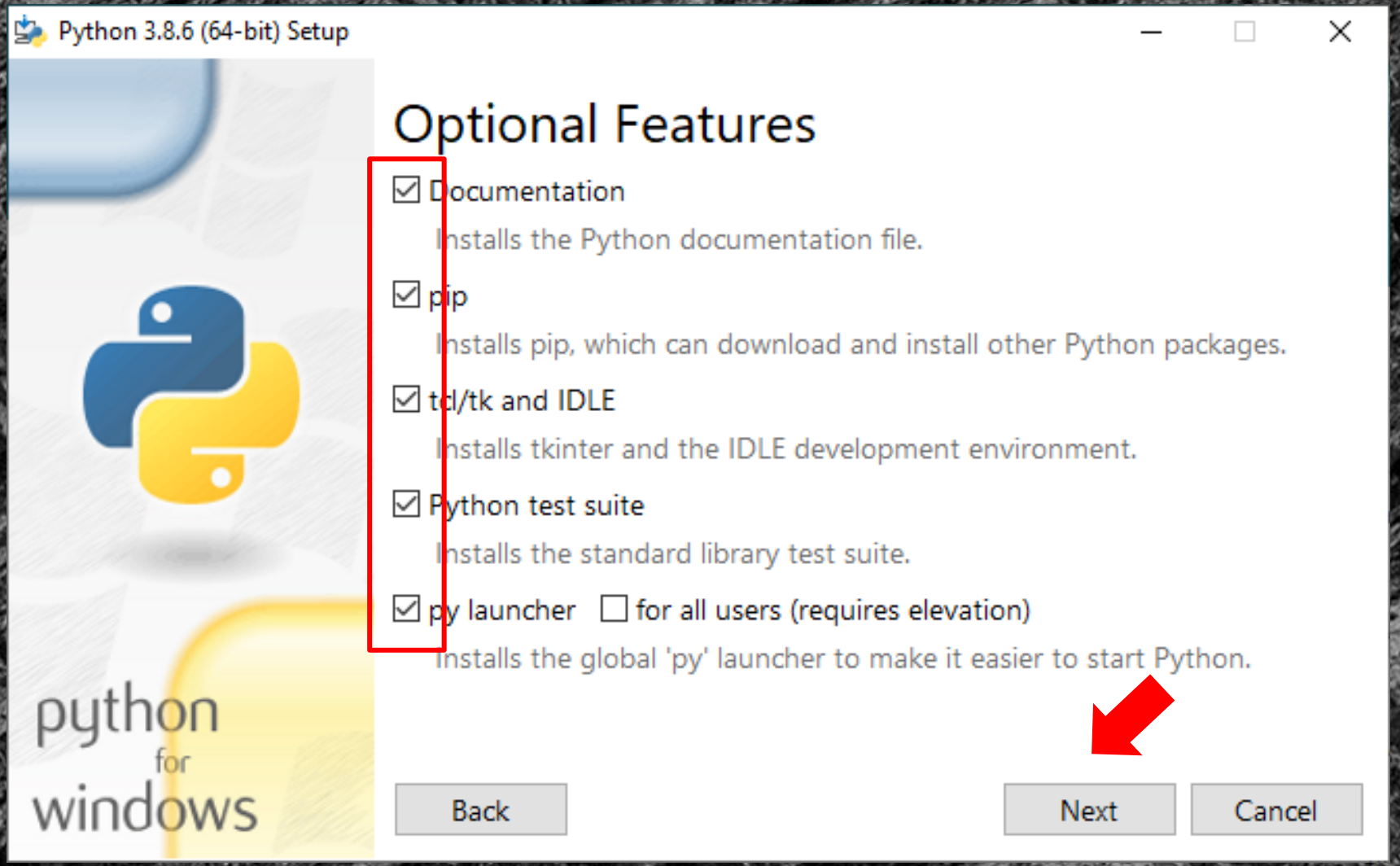
Click on Add Python 3.X to PATH and then Customize installation



Python installation

Step 3

Click on all options in the red rectangle and then Next



Python installation

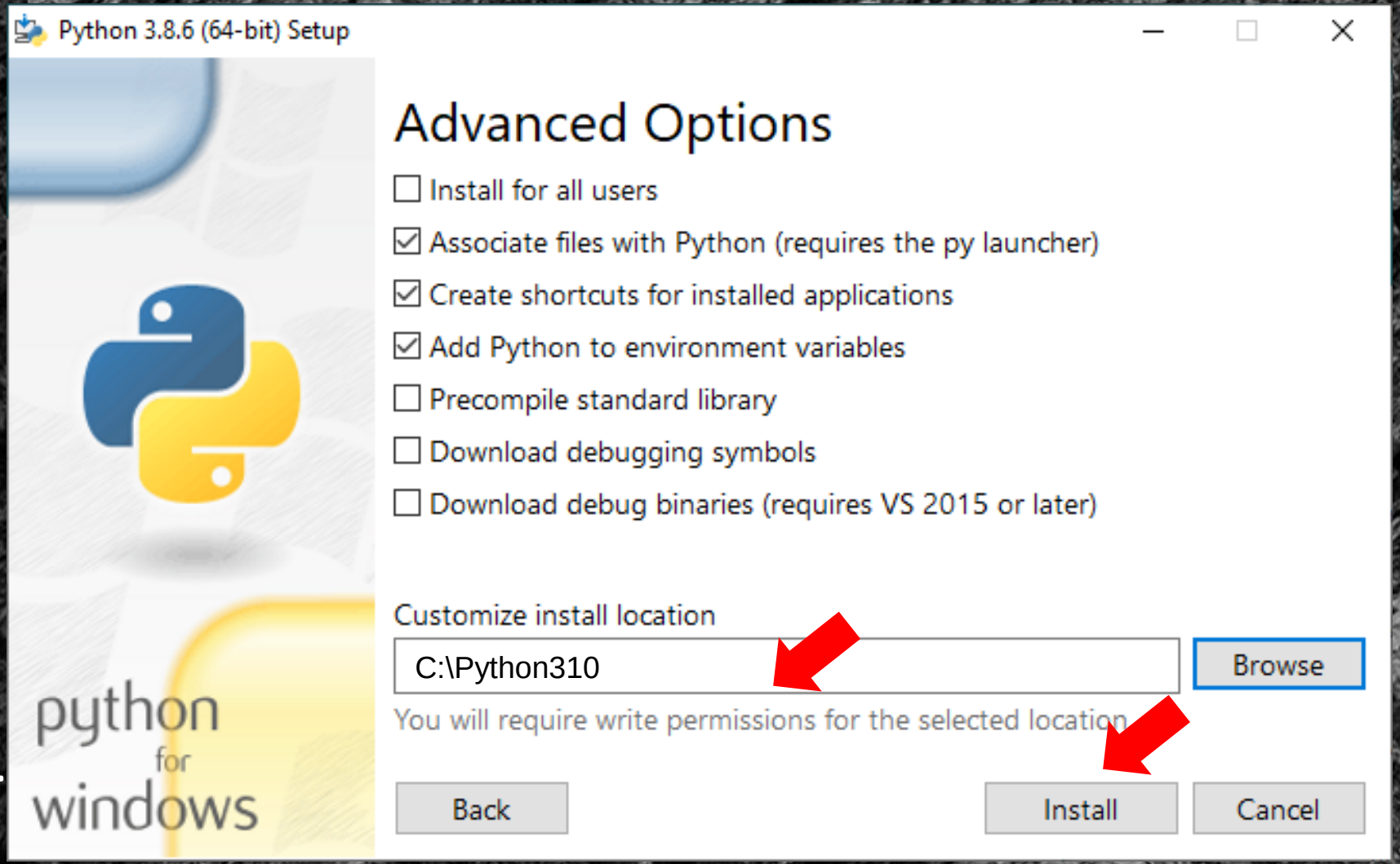
Step 4

Change the Customize install location to a simple path where you have permission to write and enough space (at least 1 GB). We recommend something like this:

C:\PythonX

Where X is the number version to be installed.

Then press the Install button. It should take ~5 minutes



Python installation

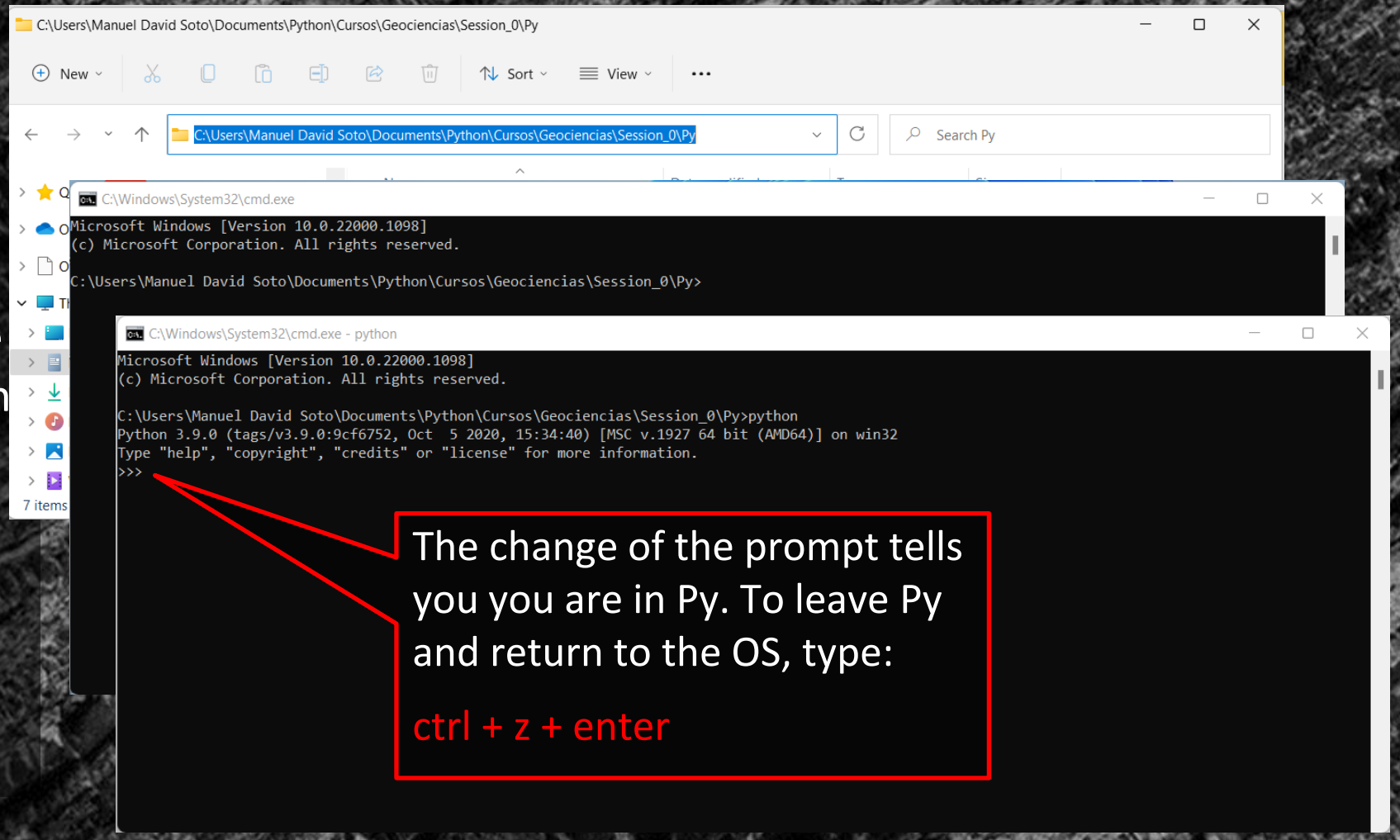
Step 5

Let's check the installation.

Open a cmd (Command Prompt or Terminal window).

A practical way to do that is to located yourself, by using the File Explorer, in the directory in which you want to work. Then highlight the path of your directory (it should turn blue) and type cmd

In the cmd type python and now you are ready to code in the cmd



Basic libraries

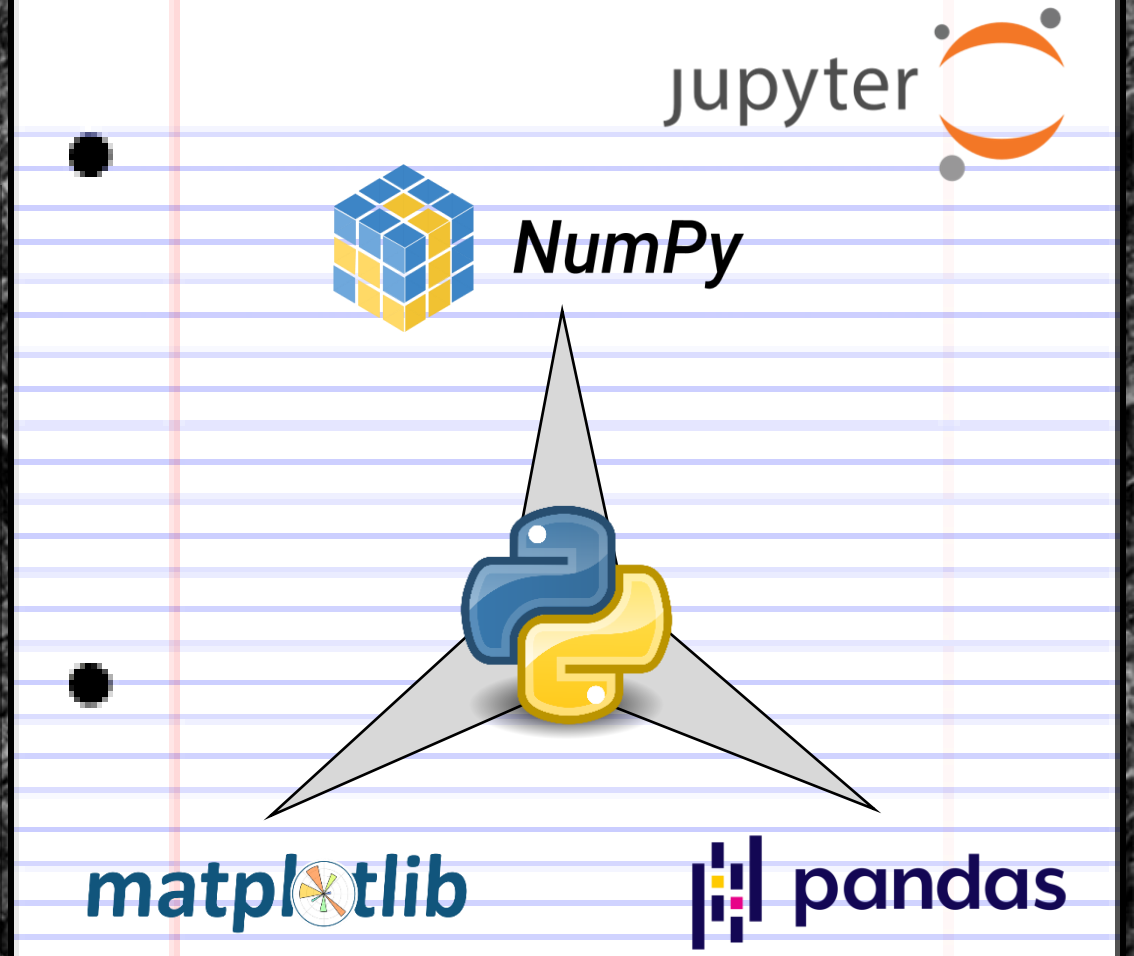
Along this course we are going to use many different libraries, from general libraries to specific ones. Apart from Jupyter that is going to be our work notebook, these are the three libraries (Py holy trinity) that will rarely be missing in a project, they are:

NumPy: scientific computing, https://numpy.org/

Matplotlib: visualization, https://matplotlib.org/

Pandas: manipulation and analysis, https://pandas.org/

Many other libraries are built or use the functions provided by this trio.



Other libraries:

SciPy: scientific and technical computing
Pillow: Images manipulation
Welly: loading, processing, and analysis of well data
ObsPy: framework for processing seismological data
Segyio: interacting with seismic data

Jupyter Notebook

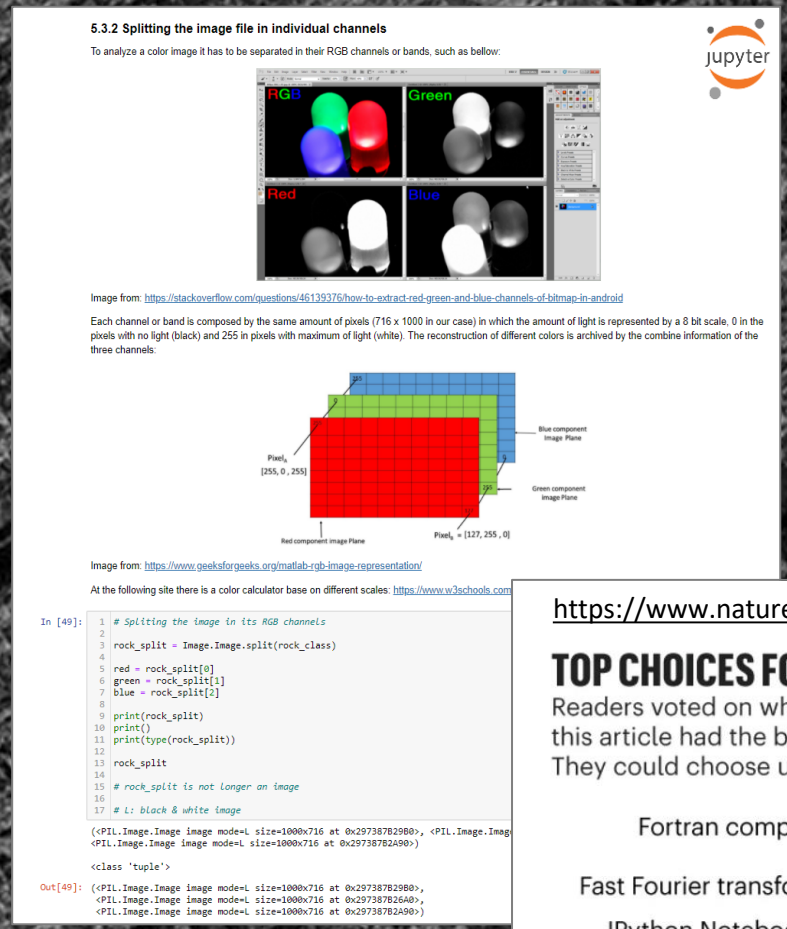
Jupyter Notebook (NB) is an IDE also focus on creating and sharing computational documents. It was initially created, twenty years ago, as Ipython by Fernando Pérez while he was doing his PHD at UC Berkeley. Later evolved to the Jupyter project that provides a collection of open-source tools such as the NB to assist users in the process of interactive computing, explore, analyze and visualize data and computational ideas.

https://data.berkeley.edu/news/project-jupyter-celebrates-20-years-fernando-perez-reflects-how-it-started-open-sciences

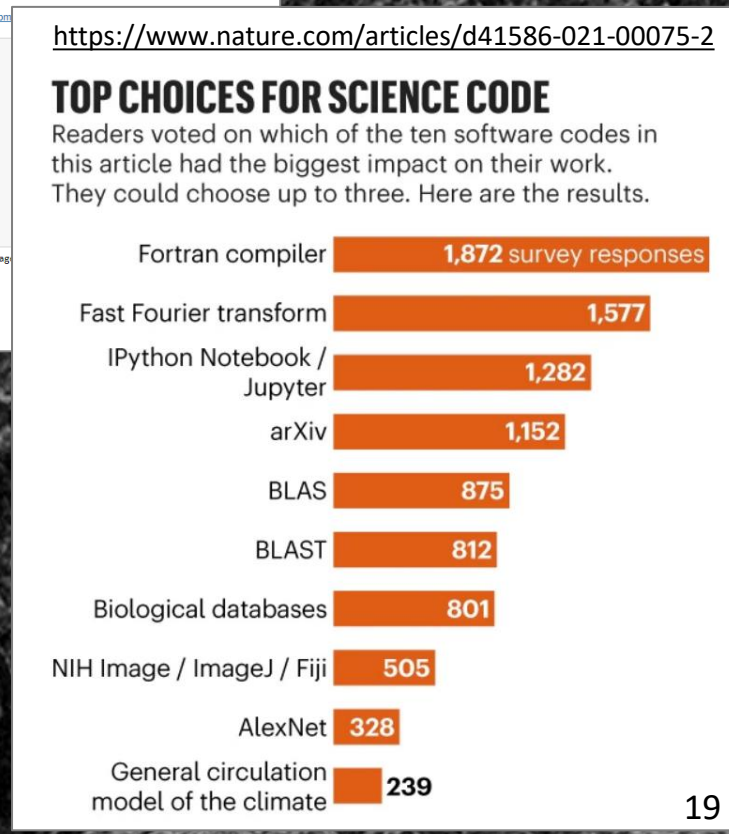
https://analyticsindiamag.com/why-jupyter-notebooks-are-so-popular-among-data-scientists/

https://jupyter.org/

To the upper right you have an example of a NB in which text, images, links to websites, data, codes, and their outputs are well integrated. NB can handle no only Py, but also other important programming languages (Jupyter stand for Julia, Python and R), and can be share easily.



https://www.nature.com/articles/d41586-021-00075-2



Python libraries installation

pip: makes everything easy for you

The command pip (package installer for Python) allows the installation of any PELI (Python's external libraries) available for Python. The command syntax is:

pip install package
pip install package2
pip install package3

Or in a sequential manner:

pip install package package2

Let's do the following installation:

pip install numpy matplotlib pandas jupyter