Introduction to Python for Geophysic

Course creators and speakers: Manuel David Soto, Ulises Berman

Course creators and speakers

Manuel David Soto. Geological Engineer from the Central University of Venezuela in Caracas, Venezuela (1997). Master of Science in Geology from the University of Texas at Austin, USA (2007). Master of Big Data and Business Analytics from the Complutense University of Madrid.

He has worked in geosciences application support and training (Schlumberger Venezuela), as an unconventional exploration geologist (Marathon, USA), as a petrophysicist, conventional and unconventional exploration geologist, and operations geologist (Repsol Colombia and Spain), and as a researcher in the area of petrophysics (GEO3BCN-CSIC, Spain).

He has used Matlab and later Python for teaching (Faculty of Geosciences, University of Barcelona, Repsol, etc.) and in petrophysics (Repsol and GEO3BCN) since 2005.

Course creators and speakers

Ulises Berman. Geophysical Engineer from Simón Bolívar University in Caracas, Venezuela (2023). Master of Science in Earth Sciences with a specialization in Machine Learning from King Abdullah University of Science and Technology (KAUST), Saudi Arabia (2025).

He has worked in mineral exploration (INGEOMIN, Venezuela), as a visiting researcher in the Deep Imaging Group (KAUST), as an intern at Aramco (Dhahran, Saudi Arabia), and his current research within the DeepWave consortium (KAUST) focuses on improving the elastic inversion of complete seismic waves through deep neural networks. His projects have been implemented primarily in Python.

Course content

Introduction (this document)

Session 1 – Python Fundamental: Quick examples. Python Interpreter, variables, data types, control flow, functions, Python external libraries, Numpy arrays, and Matplotlib plots.

Session 2 – Geophysic: segy manipulation and visualization, synthetic

Introduction content

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

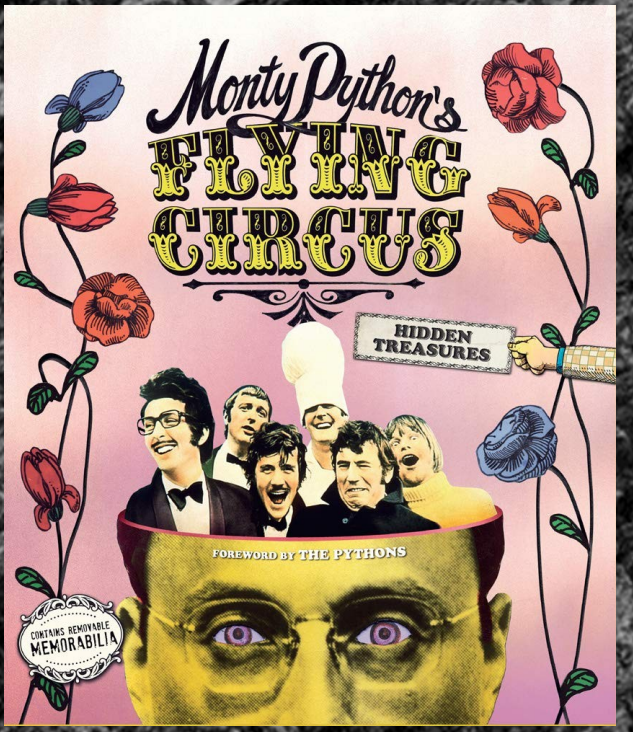
Installations must be done before Session 1. Installation issues can not be addressed during the sessions

Coding or programming

"Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit." Wikipedia

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

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 … 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 just advertising

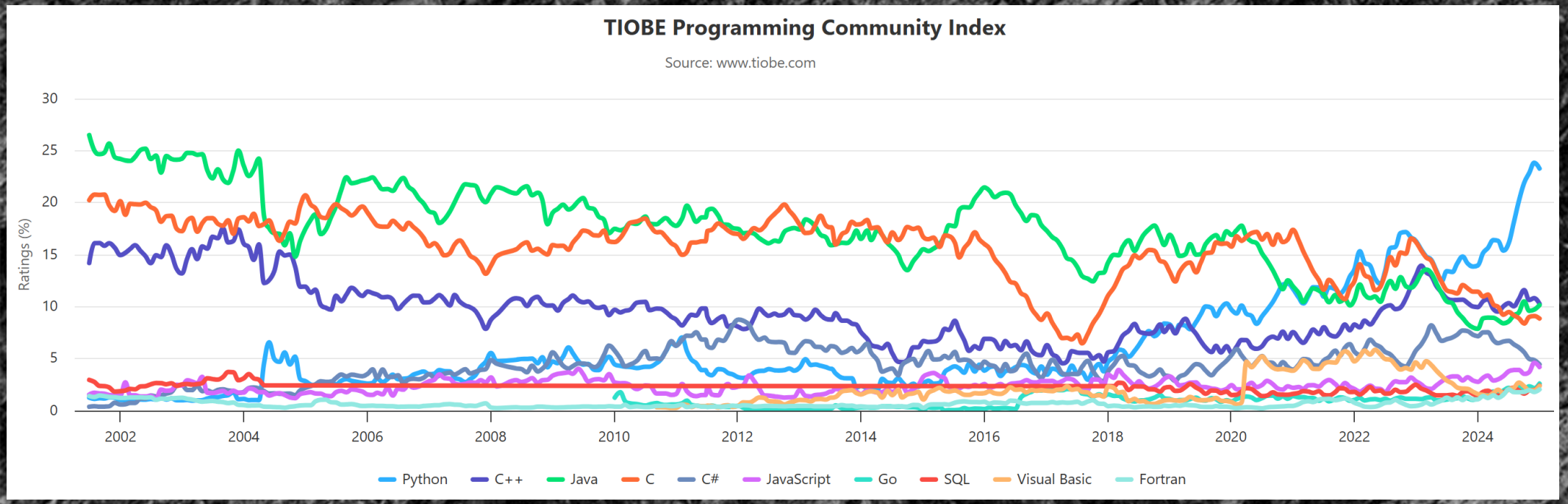
- Python (Py) is an interpreted (no compilation), high-level (user friendly) programming language with dynamic semantics (typing and binding) , multiple programming paradigms (imperative, functional, and object-oriented)
- Py is very attractive for rapid development, as well as glue language to connect existing components together.
- Py is easy to learn and read therefore reduces the cost of maintenance.
- The Py interpreter and its wide range of libraries (standard and external) are available for free for all major platforms (Windows, Mac, Linux, …).
- Py's power is based on a huge ecosystem of specialized libraries, modules or packages, which encourages program modularity and code reuse.

What is Python and why?

C, Java, JavaScript, HTML/CSS, and SQL are specific-purpose programming languages (pl), Py is a general-purpose pl

↓TIOBE is a company specialized in assessing and tracking the use and quality of software

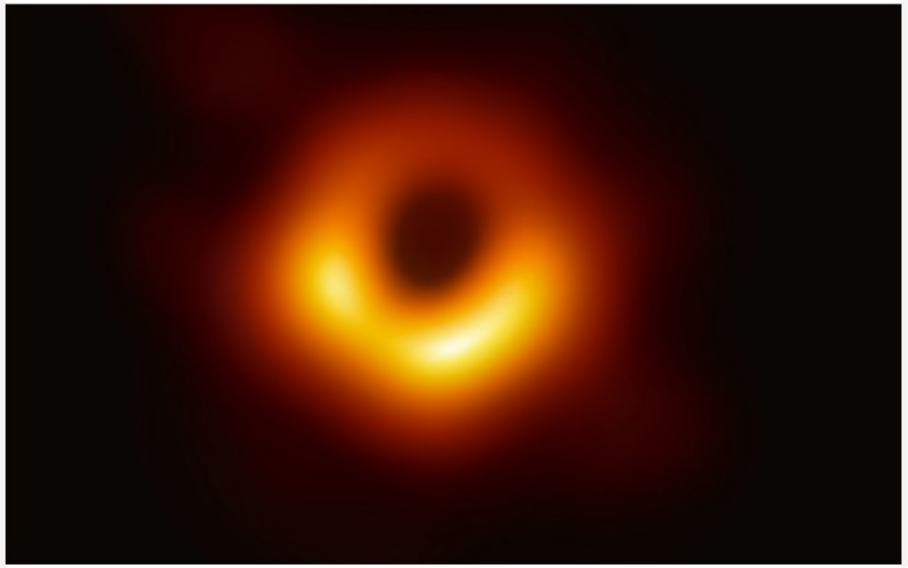
→ Stack Overflow is a question-and-answer website for professional and enthusiast programmers



1st

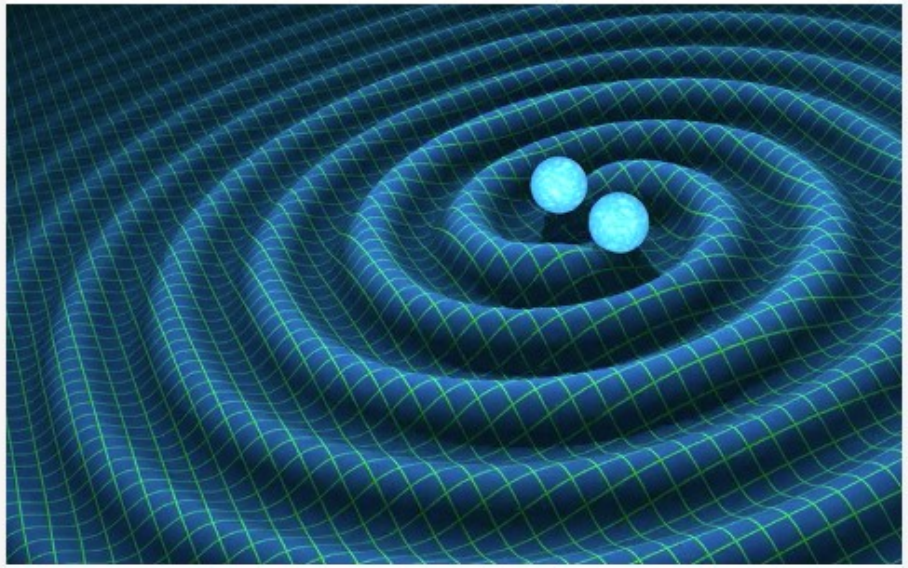
What is Python and why?

FIRST IMAGE OF A BLACK HOLE



How NumPy, together with libraries like SciPy and Matplotlib that depend on NumPy, enabled the Event Horizon Telescope to produce the first ever image of a black hole

DETECTION OF GRAVITATIONAL WAVES

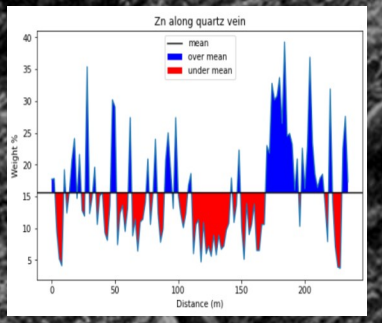


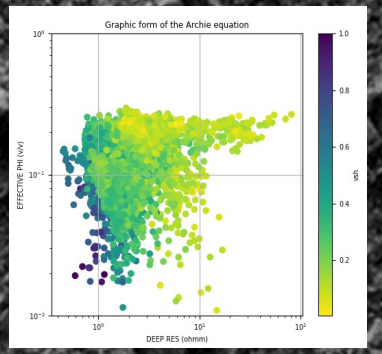
In 1916, Albert Einstein predicted gravitational waves; 100 years later their existence was confirmed by LIGO scientists using NumPy.

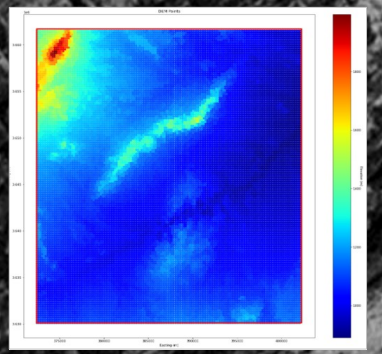
Example of projects with huge impact in science that use Py in their processes.

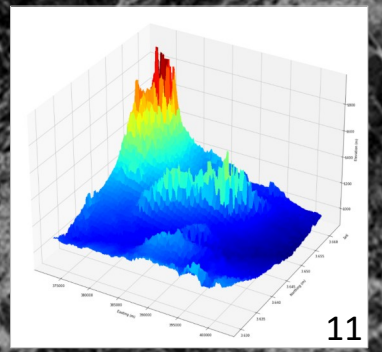
Python for geosciences

- 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) and unstructured data (text, image).
- Py allows loading, QC, manipulation, visualization, and analysis of data of a very diverse nature (1-3D) ; from measurements with the a simple compass to logs, and seismic data.
- Py allows to automatize repetitive tasks.
- Most important commercial programs for geosciences incorporate utilities to code your own Py programs.
- Advance statistical and ML methods are gaining importance in geosciences. Py is the main gate for such advance applications.



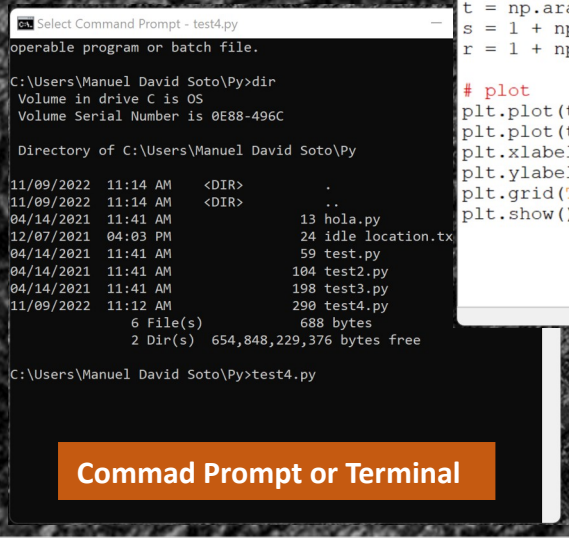


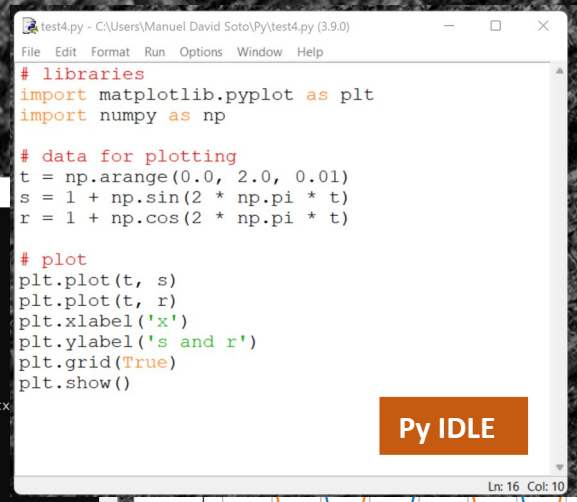


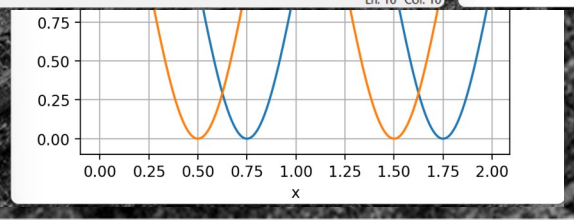


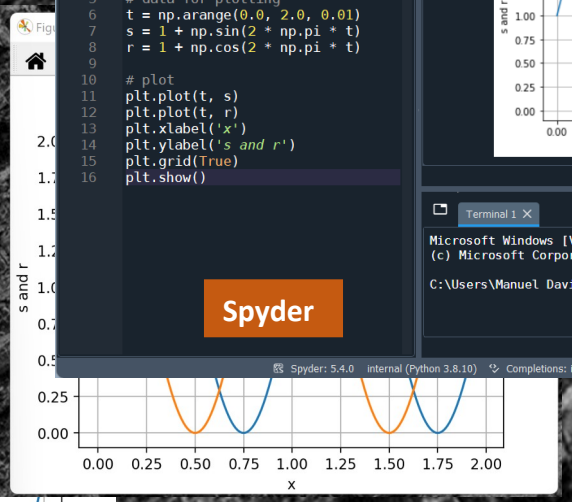
Python execution options

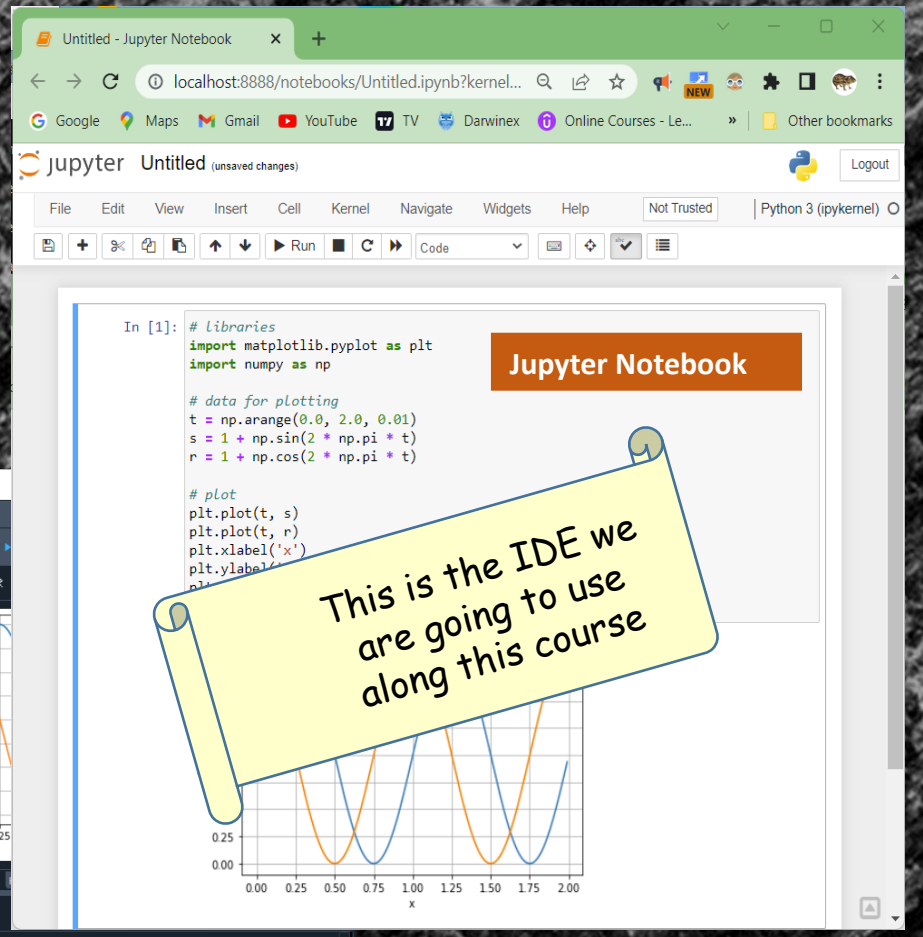
Different IDE (Integrated Development Environments) to work with Py, on different platforms, from the most simple and modest to the most complex and powerful. In these four options installation is required











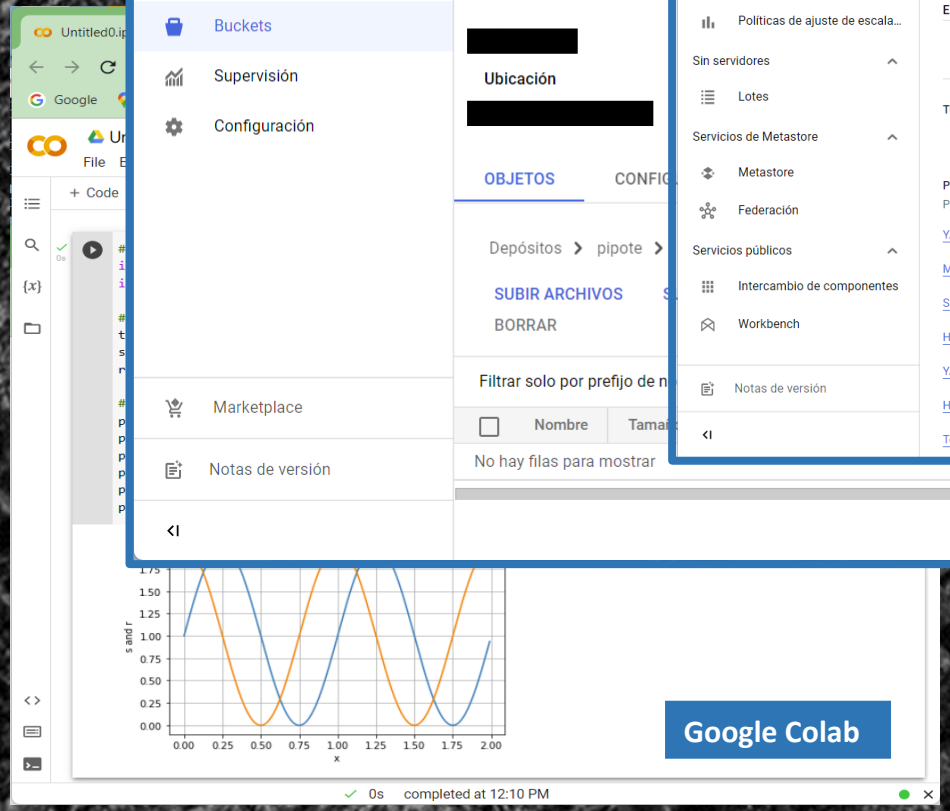
Complexity, power, resources in your PC

\+

\-

Python execution options

In these four options installation is not required

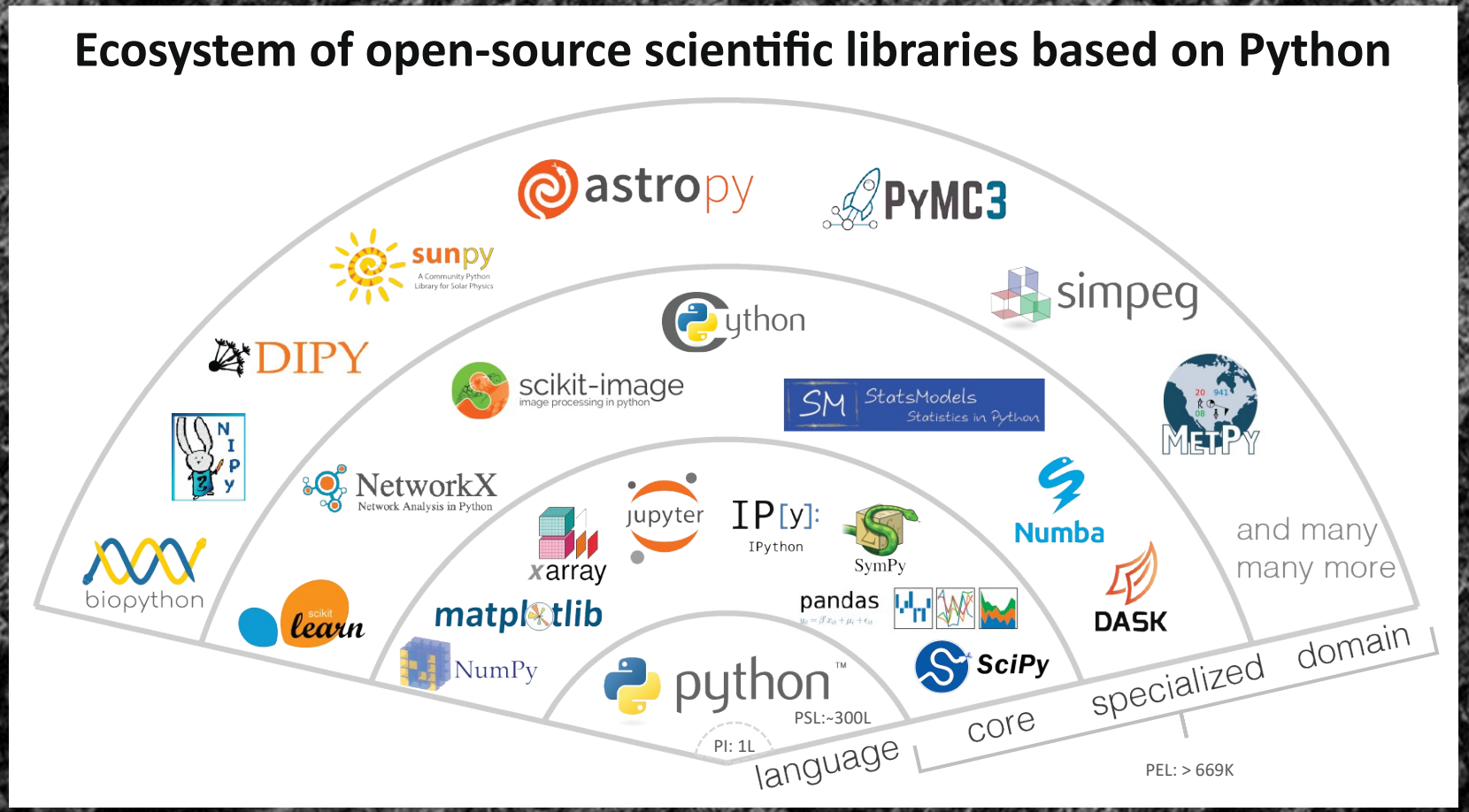


Complexity, power, resources in the could

\+

\-

What is Python and why?



Key aspects of the ecosystem:

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

PI: Py Interpreter
PSL: Py Standard Libraries
PEL: Py External Libraries

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

Python library structure

The pure Py, also know as **PI** (Py interpreter), is the central part (skull & brain) of huge system which has its own built-in functions (~70):

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

In addition to these few functions, directly available in the PI, you can define your own libraries, know here as **UDL** (user defined libraries), composed in turn of several of own functions (ideas).

Just as the bones support the skull & brain, there are the already-installed **PSL** (Py Standard Libraries, ~300) that are available only when called (import), and provide special functions in many areas support areas as systems, math, statistics, :

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

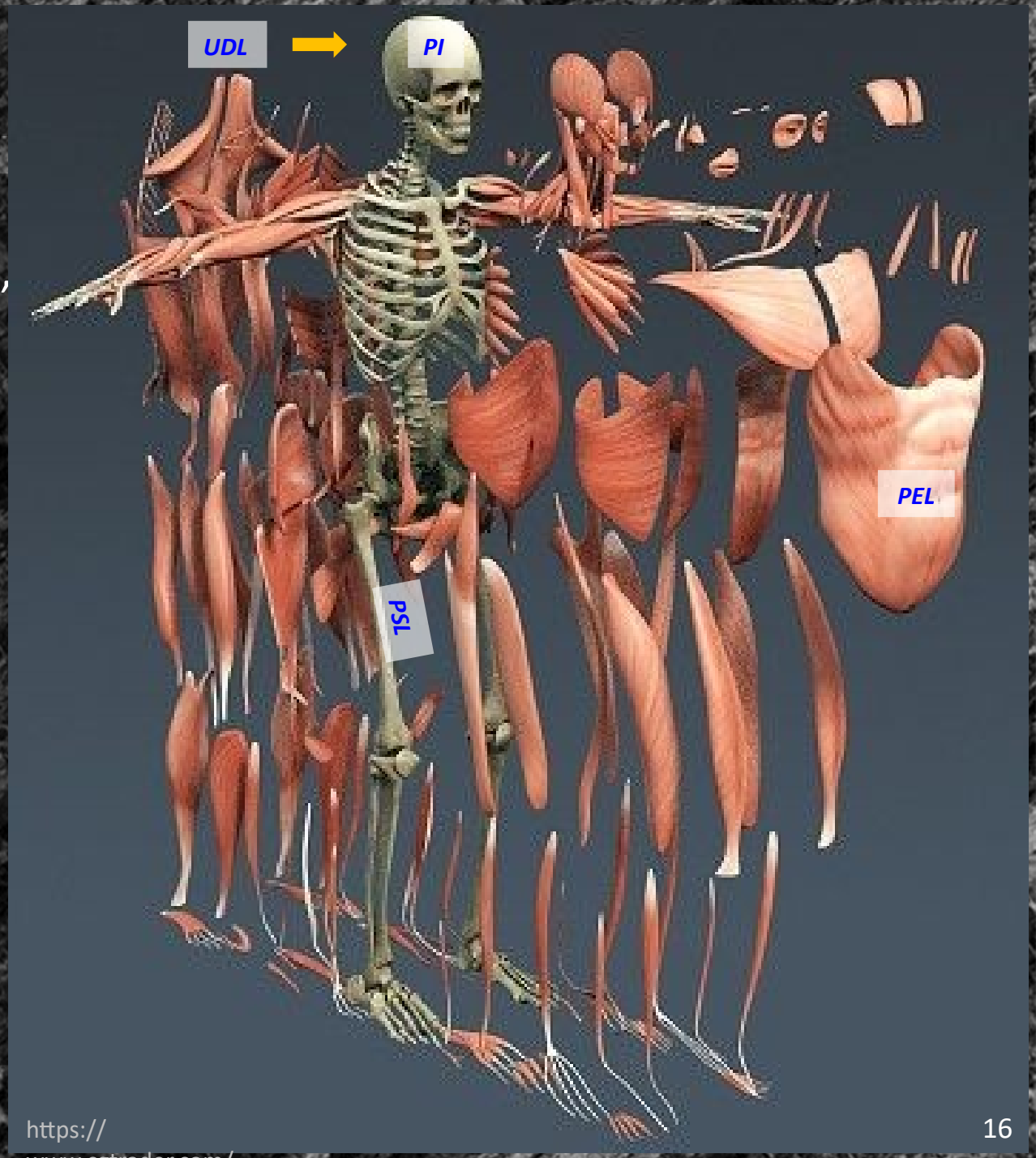
https://

Python library structure

Finally, there is a huge group (> 699K) of **PEL** (Py External libraries). Like specialized muscles or organs, they cover many area (those core, specialized and domain in slide 13), from data visualization, machine learning, genomic, …, and each one has its own functions and sometime data type.

As is indicated by the name PEL, they have to be installed (only one per Py installation) and then imported each time they are going to be used (only one per notebook or other IDE) . They have been developed by individuals, groups, public and/or private institutions. The Py Software Foundation is in charge of organize and distribute them. Explore the list of PELs is at:

https://pypi.org/



https://

Basic libraries

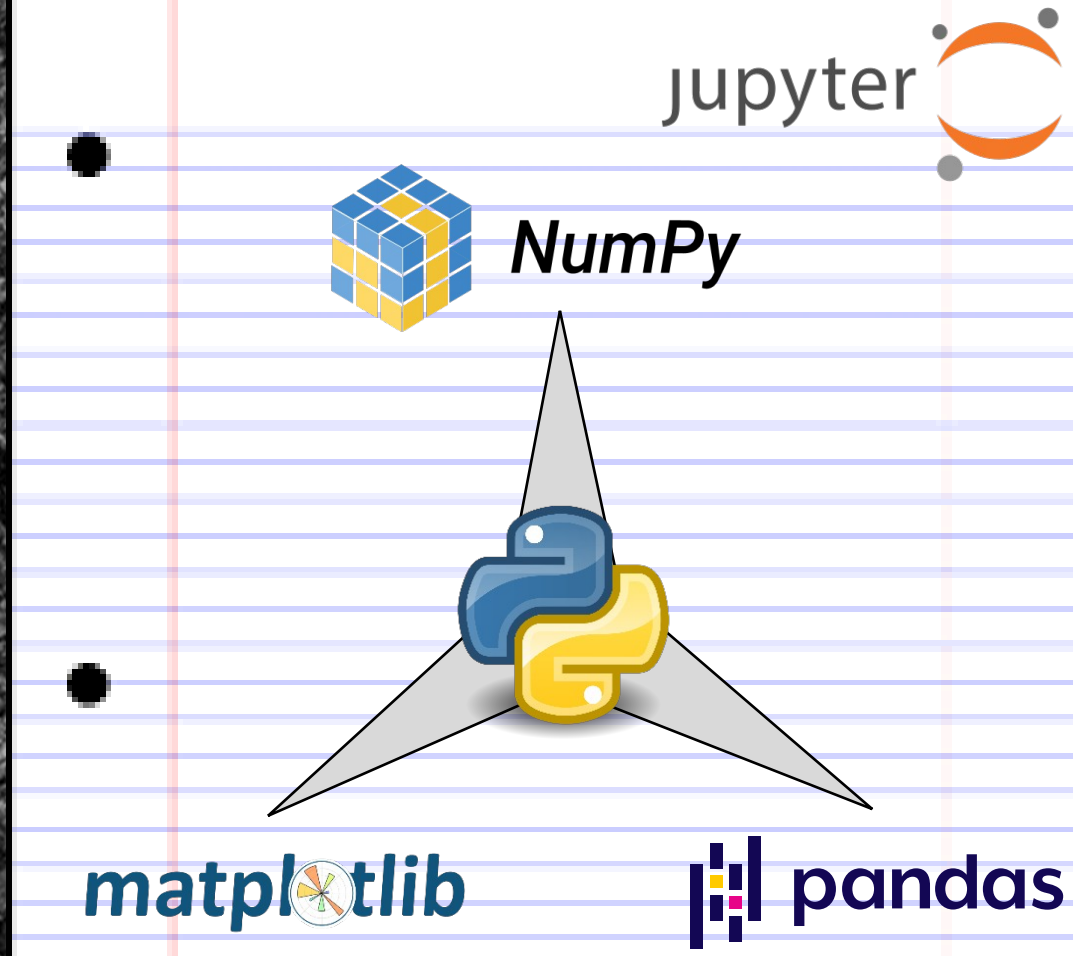
Along this course we are going to use many different libraries, from core to specialized libraries. Apart from Jupyter, that is going to be our work notebook or canvas, these are the three libraries (Holly Trinity) that will rarely be missing in a project, they are:

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

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

Pandas: manipulation and analysis of tabular data, https://pandas.org/

Many other libraries are built or use the functions provide d 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

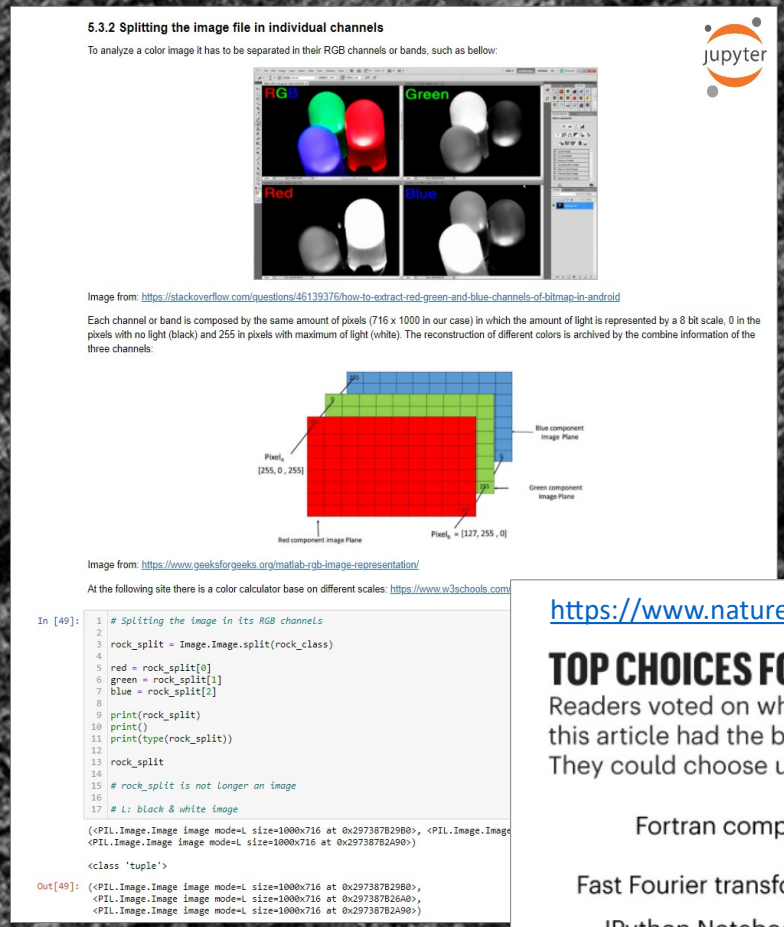
Jupyter Notebook

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.

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

More information at:

[Fernando Perez]
[Why Jupyter notebooks are so popular]
[Jupyter]



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