Jupyter, Markdown, Python

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

fit@hcmus

WHY SHOULD A DATA SCIENTIST HAVE JUPYTER NOTEBOOK IN HIS/HER TOOLBOX?

Why Jupyter notebook

- A data science process is a process of finding an answer for a question through data; this is a long process …
- During this process, we want to **write text, code, write text, code**, ...
- Documenting the process not only helps us to review and continue our work next day but also helps stakeholders to verify our results at the end
- The coding style in data science is also quite special: **exploratory** — code some lines, observe results, code some lines, observe results, …

Why Jupyter notebook

- Use code file + comment in code file?
- Comment only allows simple notes, not supported text format
- The likelihood of coding in an “exploratory” style is quite low, per se
- If you run, you will run from start to finish

Why Jupyter notebook

- Jupyter Notebook is a good tool for our need
- Jupyter Notebook is a “notebook” allowing us to:
  - Write text (using Markdown)
  - Write code and run code (using Python, but other programming languages are also supported)
  - Output of the code
  - Create interactive data visualizations
- Jupyter Notebook also allow us to code in exploratory style

Offers a single document that contains

- Visualizations
- Mathematical equations
- Statistical modeling
- Narrative text
- Any other rich media

This single document approach enables users to develop, visualize the results and add information, charts, and formulas that make work more **understandable**, **repeatable**, and **shareable**.

Two variants of the Jupyter notebook

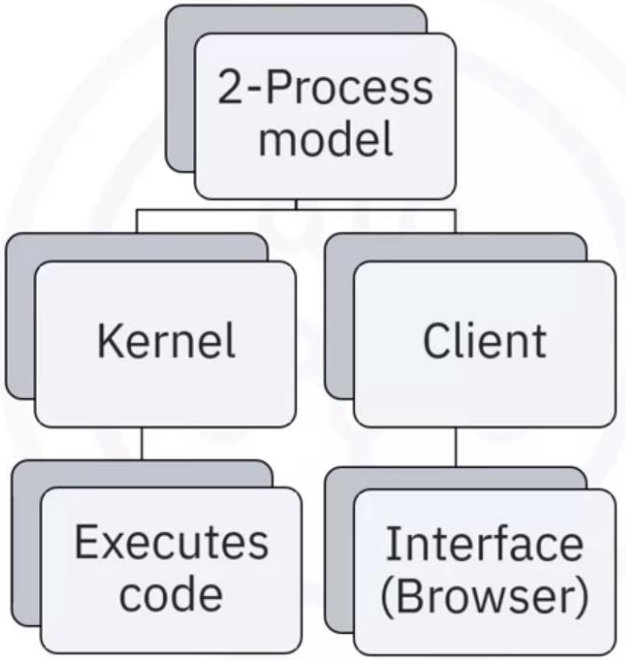
- **Jupyter Classic Notebook**, with all the capabilities mentioned above.
- **JupyterLab**, a new next-generation notebook interface designed to be much more extensible and modular, with support for a wide variety of workflows from data science, machine learning, and scientific computing.

What is Jupyter Notebook

- "notebook" or "notebook documents" denote documents that contain **both code and rich text elements**, such as figures, links, equations, ...
- Because of the mix of code and text elements, these documents are the ideal place to bring together an **analysis description**, and its results, as well as, they can be executed perform the data analysis in real time.
- The Jupyter Notebook App produces these documents.

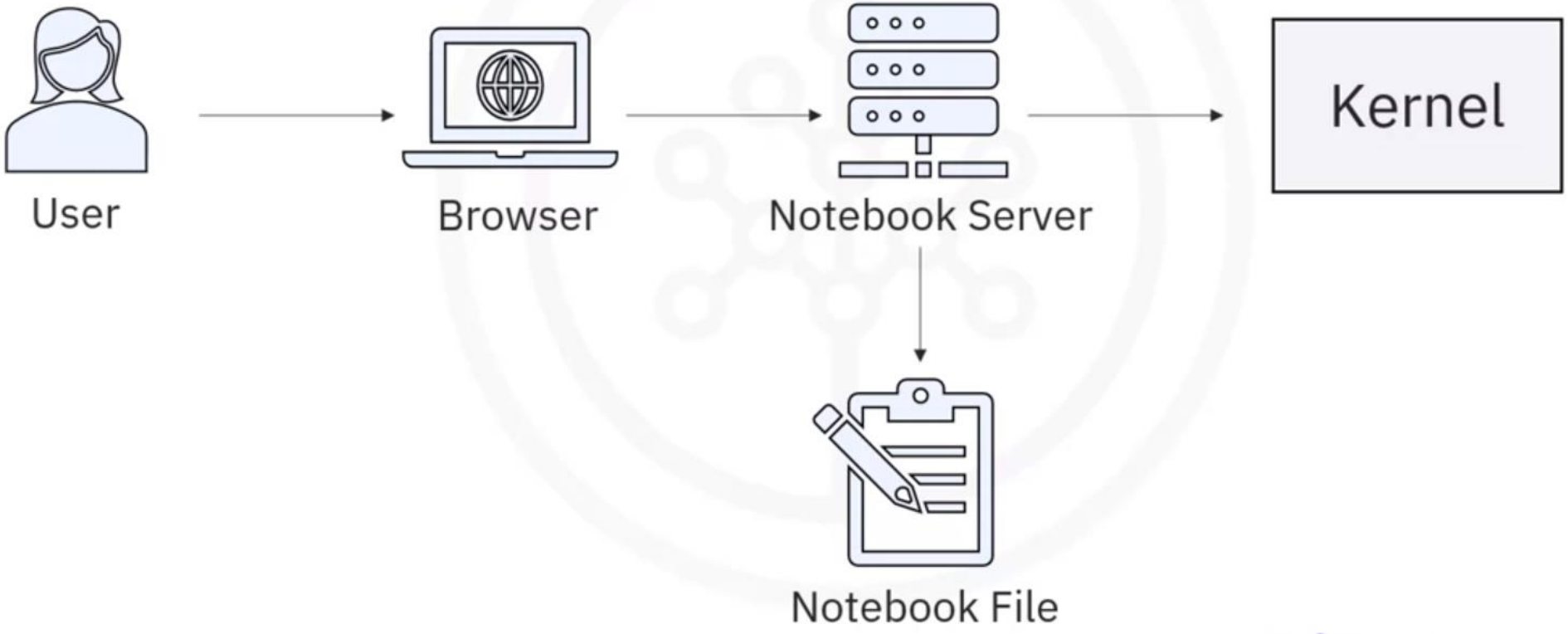
Jupyter Notebook App

- As a **server-client application**, the Jupyter Notebook App allows you to edit and run your notebooks via a web browser.
- The application can be executed on a PC without Internet access, or it can be installed on a remote server, where you can access it through the Internet.



https://www.coursera.org/learn/open-source-tools-for-data-science/

Jupyter Notebook App



Jupyter Notebook App

- Its two main components are the kernels and a dashboard.
- A **kernel** is a program that runs and introspects the user's code. The Jupyter Notebook App has a kernel for Python code, but there are also kernels available for other programming languages.
- The **dashboard** of the application not only shows you the notebook documents that you have made and can reopen but can also be used to manage the kernels: you can which ones are running and shut them down if necessary.

Use your Jupyter Notebooks

- Try to provide **comments** and **documentation** to your code. They might be a great help to others!
- Also consider a consistent naming scheme, code grouping, limit your line length, ...
- Don't be afraid to **refactor** when or if necessary.

Use your Jupyter Notebooks

- Don't forget to name your notebook documents!
- Try to keep the cells of your notebook simple: don't exceed the width of your cell and make sure that you don't put too many related functions in one cell.
- If possible, import your packages in the first code cell of your notebook
- Display the graphics inline.
- Sometimes, your notebook can become quite code-heavy, or maybe you just want to have a cleaner report. In those cases, you could consider hiding some of this code. You can already hide some of the code by using magic commands such as %run to execute a whole Python script as if it was in a notebook cell.
  - How to hide code

Share your Jupyter Notebooks

- Share .ipynb
  - Click “Cell > All Output > Clear”
  - Click “Kernel > Restart & Run All”
  - Wait for your code cells to finish executing and check they did so as expected

Share your Jupyter Notebooks

- File>Download as
- jupyter nbconvert --to html Untitled4.ipynb

AsciiDoc (.asciidoc)

HTML (.html)

LaTeX (.tex)

Markdown (.md)

Notebook (.ipynb)

PDF via LaTeX (.pdf)

PDF via HTML (.html)

PNG via HTML (.html)

reST (.rst)

Python (.py)

Reveal.js slides (.slides.html)

Jupyter Notebooks for Data Science Teams: Best Practices

- Use two types of notebooks for a data science project, namely, a **lab notebook** and a **deliverable notebook**. The difference between the two is the fact that individuals control the lab notebook, while the deliverable notebook is controlled by the whole data science team,
- Use some type of versioning control (Git, Github, ...). Don't forget to commit also the HTML file if your version control system lacks rendering capabilities, and
- Use explicit rules on the naming of your documents.
- For more information

Learn From the Best Notebooks

- Notebooks are also used to complement books, such as the Python Data Science Handbook. You can find the notebooks here.
- This matplotlib tutorial is an excellent example of how well a notebook can serve as a means of teaching other people topics such as scientific Python.

Jupyter notebook reference

- https://jupyter-notebook.readthedocs.io/en/stable/notebook.html
- https://www.edureka.co/blog/wp-content/uploads/2018/10/Jupyter_Notebook_CheatSheet_Edureka.pdf

Markdown

- Markdown allow us to write plain text with marks to specify simple and widely used formats such as heading, bullet, italic, bold, …
- It's **easy to read and write**
- Restricting users to a small set of formats also helps users to **focus more on content** when writing

Markdown

- If you just need to write text and don't need to run code, you can simply write to a Markdown file (*.md)
- From a Markdown file, you can convert to other formats using **Pandoc**
- In Jupyter Notebook, Pandoc is used behind the scene to convert a Notebook file to other formats

Markdown - Reference

- https://www.markdownguide.org/basic-syntax/
- https://www.markdowntutorial.com/

Python

- Why did Guido van Rossum create Python programming language, although C had already existed at that time?
- Guido van Rossum: “I realized that the development of system administration utilities in C was taking too long”

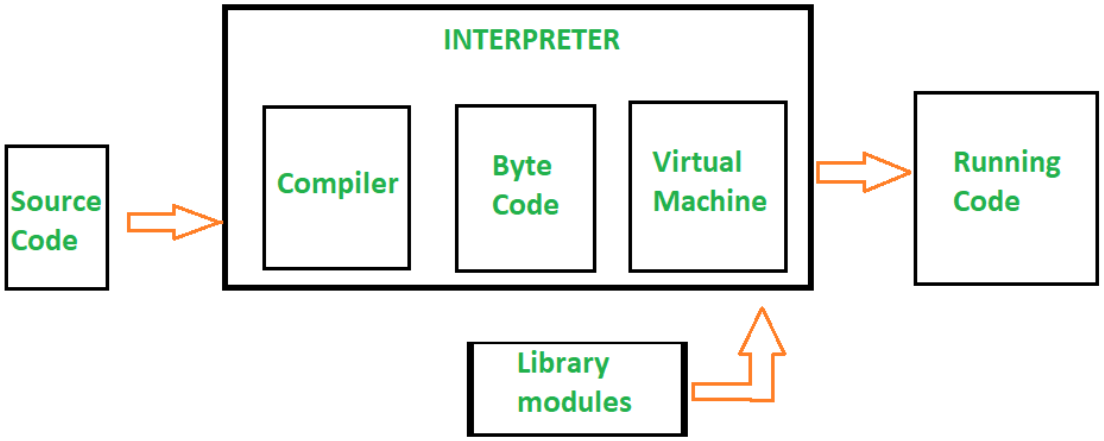
Why Python write faster

- **Interpreted language**: no need to compile all code before running; instead, programmers can code some lines of code and run immediately, and can continue to code and run from previous running state
- **Dynamic typing**

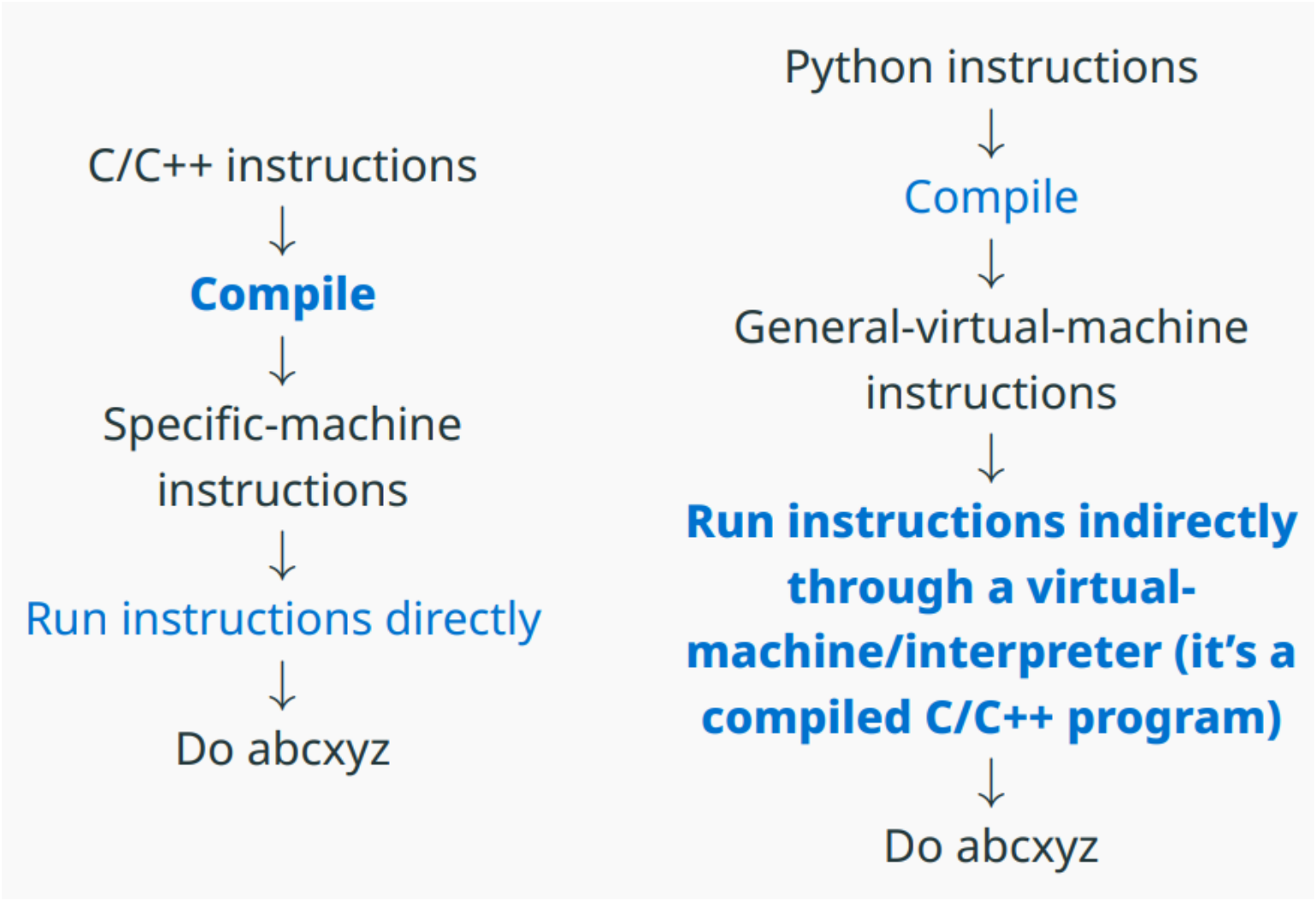
```
a = 1                   def add_two_things(a, b):
...                         return a + b
a = 'hello'             # Can call add_two_things
                        # with 2 arbitrary objects
                        # having + operation
```

- **Automatic memory management**: programmers don't have to spend time doing malloc and free memories manually as in C/C++

Why Python write faster



Why Python write faster

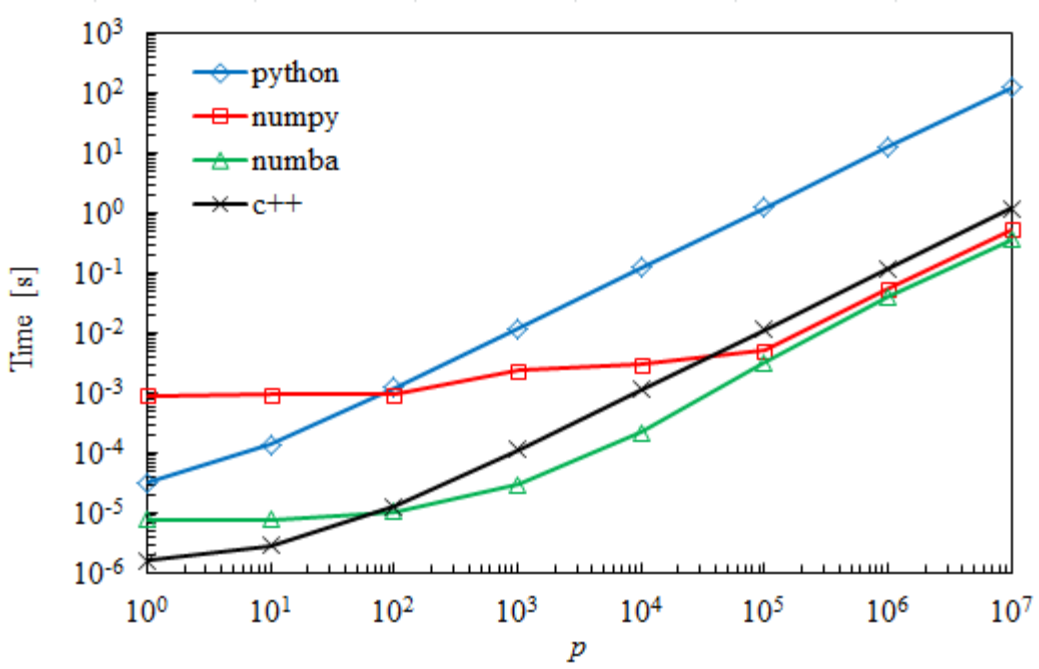


Why Python run slower

- **Interpreted language**: interpreted languages *often run slower* than compiled languages, because in compiled languages, before running, compiler will look at all source code and do optimizations
- **Dynamic typing**: to achieve this, a variable in Python is just a pointer pointing to an object, and this object contains not only value but also meta info such as data type, …; when do an operation with 2 objects, Python interpreter first have to *spend time* opening these objects to identify data type
- **Automatic memory management**: it also costs *extra work and time* to know when to free allocated memories

Homework

- Compare runtime of matrix multiplication program with: **C**, **Python**, **numpy**.
- Test with several different size of matrix.
- Write report with Jupyter notebook with code and explain.



Reference

- https://www.datacamp.com/tutorial/tutorial-jupyter-notebook
- https://www.markdowntutorial.com/