



DÀI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN  
VIETNAM - KOREA UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY

한-베정보통신기술대학교



# Chapter 1

## Introduction to Python & Jupyter Notebook

- What is Python?
- Philosophy of Python
- Brief Development History of Python
- Why Python?
- Who uses Python today?
- Python IDEs and Code Editors
- Jupyter Notebook
- The first Python Program

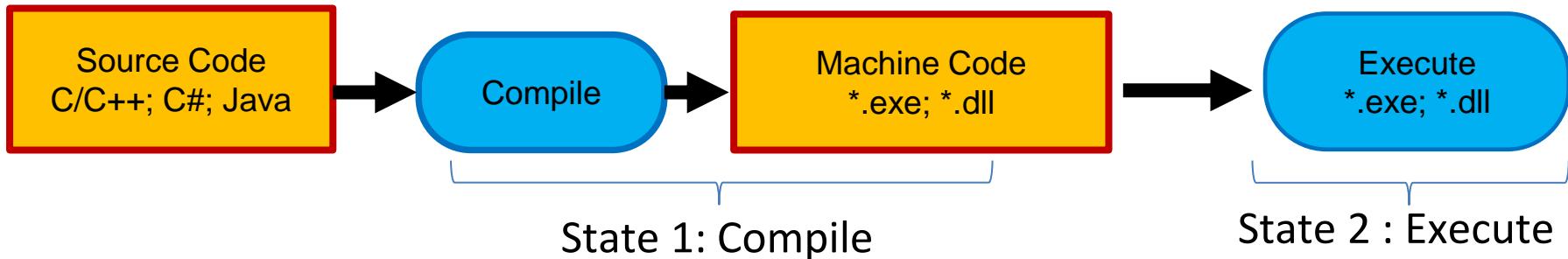
- Python is ...
  - a **scripting language** characterized by
    - **Interpreted** - is processed at runtime by the interpreter.



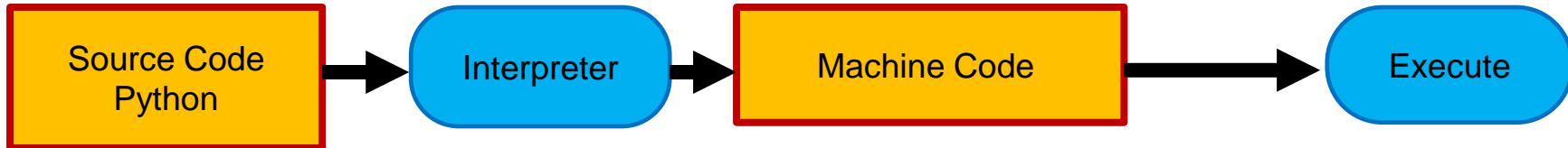
- **Interactive** – uses prompts and interact with the interpreter directly to write programs.
- **Object-Oriented** – supports Object-Oriented style or technique of programming that encapsulates code within objects

- Compiler – Interpreter: To convert computer programs from a high-level programming language to a machine language → use one of two methods:
  - **Interpreter:** At the time of running the program, the program is converted to machine language and executed (convert and execute at the same time).
  - **Compiler:**
    - State 1: Compile → computer programs is converted to machine language: .exe ; .com; .bat; .dll; ...
    - State 2 : Execute the compiled program

- Compile and Execute



- Interpret and Execute



- A high-level programming language designed emphasize
  - **Code readability** - designed to be highly readable.
  - **Developer productivity** - automates certain areas of computing systems to make the process of developing a program simple and fast.
  - **Program portability** - used to write software in a wide variety of application domains (e.g. web applications, GUI desktop applications, scientific and numeric application).

- Based on minimalistic design philosophy in which complex tasks can be done in simple ways.
- “Python is an experiment in how much freedom programmers need. Too much freedom and nobody can read another's code; too little and expressiveness is endangered.” - Guido van Rossum
- “The Zen of Python” is a collection of 19 guiding principles for writing computer programs in Python.
- (<https://zen-of-python.info>)

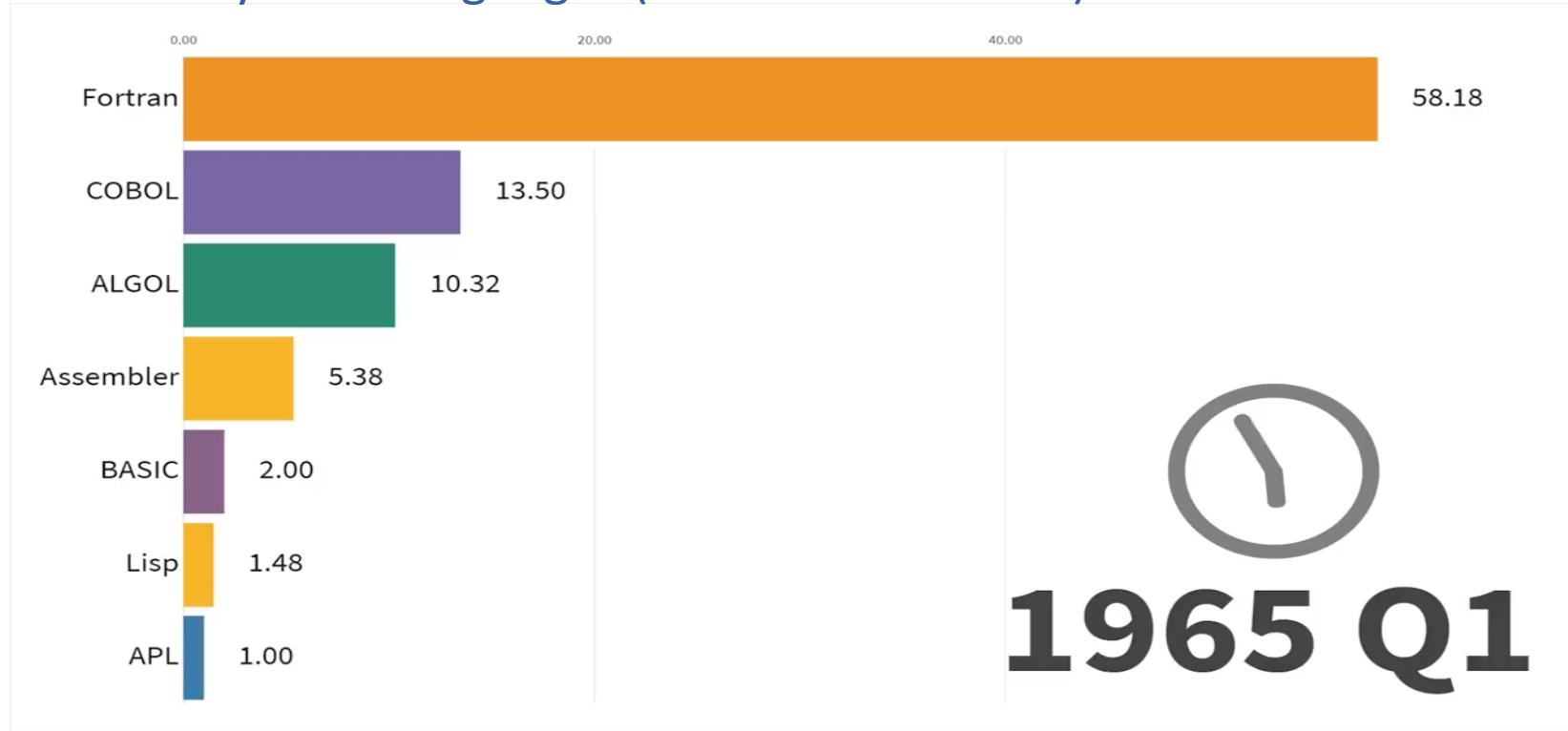


- Python was conceptualized by Guido van Rossum in the late 1980s.
- Guido van Rossum retired at October, 2019
- Currently, Guido van Rossum joined Microsoft



Version	Release time
Python 1.0 (First standard release)	1994
Python 1.6 (Final version 1.x)	2000
Python 2.0	10/2000
Python 2.7 (Final version 2.x)	2010
Python 3.0	2008
Python 3.7.3/3.7.16	3/2019
Python 3.9.6	29/6/2021
Python 3.11.4	Currently (17/8/2023)

- Commonly used languages (from 1965 to 2019)



- Python is easier and simpler to learn and use than other languages.
  - A dynamically typed language, i.e. need not declare the type of variables, declare the class like some other languages
  - simple syntaxes, i.e. need not use any opening/closing braces, semicolons, etc.

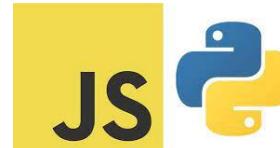
Java

```
public class Main
{
    public static void main(String[] args) {
        int a= 10, b=20;
        int result = a+b;
        System.out.println("The result a+b = " + result);  }
}
```

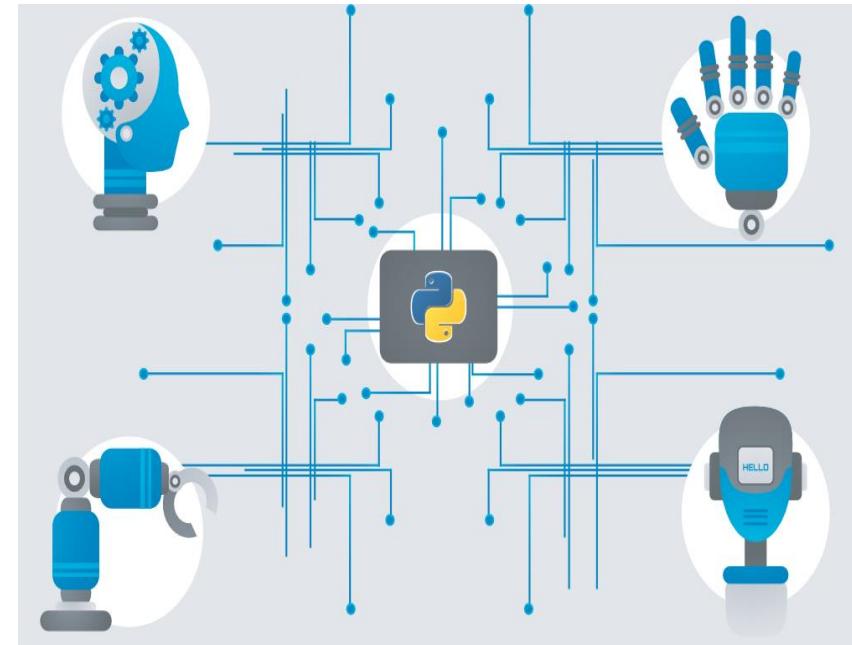
Python

```
a=10
b=20
print ('The result a+b = ', a+b)
```

- Python is open source! Free!  
Massive online support from many resources and quality documentation worldwide.
- Python projects can be integrated with other systems coded in different programming languages.

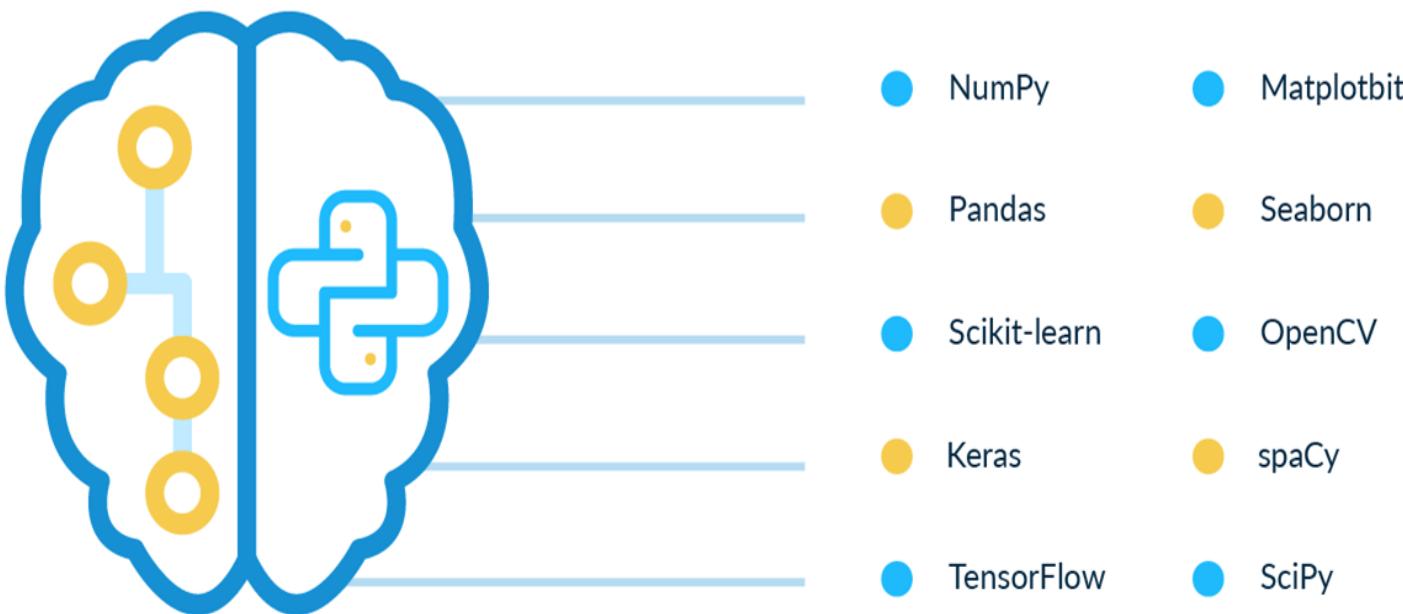


- Python is one of the best programming languages for Data science (DS) and AI.
  - Python's simplicity allows developers to put effort into solving AI problems instead of focusing on technical nuances of the language

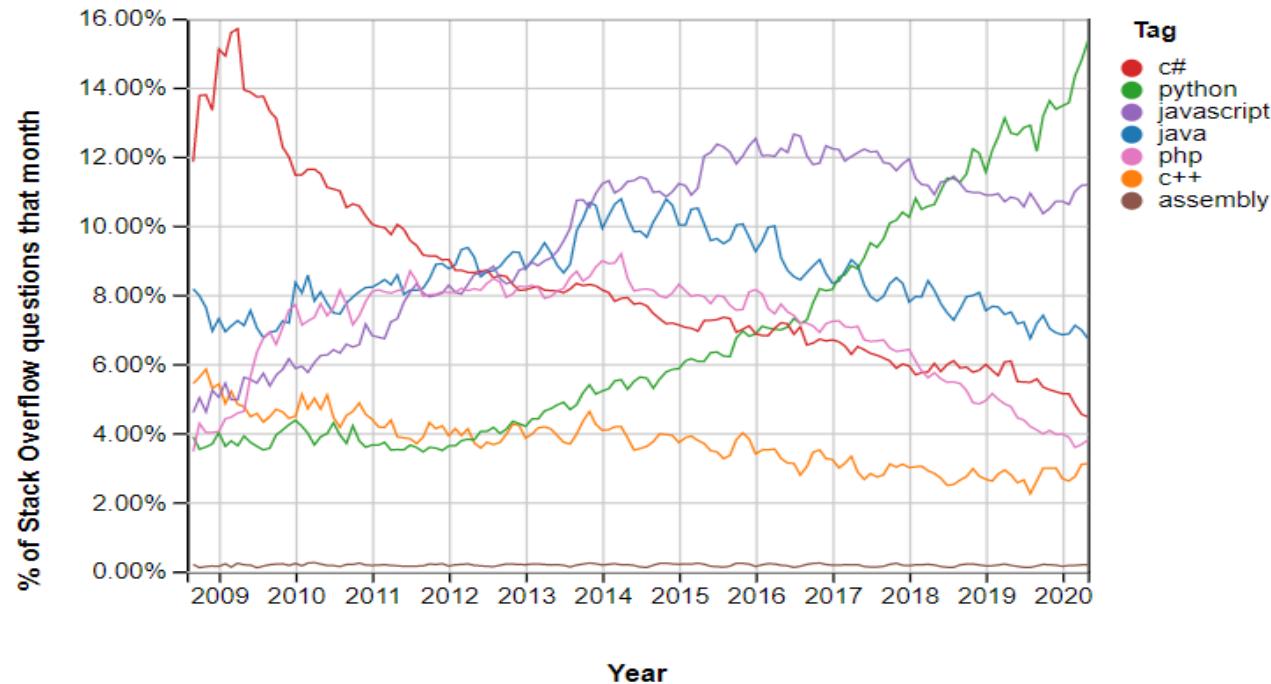


Jun 2022	Jun 2021	Change	Programming Language	Ratings	Change
1	2	▲	 Python	12.20%	+0.35%
2	1	▼	 C	11.91%	-0.64%
3	3		 Java	10.47%	-1.07%
4	4		 C++	9.63%	+2.26%
5	5		 C#	6.12%	+1.79%
6	6		 Visual Basic	5.42%	+1.40%
7	7		 JavaScript	2.09%	-0.24%
8	10	▲	 SQL	1.94%	+0.06%
9	9		 Assembly language	1.85%	-0.21%

- Python offers extensive set of libraries for Machine Learning and plenty of data processing tools to handle the data.



- The incredible growth of Python is shown very clearly by StackOverflow:



<https://www.stxnext.com>



Confidence Comes Standard.™



Massachusetts  
Institute  
of  
Technology



NOKIA  
Connecting People



YAHOO!

YouTube



LexisNexis®



mozilla  
Firefox

Google™



TRUECar.  
Know the Real Price™



EVE  
ONLINE



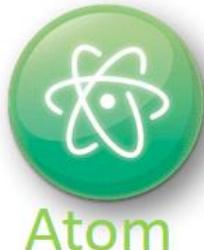
AstraZeneca



INDUSTRIAL  
LIGHT & MAGIC  
A LUCASFILM COMPANY



Source: logicfinder



.....

- Introduction
- Jupyter Architecture
- Why uses Jupyter Notebook?



- Jupyter Notebook formerly known as IPython (or Interactive Python), is an open source web application that is used to create and share documents that contain live code, equations, visualizations, and text.
- The name Jupyter is a reference to the three core languages supported by the project (Julia, Python, and R)
- Project Jupyter supports interactive data science and scientific computing across more than 40 programming languages.

jupyter IntroToJupyterPython (autosaved)

File Edit View Insert Cell Kernel Help

Not Trusted Python 2

There are many shortcuts available. Some of these include:

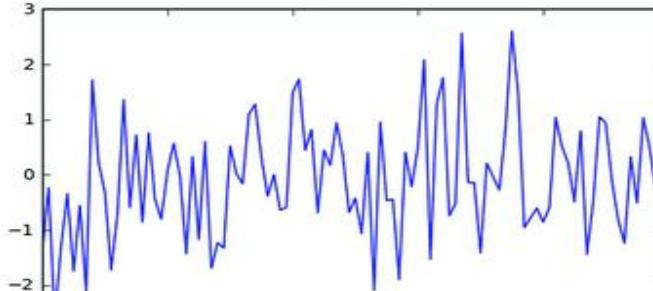
Basic navigation: up-arrow, down-arrow, enter, shift-enter, up/k, down/j Saving the notebook: s Cell types: y (code), m (markdown), 1-6 (heading level) Add cells: a (cell above), b (cell below) Cell editing: x (delete), c(copy)

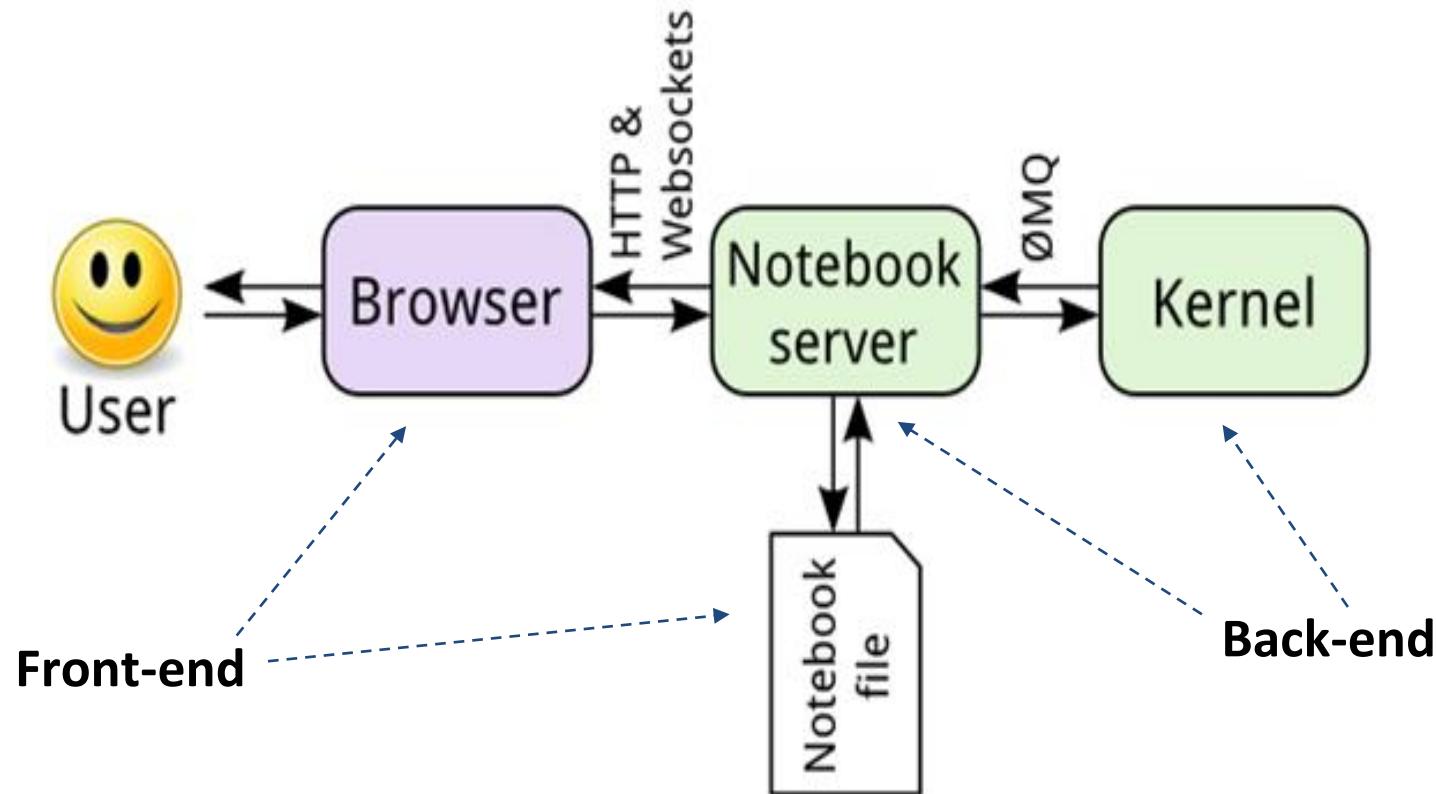
For a full list, see Help/Keyboard shortcuts on the top Menu.

Now let's try some examples! To run: double-click in the code cell below, then press Shift+Enter:

```
In [1]: # This is a code cell. Click on it and press Shift+Enter. While it's executing, the prompt will turn into In [*]:  
%pylab inline  
print "Python is easy to learn!"  
  
plot(randn(100))  
Populating the interactive namespace from numpy and matplotlib  
Python is easy to learn!
```

```
Out[1]: [<matplotlib.lines.Line2D at 0x49a0d68>]
```

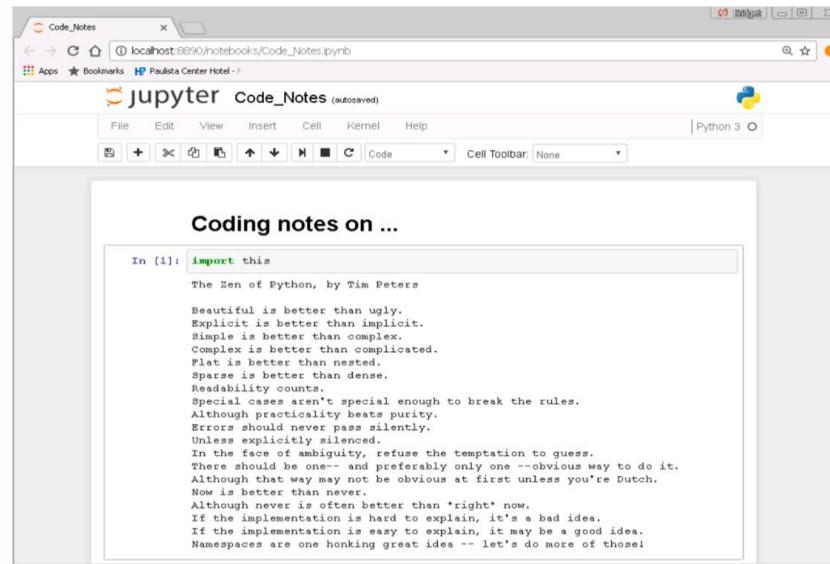




- **Front-end:** The user will work with:
  - **Web Application:** Browser-based tool for interactive development of notebook documents
  - **Notebook Document:** A representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images, and rich media representations of objects.

- **Back-end:** The user doesn't directly interact with, but should at least be aware of
  - **Kernel:** A separate process responsible for running user code. For the purposes of the course, we will be working on Python kernels, although Jupyter is capable of interfacing with other programming languages as well.
  - **Notebook Server:** Communicates with kernel and routes the Python programming language to the web browser.

- Documentation and literate programming by combining rich-text narrative concepts & machine-readable code as Coding Diary. The notebook itself is a data-structure with metadata that can be easily read and parsed.



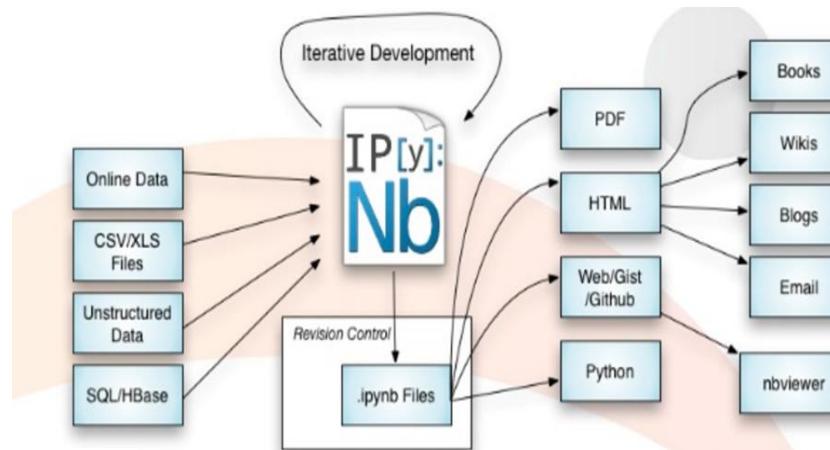
A screenshot of a Jupyter Notebook interface titled "jupyter Code\_Notes (autosaved)". The notebook shows a single code cell with the following content:

```
In [1]: import this
```

The Zen of Python, by Tim Peters

Beautiful is better than ugly.  
Explicit is better than implicit.  
Simple is better than complex.  
Complex is better than complicated.  
Flat is better than nested.  
Sparse is better than dense.  
Readability counts.  
Special cases aren't special enough to break the rules.  
Although practicality beats purity.  
Errors should never pass silently.  
Unless explicitly silenced.  
In the face of ambiguity, refuse the temptation to guess.  
There should be one-- and preferably only one --obvious way to do it.  
Although that may not be obvious at first unless you're Dutch.  
Now is better than never.  
Although never is often better than "right" now.  
If the implementation is hard to explain, it's a bad idea.  
If the implementation is easy to explain, it may be a good idea.  
Namespaces are one honking great idea -- let's do more of those!

- Exploration & development: Intermediate steps are saved in a clean, well documented format
- Communication/Collaboration: sharing research with peers, collaborators, reviewers, public
- Publishing: It is simple and quick switch between the development & publishing stage



- Install Jupyter Notebook:
  - At a Command Prompt: Pip install Jupyter Notebook → Press Enter key
- Using Jupyter Notebook:
  - At a Command Prompt: Jupyter Notebook → Press Enter key
  - Folders and Files which are created by Jupyter Notebook, they are in install folder

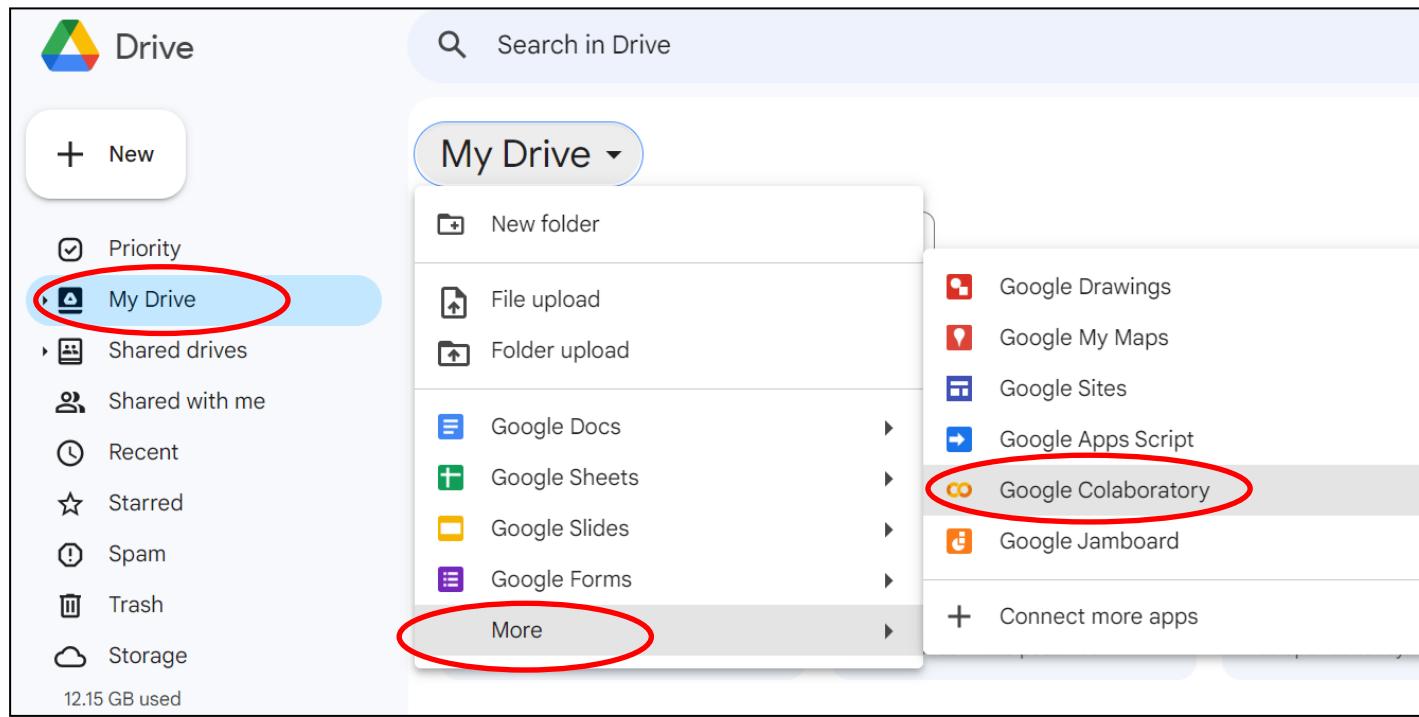
- just a cloud-based version of the Jupyter Notebook
- is a product from Google Research allowing anybody to write and execute arbitrary python code through the browser
- a hosted Jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs.

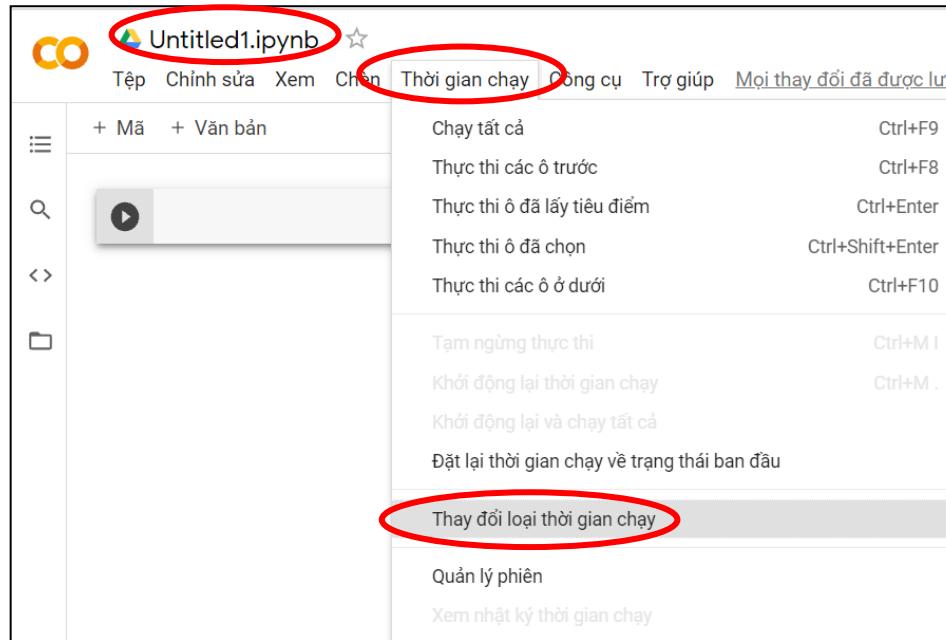
<https://colab.research.google.com/>



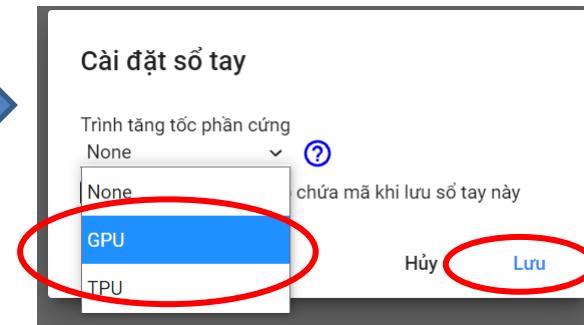
- Collaboratory also known as Google Colab is a production of Google Research
- Allows running code which is written by python via Web Browser
- Provide computer resources as Virtual Machine → GPU or TPU
- For the free service, there will be limited usage time, up to 12 hours → Therefore, combine Google Drive to save resources

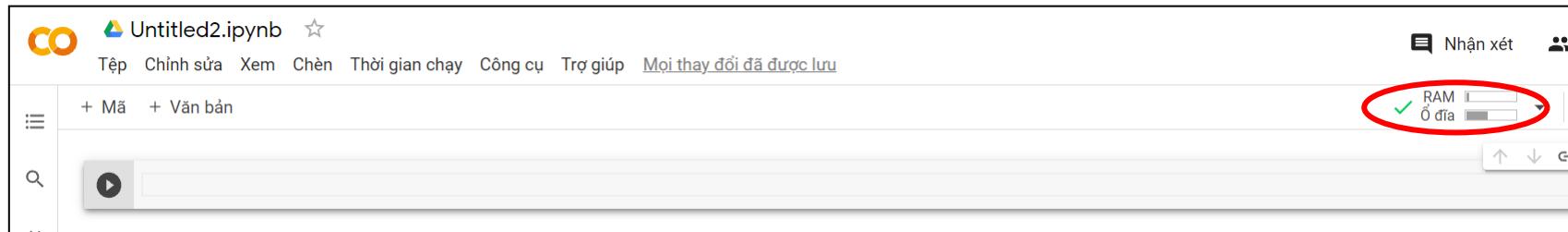
- Step 1: Log in Drive Google
- Step 2: My Drive → More → Google Colaboratory





- Step 3: Rename for file
- Step 4: Runtime → Change a type of Runtime → Choice CPU, GPU or TPU → Click Save button
- Step 5: Click Connect





- Step 6: Connect Colab to Drive Google → use 02 comand

A screenshot of a Google Colab notebook titled "Untitled2.ipynb". The code cell contains the following Python code:

```
from google.colab import drive  
drive.mount('/content/gdrive')
```

A red arrow points from the text "Step 6" in the previous slide to the play button icon in the toolbar above the code cell.

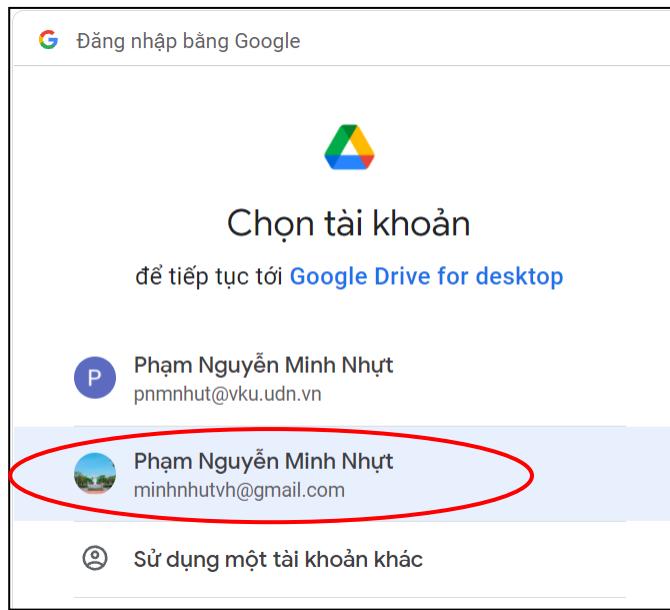
- Step 7: Click Triangle button → Click link

A screenshot of a Google Colab notebook titled "Untitled2.ipynb". The code cell contains the same Python code as the previous screenshot:

```
from google.colab import drive  
drive.mount('/content/gdrive')
```

Below the code cell, there is a message: "Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com](https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com)". A red arrow points from the text "Step 7" in the previous slide to this URL link.

- Step 8: Log in mail google



Đảm bảo rằng bạn đã tải ứng dụng này xuống từ Google

minhnhtv@gmail.com

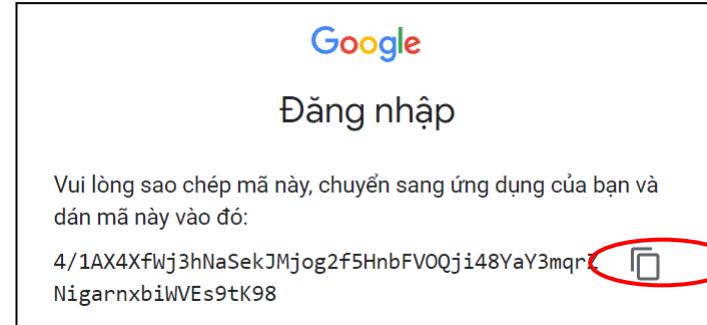
Đừng đăng nhập vào [Google Drive for desktop](#) trừ khi bạn chắc chắn mình đã tải ứng dụng này xuống từ Google.

Nếu bạn đã tải [Google Drive for desktop](#) xuống từ Google, ứng dụng này có thể yêu cầu bạn đăng nhập lại ngay bây giờ vì ứng dụng này đã được cập nhật gần đây.

Nếu bạn không chắc mình có tải [Google Drive for desktop](#) xuống từ Google hay không, bạn nên xóa ứng dụng này.

Hủy Đăng nhập

- Step 9: Copy Code



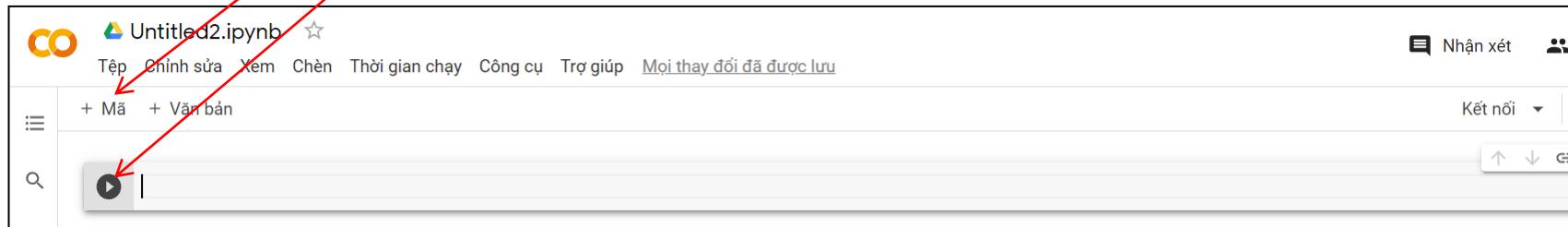
- Step 10: Paste into Enter your authorization Code → Press enter key

The image shows the Google Colab interface. In the code editor, the following Python code is run:

```
from google.colab import drive
drive.mount('/content/gdrive')
```

Below the code, a message says: "Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect\\_uri=https://colab.research.google.com/notebooks/api/oauthcallback](https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=https://colab.research.google.com/notebooks/api/oauthcallback)". An input field labeled "Enter your authorization code:" is present, with a red box highlighting the text input area.

- Add Code: Click Code button
- Run: Click Triangle button



- Into a folder: `%cd "Path/folder"`
- Install library : `!pip install <library>`
- Run file python: `!python <file.py>`
- Note: If we run a file python with parameter, we run same as at terminal of PC and add (!) at front of the command

---

## PRACTICE

Install Python, Jupyter Notebook and use Google Colab. Some basic Python exercises.