



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

- 1. What is Python?
- 2. Why Python?
- 3. Brief Development History of Python
- 4. Python IDEs and Code Editors
- 5. Jupyter Notebook
- 6. Examples

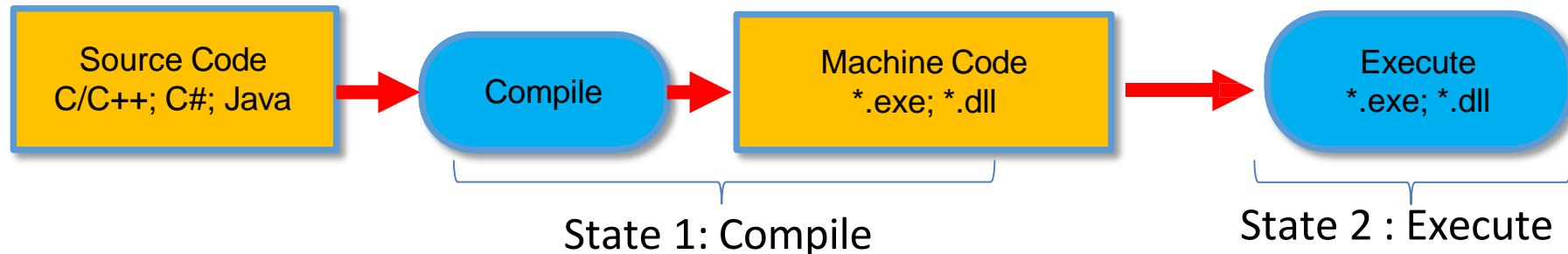
- Python is ...
  - a **scripting language** characterized by
    - **Interpreted** - is processed at runtime by the interpreter.



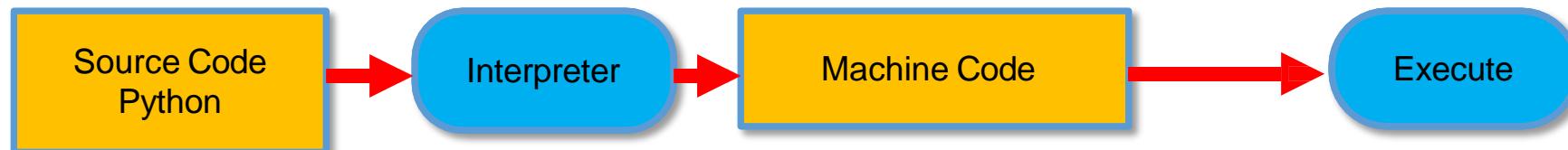
- **Interactive** – uses prompts and interact with the interpreter directly to write programs.
- **Object-Oriented** – supports Object-Oriented style or technique of programming that encapsulates code within objects

- 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

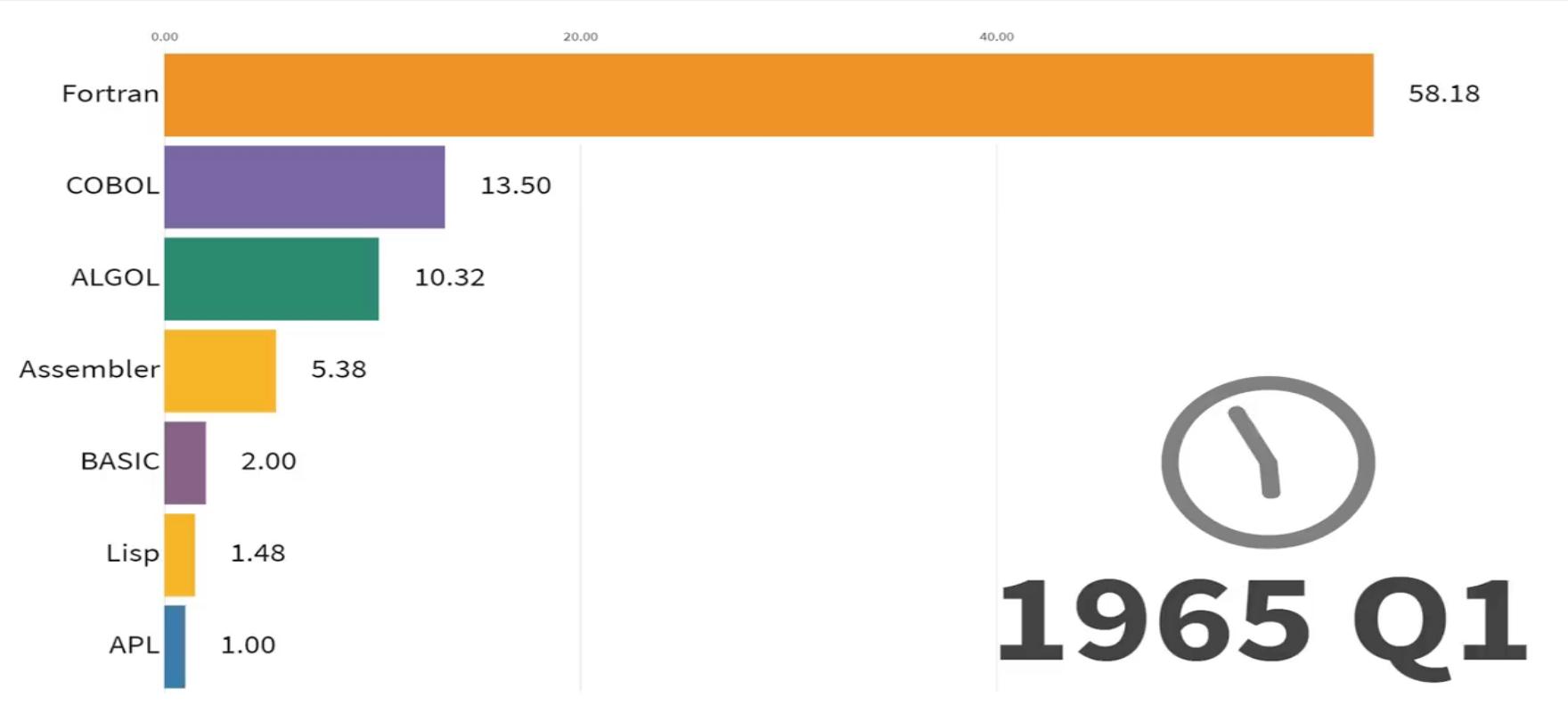


- Interpret and Execute



- 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).

- Commonly used languages (from 1965 to 2019)



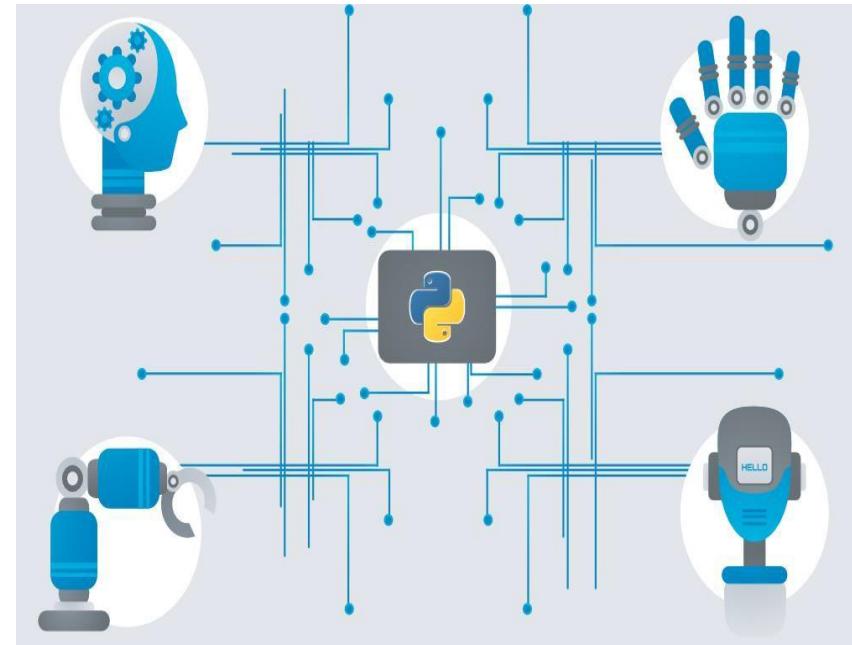
- Python is easier and simpler to learn and use than other languages.
  - A dynamically typed language, i.e. need not declare the type of variables, declare the class like some other languages
  - simple syntaxes, i.e. need not use any opening/closing braces, semicolons, etc.

Java	Python
<pre>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);     } }</pre>	<pre>a=10 b=20 print ('The result a+b = ', a+b)</pre>

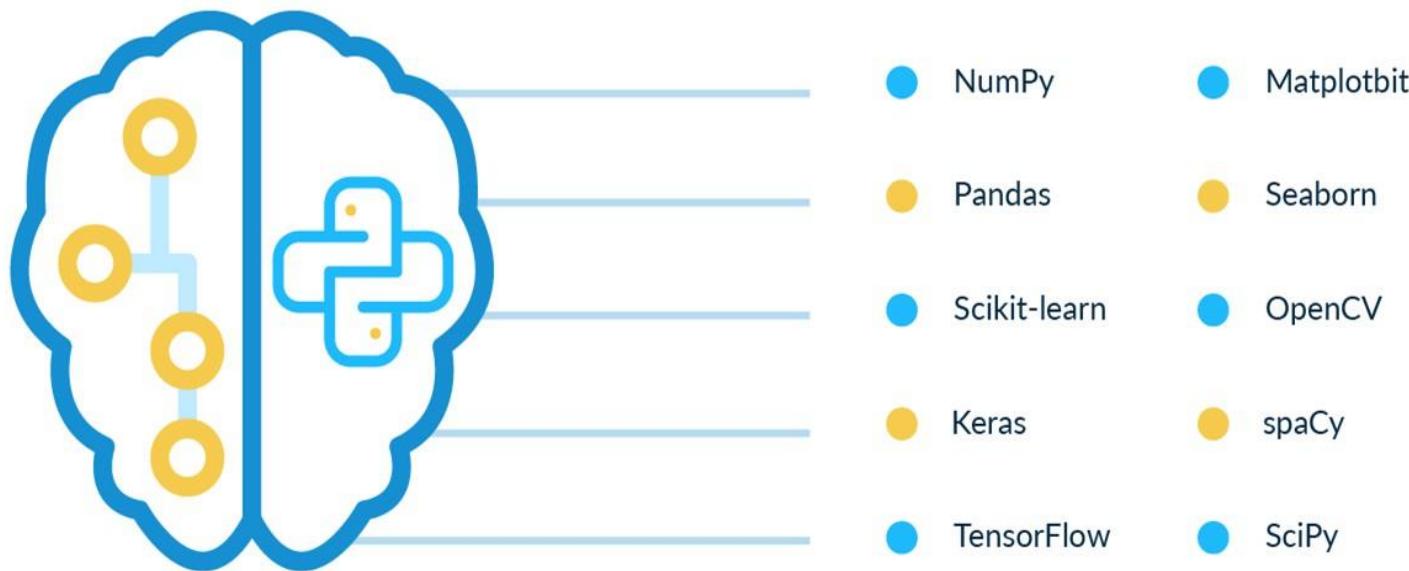
- Python is open source! Free!  
Massive online support from many resources and quality documentation worldwide.
- Python projects can be integrated with other systems coded in different programming languages.



- 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



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



## 2. Why Python?



Confidence Comes Standard.



Massachusetts  
Institute  
of  
Technology



LexisNexis®



mozilla  
Firefox

Google™



TRUECar.  
Know the Real Price™



AstraZeneca



INDUSTRIAL  
LIGHT & MAGIC  
A LUCASFILM COMPANY

Source: logicfinder

### 3. Brief Development History of Python

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

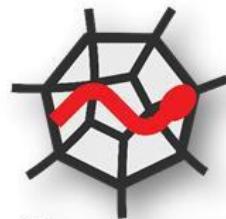


Version	Release time
Python 1.0 (First standard release)	1994
Python 1.6 (Final version 1.x)	2000
Python 2.0	10/2000
Python 2.7 (Final version 2.x)	2010
Python 3.0	2008
Python 3.7.3/3.7.16	3/2019
Python 3.9.6	29/6/2021
Python 3.11.4	Currently (17/8/2023)

## 4. Python IDEs & Code Editors



IDLE



SPYDER



PyCharm



Sublime Text



Atom



jupyter



Visual Studio Code



**Thonny**  
Python IDE for beginners

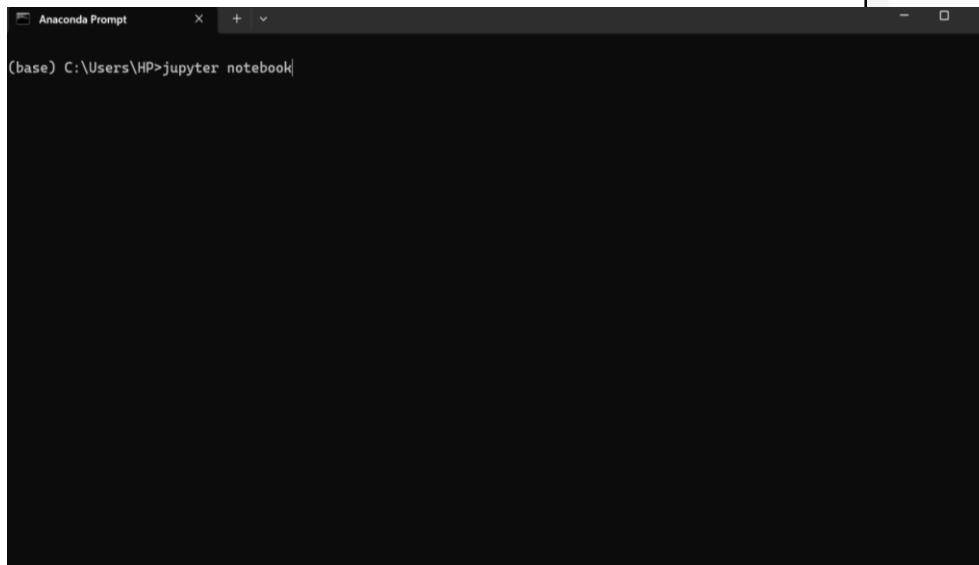
.....

- Jupyter Notebook is an open-source web app for creating and sharing documents with live code, equations, visuals, and text.
- Jupyter refers to its core languages: Julia, Python, and R.
- Supports interactive data science and scientific computing in 40+ languages.
- Documents: Serves as a Coding Diary, combining narrative text with machine-readable code in a metadata-rich, easily readable structure.

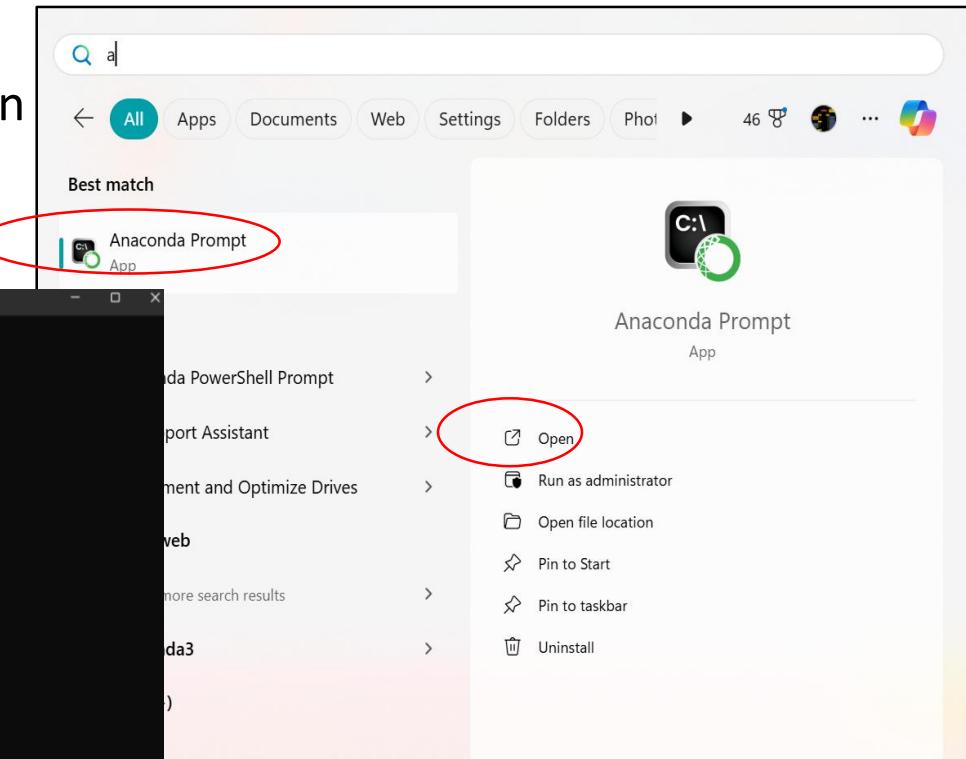


- Installing Jupyter using Anaconda
  - Download [Anaconda](https://www.anaconda.com/download) (<https://www.anaconda.com/download>). (currently Python 3.9).
  - Install the version of Anaconda which you downloaded, following the instructions on the download page.

- To run the notebook:
  - Start → Anacoda prompt → Open
  - jupyter notebook → enter

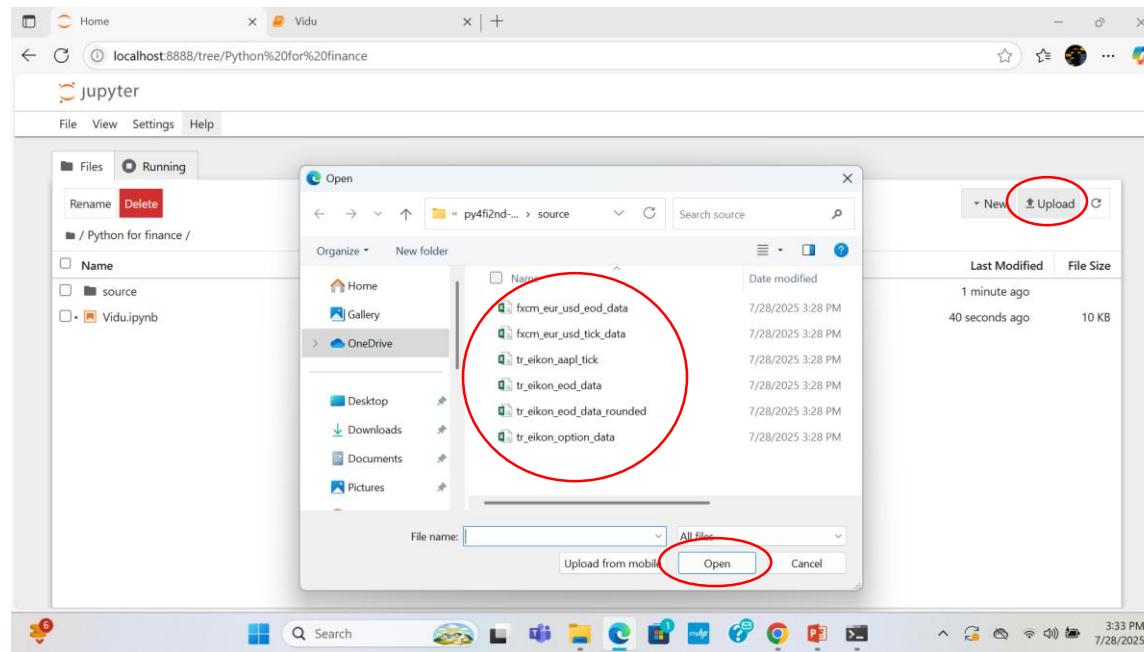


```
(base) C:\Users\HP>jupyter notebook
```

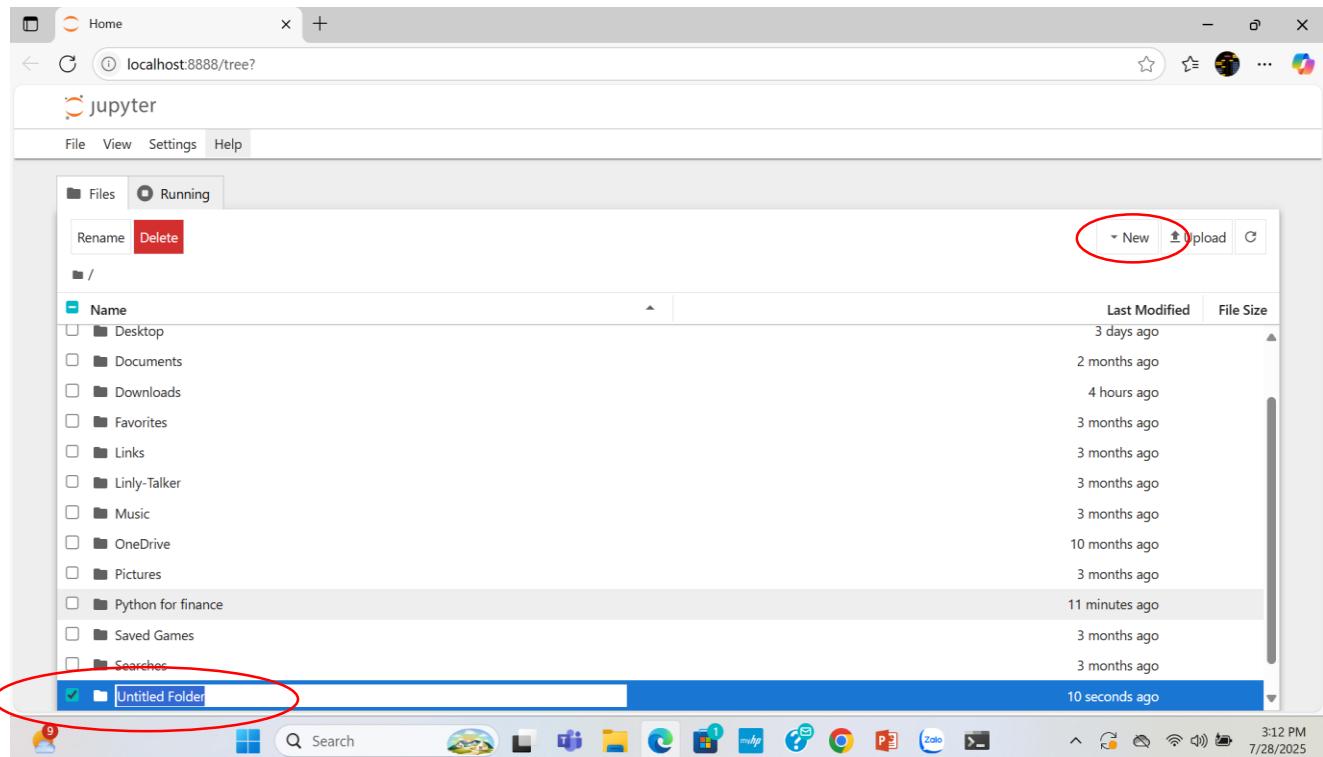


- Upload file

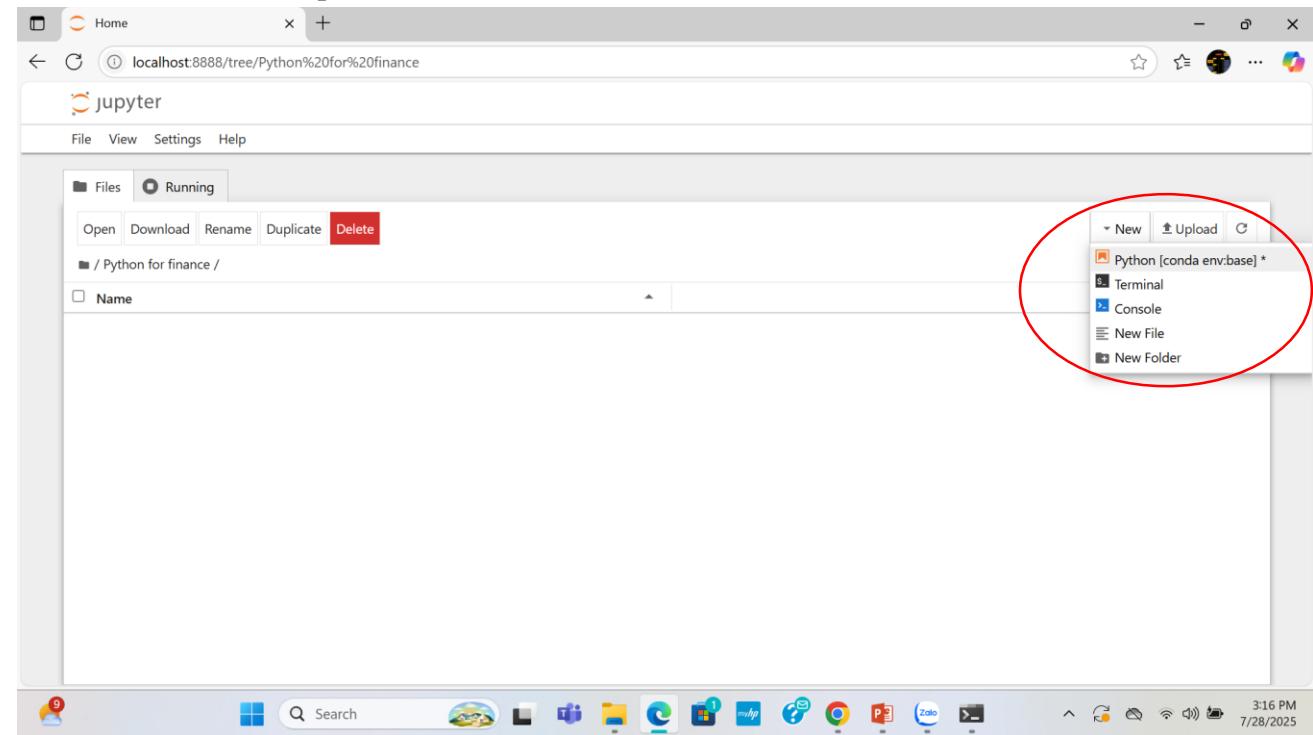
- Double Click folder to open folder
- Upload → Choice file → Open



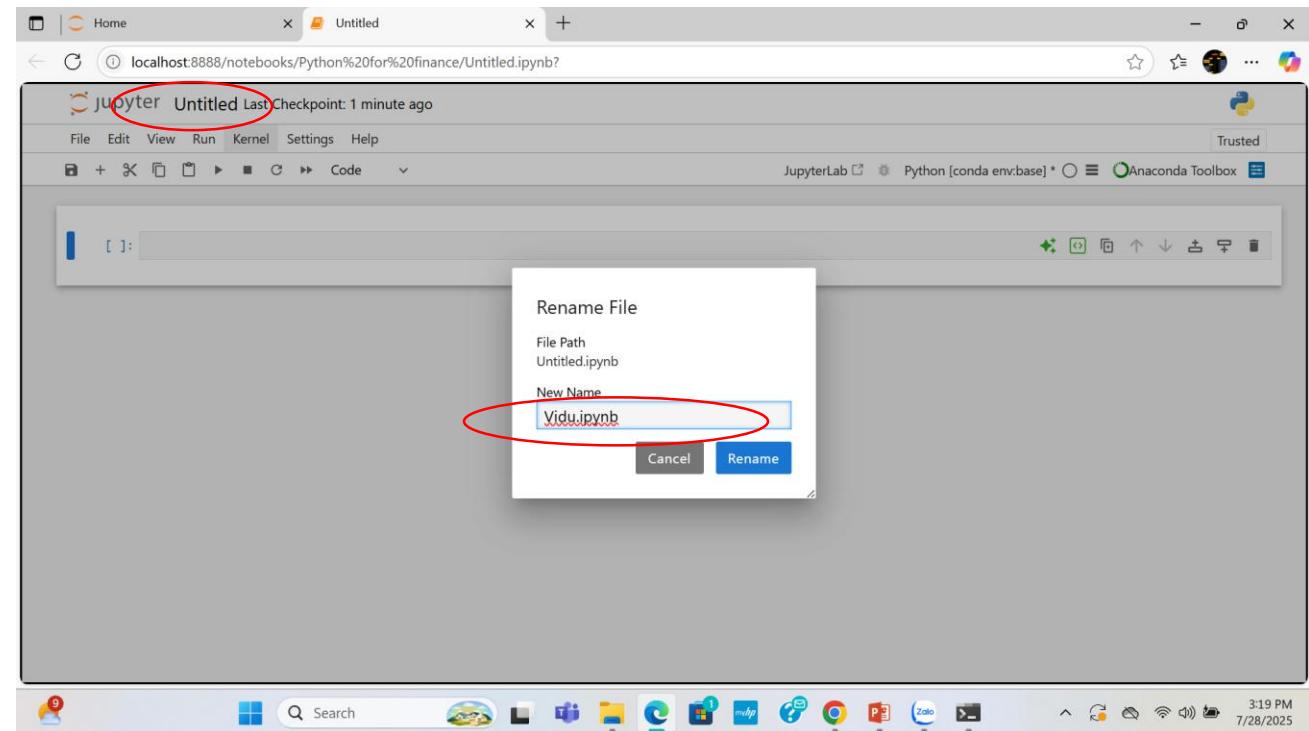
- Create folder
  - New → Folder
  - Enter folder name



- Create notebook file
  - New → Python [conda env: base]

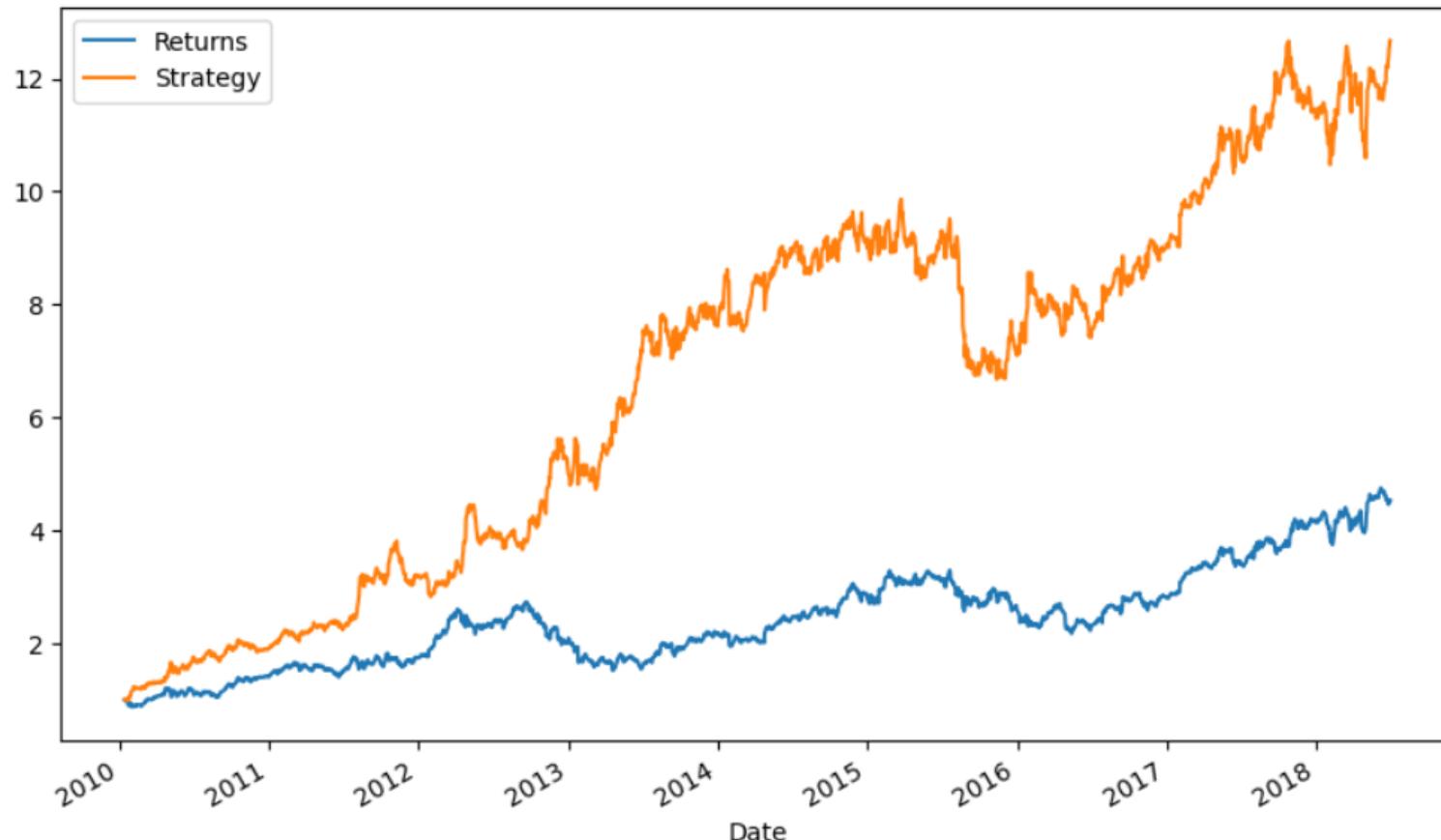


- Rename Filename
  - Click Untitled → Enter Filename → Rename



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Date	AAPL.O	MSFT.O	INTC.O	AMZN.O	GS.N	SPY	.SPX	.VIX	EUR=	XAU=	GDX	GLD	
2	1/1/2010									1.4323	1096.35			
3	1/4/2010	30.57283	30.95	20.88	133.9	173.08	113.33	1132.99	20.04	1.4411	1120	47.71	109.8	
4	1/5/2010	30.62568	30.96	20.87	134.69	176.14	113.63	1136.52	19.35	1.4368	1118.65	48.17	109.7	
5	1/6/2010	30.13854	30.77	20.8	132.25	174.26	113.71	1137.14	19.16	1.4412	1138.5	49.34	111.51	
6	1/7/2010	30.08283	30.452	20.6	130	177.67	114.19	1141.69	19.06	1.4318	1131.9	49.1	110.82	
7	1/8/2010	30.28283	30.66	20.83	133.52	174.31	114.57	1144.98	18.13	1.4412	1136.1	49.84	111.37	
8	1/11/2010	30.01568	30.27	20.95	130.308	171.56	114.73	1146.98	17.55	1.4513	1152.6	50.17	112.85	
9	1/12/2010	29.67426	30.07	20.608	127.35	167.82	113.66	1136.22	18.25	1.4494	1127.3	48.35	110.49	
10	1/13/2010	30.09283	30.35	20.96	129.11	169.07	114.62	1145.68	17.85	1.451	1138.4	48.86	111.54	
11	1/14/2010	29.91854	30.96	21.48	127.35	168.53	114.93	1148.46	17.63	1.4502	1142.85	48.6	112.03	
12	1/15/2010	29.41854	30.86	20.8	127.14	165.21	113.64	1136.03	17.91	1.4382	1129.9	47.42	110.86	
13	1/18/2010									1.4385	1132.5			
14	1/19/2010	30.71997	31.1	21.05	127.61	166.86	115.06	1150.23	17.58	1.4298	1137.7	47.69	111.52	
15	1/20/2010	30.2464	30.585	21.08	125.78	167.79	113.89	1138.04	18.68	1.4101	1111.3	45.73	108.94	
16	1/21/2010	29.72454	30.01	20.84	126.62	160.87	111.7	1116.48	22.27	1.409	1094.75	43.75	107.37	
17	1/22/2010	28.24997	28.96	19.91	121.43	154.12	109.21	1091.76	27.31	1.4137	1092.6	43.79	107.17	
18	1/25/2010	29.01069	29.32	20.32	120.31	154.98	109.77	1096.78	25.41	1.415	1097.85	43.19	107.48	
19	1/26/2010	29.41997	29.5	20	119.48	150.88	109.31	1092.17	24.55	1.4073	1097.65	43.13	107.56	
20	1/27/2010	29.69768	29.67	20.24	122.75	151.5	109.83	1097.5	23.14	1.4017	1086.35	42.77	106.528	
21	1/28/2010	28.46997	29.16	19.92	126.03	153.29	108.57	1084.53	23.73	1.3962	1085.9	42.17	106.48	
22	1/29/2010	27.43754	28.18	19.4	125.41	148.72	107.39	1073.87	24.62	1.3862	1081.05	40.72	105.96	

```
import numpy as np
import pandas as pd
data = pd.read_csv('source/tr_eikon_eod_data.csv',
                    index_col=0, parse_dates=True)
data = pd.DataFrame(data['AAPL.O'])
data['Returns'] = np.log(data / data.shift()) data.dropna(inplace=True)
lags = 6
cols = []
for lag in range(1, lags + 1):
    col = 'lag_{}'.format(lag)
    data[col] = np.sign(data['Returns'].shift(lag))
    cols.append(col) data.dropna(inplace=True)
from sklearn.svm import SVC
model = SVC(gamma='auto')
model.fit(data[cols], np.sign(data['Returns']))
data['Prediction'] = model.predict(data[cols])
data['Strategy'] = data['Prediction'] * data['Returns']
data[['Returns', 'Strategy']].cumsum().apply(np.exp).plot(figsize=(10, 6))
```



---

## PRACTICE

Install Python, Jupyter Notebook. Some  
basic Python exercises.