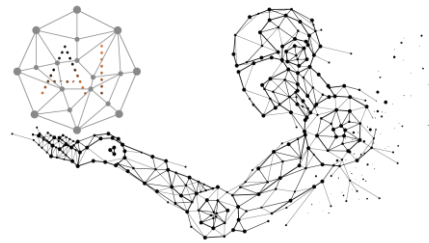
Applied Machine Learning

Chapter 1- Introduction to Python

دانشگاه تربیت مدرس
Tarbiat Modares University

Hossein Homaei
Department of Electrical & Computer Engineering

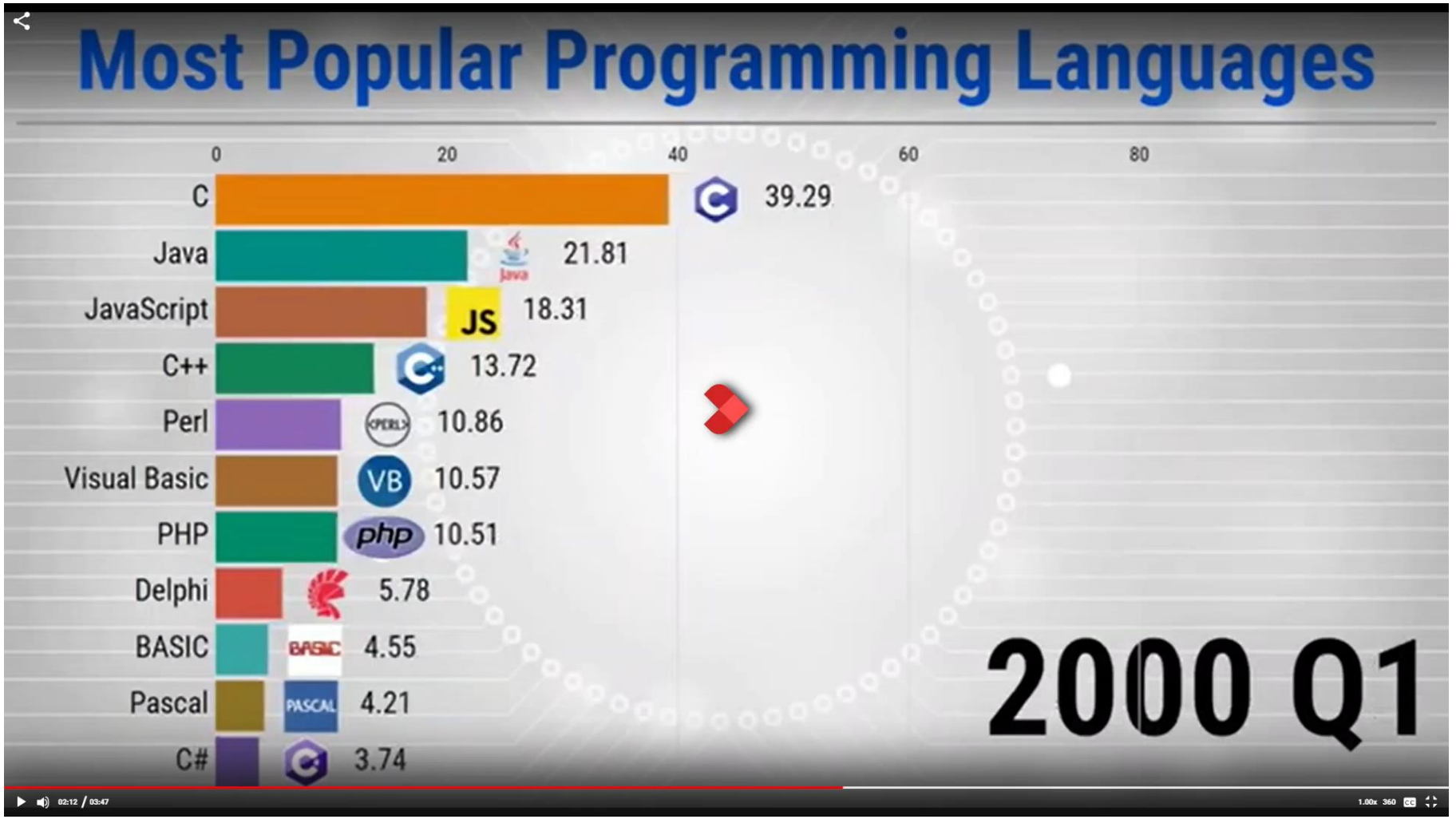


What is Python?

- High-level Interpreted programming language
- Created in 1991 by Guido van Rossum
- Supports
  - Object-oriented programming
- Dynamic typing
- Multipurpose
  - Backend web apps
  - Desktop apps
  - Games
  - Task automation
  - Machine learning and Data science
  - …

"I was looking for a hobby programming project to keep me occupied during a slow work week around Christmas. I used my home computer to write an interpreter for a new scripting language, which would become known as Python."
Guido van Rossum

Why Python?


Most Popular Programming Languages
0
20
40
60
80
C
39.29
Java
21.81
JavaScript
18.31
C++
13.72
Perl
10.86
Visual Basic
10.57
PHP
10.51
Delphi
5.78
BASIC
4.55
Pascal
4.21
C#
3.74
2000 Q1
02:12 / 03:47
1.00x 360

...Why Python?

- Well-connected and supportive community.
- Efficient language
  - Do more in fewer lines of code
  - Write clean code (because of Python's syntax)
  - Easy to read, debug, and extend programs
  - Like natural human-language
- Easy to learn yet powerful
- Comes with a large library of useful modules
  - Motto of the Python programming lanhuage: Batteries Included

```
print("Hello world.")
```

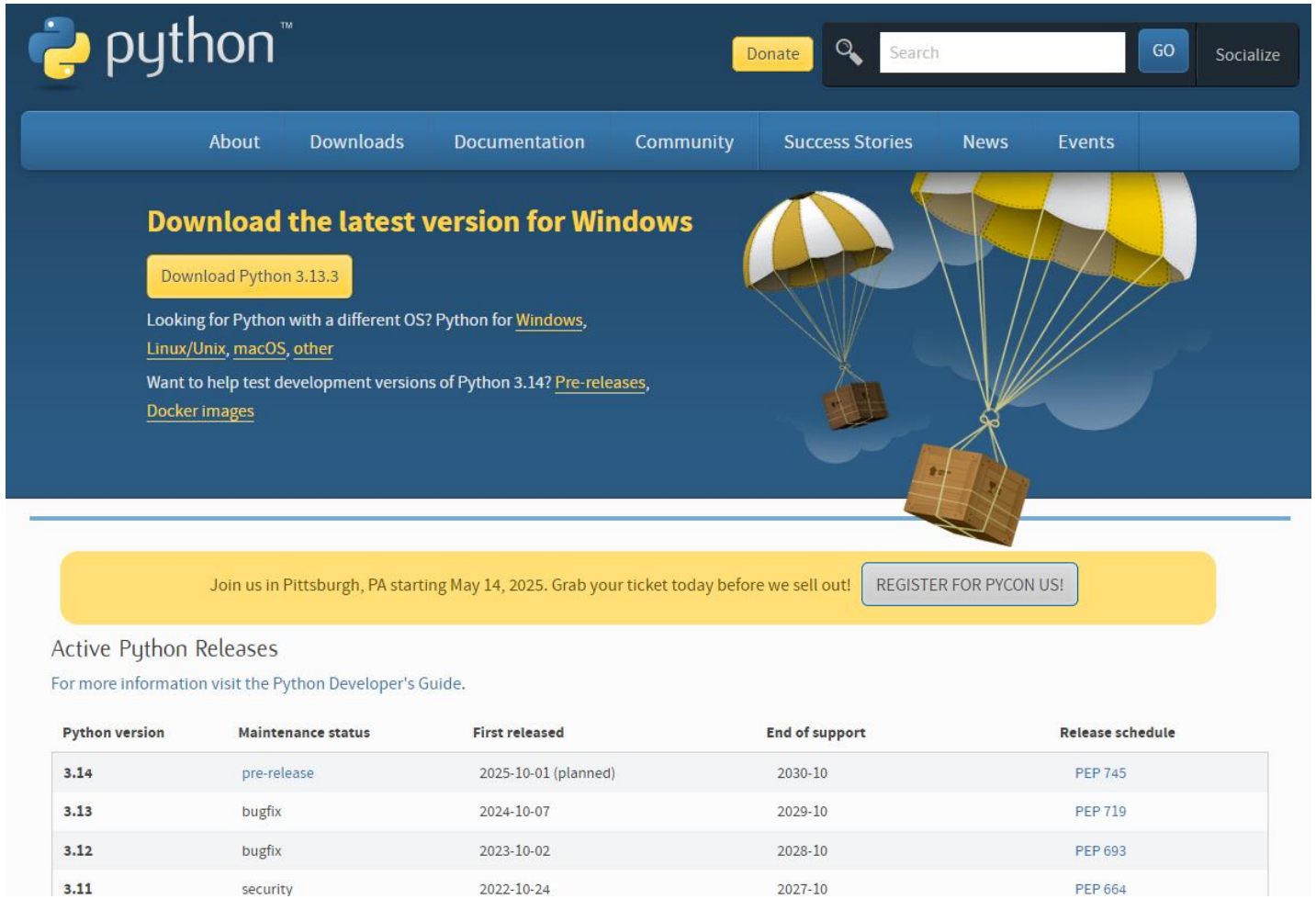
```
public class HelloWorld {
    public static void main (String[]args) {
      System.out.println.("Hello world");
    }
}
```

- Is used heavily in scientific fields for academic research and applied work.

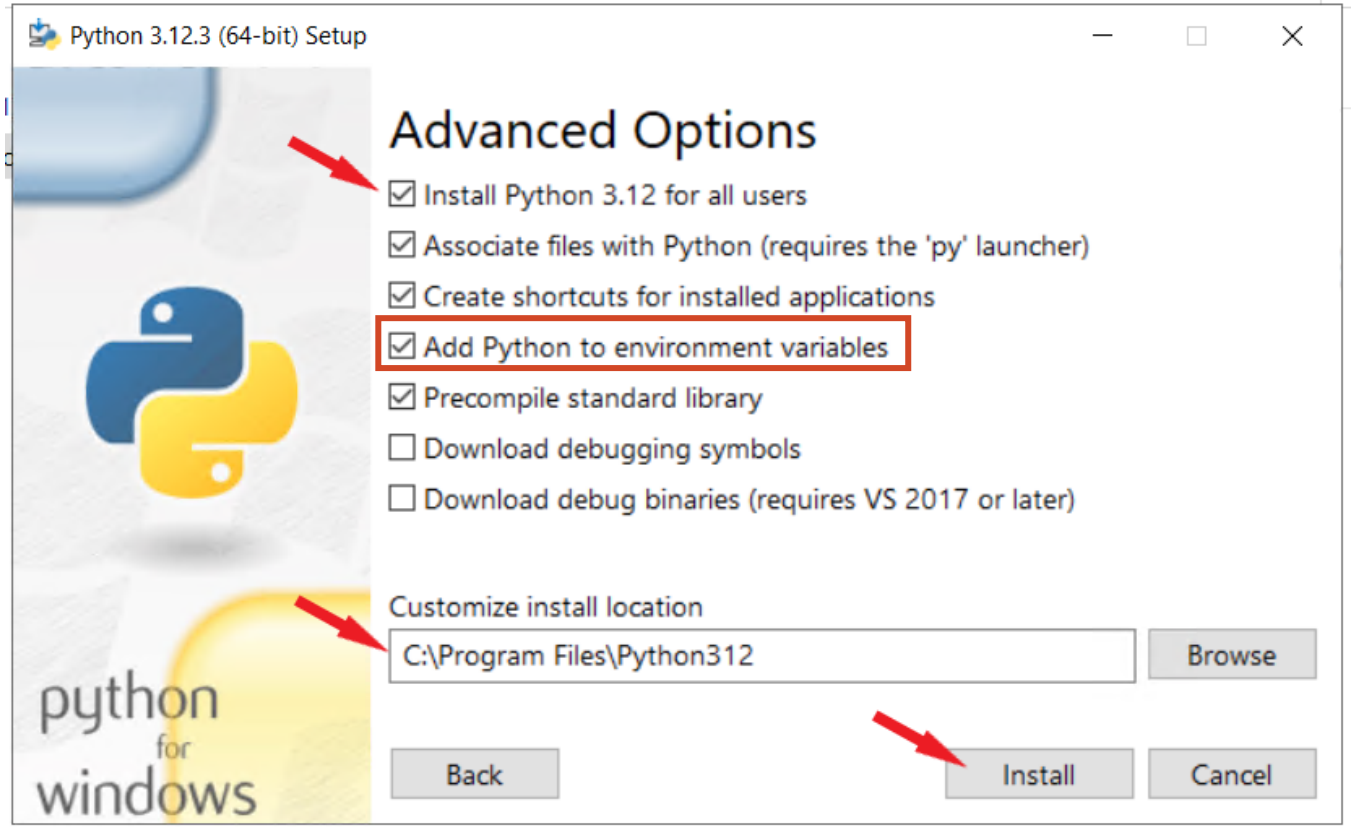
Set Up The Programming Environment

Install Python

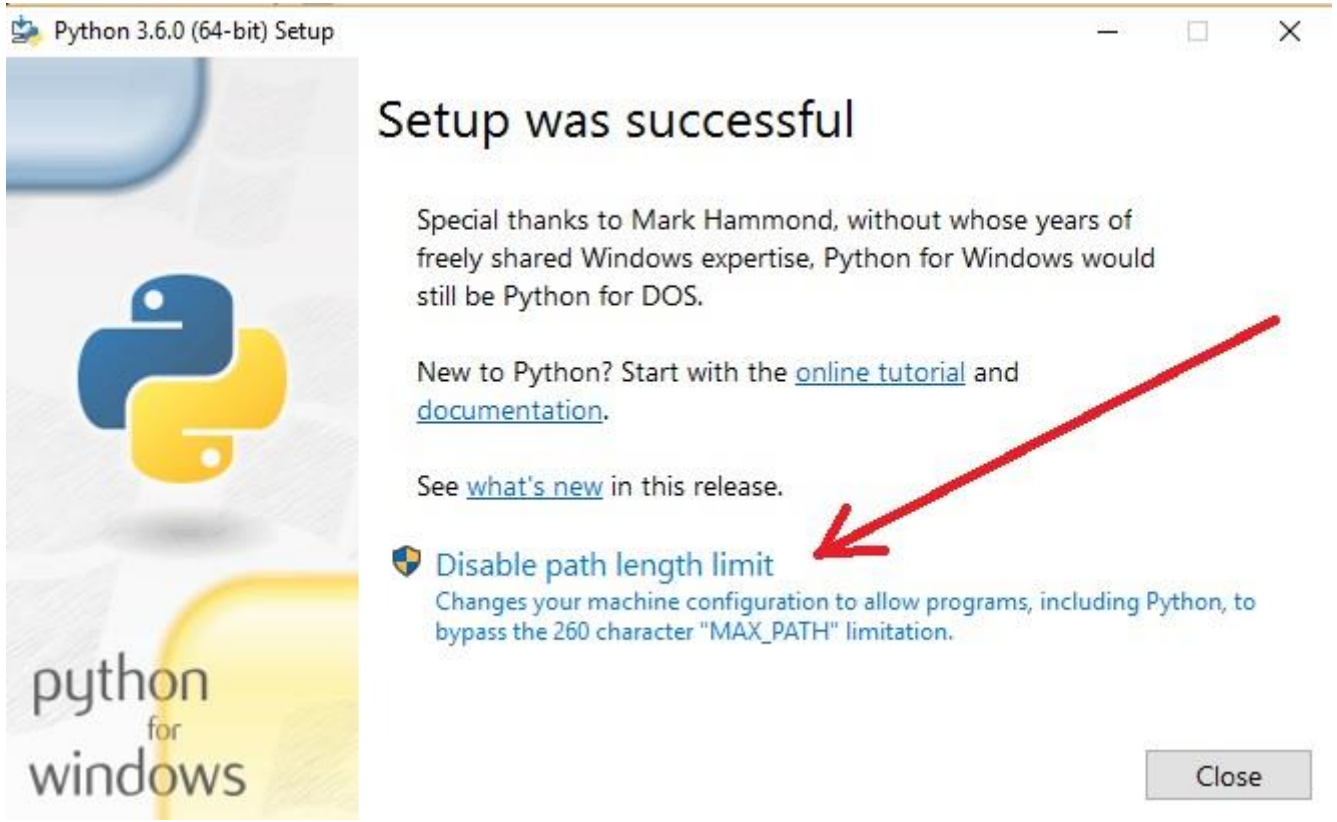
- https://www.python.org/downloads/



...Install Python



...Install Python



...Install Python

- Other options in Windows OS
  - Install Anaconda
    - Python interpreter + useful libraries and tools for data science
  - Use Windows Subsystem for Linux (WSL)
    - If you are working on Windows and want a Linux environment for working with Python

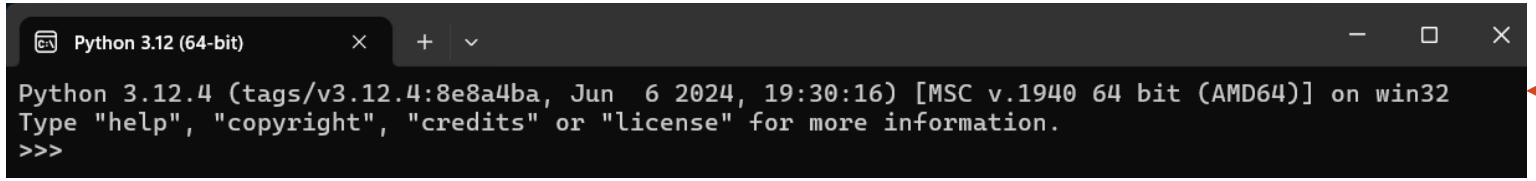
```
python3 --version
```

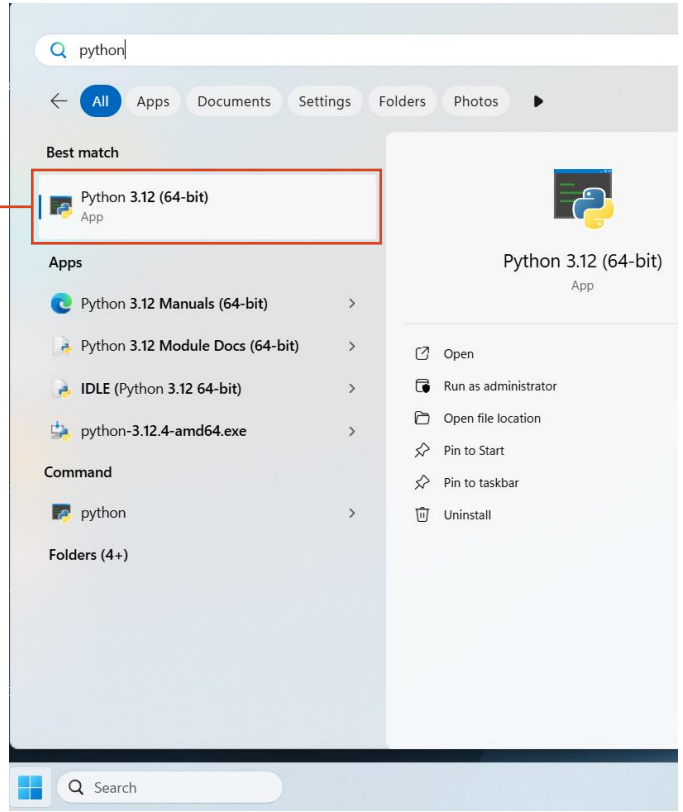
Test

Coding

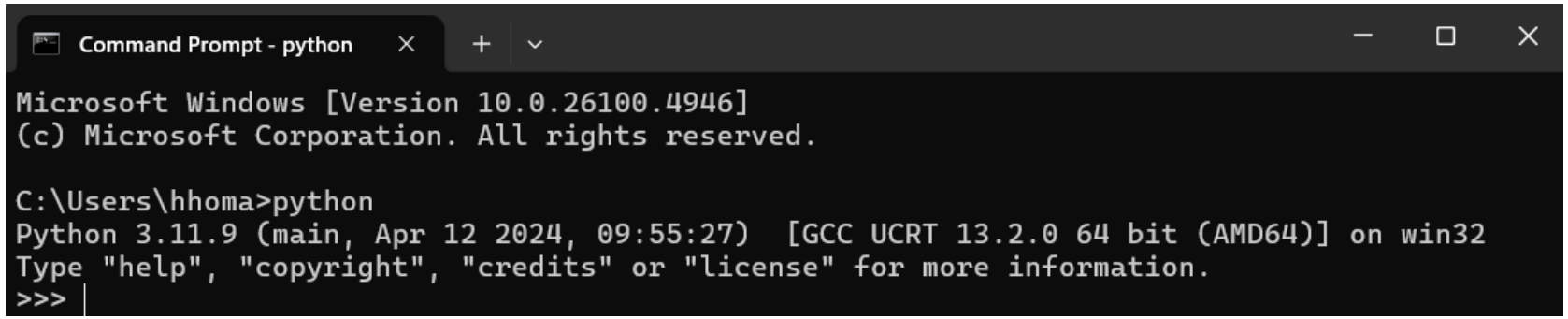
- You can start coding by the default Python REPL (Read-Eval-Print Loop)
  - Interactive programming environment
    - Write and execute code in a step-by-step manner and get immediate feedback

Run from Windows Strat Menu





Run from Command Prompt



...Coding

- Single line example

```
Python 3.12 (64-bit)
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello world!")
Hello world!
>>>
```

- Multi-line examples

Difficult! Isn't it?

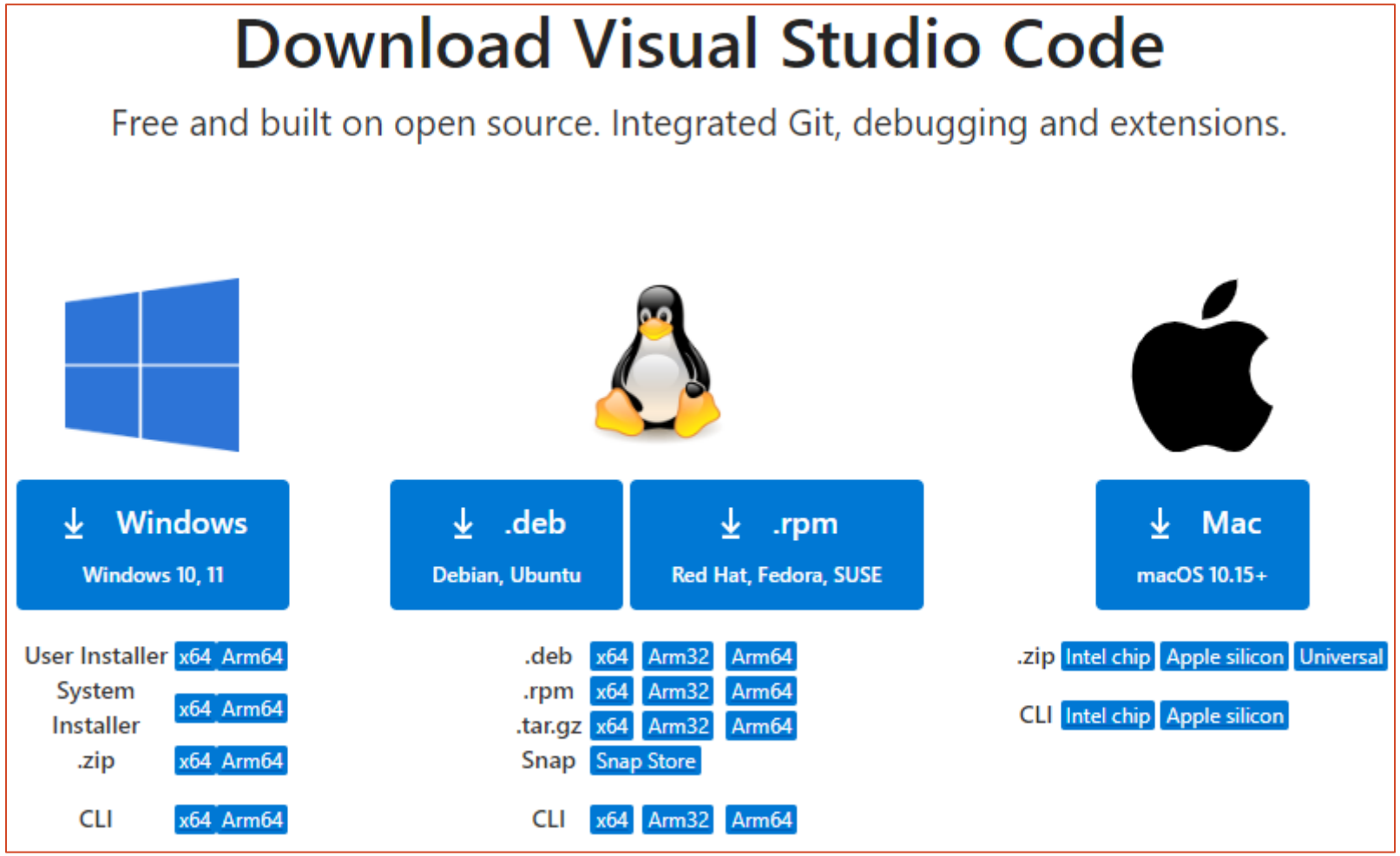
```
>>> print \
... ("My name is Hossein")
My name is Hossein
>>>
```

```
>>> x = "What is your name?"
>>> print(x)
What is your name?
>>>
```

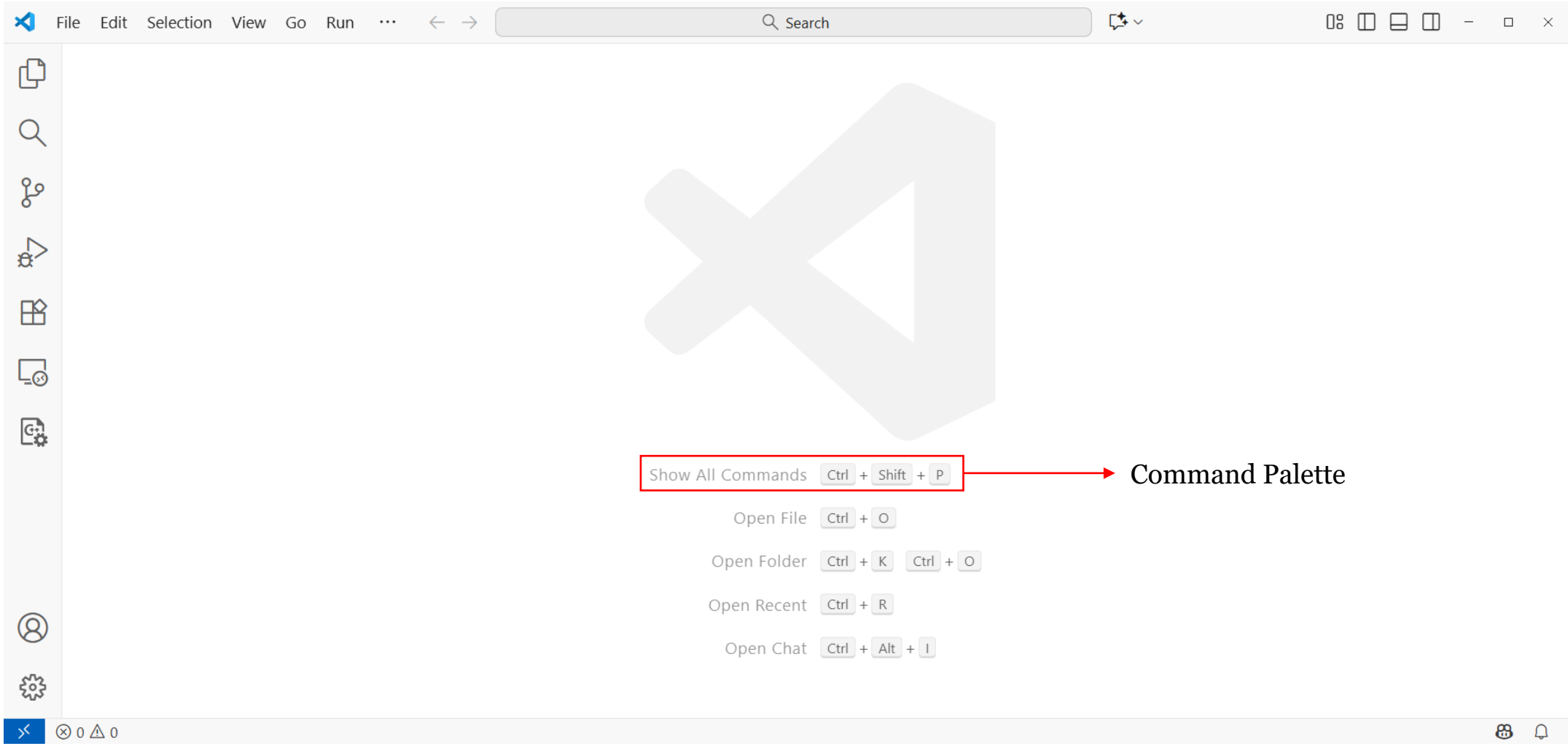
```
>>> print("Hello!"); print("How are you?")
Hello!
How are you?
```

```
>>> i = 1
>>> while i <= 5:
...     print(i)
...     i = i + 1
...
1
2
3
4
5
>>>
```

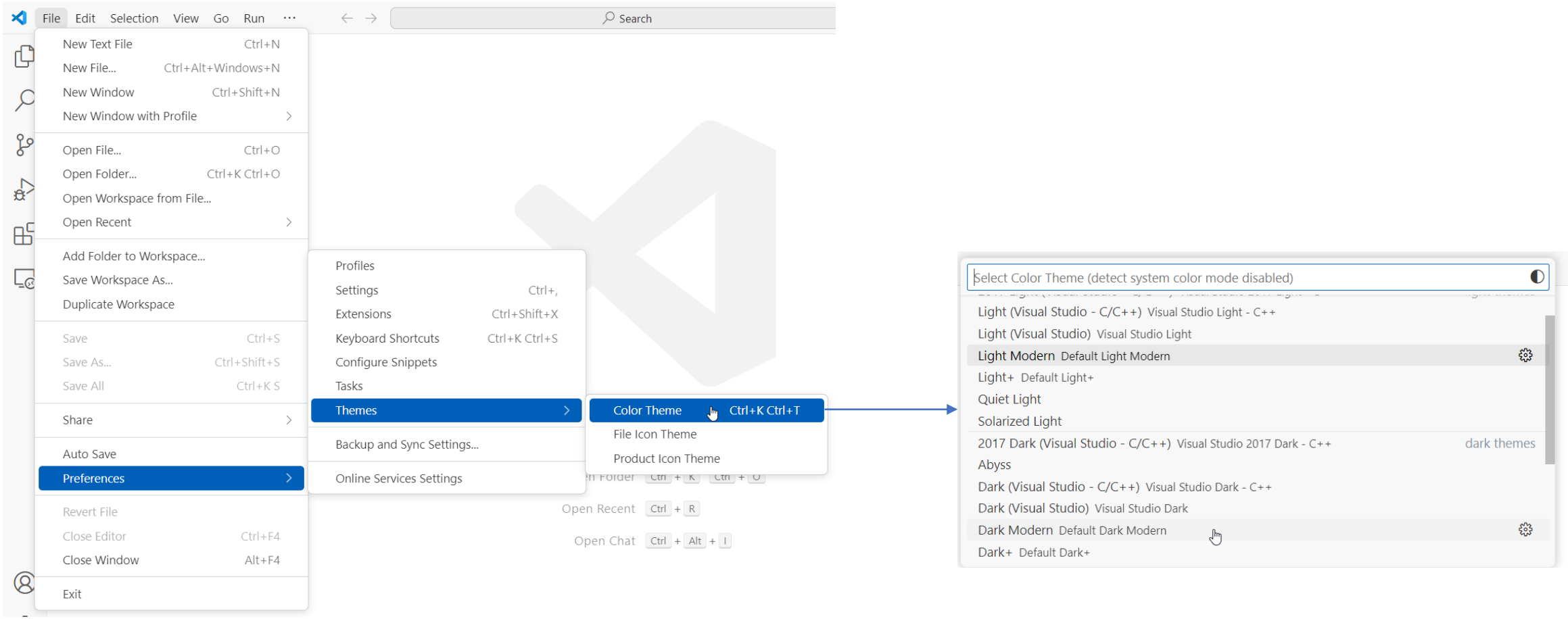
Install VS Code



VS Code Environment

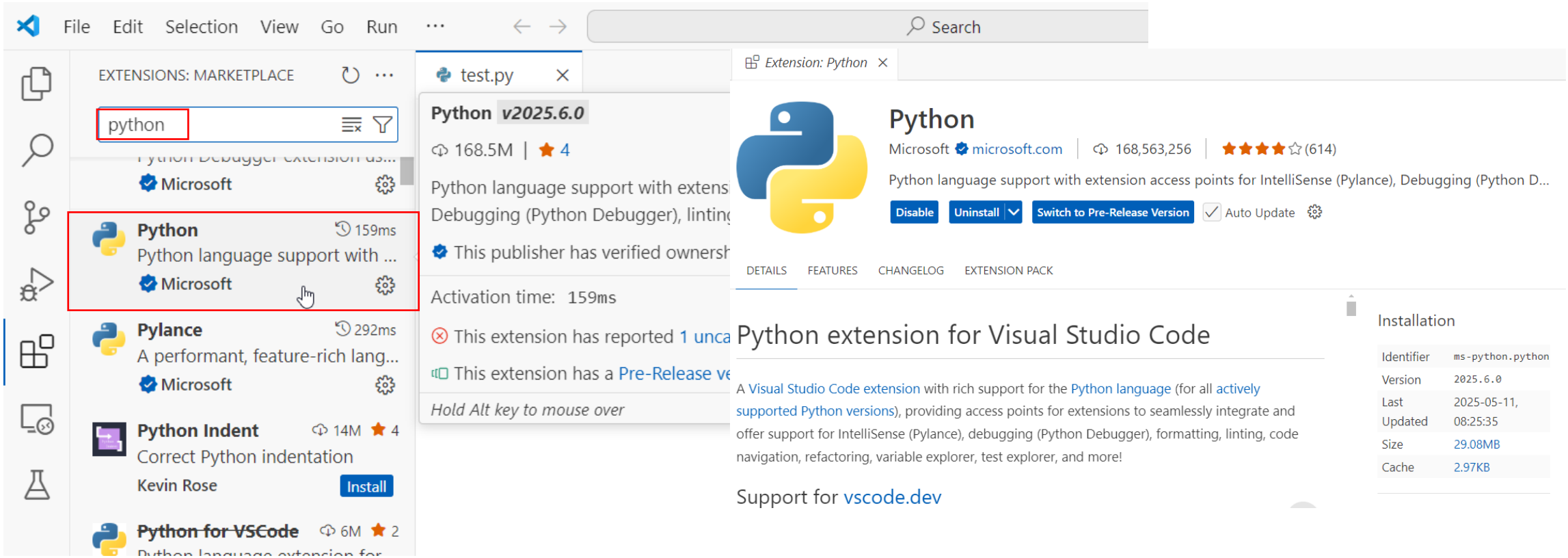


...VS Code Environment

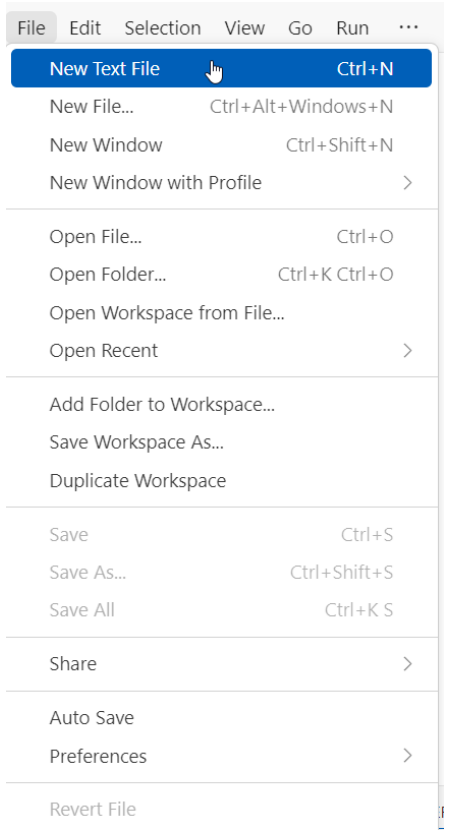


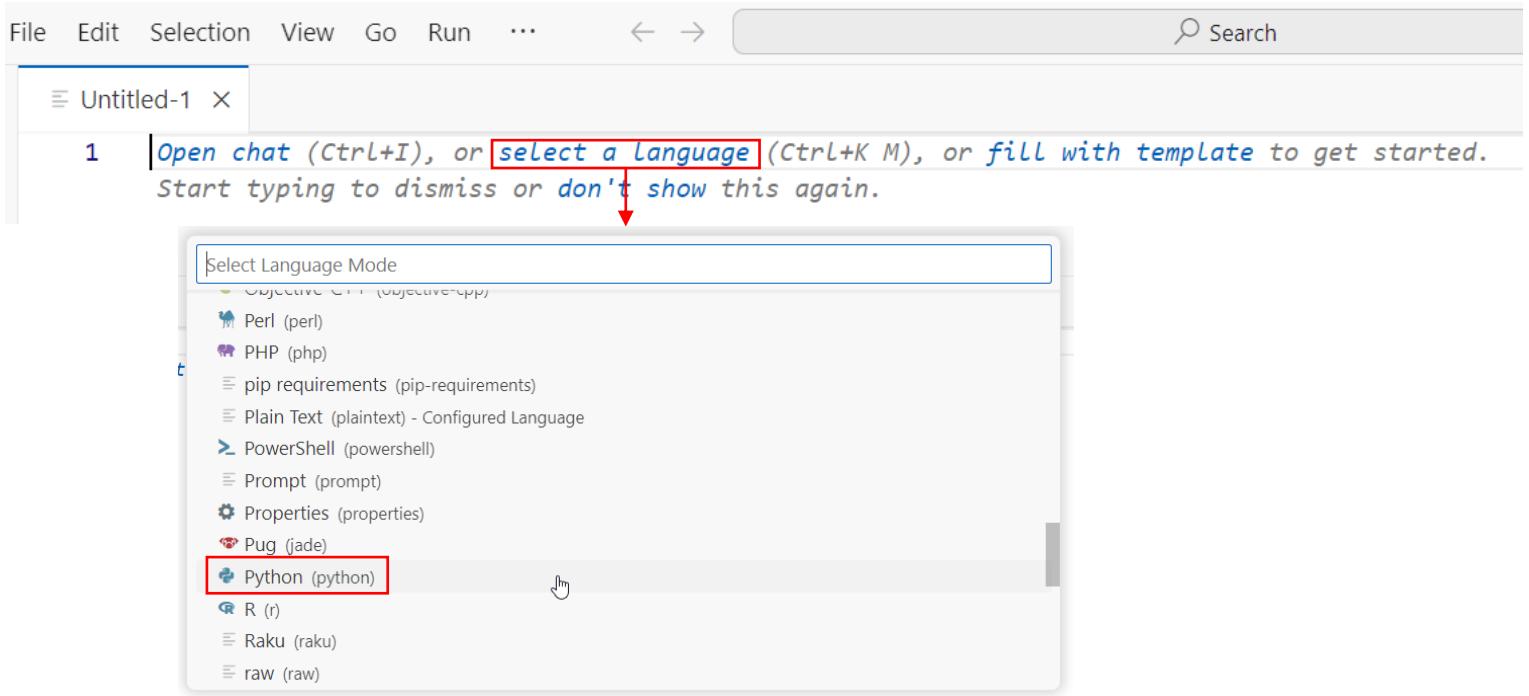
...VS Code Environment

- Install Python Extension on VS Code
  - Left side bar → Extensions → Search for Python

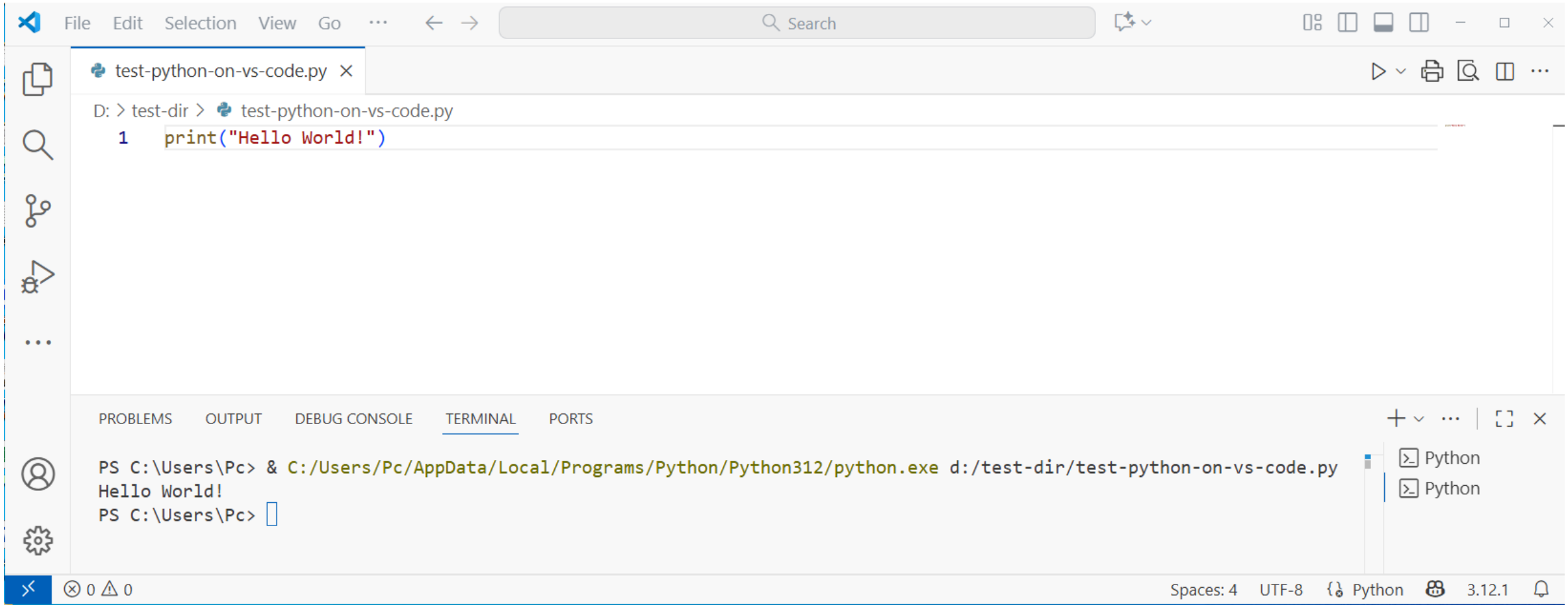


Testing Python on VS Code



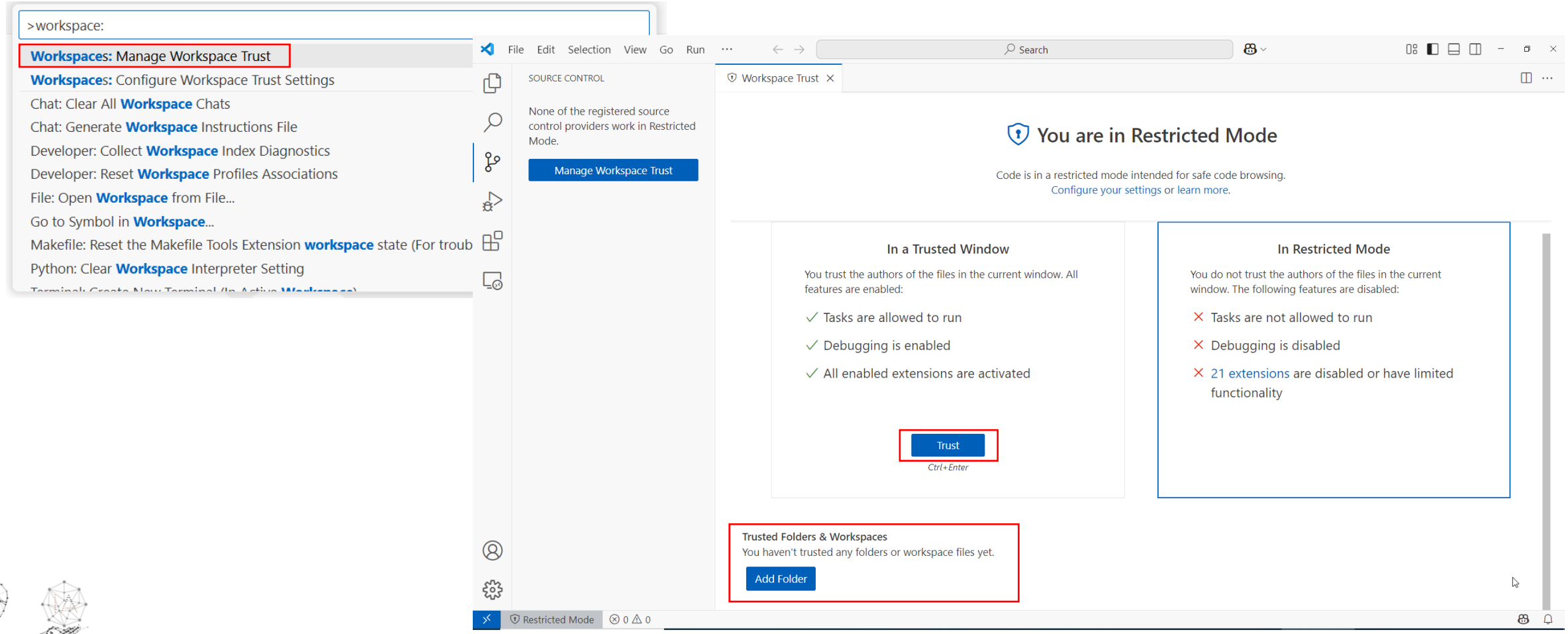


...Testing Python on VS Code



...Testing Python on VS Code

- It may require to be run from a trusted folder!



Main Python Libraries for Machine Learning

Array manipulation

Matrix manipulation

Series

Data-Frames

Drawing charts and plots

(Linear, Scatter and Pie plots, Bar Charts, Histograms, Boxplots, Heatmaps

Machine Learning Algorithms

Install the Required Libraries

- On the
  - Windows command prompt (cmd) or
  - command terminal of VS Code (ctrl+`)
- write `pip install` (or `python -m pip install`)
    - `numpy`
    - `pandas`
    - `scikit-learn`
    - `matplotlib`

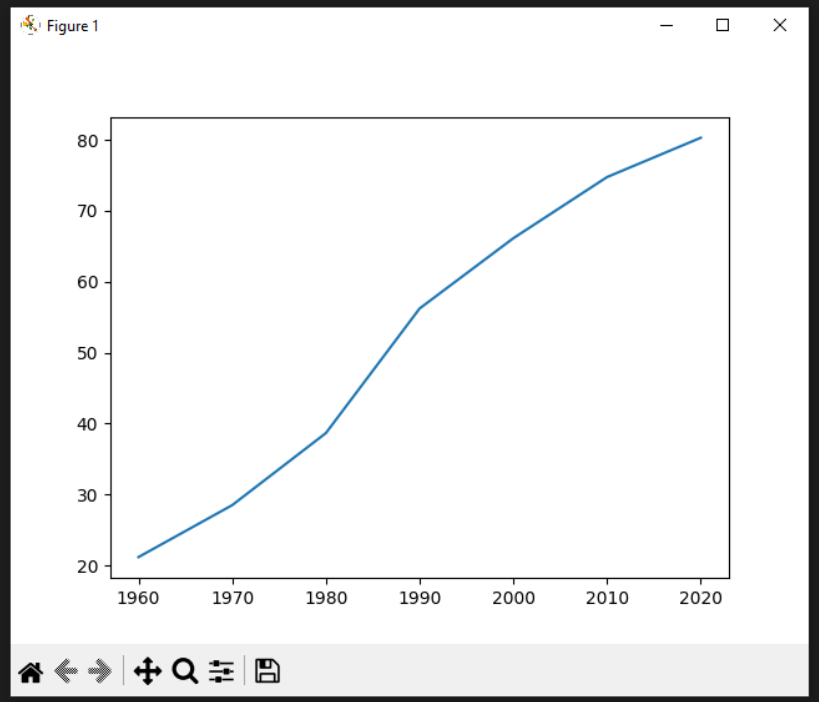
Troubleshooting

- NOTE-1: Sometimes previous versions work but the latest one does not.
  - Install an specific version:
    - `pip install “matplotlib=3.8.2”`

- NOTE-2: You may require GCC compiler on your system for working with Matplotlib
  - Follow the instruction explained by Microsoft: “C/C++ for Visual Studio Code”
    - Install MinGW-x64 on Windows: https://code.visualstudio.com/docs/languages/cpp

Test Libs

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn.datasets as skdb

myArray = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]])
print ("myArray =\n", myArray, "\ndata type: ", myArray.dtype)

myDataFrame = pd.DataFrame(myArray,
                           index=['row1', 'row2', 'row3', 'row4'],
                           columns = ['col1', 'col2', 'col3'])
print("myDataFrame:")
print(myDataFrame)

iris=skdb.load_iris()
print("Iris dataset was loaded")

years = [1960, 1970, 1980, 1990, 2000, 2010, 2020]
iranpop = [21.19, 28.51, 38.67, 56.23, 66.13, 74.75, 80.29]

plt.plot(years, iranpop)
plt.show()
```



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
myDataFrame:
      col1  col2  col3
row1     1     2     3
row2     4     5     6
row3     7     8     9
row4    10    11    12
Iris dataset was loaded
```

Jupyter Notebook

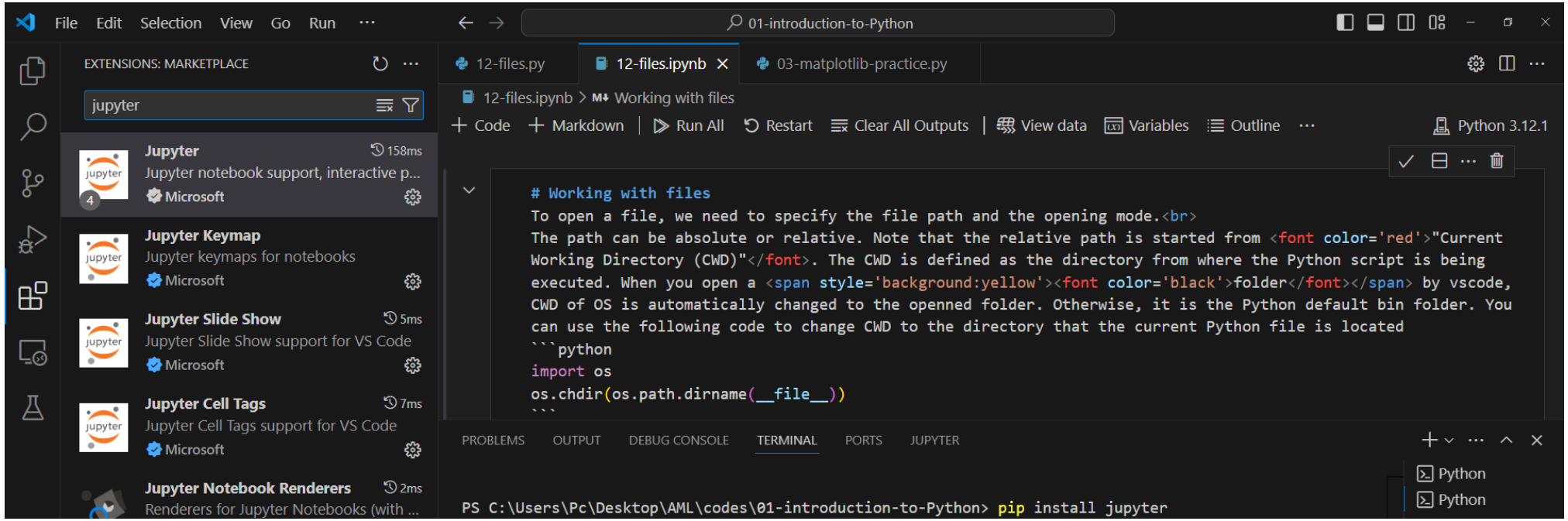
- Being familiar with Jupyter notebook is strongly recommended.
  - Why?
    - Many code examples of books and online courses at GitHub are written as Jupyter notebooks.

- What is it?
  - A powerful tool to create interactive documents containing text, images, and executable code snippets (Python in our case).
  - Jupyter's three core programming languages are Julia, Python and R.
  - Now, it supports various programming languages.

- Format of files = .ipynb
  - Why ipynb?
    - It was spun off from IPython project in 2014.

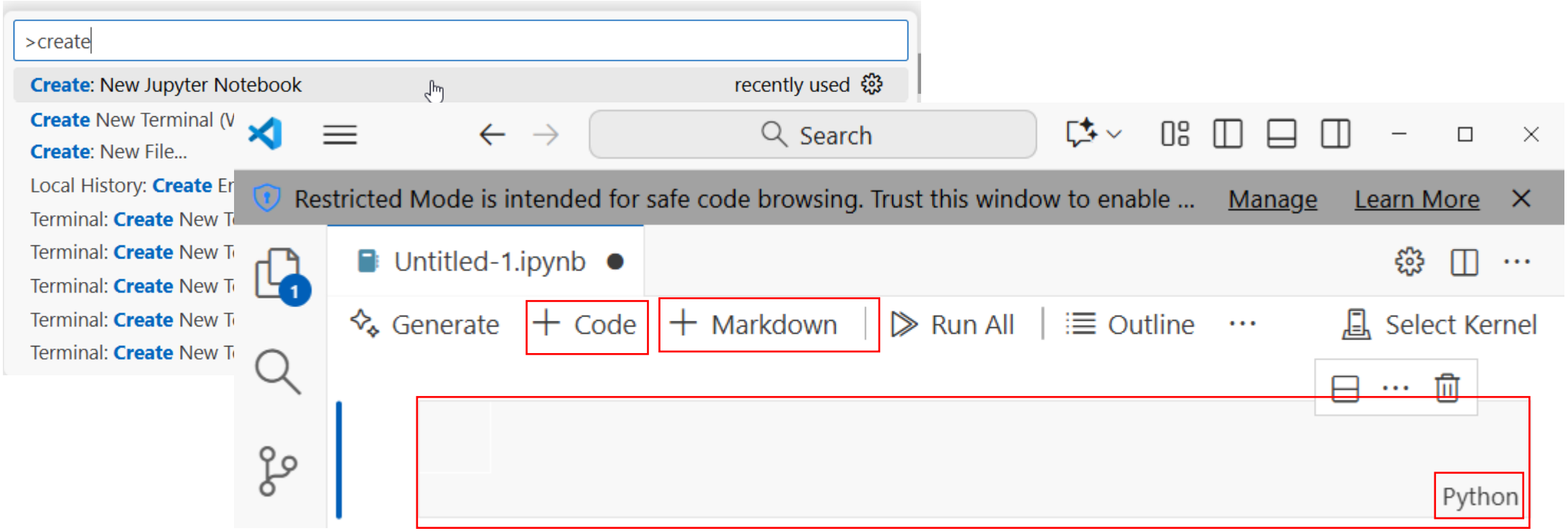
...Jupyter Notebook

- Install
  1. From the Command line: `pip install Jupyter`
  2. Install Jupyter Notebook extension on VS Code



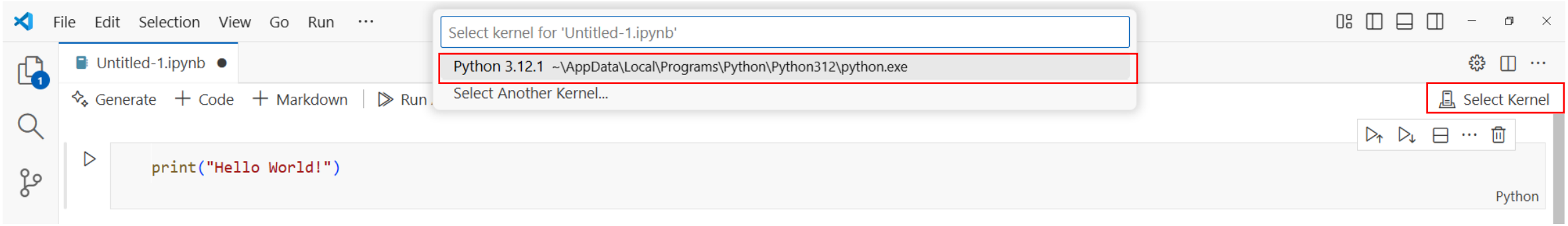
...Jupyter Notebook

- On VS Code

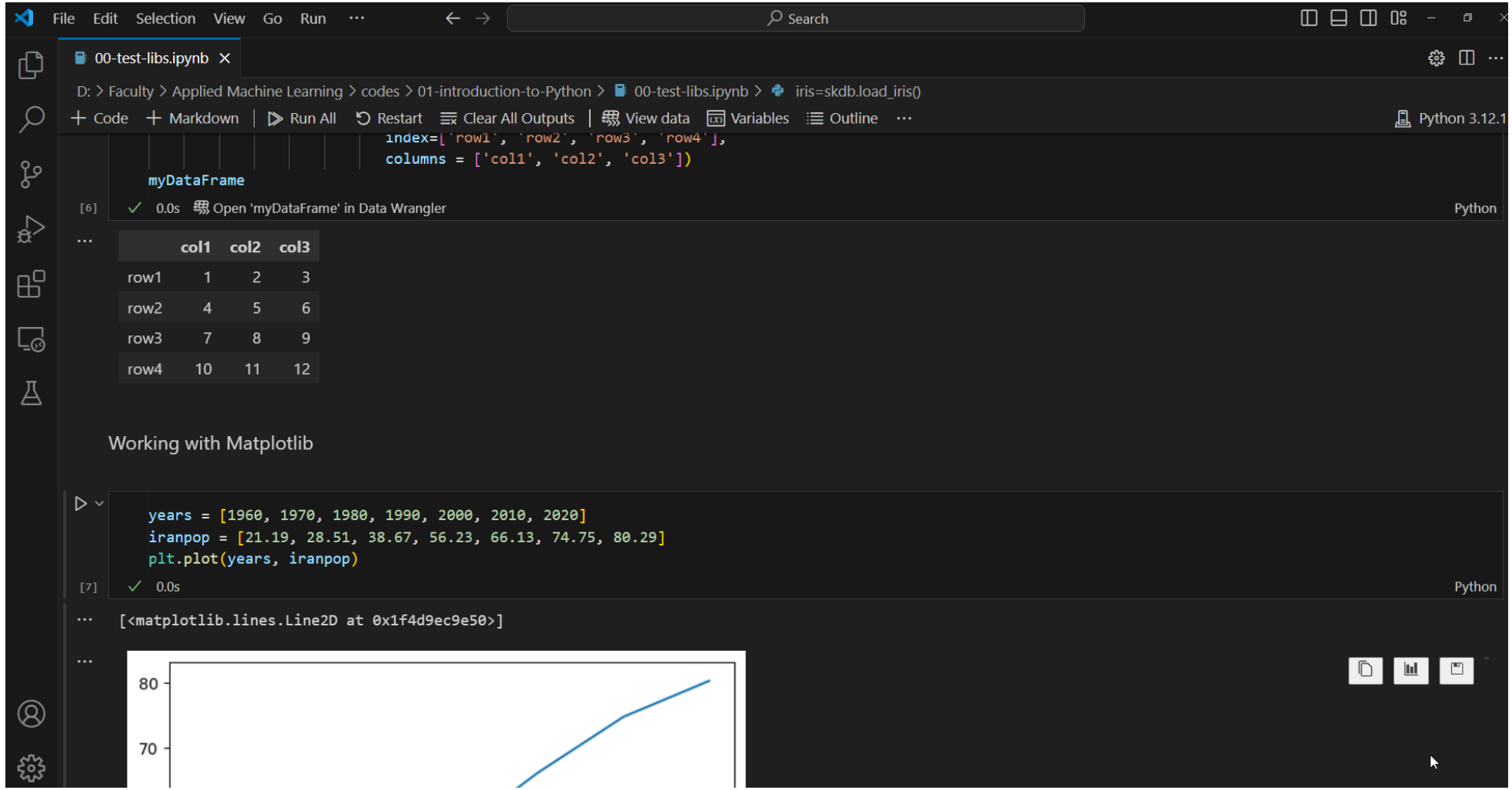


...Jupyter Notebook

- Example



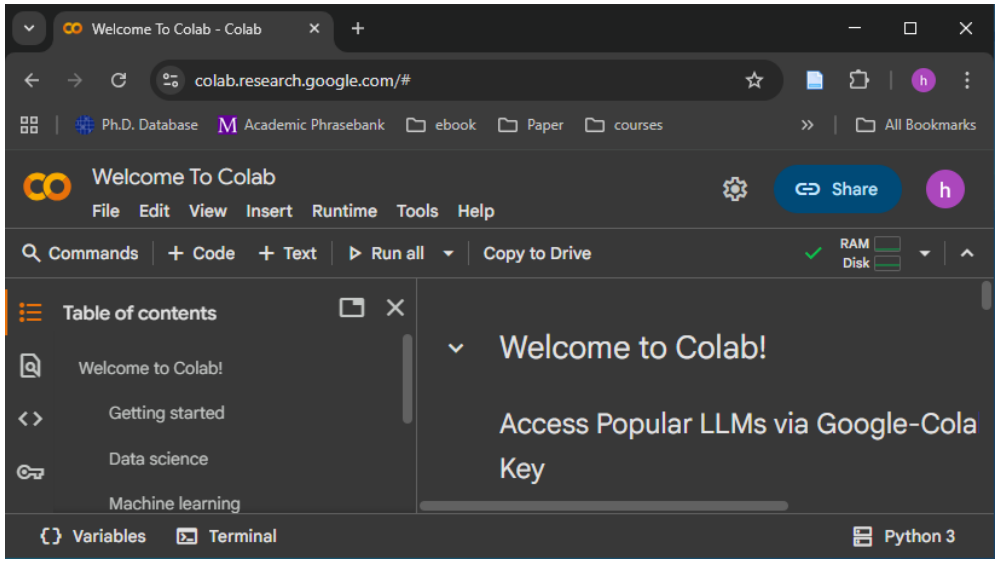
...Jupyter Notebook



...Jupyter Notebook

- Google Colab
  - Online Integrated Development Environment (IDE) for Python
  - No setup is required → Get rid of installation
  - Offers free access to computing resources, including GPUs and TPUs, making it popular among researchers and students working on deep learning and data science projects.
    - Access to GPUs is restricted and is NOT guaranteed
    - Notebooks can run for at most 12 hours

https://colab.research.google.com



Python Virtual Environment

- What is it?
  - A folder structure that gives you everything you need to run a lightweight yet isolated Python environment.
    - Created on top of an existing Python installation (base) and are isolated from the packages in the base environment.
  - When used from within a virtual environment, common installation tools such as pip, install Python packages into a virtual environment

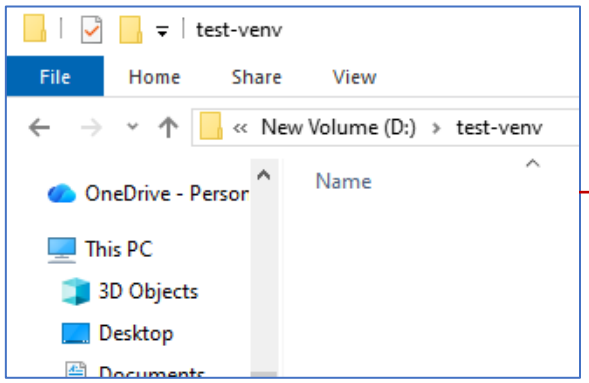
- Benefits
  - Manage dependencies separately for different projects
    - Minimize reproducibility issues
  - Preventing conflicts
  - Maintaining cleaner setups

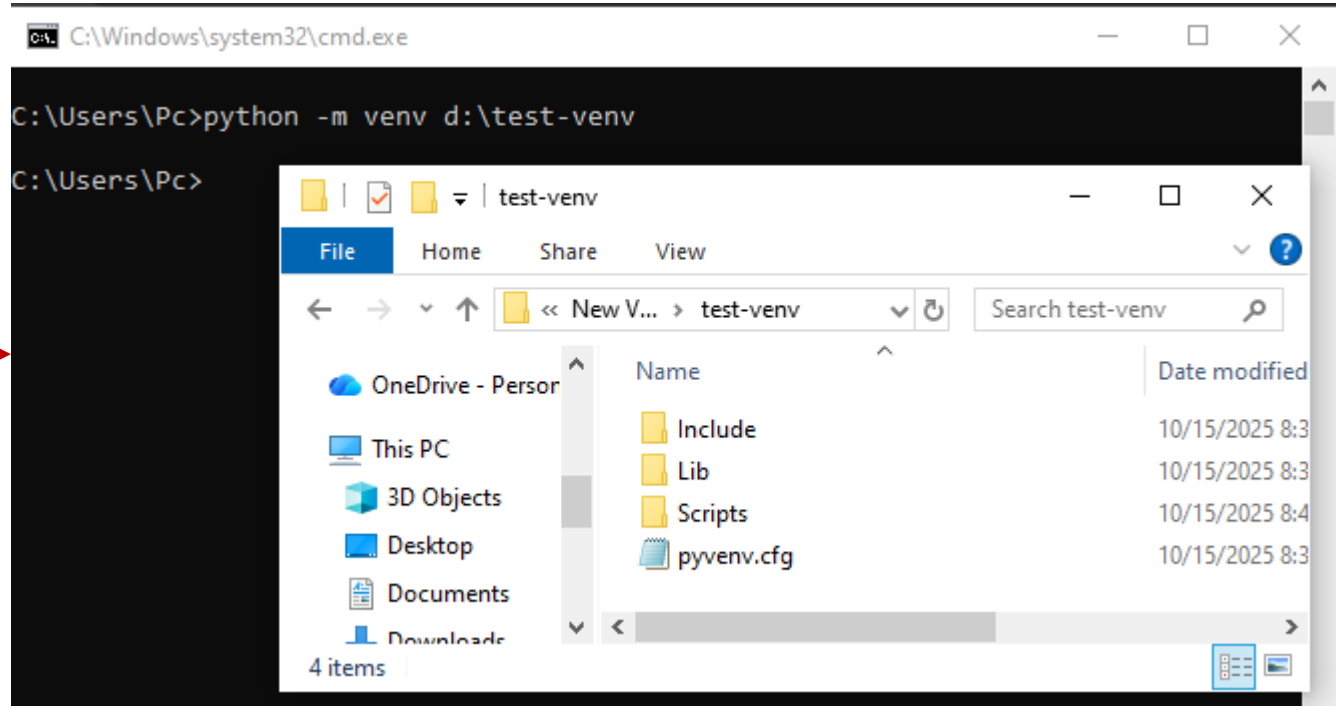
…Python Virtual Environment

- From the command line
  - `python –m venv ENV_DIR_PATH`
  - Note
    - If you have installed multiple Pythons on a single system and want to use a non-default version (not set to the PATH environmental variable as Python) as the base, use the entire path of the appropriate Python. Example:
      - `C:\Users\Name\AppData\Local\Programs\Python\Python312\python -m venv ENV_DIR_PATH`
- Activation
  - Go to the created virtual environment directory and run the activate script
    - `ENV_DIR_PATH\Scripts\Activate`

...Python Virtual Environment

- ...From the command line
  - Example





...Python Virtual Environment

- ...From the command line
  - ...Example

Activation

```
D:\test-venv>Scripts\activate

(test-venv) D:\test-venv>pip list
Package Version
------- -------
pip     23.2.1

[notice] A new release of pip is available: 23.2.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Install a package

```
(test-venv) D:\test-venv>python -m pip install rptree
Collecting rptree
  Obtaining dependency information for rptree from https://files.
  Downloading rptree-0.1.1-py3-none-any.whl.metadata (2.0 kB)
Downloading rptree-0.1.1-py3-none-any.whl (5.0 kB)
Installing collected packages: rptree
Successfully installed rptree-0.1.1

[notice] A new release of pip is available: 23.2.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip

(test-venv) D:\test-venv>pip list
Package Version
------- -------
pip     23.2.1
rptree  0.1.1

[notice] A new release of pip is available: 23.2.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Works on the venv

```
(test-venv) D:\test-venv>rptree d:\test-dir
d:\test-dir\
├── New folder\
│   ├── New Text Document 2.txt
│   └── New Text Document.txt
│
├── New folder 2\
│
└── New Text Document.txt
```

Deactivate

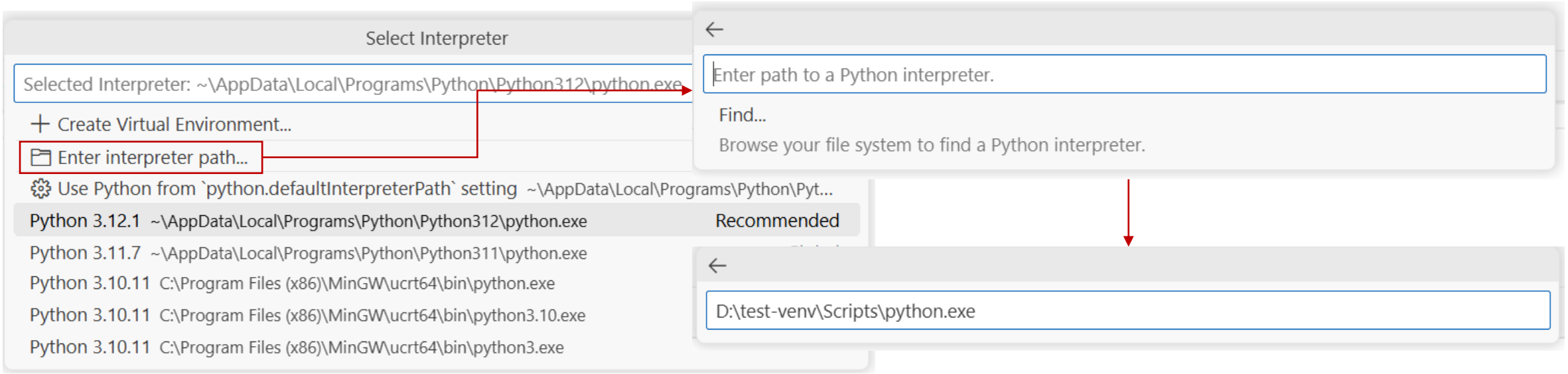
```
(test-venv) D:\test-venv>Scripts\deactivate
D:\test-venv>
```

Does not work on the venv

```
(test-venv) D:\test-venv>deactivate
D:\test-venv>rptree d:\test-dir
'rptree' is not recognized as an internal or external command,
operable program or batch file.
```

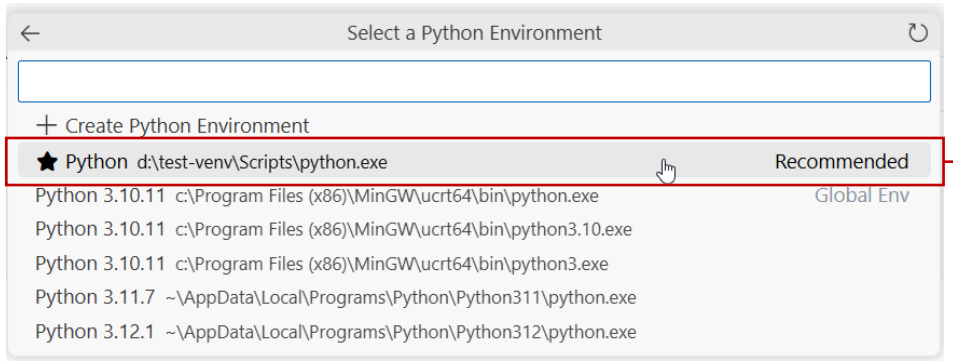
...Python Virtual Environment

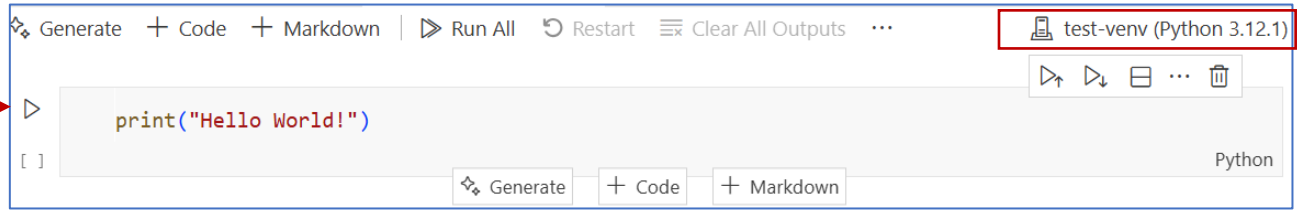
- From VS Code
  - Go to Command Palette and select `Python: Select Interpreter`
    - Enter previously created interpreter path or create a new one



...Python Virtual Environment

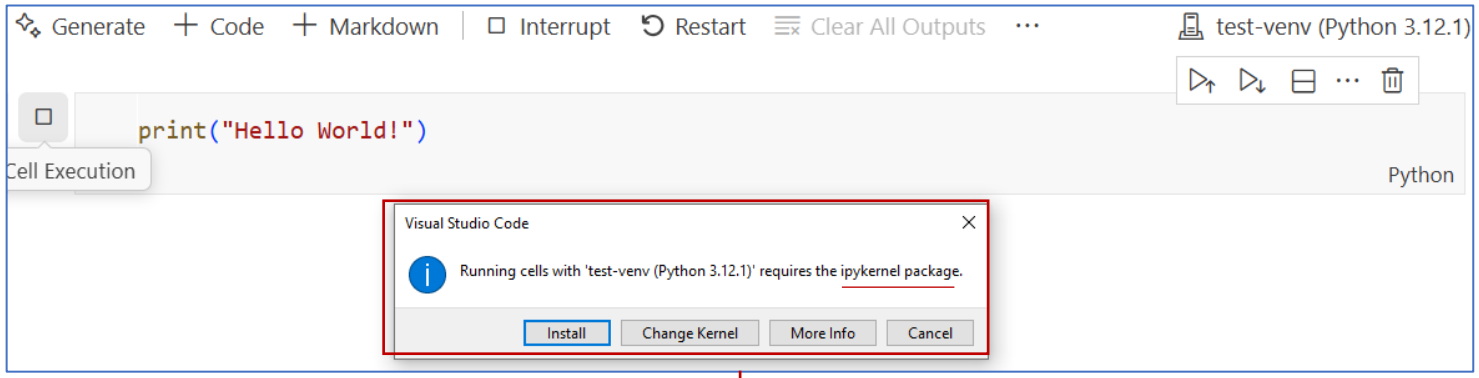
- ...From VS Code
  - On kernel selection step, select your virtual environment

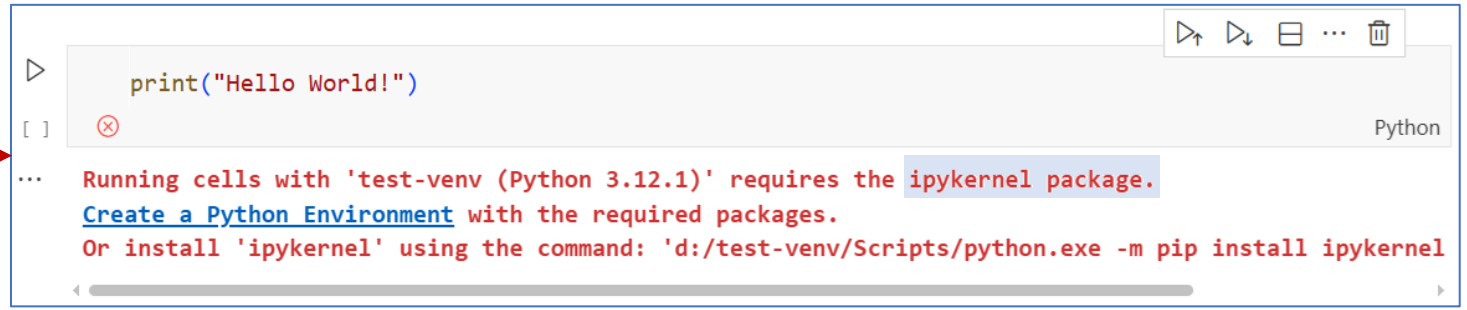




...Python Virtual Environment

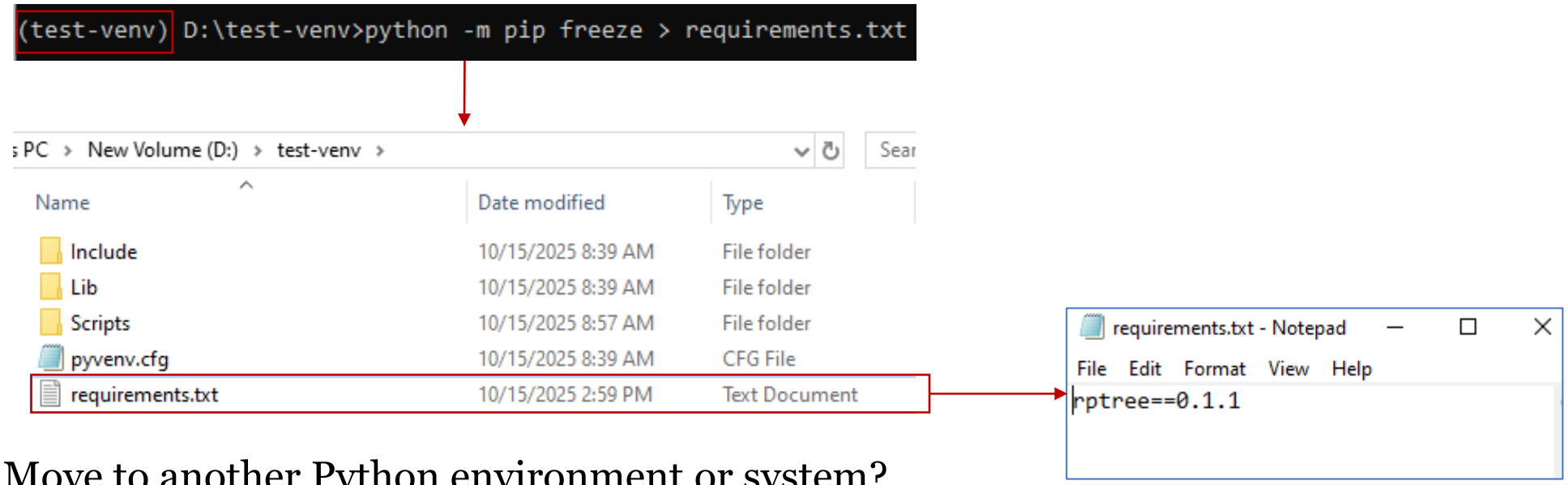
- ...From VS Code
  - Example





...Python Virtual Environment

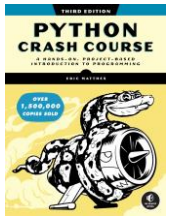
- Pin your dependencies
  - Make your virtual environment reproducible
  - Create requirements.txt which contains the list of installed packages
    - Use `pip freeze` command while your virtual environment is active



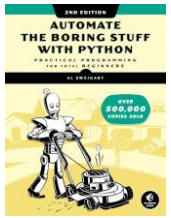
    - Move to another Python environment or system?
      - You have all the required packages names and versions Install them by a single simple command!
        - `python -m pip install -r requirements.txt`

How to Learn Python

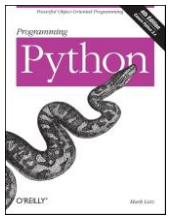
- Books



E. Matthes, Python crash course: A hands-on, project-based introduction to programming, 3rd ed. No Starch Press, 2023.



A. Sweigart, Automate the boring stuff with Python: practical programming for total beginners, 2nd ed. No Starch Press, 2019.



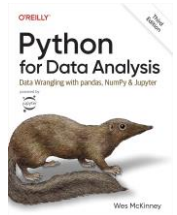
M. Lutz, Programming Python, 4th ed. O'Reilly Media, 2011.

+ Lots of online courses!

How to Learn Python Libraries

- Online user guides
  - https://numpy.org
  - https://pandas.pydata.org
  - https://matplotlib.org
  - https://scikit-learn.org

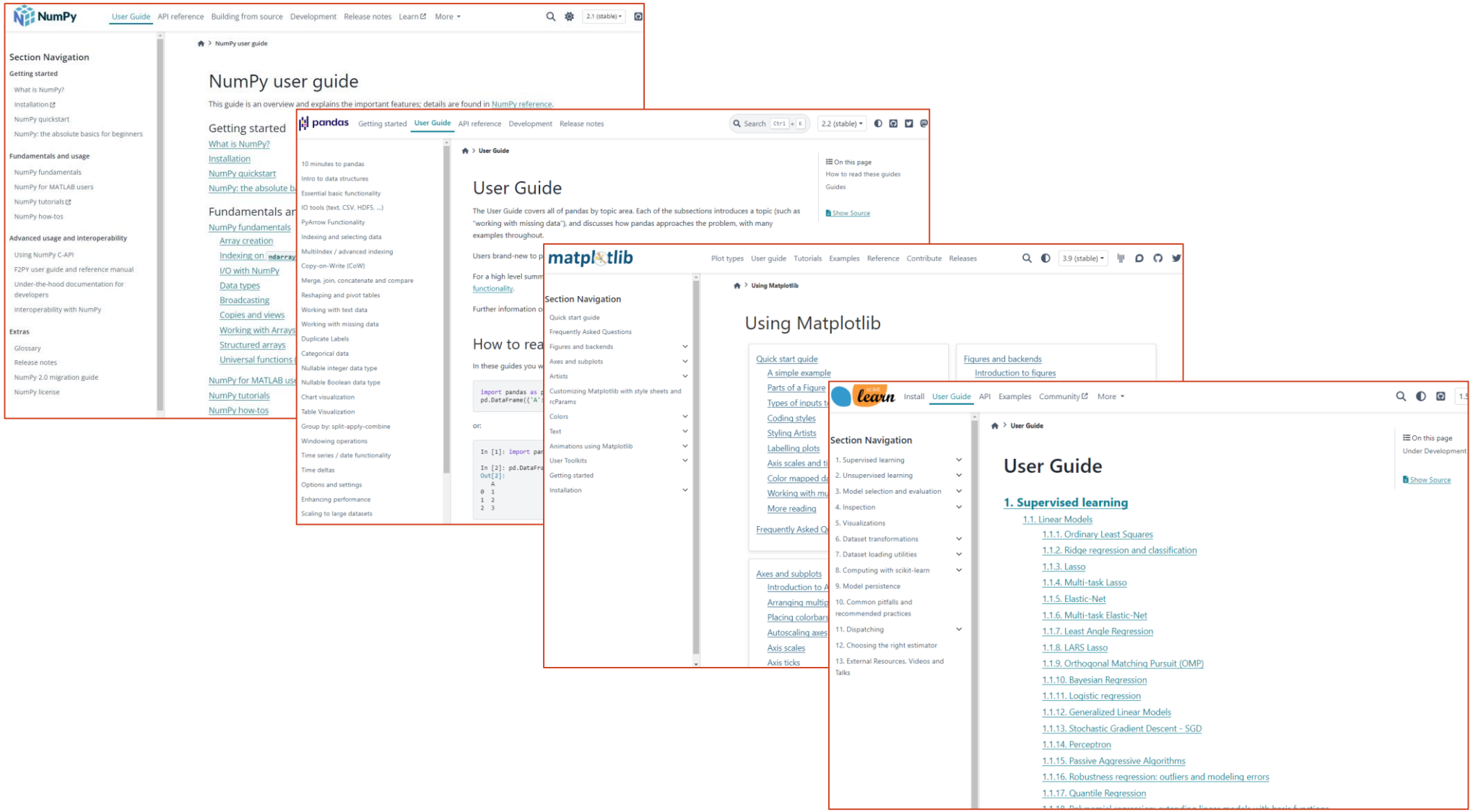
- W. McKinney, *Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter*, 3rd ed. O'Reilly Media, 2022.



- Online course
  - Applied Data Science with Python- Coursera Specialization,
    - Instructors: Christopher Brooks, Kevyn Collins-Thompson, Daniel Romero, and V. G. Vinod Vydiswaran
    - University of Michigan

+ Lots of online courses!

User guides


NumPy
User Guide
API reference
Building from source
Development
Release notes
Learn
More
Section Navigation
Getting started
What is NumPy?
Installation
NumPy quickstart
NumPy: the absolute basics for beginners
Fundamentals and usage
NumPy fundamentals
NumPy for MATLAB users
NumPy tutorials
NumPy how-tos
Advanced usage and interoperability
Using NumPy C-API
F2PY user guide and reference manual
Under-the-hood documentation for developers
Interoperability with NumPy
Extras
Glossary
Release notes
NumPy 2.0 migration guide
NumPy license
NumPy user guide
This guide is an overview and explains the important features; details are found in NumPy reference.
Getting started
What is NumPy?
Installation
NumPy quickstart
Fundamentals
NumPy fundamentals
Array creation
I/O with NumPy
Data types
Broadcasting
Copies and views
Working with Arrays
Structured arrays
Universal functions
NumPy for MATLAB
NumPy tutorials
NumPy how-tos
pandas
Getting started
User Guide
API reference
Development
Release notes
User Guide
10 minutes to pandas
Intro to data structures
Essential basic functionality
IO tools (text, CSV, HDF5, ...)
PyArrow Functionality
Indexing and selecting data
MultiIndex / advanced indexing
Copy-on-Write (CoW)
Merge, join, concatenate and compare
Reshaping and pivot tables
Working with text data
Working with missing data
Duplicate Labels
Categorical data
Nullable integer data type
Nullable Boolean data type
Chart visualization
Table Visualization
Group by: split-apply-combine
Windowing operations
Time series / date functionality
Time deltas
Options and settings
Enhancing performance
Scaling to large datasets
User Guide
The User Guide covers all of pandas by topic area. Each of the subsections introduces a topic (such as "working with missing data"), and discusses how pandas approaches the problem, with many examples throughout.
On this page
How to read these guides
Guides
Show Source
matplotlib
Plot types
User guide
Tutorials
Examples
Reference
Contribute
Releases
Section Navigation
Quick start guide
Frequently Asked Questions
Figures and backends
Axes and subplots
Artists
Customizing Matplotlib with style sheets and rcParams
Colors
Text
Animations using Matplotlib
User Toolkits
Getting started
Installation
Using Matplotlib
Quick start guide
A simple example
Parts of a Figure
Coding styles
Styling Artists
Labelling plots
More reading
Figures and backends
Introduction to figures
Axes and subplots
learn
Install
User Guide
API
Examples
Community
More
Section Navigation
1. Supervised learning
2. Unsupervised learning
3. Model selection and evaluation
4. Inspection
5. Visualizations
6. Dataset transformations
7. Dataset loading utilities
8. Computing with scikit-learn
9. Model persistence
10. Common pitfalls and recommended practices
11. Dispatching
12. Choosing the right estimator
13. External Resources, Videos and Talks
User Guide
On this page
Under Development
Show Source
1. Supervised learning
1.1. Linear Models
1.1.1. Ordinary Least Squares
1.1.2. Ridge regression and classification
1.1.3. Lasso
1.1.4. Multi-task Lasso
1.1.5. Elastic-Net
1.1.6. Multi-task Elastic-Net
1.1.7. Least Angle Regression
1.1.8. LARS Lasso
1.1.9. Orthogonal Matching Pursuit (OMP)
1.1.10. Bayesian Regression
1.1.11. Logistic regression
1.1.12. Generalized Linear Models
1.1.13. Stochastic Gradient Descent - SGD
1.1.14. Perceptron
1.1.15. Passive Aggressive Algorithms
1.1.16. Robustness regression: outliers and modeling errors
1.1.17. Quantile Regression

Python Syntax

Indentation
Comments
Variables
Strings
Inputs
Conditionals
Lists
Loops

Functions
Tuples
Sets
Dictionaries
Modules
Classes
Files
Exceptions

All the codes are available online:
https://github.com/hhomaei/aml

let's start coding…