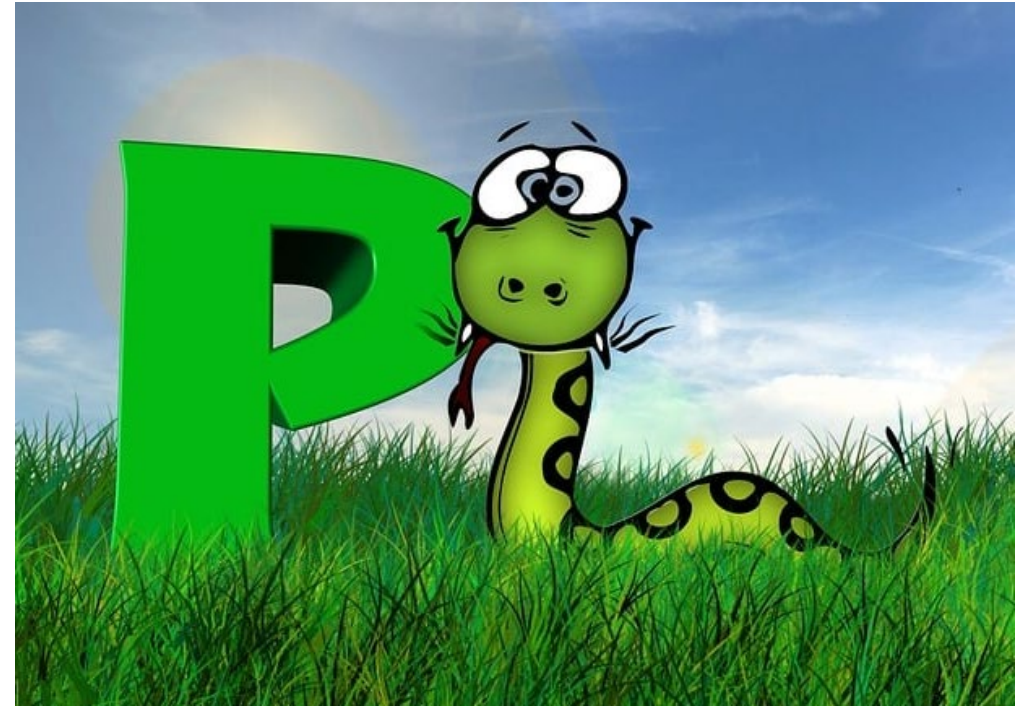


Python - Introduction

By: Winston Yeung
Department of Engineering
IVE(Lee Wai Lee)

Revised: April 2025



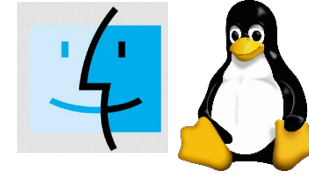
Learning Outcomes

After completion of the 2-hour taster programme, students should be able to:

- Install Python and IDE on PC and use them to write codes
- Understand the Python basics (e.g. syntax, comments, variables, data types, numbers, Booleans, strings, lists)
- Understand simple conditionals (e.g. if statement, for and while loops)
- write a Python code to solve simple math problem

Why Python?

- Easy to use
- Active community
- Compatible with many platforms
- Excellent ranges of libraries, packages
- Data science
- Machine learning
- Web development
- Career opportunities



Just want to try Python without installing anything?

- Go to <https://jupyter.org/>

File → New → Notebook

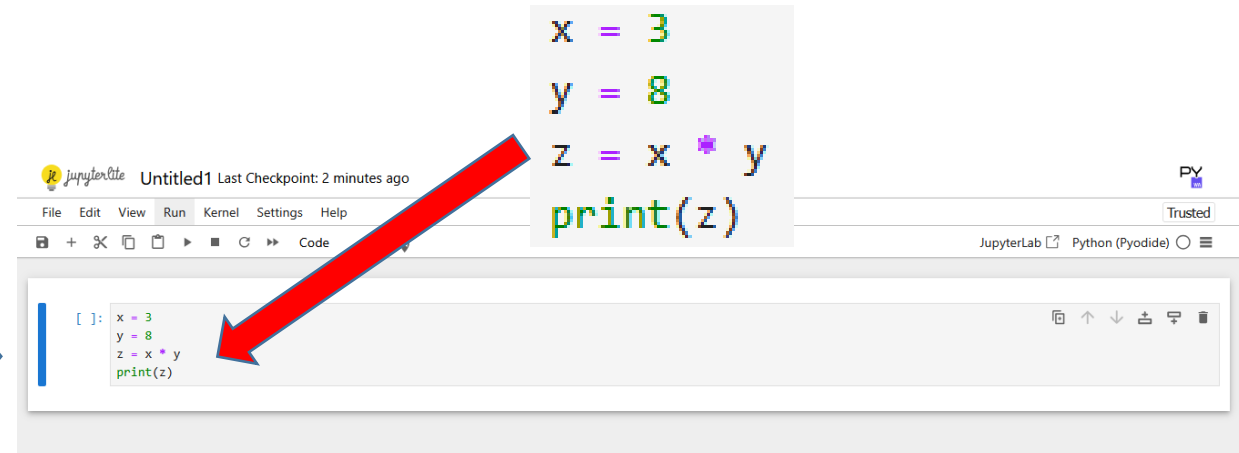
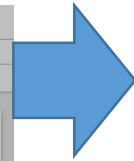
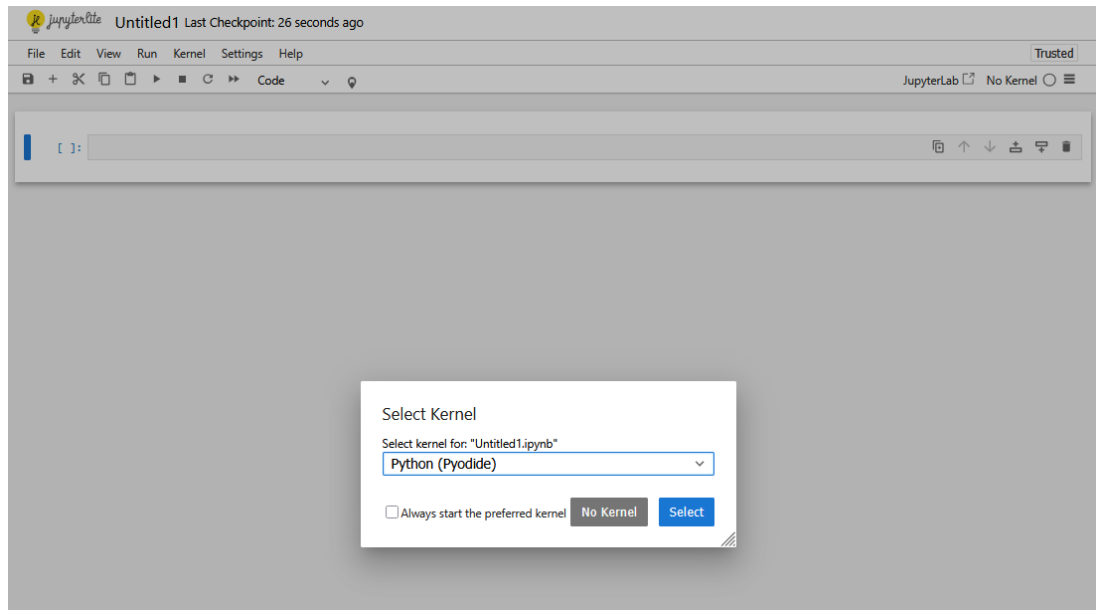
The image is a composite of three parts illustrating how to try Python without installation:

- Top Left:** The Jupyter website homepage. It features the Jupyter logo and the text "Free software, open standards, and web services for interactive computing across all programming languages". Below this, there are several screenshots of JupyterLab interfaces. A red arrow points from the "Try on browser" button to the Jupyter Notebook interface.
- Top Center:** A blue box with the Python logo and the text "The original web application for creating and sharing computational documents".
- Top Right:** A screenshot of the Jupyter Notebook interface. The "File" menu is open, and the "New" submenu is visible. A red arrow points from the "File" menu to the "New" submenu, and then to the "Notebook" option.

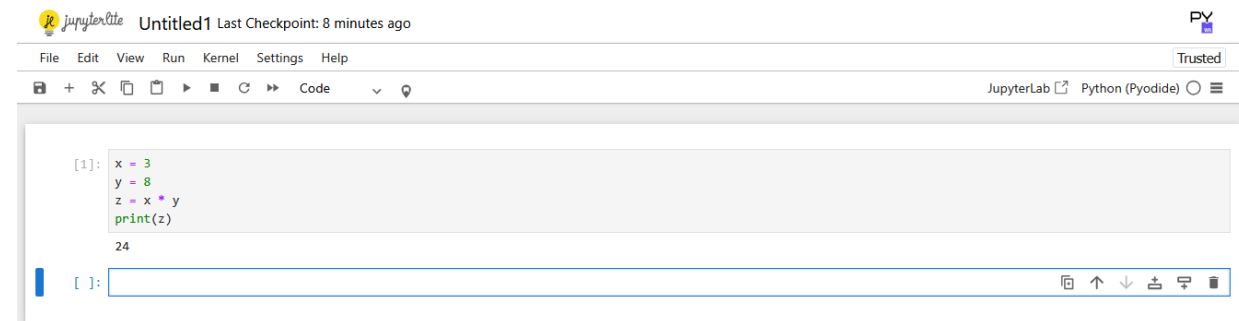
Below the screenshots, the text "Try on browser" is displayed.

Using Jupyter Notebook online (no download)

- Select the right kernel



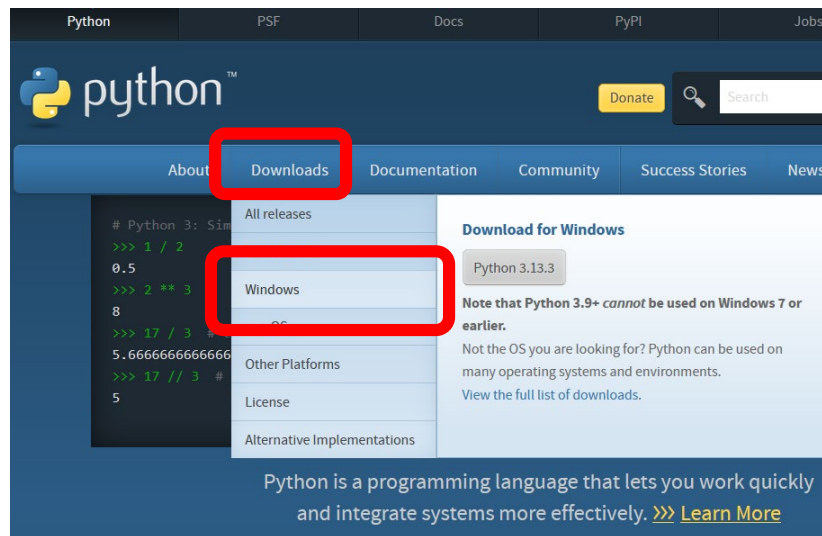
Run the selected cell by pressing
“Shift” + “Enter” or ►



Python on your PC

- <https://www.python.org/>

1. Downloads → Windows



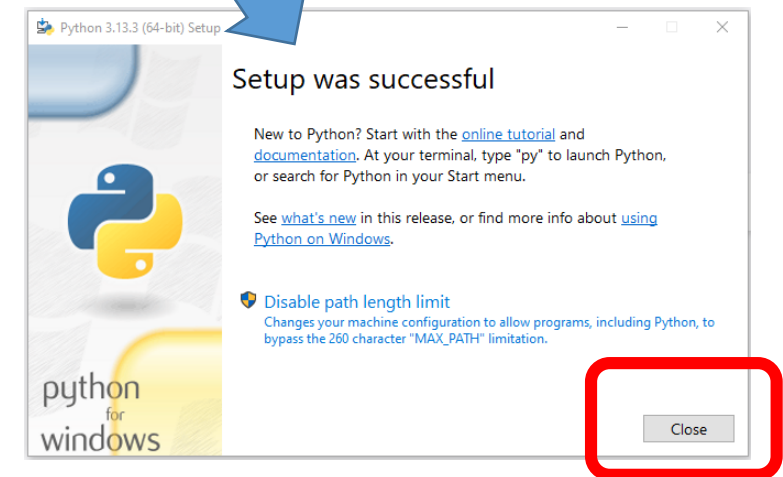
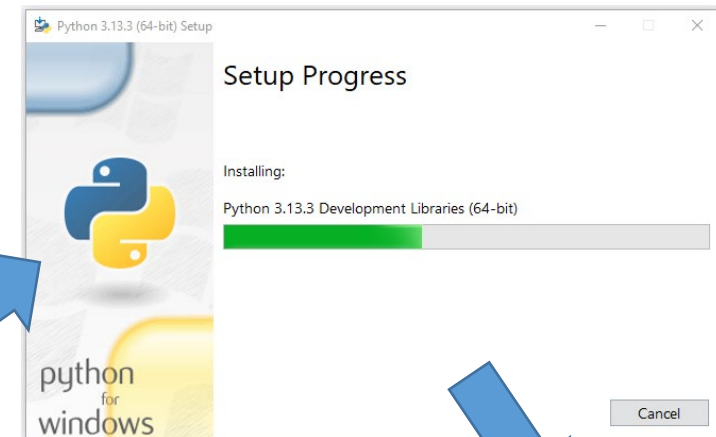
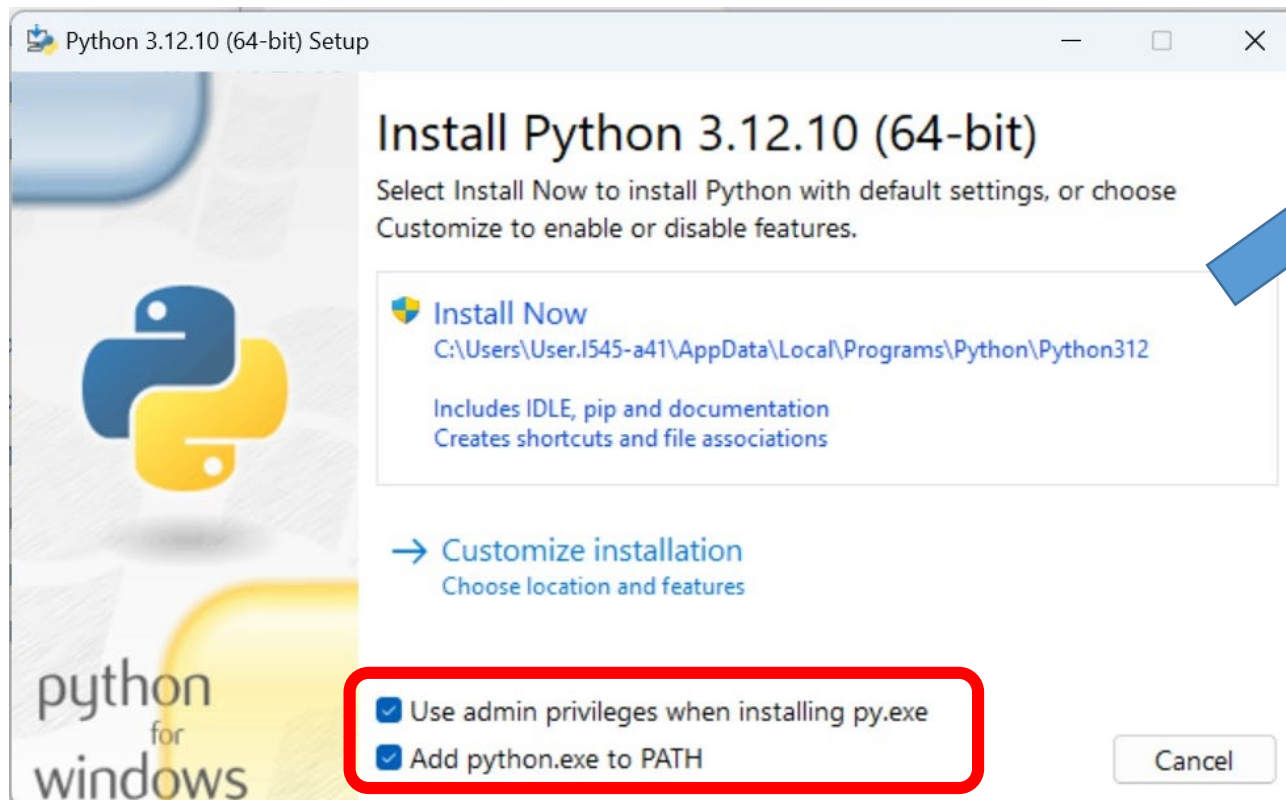
2. Scroll down and find Python 3.12.10, download the installer for Windows (64-bit)

- Download [Windows embeddable package \(ARM64\)](#)
- [Python 3.12.10 - April 8, 2025](#)
Note that Python 3.12.10 *cannot* be used on Windows 7 or earlier.
 - Download [Windows installer \(64-bit\)](#)
 - Download [Windows installer \(32-bit\)](#)
 - Download [Windows installer \(ARM64\)](#)
 - Download [Windows embeddable package \(64-bit\)](#)
 - Download [Windows embeddable package \(32-bit\)](#)
 - Download [Windows embeddable package \(ARM64\)](#)
- [Python 3.10.17 - April 8, 2025](#)
Note that Python 3.10.17 *cannot* be used on Windows 7 or earlier.

3. From your Download folder, double click the application file (**python-3.12.10-amd64.exe**) that you just downloaded

Install Python 3.12.10 (64-bit)

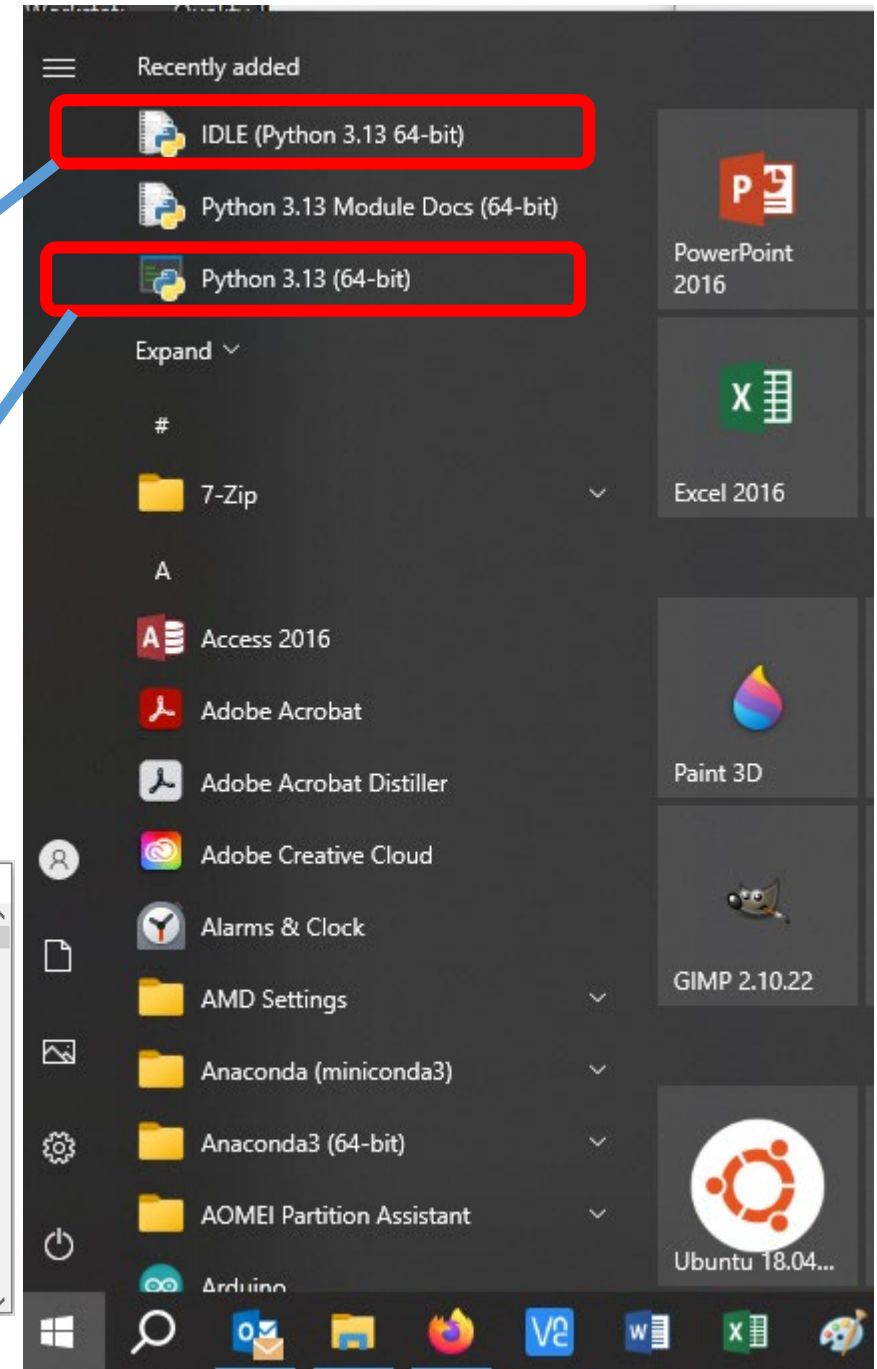
- Make sure “Add python.exe to PATH” is checked ☒
- Then, click “Install Now”



Python installed

For Windows 10

- Try the **IDLE** and Python Shell that are installed together
- **IDLE** is Python's **I**ntegrated **D**evelopment and **L**earning **E**nvironment
- Python Shell window

A screenshot of the IDLE Shell 3.13.3 window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the Python 3.13.3 startup message and a series of commands and their output:

```
>>> x = 3
>>> y = 8
>>> z = x * y
>>> print(z)
24
>>>
```

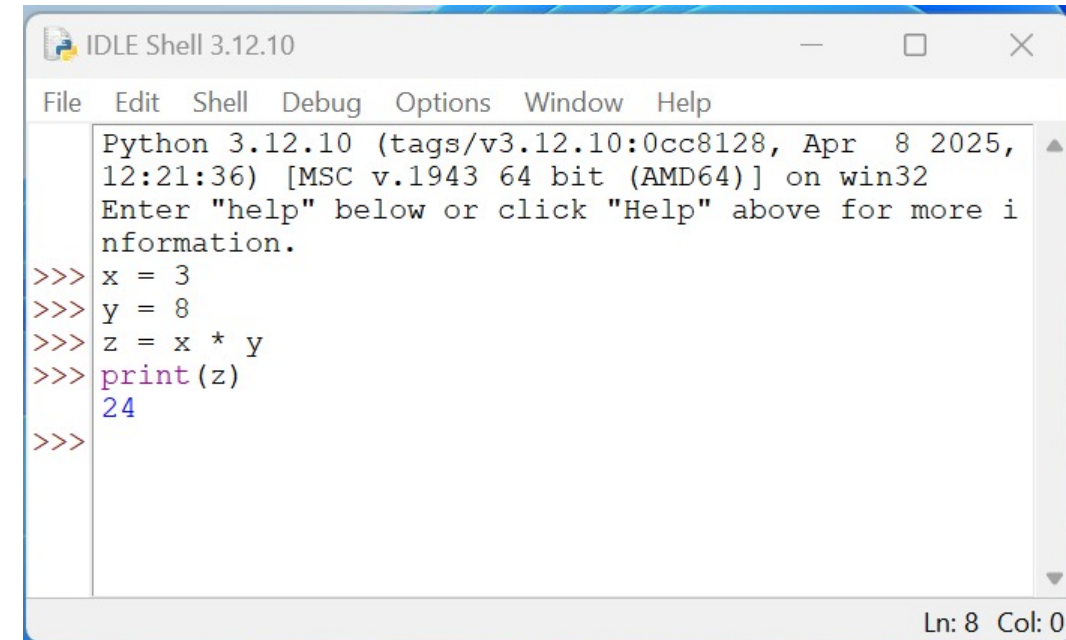
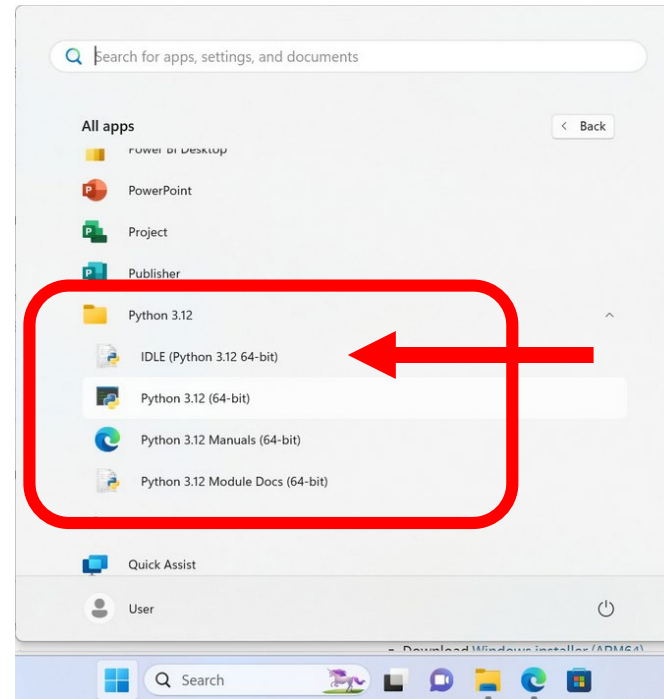
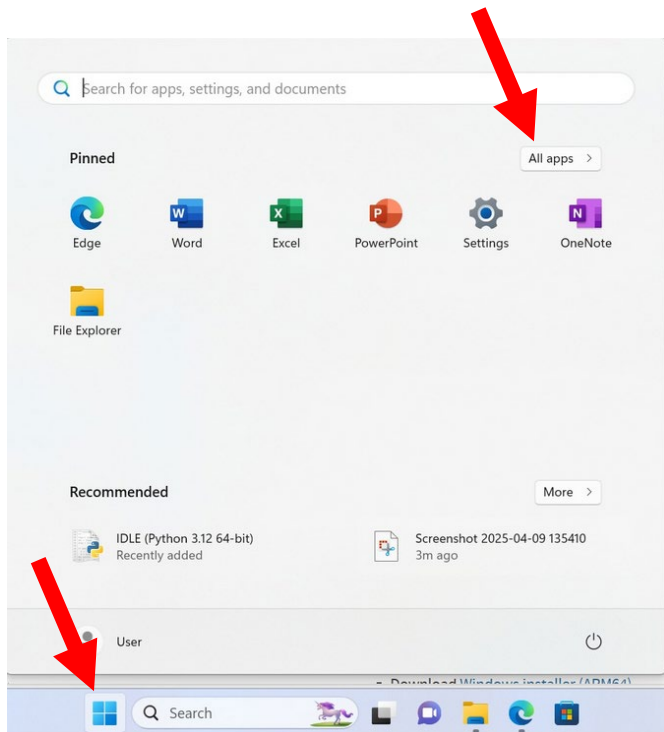
The status bar at the bottom right indicates 'Ln: 8 Col: 0'.A screenshot of the Python 3.13 (64-bit) Shell window. The window title is 'Python 3.13 (64-bit)'. The main text area shows the Python 3.13.3 startup message and a series of commands and their output:

```
Python 3.13.3 (tags/v3.13.3:6280bb5, Apr 8 2025, 14:47:33) [MSC v.1943 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 3
>>> y = 8
>>> z = x * y
>>> print(z)
24
>>>
```


Run Python

For Windows 11

- Run the Python IDLE Shell after Python is installed



Some good IDE (Integrated Development Environment)

- There are several popular Python IDEs that you can use for free



PyCharm

<https://www.jetbrains.com/pycharm/>



VS Code

<https://code.visualstudio.com/>



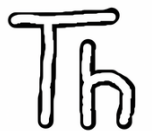
Sublime Text

<https://www.sublimetext.com/>



Atom

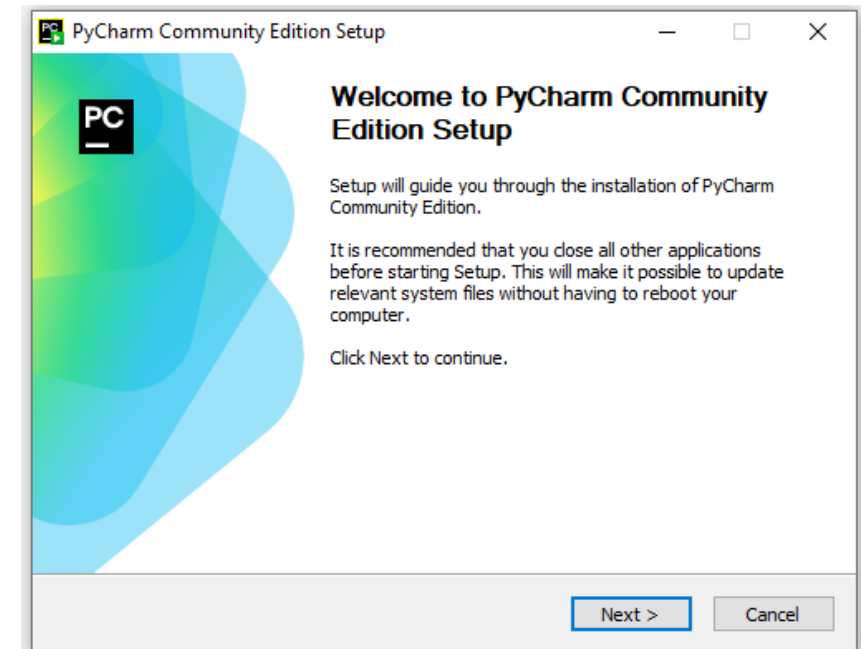
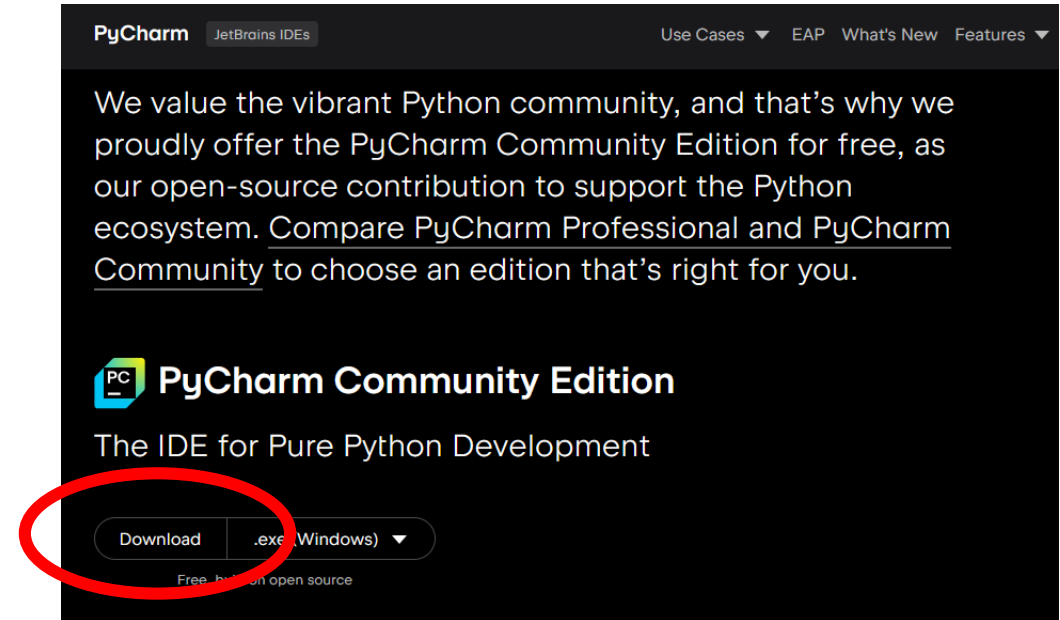
<https://atom-editor.cc/>



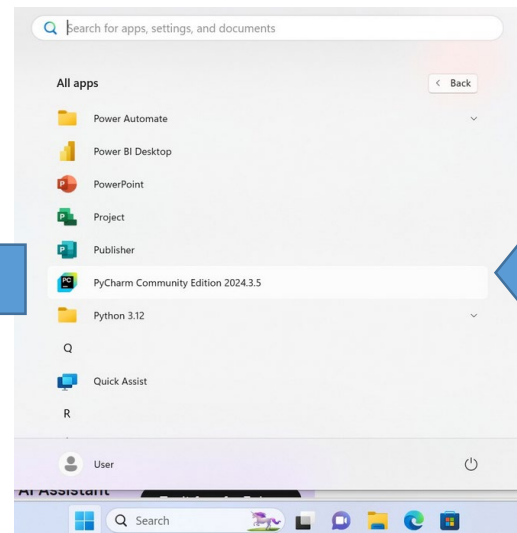
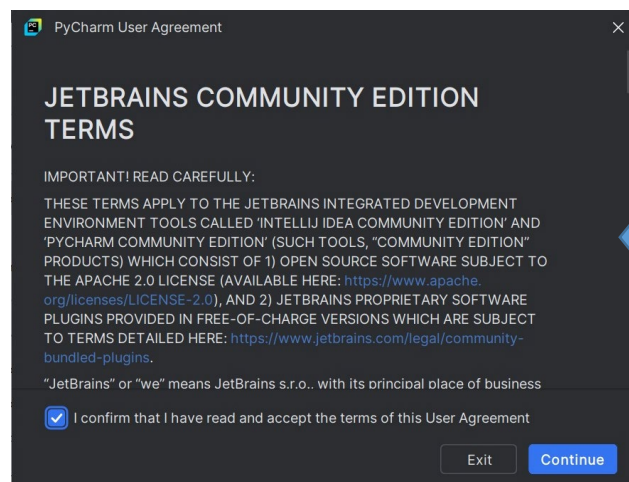
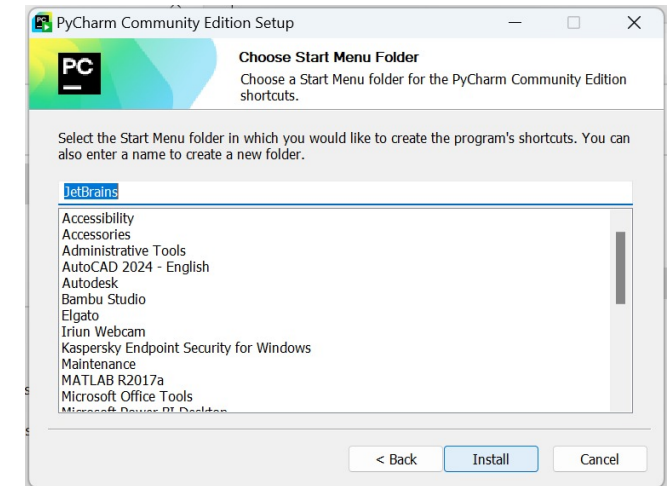
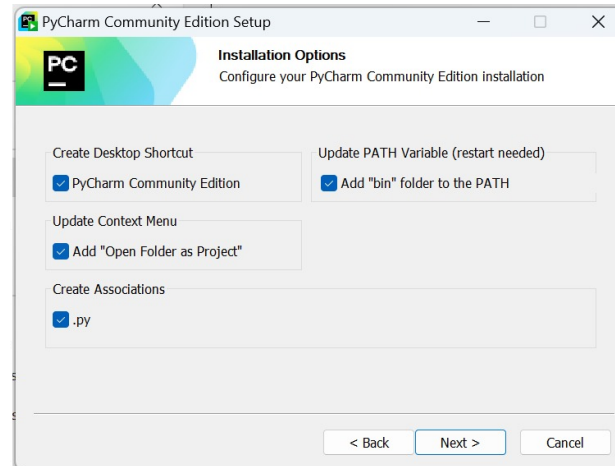
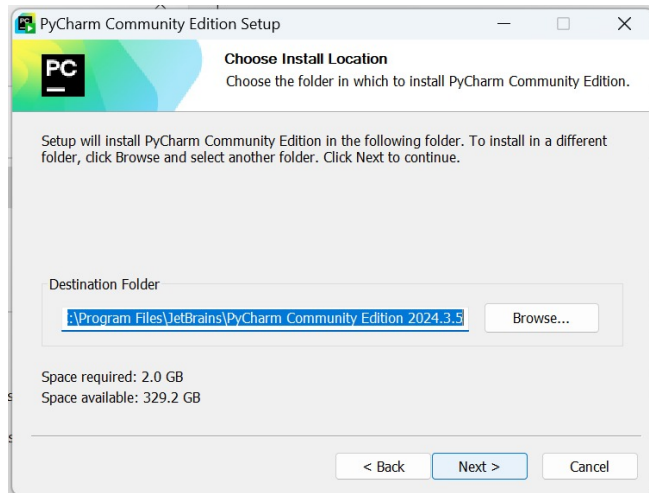
Thonny

PyCharm

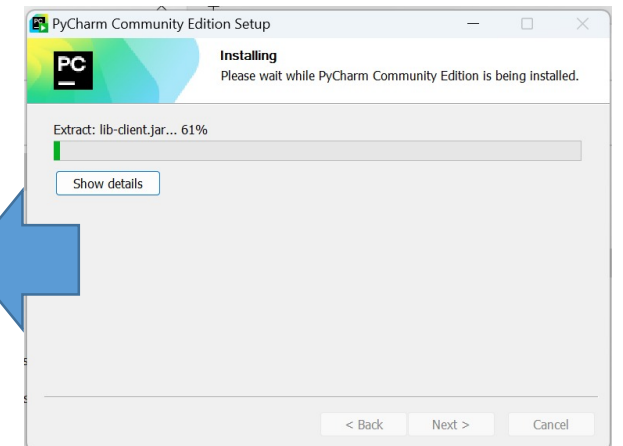
- Download PyCharm Community Edition
 1. Scroll down and find the PyCharm Community Edition. It is a free open source software. Click “Download”
 2. Go to Downloads folder and double click the **pycharm-community-2024.3.5.exe** file to start installation



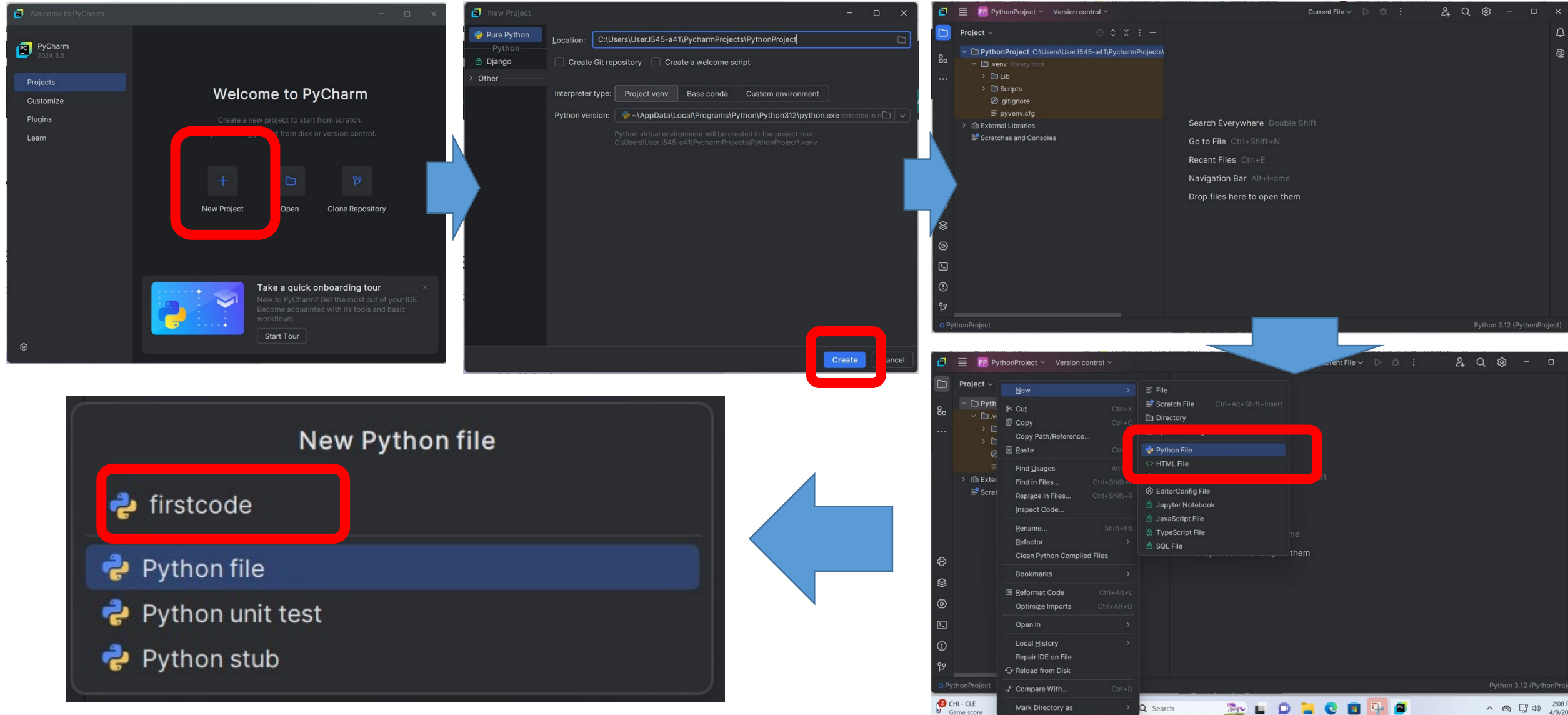
Install and run PyCharm (1)



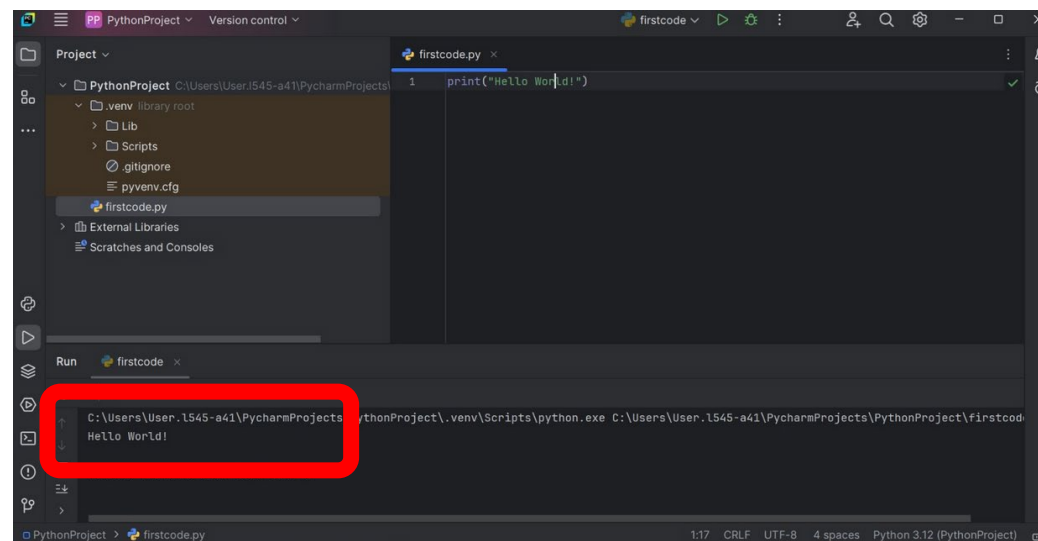
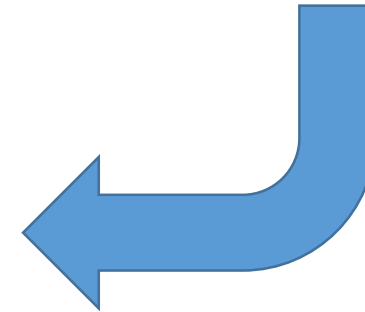
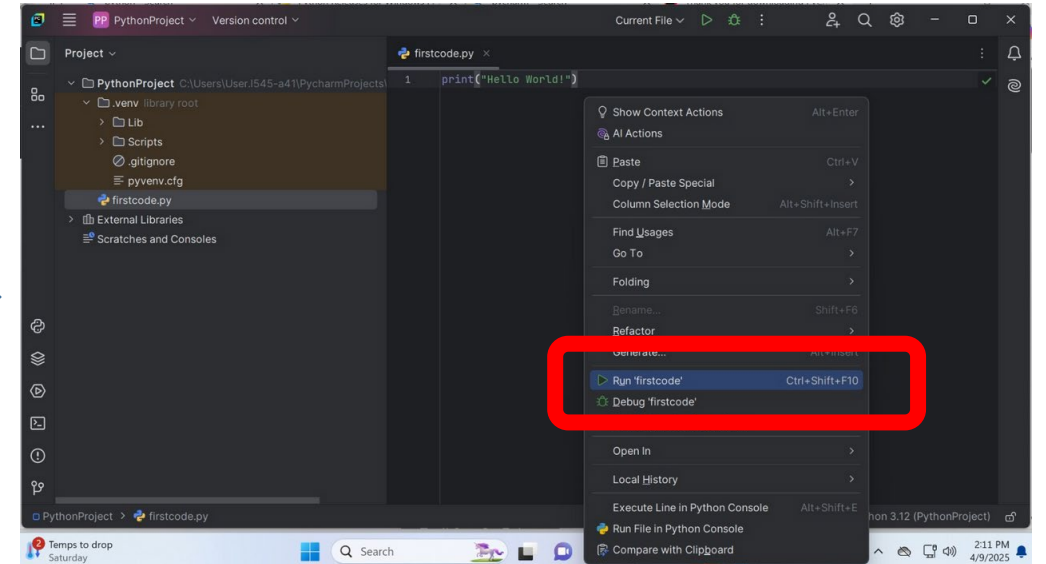
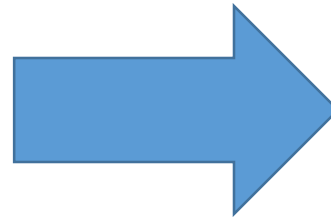
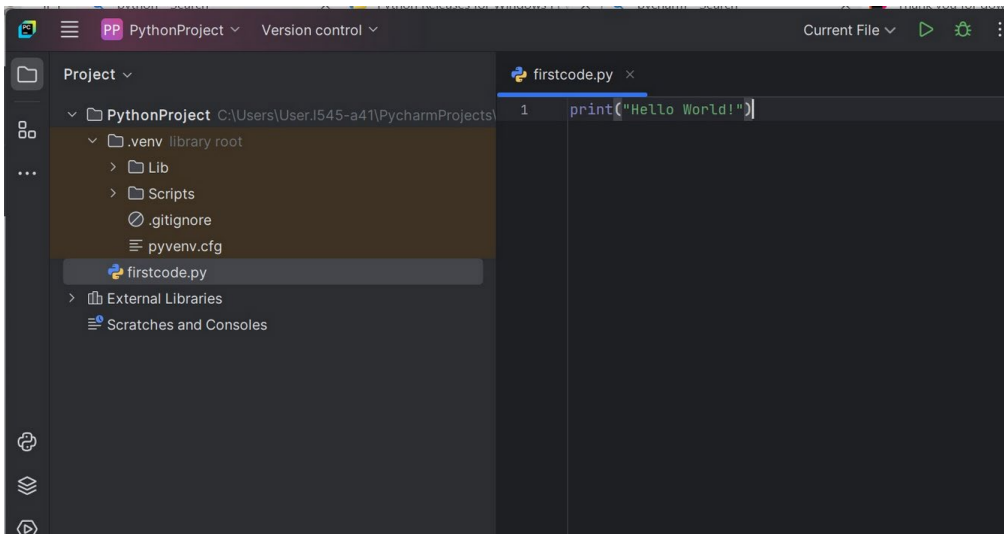
DO NOT
RESTART
YOUR
COMPUTER



Install and run PyCharm (2)

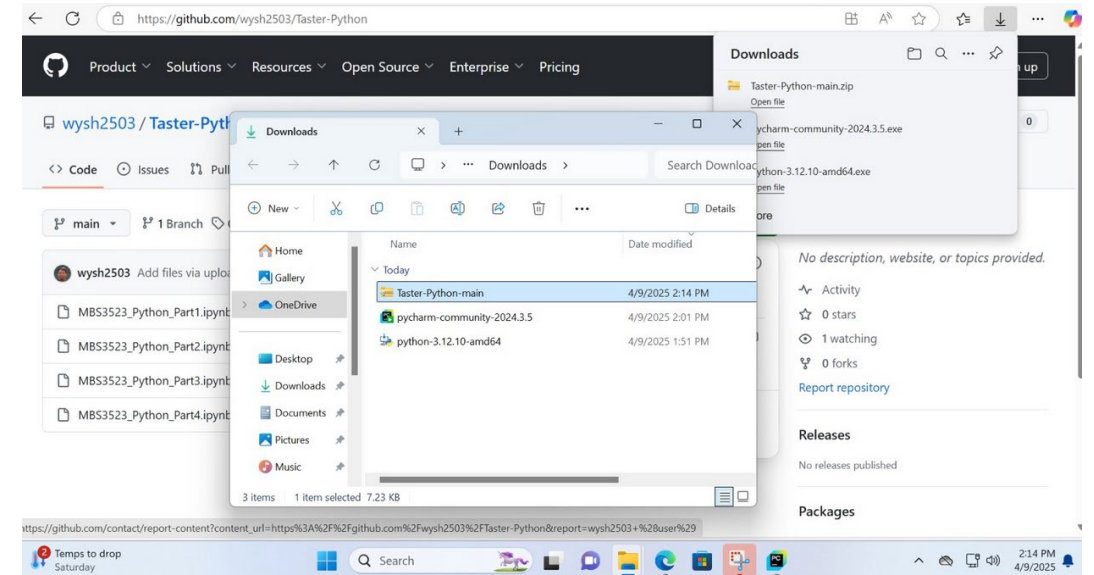
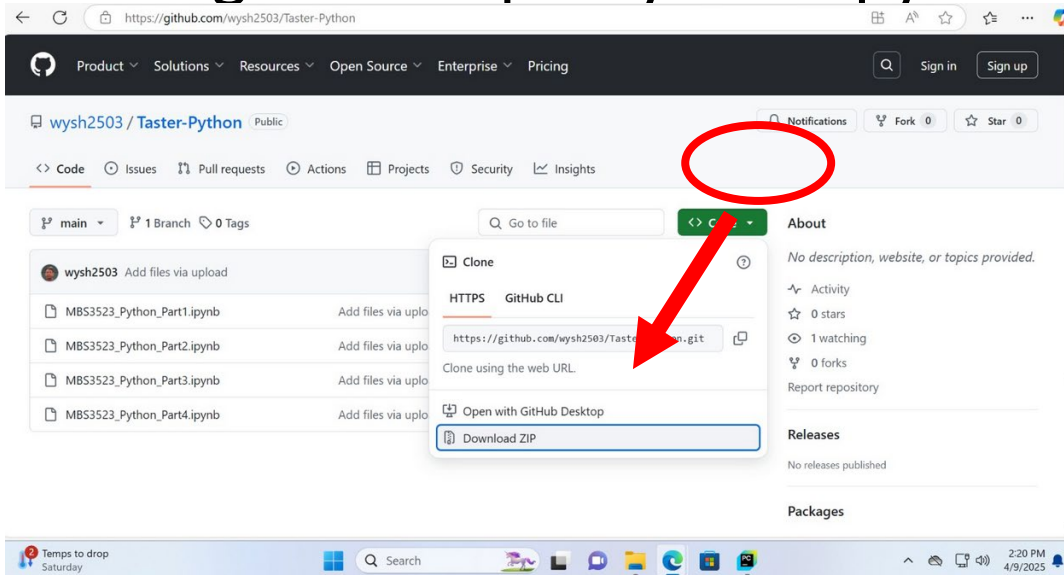


Install and run PyCharm (3)



Time for practice

- Get the file from <https://github.com/wysh2503/Taster-Python>
- Download the 4 files (Code → Download ZIP), open the Download folder, right click the Taster-Python-main ZIP file and select “Extract All...” to unzip the folder, save them on your Desktop
- Drag and drop to your Jupyter Notebook



End of Taster test

Quadratic Equation

- Given: $y = 1.15x^2 + 0.1x - 6.2$
- Write a Python code to find the roots (when $y = 0$) of the above quadratic equation

Reference:

<https://www.mathsisfun.com/algebra/quadratic-equation-graph.html>

Suggested answer

```
import math

# Define coefficients
a = 1.15 # coefficient of x2
b = 0.1  # coefficient of x
c = -6.2 # constant term

# Method 1: Using quadratic formula
# Quadratic formula:  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ 
discriminant = b**2 - 4*a*c

if discriminant < 0:
    print("No real roots exist (discriminant < 0)")
elif discriminant == 0:
    x = -b / (2*a)
    print(f"One real root: x = {x:.4f}")
```

```
else:
    x1 = (-b + math.sqrt(discriminant)) / (2*a)
    x2 = (-b - math.sqrt(discriminant)) / (2*a)
    print("Two real roots:")
    print(f"x1 = {x1:.4f}")
    print(f"x2 = {x2:.4f}")
```