



GETTING STARTED WITH PYTHON

Basic Extensions & Libraries



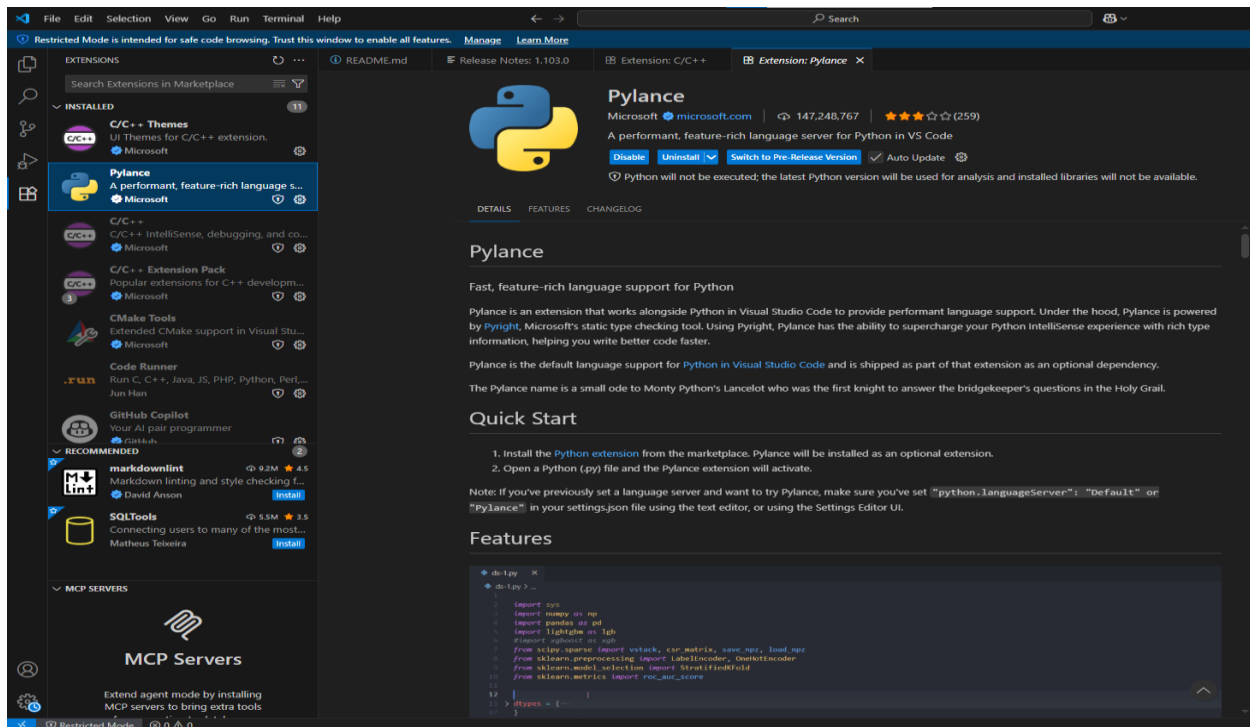
AUGUST 14, 2025

SYED MUHAMMAD SAJAWAL HUSSAIN

0319-7341432 / 0328-0841432

In Python development, **extensions** and **libraries** play a vital role in enhancing productivity and functionality. In **Visual Studio Code (VS Code)**, extensions are add-ons that provide extra features such as syntax highlighting, debugging support, linting, code completion, and integration with tools like Git or virtual environments—making coding faster, easier, and more efficient. Python libraries, on the other hand, are collections of pre-written code that developers can use to perform specific tasks without building everything from scratch, such as data analysis (Pandas), numerical computation (NumPy), or web development (Flask, Django). Together, VS Code extensions help create a smooth development environment, while Python libraries provide ready-made tools to implement powerful features in programs quickly.

Extensions Page in VS Code



Useful Extensions

Here's a list of **useful VS Code extensions** for Python development and why they're helpful:

1. Python (by Microsoft)

- **Purpose:** Core extension for Python in VS Code.
 - **Features:** Syntax highlighting, IntelliSense (code completion), linting, debugging, and virtual environment support.
-

2. Pylance

- **Purpose:** Improves IntelliSense performance.
 - **Features:** Fast and accurate type checking, better auto-completion, and code navigation.
-

3. Jupyter

- **Purpose:** Run and edit Jupyter Notebooks directly in VS Code.
 - **Features:** Useful for data science, AI, and quick code testing.
-

4. Python Docstring Generator

- **Purpose:** Automatically creates docstrings for functions and classes.
 - **Features:** Helps document your code quickly and neatly.
-

5. Python Indent

- **Purpose:** Fixes indentation issues automatically.
 - **Features:** Keeps code formatting clean and consistent.
-

6. GitLens

- **Purpose:** Advanced Git integration in VS Code.
 - **Features:** View commit history, authorship, and file changes directly in the editor.
-

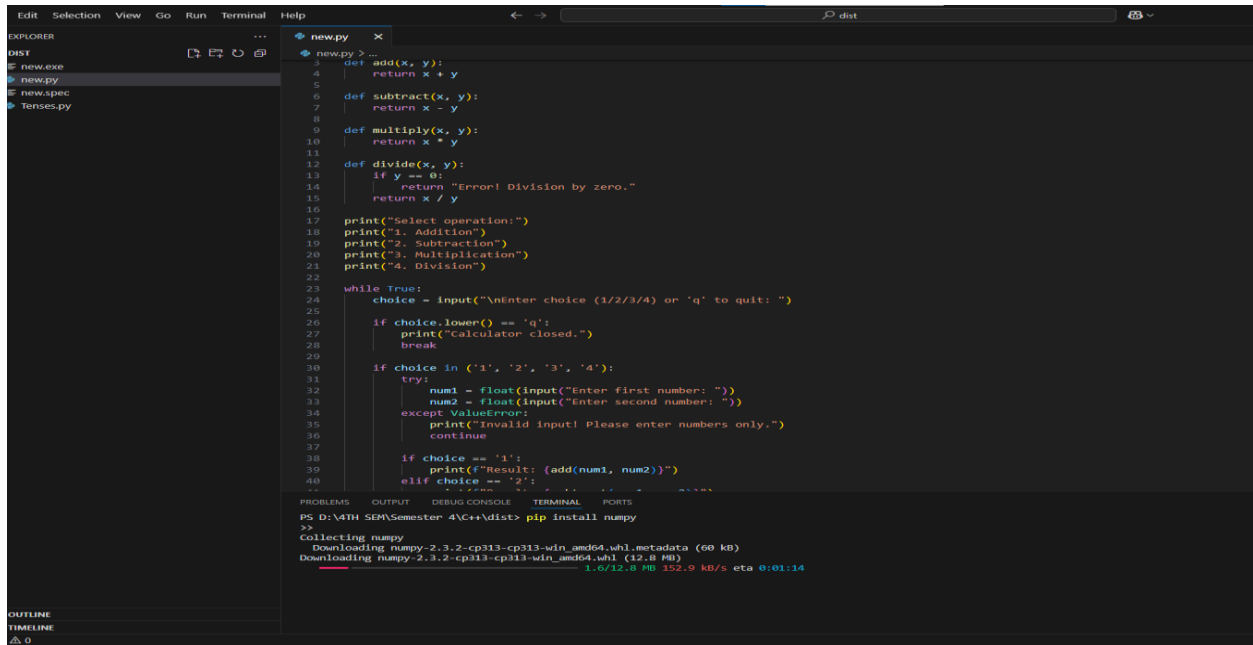
7. Code Runner

- **Purpose:** Run Python (and other language) code snippets instantly.
 - **Features:** Executes code without needing to set up debugging.
-

8. Autopep8

- **Purpose:** Automatically formats Python code according to PEP 8 standards.
 - **Features:** Keeps code clean and readable.
-

Libraries installation Page



The screenshot shows a Python IDE with a file explorer on the left containing 'new.exe', 'new.py', 'new.spec', and 'Tenses.py'. The main editor displays a Python script named 'new.py' which implements a simple calculator with functions for addition, subtraction, multiplication, and division, along with a menu-driven interface. The terminal at the bottom shows the command 'pip install numpy' being executed, with output indicating the collection and downloading of NumPy version 2.3.2.

```
new.py
3 def add(x, y):
4     return x + y
5
6 def subtract(x, y):
7     return x - y
8
9 def multiply(x, y):
10    return x * y
11
12 def divide(x, y):
13     if y == 0:
14         return "Error! Division by zero."
15     return x / y
16
17 print("Select operation:")
18 print("1. Addition")
19 print("2. Subtraction")
20 print("3. Multiplication")
21 print("4. Division")
22
23 while True:
24     choice = input("\nEnter choice (1/2/3/4) or 'q' to quit: ")
25
26     if choice.lower() == 'q':
27         print("Calculator closed.")
28         break
29
30     if choice in ('1', '2', '3', '4'):
31         try:
32             num1 = float(input("Enter first number: "))
33             num2 = float(input("Enter second number: "))
34         except ValueError:
35             print("Invalid input! Please enter numbers only.")
36             continue
37
38         if choice == '1':
39             print(f"Result: {add(num1, num2)}")
40         elif choice == '2':
41             print(f"Result: {subtract(num1, num2)}")
42         elif choice == '3':
43             print(f"Result: {multiply(num1, num2)}")
44         elif choice == '4':
45             print(f"Result: {divide(num1, num2)}")
46
47     else:
48         print("Invalid choice. Please enter a valid choice.")
49
50     print("\n")
```

```
PS D:\4TH SEM\Semester 4\C++\dist> pip install numpy
>>
Collecting numpy
  Downloading numpy-2.3.2-cp313-cp313-win_amd64.whl.metadata (60 kB)
  Downloading numpy-2.3.2-cp313-cp313-win_amd64.whl (12.8 MB)
  Installing build dependencies: started 1.9/12.3 MB 152.9 kB/s eta 0:01:14
```

General Format

`pip install library_name`

Useful Libraries

Here's a list of **commonly used Python libraries** that are widely used for various purposes:

1. Data Analysis & Data Science

- **NumPy** – For numerical computations.
 - **Pandas** – For data manipulation and analysis.
 - **Matplotlib** – For data visualization (plots, charts).
 - **Seaborn** – For advanced and beautiful visualizations.
-

2. Machine Learning & AI

- **Scikit-learn** – For machine learning algorithms.
 - **TensorFlow** – For deep learning models.
 - **PyTorch** – Another popular deep learning framework.
-

3. Web Development

- **Flask** – Lightweight web framework.
 - **Django** – Full-featured web framework for large applications.
-

4. Automation & Scripting

- **Requests** – For sending HTTP requests to APIs or websites.
 - **BeautifulSoup** – For web scraping HTML data.
 - **Selenium** – For automating web browsers.
-

5. Others

- **OpenCV** – For image and video processing.
 - **Pillow** – For image manipulation.
 - **PyGame** – For making simple games.
-

Installation Commands

Here's that same list with **installation commands** so you can set up everything easily in Python:

1. Data Analysis & Data Science

- pip install numpy
 - pip install pandas
 - pip install matplotlib
 - pip install seaborn
-

2. Machine Learning & AI

- pip install scikit-learn
 - pip install tensorflow
 - pip install torch
-

3. Web Development

- pip install flask
 - pip install django
-

4. Automation & Scripting

- pip install requests
 - pip install beautifulsoup4
 - pip install selenium
-

5. Others

- pip install opencv-python
 - pip install pillow
 - pip install pygame
-

◆ Installing Python Libraries in VS Code using Terminal

1. Open VS Code and make sure Python is installed on your system.
2. Open the Terminal in VS Code by pressing:
 - Windows/Linux: `Ctrl + ~`
 - Mac: `Cmd + ~`
3. In the terminal, type the following command to install a library:

```
bash
```

[Copy](#) [Edit](#)

```
pip install library_name
```

Example:

```
bash
```

[Copy](#) [Edit](#)

```
pip install numpy
```

4. Wait for the installation to complete.
5. To confirm the installation, run:

```
bash
```

[Copy](#) [Edit](#)

```
pip show library_name
```

Tip: You can install multiple libraries at once, e.g.:

```
bash
```

[Copy](#) [Edit](#)

```
pip install numpy pandas matplotlib
```