

# Primer on Python & VSCode

We will transition from using Anaconda and Jupyter Notebook to **Visual Studio Code (VSCode)** and .py files for Python programming. VSCode provides a powerful, lightweight environment for writing and running Python code, offering more flexibility and features for larger projects. This guide will walk you through installing VSCode on macOS, Windows, and Linux, as well as creating and running your first Python file in VSCode. If you already have Anaconda installed on your computer, you can use its Python interpreter. However, if you do not have Anaconda, we've included steps to install Python separately.

## Sections:

1. VSCode Installation
2. Creating and Running a Python File in VSCode
3. Installing Python (If Anaconda is Not Installed)

Follow the instructions in the appropriate section for your operating system, and you will be up and running with Python in no time!

## What is VSCode, an Editor, and an IDE?

**VSCode (Visual Studio Code)** is a free, open-source code editor developed by Microsoft that supports multiple programming languages, including Python. It's lightweight and highly customizable, offering features like syntax highlighting, code suggestions, debugging, and built-in Git integration, which makes it a popular choice among developers.

An **editor** is a tool that allows you to write and edit code but doesn't provide many additional features beyond text manipulation, like debugging or version control. Examples include Notepad or Sublime Text.

An **IDE** (Integrated Development Environment) is a more comprehensive tool that combines an editor with additional tools like a debugger, compiler, and project management, providing a full-featured environment for development. While VSCode is technically an editor, it can function like an IDE when combined with extensions, offering many features of a traditional IDE without the overhead.

## Installation of VSCode

### For Mac:

1. **Download VSCode:** Go to [Visual Studio Code](#) and download the macOS version.
2. **Install VSCode:** Open the `downloaded.dmg` file and drag Visual Studio Code.app to the Applications folder.

3. **Open VSCode:** Launch VSCode from the Applications folder or Spotlight.

*If you face any difficulties, please refer to the official documentation :*

<https://code.visualstudio.com/docs/setup/mac>

#### **For Windows:**

1. **Download VSCode:** Go to [Visual Studio Code](https://code.visualstudio.com) and download the Windows version.
2. **Install VSCode:** Run the downloaded .exe file and follow the installation prompts.
3. **Open VSCode:** Launch VSCode from the Start menu.

*If you face any difficulties, please refer to the official documentation :*

<https://code.visualstudio.com/docs/setup/windows>

#### **For Linux:**

1. **Download VSCode:** Go to [Visual Studio Code](https://code.visualstudio.com) and download the Linux version (.deb for Debian/Ubuntu or .rpm for Fedora/Red Hat).
2. **Install VSCode:**
  - o **Debian/Ubuntu:** Install the .deb package using `sudo dpkg -i <file>.deb`.
  - o **Fedora/Red Hat:** Install the .rpm package using `sudo rpm -i <file>.rpm`.
3. **Open VSCode:** Launch VSCode from the Applications menu or by running code in the terminal.

*If you face any difficulties, please refer to the official documentation :*

<https://code.visualstudio.com/docs/setup/linux>

## **Creating and Running a .py File in VSCode**

1. **Open VSCode:** Launch VSCode from your installed location.
2. **Create a New Python File:**
  - o Click on **File > New File** to create a new file.
  - o Save the file with a .py extension (e.g., `hello.py`).
3. **Write Python Code:**

```
# hello.py
print("Hello, CS103!")
```
4. Save your file (CTRL + S)

## **Running the First .py File in VSCode**

Once you have written your Python code in VSCode, there are two main ways to run it: *using the terminal* or *using the built-in Run button*. Before running the code, it's important to ensure that VSCode is using the correct Python interpreter, especially if you have multiple Python versions installed (e.g., from Anaconda or Python.org).

## 1. Choosing the Python Interpreter

Step 1: Open VSCode and click on the View menu, then choose Command Palette (or press Ctrl+Shift+P / Cmd+Shift+P).

Step 2: In the Command Palette, type "Python: Select Interpreter" and select it.

Step 3: VSCode will show a list of available Python interpreters on your machine. If you have Anaconda installed, choose the interpreter from the Anaconda environment (e.g., Python 3.x.x ('base': conda)).

Step 4: Once selected, the interpreter will appear at the bottom-left corner of VSCode.

## 2. Running Python Code Using the Terminal

Step 1: After choosing the correct Python interpreter, save your file with a .py extension (e.g., `hello.py`).

Step 2: Open the integrated terminal in VSCode by going to View > Terminal

Step 3: Navigate to the folder where your Python file is saved using the `cd` command in the terminal.

```
cd path/to/your/file
```

Step 4: Run your Python file by typing the following command and pressing Enter:

```
python hello.py
```

Step 5: You should see the output of your Python code printed in the terminal.

*\*\*\* If you are having difficulties to locate your files in the terminal, please refer to the Linux command section at the end of this primer.*

## 3. Running Python Code Using the Run Button

Step 1: Open your .py file in the VSCode editor.

Step 2: At the top-right corner of the VSCode window, you'll see a green Run button (a play icon). Click this button.

Step 3: VSCode will automatically run the Python file using the selected interpreter, and the output will appear in the integrated terminal.

Step 4: If there are multiple Python files open, make sure the correct one is active in the editor before clicking the Run button.

## Which Method to Use?

*Run Button:* Ideal for quickly running small scripts without typing commands manually. It's more user-friendly for beginners.

*Terminal:* Provides more control and flexibility, especially for working with larger projects, virtual environments, or when passing arguments to your Python script.

## Installing Python (If Anaconda is Not Installed)

For Mac, Windows, and Linux:

1. **Download Python:** Go to [Python Downloads](#) and download the latest version of Python.
2. **Install Python:**
  - **Mac and Linux:** Run the downloaded .pkg (Mac) or .tar.gz (Linux) file and follow the installation instructions.
  - **Windows:** Run the downloaded .exe file and select "Add Python to PATH" during installation.
3. **Verify Installation:**
  - Open a new terminal or command prompt.
  - Type `python --version` to check if Python is installed correctly.
  - Type `pip --version` to check if pip (Python package installer) is installed.

## Basic Terminal Commands for First-Time Users

If this is your first time using the terminal, don't worry! The terminal is a powerful tool that allows you to interact with your computer using text commands. Below are some essential Linux/Unix commands (these also work on macOS and with slight differences on Windows using PowerShell or Command Prompt) to help you navigate and manage files in your system. You will use these commands to find, create, and run your Python files.

### 1. Navigating Directories

- **cd (Change Directory):** This command is used to navigate to a different folder or directory in your file system.
  - **Usage:**  
`cd path/to/directory`
  - **Example:**  
`cd Documents/PythonProjects`
  - **Go back to the previous directory:**  
`cd ..`

### 2. Viewing Contents of a Directory

- **ls (List):** This command shows the contents of the current directory.
  - **Usage:**  
`ls`

### 3. Creating a Directory

- **mkdir (Make Directory):** This command creates a new folder or directory.
  - **Usage:**

```
mkdir directory_name
○ Example:
mkdir PythonProjects
```

#### 4. Creating a New File

- **touch:** This command is used to create an empty file.
  - **Usage:**  
touch filename.py
  - **Example:**  
touch hello.py

#### 5. Checking Your Current Directory

- **pwd (Print Working Directory):** This command shows the full path of your current directory.
  - **Usage:**  
pwd
  - **Example:**  
/home/username/Documents/PythonProjects

#### 6. Running Python Scripts

Once you've navigated to the folder containing your Python file using the `cd` command, you can run the Python script.

- **Usage:**  
python filename.py
- **Example:**  
python hello.py

#### 7. Deleting Files and Directories

- **rm (Remove):** This command deletes a file.
  - **Usage:**  
rm filename.py
  - **Example:**  
rm hello.py
- **rm -r:** To delete a directory and all of its contents, use the `-r` flag (recursive).
  - **Usage:**  
rm -r directory\_name
  - **Example:**  
rm -r PythonProjects

#### Example Walkthrough:

1. Open your terminal.
2. Navigate to the folder where you want to create your Python file:  
cd Documents
3. Create a new directory for your Python projects:  
mkdir PythonProjects

4. Navigate into this new folder:  
`cd PythonProjects`
5. Create a new Python file:  
`touch hello.py`
6. Open this file in VSCode or any editor, write your Python code, and save it. After that, run it in the terminal:  
`python hello.py`

By mastering these basic commands, you'll be able to efficiently manage your files and navigate your system using the terminal!

Useful Links:

<https://code.visualstudio.com/docs/setup/setup-overview>

<https://marketplace.visualstudio.com/items?itemName=ms-python.python>