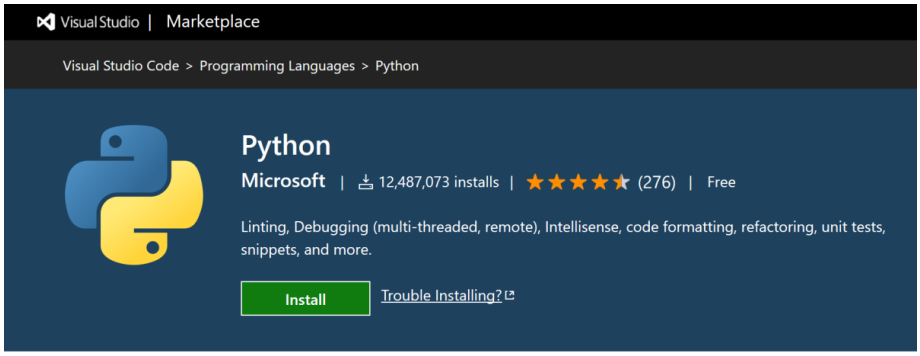
Quick guide to developing software in Visual Studio using Python. This is based on https://code.visualstudio.com/docs/python/python-tutorial

1. If you have not already done so, install VS Code.

2. Next, install the Python extension for VS Code from the Visual Studio Marketplace. For additional details on installing extensions, see Extension Marketplace. The Python extension is named **Python** and it's published by Microsoft.

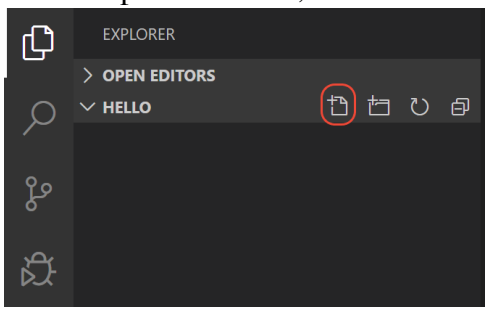


3. Install a Python interpreter. The system install of Python on macOS is not supported. Instead, an installation through Homebrew is recommended. To install Python using Homebrew on macOS use `brew install python3` at the Terminal prompt.

4. Start a new workspace. Open the Command Palette via (⇧⌘P) and type shell command to find the Shell Command: Install 'code' command in PATH** command.

5. Using a command prompt or terminal, create an empty folder, navigate into it, and open VS Code (`code`) in that folder (`.`) By starting VS Code in a folder, that folder becomes your "workspace". VS Code stores settings that are specific to that workspace in `.vscode/settings.json`, which are separate from user settings that are stored globally.

6. From the File Explorer toolbar, select the **New File** button on the `hello` folder



7. It's simple to run your program with Python. Just click the **Run Python File in Terminal** play button in the top-right side of the editor.

8. Let's now try **debugging** our program. Just click in the editor's left gutter, next to the line numbers. When you set a breakpoint, a red circle appears in the gutter.

9. Next, to initialize the debugger, press F5, a configuration menu will open from the Command Palette allowing you to select the type of debug configuration you would like for the opened file. For a simple file just select **Python File**. The debugger will stop at the first line of the file breakpoint.

10. To install a package (which may also install other as a dependency), use the Command Palette to run **Terminal: Create New Integrated Terminal** (^⇧`). This command opens a command prompt for your selected interpreter.

11. A best practice among Python developers is to avoid installing packages into a global interpreter environment. You instead use a project-specific **`virtual environment`** that contains a copy of a global interpreter. Once you activate that environment, any packages you then install are isolated from other environments. Such isolation reduces many complications that can arise from conflicting package versions. To create a *virtual environment* and install the required packages, enter the following commands as appropriate for your operating system:

    a. Create and activate the virtual environment in MacOS:

    ```
    python3 -m venv .venv
    source .venv/bin/activate
    ```

    b. Select your new environment by using the **Python: Select Interpreter** command from the **Command Palette**.

    c. Install the packages with

    ```
    python3 -m pip install <package>
    ```

    d. Rerun the program

    e. Once you are finished, type `deactivate` in the terminal window to deactivate the virtual environment.