**Assignment; setting up your developer environment.**

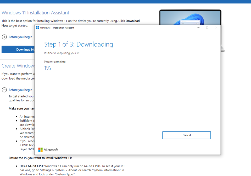
**1. Select your operating system (OS);**

Installing Windows 11 on your computer involves several steps, and it's essential to ensure that your hardware meets the minimum requirements for Windows 11. Here's a general guide on how to install Windows 11:

### Before You Begin:

1. **Check System Requirements**:
   * Verify that your computer meets the minimum hardware requirements for Windows 11. These include:
     + **Processor**: 1 gigahertz (GHz) or faster with 2 or more cores on a compatible 64-bit processor or system on a chip (SoC).
     + **RAM**: 4 gigabytes (GB) or more.
     + **Storage**: 64 GB or larger storage device.
     + **TPM**: Trusted Platform Module (TPM) version 2.0.
     + **UEFI**: Secure Boot capable.
     + **Graphics**: DirectX 12 compatible graphics / WDDM 2.x.
     + For the latest requirements, visit the [official Microsoft Windows 11 specifications](https://www.microsoft.com/en-us/windows/windows-11-specifications).
2. **Backup Your Data**:
   * Before installing Windows 11, it's advisable to back up your important files and documents to an external drive or cloud storage.
3. **Create Installation Media**:
   * You will need a USB flash drive (at least 8 GB) to create installation media for Windows 11. You can use the [Media Creation Tool](https://www.microsoft.com/software-download/windows11) provided by Microsoft to create a bootable USB drive.

### Installing Windows 11:

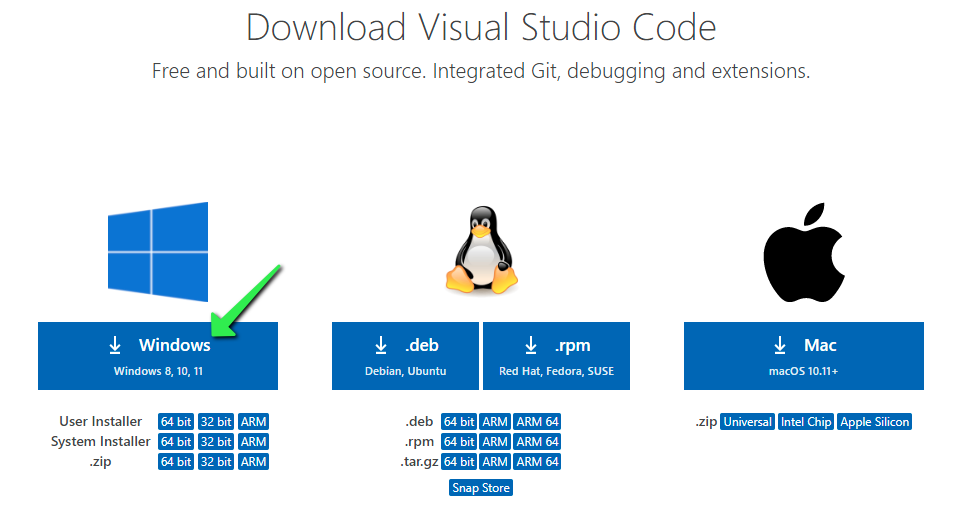
1. **Download Windows 11 ISO**:
   * Visit the [Microsoft Windows 11 download page](https://www.microsoft.com/en-us/software-download/windows11) and download the Windows 11 installation media creation tool.
   * Run the tool and follow the prompts to create a bootable USB drive with Windows 11 installation files.
2. **Boot from USB**:
   * Insert the bootable USB drive into your computer.
   * Restart your computer and access the BIOS/UEFI settings (usually by pressing a key like F2, F12, Delete, or Esc during startup).
   * Set the USB drive as the primary boot device in the BIOS/UEFI settings and save the changes.
3. **Install Windows 11**:
   * Your computer should boot from the USB drive into the Windows 11 setup process.
   * Follow the on-screen instructions to select language, time, and keyboard preferences.
   * Click "Install Now" and follow the prompts to proceed with the installation.
   * You may need to enter your product key during the installation (if you don't have one, you can skip this step and activate Windows later).
4. **Configure Settings**: During installation, you will be prompted to choose between upgrading an existing installation and performing a clean installation (which wipes your current data). Choose accordingly.
5. **Complete Installation**; after installation completes, follow the setup wizard to personalize your settings, including creating user accounts and adjusting privacy settings.
6. **Install Drivers and Updates**: After Windows 11 is installed, it's recommended to install device drivers for your hardware components (graphics card, network adapter, etc.) and run Windows Update to ensure you have the latest security patches and drivers.
7. **Activate Windows**:
   * If prompted, activate Windows 11 using your product key. You can find this key in the packaging or confirmation email if you purchased a copy of Windows 11.

**2 .Install a text editor or Integrated Development Environment**.

For windows, visit the official visual studio website: code.visualstudio.com.

Click on the download for windows button to download the installer.

Once the installer is downloaded, find the installation file on your pc and run it as an administrator. Follow the prompts on the installer unless you want to choose a specific preference.

After the installation, you can open VS code from the start menu or by searching for “visual studio code” in the search bar.

**3. Set up a version control system.**

Installing Git and configuring it on your windows machine.

Visit the official Git website; git-scm.com.

Click on the “download” button to download the installer.

Once the installer is downloaded, run it as an administrator on your pc. Follow the prompts displayed on the screen to set up the installer.

Once the installation is complete, we have to adjust the path environment. It is optional but recommended. During the installation, you might be asked to “adjust your PATH environment” settings. Choose the option “use Git from the Windows Command Prompt” or “use Git and optional Unix tool from the Command Prompt” to make Git accessible from the command line.

Creating a Git hub account for hosting your repositories.

Open your browser and go to github.com, sign up and fill in your details to create an account.

Choose a plan of your choice set up remote origin {free or paid}.

Verify your email and complete your profile.

You’ll have to link local Git repository to Github;

1. Initialize your git repository if not already done.

* Open git terminal on windows
* Navigate to your project directory “cd path/to/your/project”.
* Initialize Git repository by typing ‘git init’

2. Set Up Remote Origin:

After creating the repository on GitHub, you will see instructions on how to push an existing repository from the command line.

Copy the commands under the section "…or push an existing repository from the command line".

In your terminal or command prompt where your local repository is located, paste the commands you copied and press Enter.

Replace https://github.com/yourusername/repository-name.git with the URL provided by GitHub for your repository.

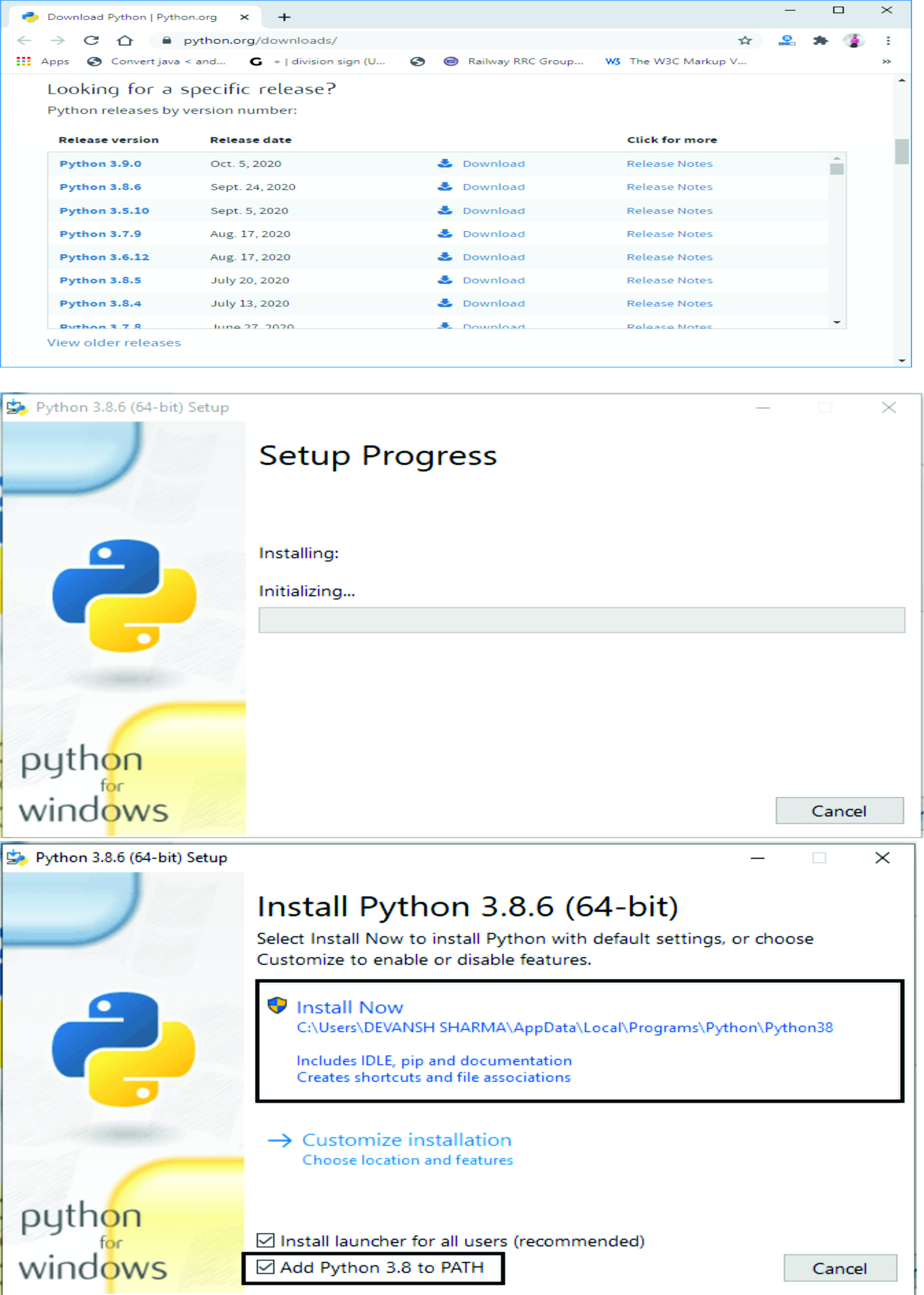
This command sets up a remote connection named "origin" to your GitHub repository and pushes your local commits to GitHub.

Verify on GitHub - Refresh your GitHub repository page. You should see your files from the local repository now synced to GitHub.

Create a New Repository on GitHub:

* Log in to your GitHub account.
* Click on the "+" sign in the upper-right corner of the page and select "New repository."
* Give your repository a name, optionally add a description, choose whether it should be public or private, and click "Create repository."

**4. Install necessary programming language**

**Download Python Installer:**

* + Visit the official Python website at python.org.
  + Navigate to the Downloads section and click on the latest version of Python for Windows (e.g., Python 8.0.3x).
  + Scroll down to find the Windows installer (usually labeled as "Windows installer (64-bit)" or "Windows installer (32-bit)" depending on your system).

**Run the Installer:**

* + Once downloaded, run the installer.
  + Ensure to check the box that says "Add Python X.X to PATH" during the installation process (where X.X is the version number). This step is crucial for being able to use Python from the command line.

**Verify Version -** Open Command Prompt (cmd.exe) or PowerShell and type: python –version. This should display the installed Python version.

**5. Configure a data base (MySQL)**

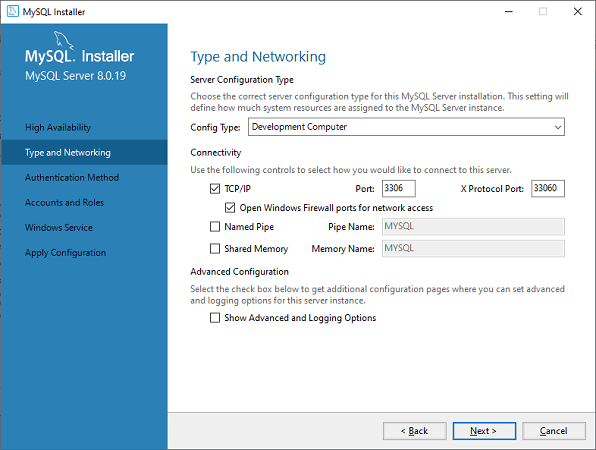
Installing MySQL involves several steps depending on your operating system. Here's a guide on how to install MySQL on Windows.

**Download MySQL Installer:**

* + Visit the MySQL Community Downloads page: MySQL Community Downloads.
  + Scroll down to find the MySQL Installer for Windows. Choose the appropriate version (typically MySQL Installer for Windows, which includes MySQL Server, MySQL Workbench, connectors, etc.).

**Run the Installer:**

* + Once downloaded, run the installer.
  + Choose the setup type. For most users, the default setup (Developer Default) is suitable, which includes MySQL Server, MySQL Workbench, connectors, etc.
  + Follow the installer prompts to complete the installation.

**Configuration:** During the installation process, you'll be prompted to configure MySQL. Set a root password for MySQL server and choose other configuration options as per your requirements. 

**Verify Installation:** Open MySQL Workbench from the Start menu. Connect to your MySQL Server using the root account and the password you set during installation.

**6. Explore Extensions and Plugins.**

Visual Studio Code (VS Code) is highly extensible with a vast ecosystem of extensions, plugins, and add-ons that enhance its functionality and cater to various programming languages, frameworks, and workflows. Here are some popular categories and examples of extensions for VS Code:

* General Productivity and UI Enhancements:

**Prettier** - Code formatter that supports various languages.

* Language support:

1. Python - Adds support for Python development.
2. Java Extension Pack - Includes essential tools for Java development (e.g., debugger, Maven support).

* Framework and libraries:

1. Reactjs code snippets - Adds snippets for React development.
2. Angular Essentials - Includes tools for Angular development.
3. **Docker** - Adds syntax highlighting, commands, and more for Dockerfiles.

* Devops and deployment;

1. **Docker** - Provides Docker support within VS Code.
2. **GitHub Pull Requests and Issues** - Manage GitHub pull requests and issues directly in VS Code.

* Theme and customization

1. **Material Theme** - Popular theme inspired by Google's Material Design.
2. **Dracula Official** - A dark theme for VS Code.

**7. Install package managers.**

In the context of Visual Studio Code (VS Code), package managers are tools that help manage **Python Package Index (PyPI)**: For Python development, PyPI is the standard repository for installing Python packages and libraries. VS Code extensions related to Python development often require Python packages, which are managed through pip (Python's package installer).

**Chocolatey**: Chocolatey is a package manager for Windows that can be used to install various software packages, including tools and utilities that may complement your VS Code setup.

Dependencies and extensions for your development environment. Here are some package managers commonly used with VS Code:

**Python Package Index (PyPI)**: For Python development, PyPI is the standard repository for installing Python packages and libraries. VS Code extensions related to Python development often require Python packages, which are managed through pip (Python's package installer).

**Chocolatey**: Chocolatey is a package manager for Windows that can be used to install various software packages, including tools and utilities that may complement your VS Code setup.

Python package index (pypl)

Python extensions often require specific Python packages. You can install these packages using pip, which you typically manage outside of VS Code but may involve the terminal within VS Code for convenience.

To install Python packages from the Python Package Index (PyPI), you typically use pip, which is the standard package installer for Python. Here's a step-by-step guide to installing Python packages using pip:

### Prerequisites:

1. **Python Installation**: Ensure Python is installed on your system. You can download Python from [python.org](https://www.python.org/downloads/) and follow the installation instructions.
2. **Set up PATH Environment Variable**: During Python installation, make sure to check the option "Add Python to PATH" so that you can run Python and pip commands from the command line.

### Installing Packages:

#### 1. Open a Command Prompt or Terminal:

* **Windows**: You can open Command Prompt or PowerShell.
* **MacOS/Linux**: Open Terminal.

#### 2. Verify pip Installation: Type pip --version or pip3 --version (depending on your Python installation) and press Enter. This command verifies that pip is installed and displays its version.

#### 3. Install a Package: To install a package from PyPI, use the following command: pip install package\_name

Replace package\_name with the name of the package you want to install. For example, to install requests, a popular HTTP library, you would type: *pip install requests*

Verify Installation: After installation, you can verify that the package is installed correctly by importing it in a Python script or running pip show package\_name to display information about the installed package.

Challenges faced during software set ups.

* App not installed properly; this was solved by using online video tutorials to check for the cause of the problem.
* In some cases I had to reconfigure the environment variables to ensure the software is installed in the correct path.