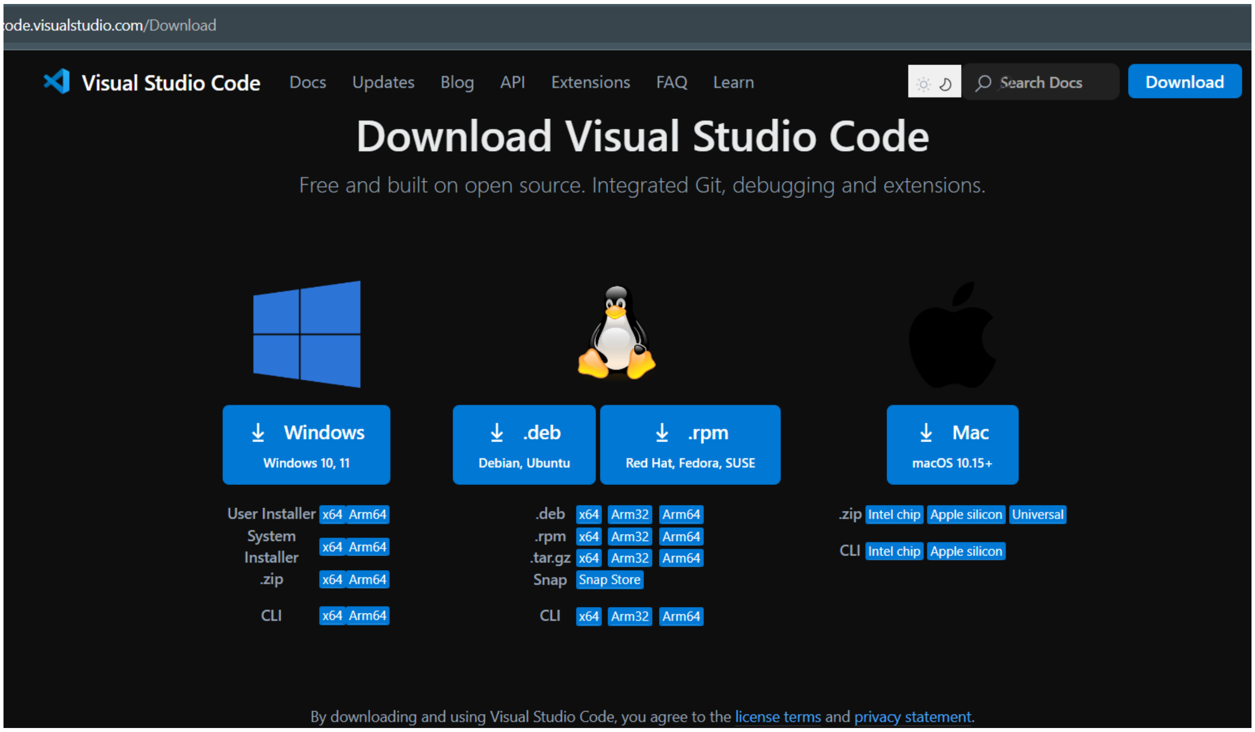
Installation and Navigation of Visual Studio Code (VS Code) Instructions

Installation of VS Code

Steps to Download and Install Visual Studio Code on Windows 11:

1. **Download VS Code:**
    - Visit the official Visual Studio Code website: Visual Studio Code.
    - Click on the "Download for Windows" button. This will download the installer for Windows.



Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

By downloading and using Visual Studio Code, you agree to the license terms and privacy statement.

2. **Run the Installer:**
    - Navigate to your Downloads folder and double-click on the downloaded file (e.g., VSCodeSetup-x64-1.x.x.exe).
3. **Accept the License Agreement:**
    - Read through the license agreement and, if you agree, check the "I accept the agreement" option and click "Next."
4. **Choose Installation Location:**
    - You can choose the default installation directory or specify a different one. Click "Next" to proceed.
5. **Select Additional Tasks:**

   - Options may include creating a desktop icon, adding to PATH (useful for command-line access), and associating VS Code with supported file types. It is recommended to add to PATH for easier access. Click "Next" to continue.

6. **Install:**
   - Click "Install" to begin the installation process.
7. **Launch VS Code:**
   - After the installation is complete, you can choose to launch VS Code immediately by checking the appropriate box and clicking "Finish."

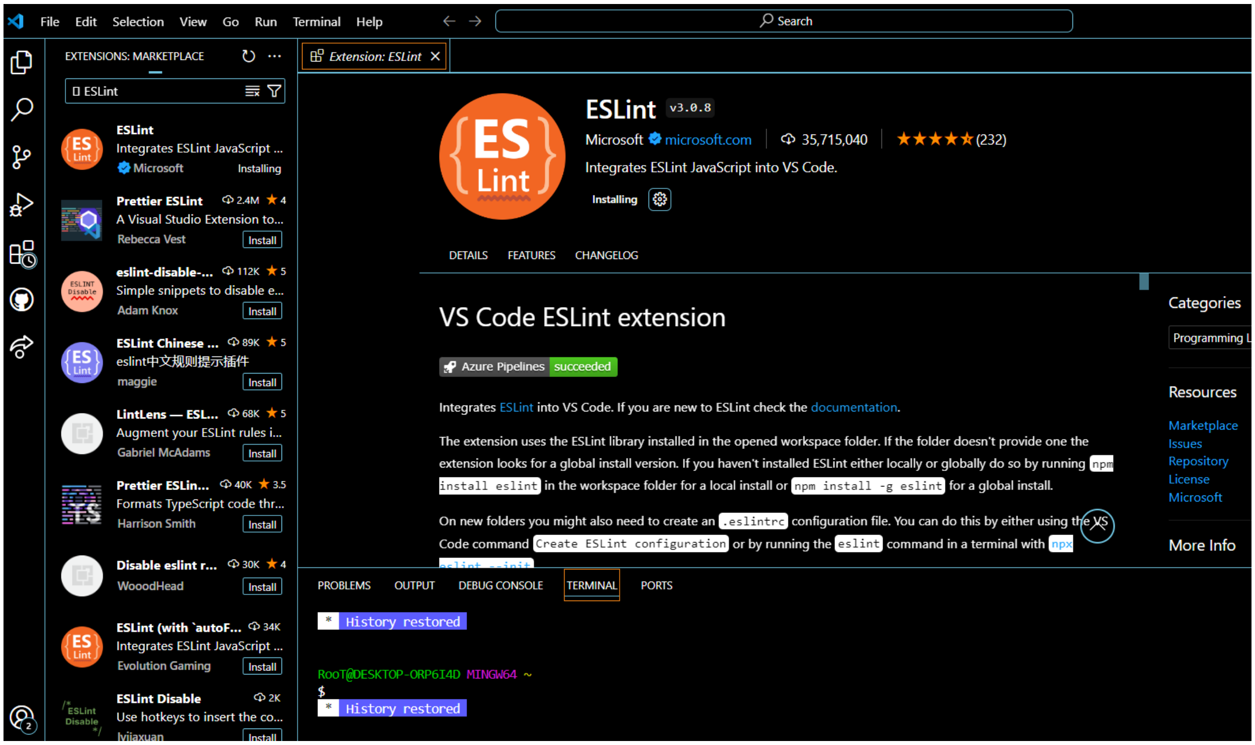
Prerequisites:

- **Operating System:** Windows 11.
- **Permissions:** Administrative privileges may be required for installation.
- **Disk Space:** Ensure sufficient disk space is available (at least 200 MB for VS Code itself).

First-time Setup

Initial Configurations and Settings for Optimal Coding Environment:

1. **Theme and Appearance:**
   - Go to File > Preferences > Color Theme to choose a theme that suits your preference, such as "Dark" or "Light."
2. **Extensions:**
   - Essential extensions for a good start include:
     - **Python** (for Python development)
     - **Prettier - Code Formatter** (for automatic code formatting)
     - **Live Server** (for a quick web server with live reload)
     - **ESLint** (for JavaScript/TypeScript linting)
     - **GitLens** (for enhanced Git capabilities)



VS Code ESLint extension

Integrates ESLint into VS Code. If you are new to ESLint check the documentation.

The extension uses the ESLint library installed in the opened workspace folder. If the folder doesn't provide one the extension looks for a global install version. If you haven't installed ESLint either locally or globally do so by running `npm install eslint` in the workspace folder for a local install or `npm install -g eslint` for a global install.

On new folders you might also need to create an `.eslintrc` configuration file. You can do this by either using the VS Code command `Create ESLint configuration` or by running the `eslint` command in a terminal with `npx eslint --init`

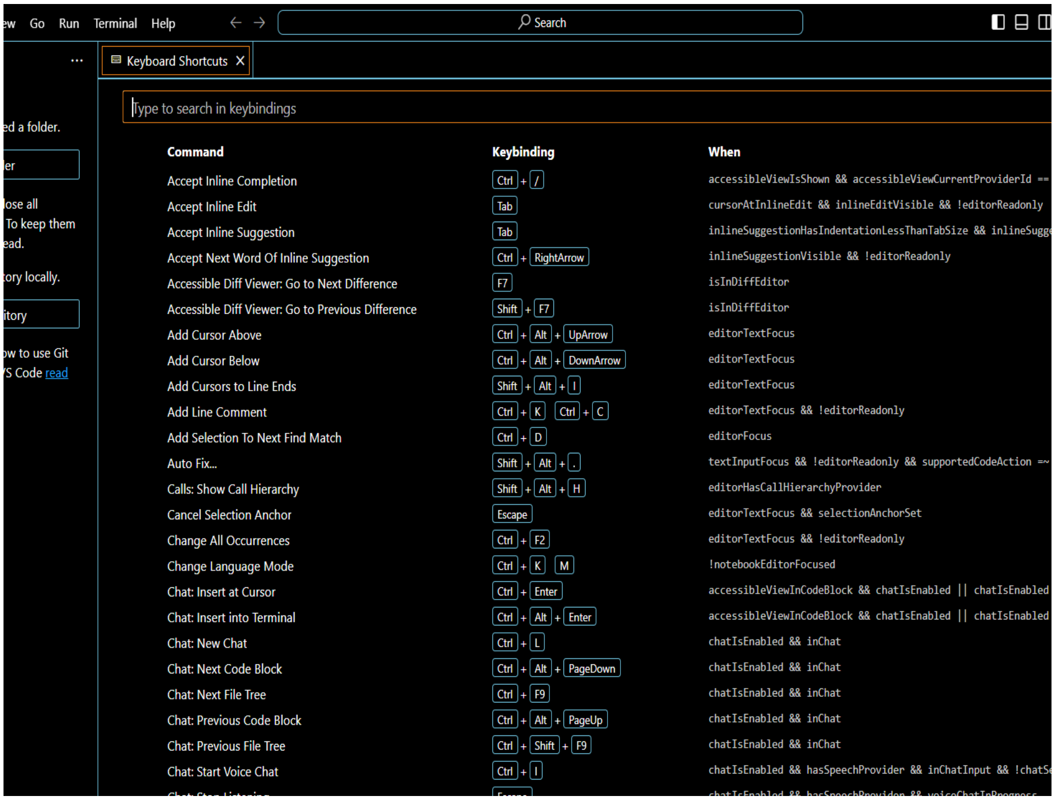
```
 *  History restored

RooT@DESKTOP-ORP6I4D MINGW64 ~
$
 *  History restored
```

3. **Settings Sync:**
   - Enable settings sync from the Manage gear icon in the lower left corner to synchronize your VS Code settings across devices.

4. **Workspace Settings:**
   - Configure settings specific to your workspace or project by going to File > Preferences > Settings and switching to Workspace settings.

5. **Keybindings:**
   - Customize keybindings via File > Preferences > Keyboard Shortcuts to tailor the shortcuts to your workflow.



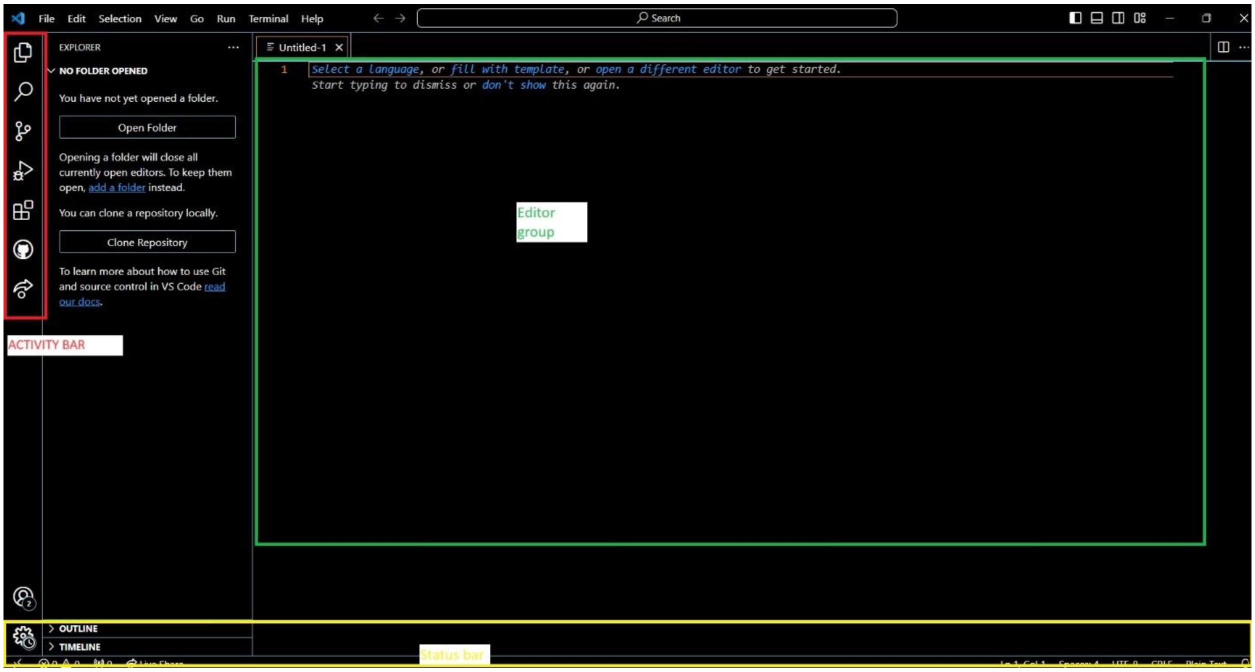
Type to search in keybindings

Command	Keybinding	When
Accept Inline Completion	Ctrl + /	accessibleViewIsShown && accessibleViewCurrentProviderId ==
Accept Inline Edit	Tab	cursorAtInlineEdit && inlineEditVisible && !editorReadonly
Accept Inline Suggestion	Tab	inlineSuggestionHasIndentationLessThanTabSize && inlineSugge
Accept Next Word Of Inline Suggestion	Ctrl + RightArrow	inlineSuggestionVisible && !editorReadonly
Accessible Diff Viewer: Go to Next Difference	F7	isInDiffEditor
Accessible Diff Viewer: Go to Previous Difference	Shift + F7	isInDiffEditor
Add Cursor Above	Ctrl + Alt + UpArrow	editorTextFocus
Add Cursor Below	Ctrl + Alt + DownArrow	editorTextFocus
Add Cursors to Line Ends	Shift + Alt + I	editorTextFocus
Add Line Comment	Ctrl + K Ctrl + C	editorTextFocus && !editorReadonly
Add Selection To Next Find Match	Ctrl + D	editorFocus
Auto Fix...	Shift + Alt + .	textInputFocus && !editorReadonly && supportedCodeAction =~
Calls: Show Call Hierarchy	Shift + Alt + H	editorHasCallHierarchyProvider
Cancel Selection Anchor	Escape	editorTextFocus && selectionAnchorSet
Change All Occurrences	Ctrl + F2	editorTextFocus && !editorReadonly
Change Language Mode	Ctrl + K M	!notebookEditorFocused
Chat: Insert at Cursor	Ctrl + Enter	accessibleViewInCodeBlock && chatIsEnabled || chatIsEnabled
Chat: Insert into Terminal	Ctrl + Alt + Enter	accessibleViewInCodeBlock && chatIsEnabled || chatIsEnabled
Chat: New Chat	Ctrl + L	chatIsEnabled && inChat
Chat: Next Code Block	Ctrl + Alt + PageDown	chatIsEnabled && inChat
Chat: Next File Tree	Ctrl + F9	chatIsEnabled && inChat
Chat: Previous Code Block	Ctrl + Alt + PageUp	chatIsEnabled && inChat
Chat: Previous File Tree	Ctrl + Shift + F9	chatIsEnabled && inChat
Chat: Start Voice Chat	Ctrl + I	chatIsEnabled && hasSpeechProvider && inChatInput && !chatS

User Interface Overview

Main Components of the VS Code User Interface:

1. **Activity Bar:**
   - Located on the far left side, it provides quick access to core functionalities such as the Explorer, Search, Source Control, Run and Debug, and Extensions. You can switch between different views and functionalities here.
2. **Side Bar:**
   - Displays detailed information and tools for the selected view from the Activity Bar. For example, in the Explorer view, it shows a list of files and folders in your project.
3. **Editor Group:**
   - The main area where you write and edit your code. Multiple files can be open in tabs here, and you can split the editor to view and edit multiple files side by side.
4. **Status Bar:**
   - Located at the bottom of the window, it provides useful information about your workspace and files, such as Git branch, line and column number, programming language of the current file, and errors/warnings.



You have not yet opened a folder.

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

You can clone a repository locally.

To learn more about how to use Git and source control in VS Code read our docs.

```
Select a language, or fill with template, or open a different editor to get started.
Start typing to dismiss or don't show this again.
```

Command Palette

What is the Command Palette and How to Access It:

- **Accessing the Command Palette:**
  - Press Ctrl + Shift + P (or F1) to open the Command Palette.
- **Common Tasks:**
  - **Open Settings:** Type "settings" and select "Preferences: Open Settings."
  - **Install Extensions:** Type "extension" and select "Extensions: Install Extensions."
  - **Toggle Sidebar Visibility:** Type "toggle sidebar" and select the relevant command.
  - **Git Commands:** Quickly access commands like "Git: Commit" or "Git: Push."

Extensions in VS Code

Role of Extensions in VS Code:

- Extensions extend the functionality of VS Code to support additional languages, debuggers, and tools, enhancing productivity and customization.

Finding, Installing, and Managing Extensions:

1. **Finding Extensions:**
   - Click the Extensions view icon in the Activity Bar or press Ctrl + Shift + X.
2. **Installing Extensions:**
   - Search for the extension you need, click on it, and then click the "Install" button.

3. **Managing Extensions:**
    - Installed extensions can be managed from the Extensions view. Click the gear icon next to an extension to access options like enabling/disabling, uninstalling, or configuring.

Essential Extensions for Web Development:

- **HTML Snippets:** Provides quick HTML boilerplate and snippets.
- **CSS IntelliSense:** Enhances CSS autocompletion and validation.
- **JavaScript (ES6) code snippets:** Provides ES6 code snippets for JavaScript.
- **Live Server:** Launches a local development server with live reload.
- **VS Code Icons:** Adds file icons to your file tree for better visibility.

Integrated Terminal

How to Open and Use the Integrated Terminal:

1. **Opening the Terminal:**
    - Use Ctrl + (backtick) or go to View > Terminal.
2. **Using the Terminal:**
    - The integrated terminal opens at the bottom of the editor, allowing you to run command-line tools directly within VS Code.
    - You can open multiple terminals and switch between them using tabs.

Advantages of Using the Integrated Terminal:

- **Convenience:** No need to switch between different applications.
- **Context Awareness:** The terminal operates within the context of the currently opened project or folder.
- **Customization:** Customize the terminal with different shells (e.g., PowerShell, Command Prompt, Git Bash).

File and Folder Management

Creating, Opening, and Managing Files and Folders:

1. **Creating Files/Folders:**
    - Right-click in the Explorer view and choose New File or New Folder.
    - Use Ctrl + N to create a new file quickly.
2. **Opening Files/Folders:**
    - Use Ctrl + O to open a file.

    - Use Ctrl + K, Ctrl + O to open a folder or workspace.

3. **Managing Files/Folders:**
    - Files and folders can be dragged and dropped to organize them.
    - Rename files by right-clicking and selecting Rename or pressing F2.

Navigating Between Files and Directories:

- Use Ctrl + P to quickly search and navigate to files.
- Use the file tabs at the top of the editor group to switch between open files.
- Use breadcrumbs (enabled via View > Appearance > Show Breadcrumbs) to navigate through file paths and symbols within a file.

Settings and Preferences

Where to Find and Customize Settings:

- **Accessing Settings:**
    - Go to File > Preferences > Settings or press Ctrl + ,.

Changing Themes, Font Size, and Keybindings:

1. **Change Theme:**
    - Go to File > Preferences > Color Theme and choose a new theme.
2. **Change Font Size:**
    - In the settings, search for "font size" and adjust the value in the Editor: Font Size field.
3. **Customize Keybindings:**
    - Go to File > Preferences > Keyboard Shortcuts or press Ctrl + K, Ctrl + S to view and modify keybindings.

Debugging in VS Code

Steps to Set Up and Start Debugging a Simple Program:

1. **Open Your Project:**
    - Open the folder containing your project files.
2. **Configure Debugging:**
    - Click on the Run and Debug icon in the Activity Bar or press Ctrl + Shift + D.
    - Click Create a launch.json file and select the environment (e.g., Node.js, Python).
3. **Set Breakpoints:**

  - Click in the margin next to the line number where you want to add a breakpoint.

4. **Start Debugging:**
   - Click the green play button in the Run and Debug view or press F5.

Key Debugging Features:

- **Breakpoints:** Set and manage breakpoints to pause code execution.
- **Step Over/Into/Out:** Navigate through your code line by line.
- **Watch Variables:** Monitor the values of variables.
- **Call Stack:** View the call stack to understand the flow of execution.
- **Debug Console:** Execute commands and evaluate expressions while debugging.

Using Source Control

Integrating Git with VS Code:

1. **Initialize a Repository:**
   - Open your project folder.
   - Click on the Source Control icon in the Activity Bar.
   - Click Initialize Repository.
2. **Making Commits:**
   - Stage changes by clicking the + icon next to the modified files.
   - Write a commit message in the message box and click the checkmark icon to commit.
3. **Pushing Changes to GitHub:**
   - Ensure you have a remote repository set up on GitHub.
   - Add the remote by typing git remote add origin <repository_url> in the integrated terminal.
   - Push your changes using git push -u origin main.
4. **Additional Features:**
   - Use the built-in Git support to handle branches, pull requests, and merge conflicts directly from within VS Code.