1. Installation of VS Code

Open your web browser and go to the [Visual Studio Code website](https://code.visualstudio.com/).

Click on the "Download for Windows" button. This will download the VS Code installer for Windows.

Once the download is complete, locate the installer file.

Double-click on the installer file to run it.

You will be greeted by the Visual Studio Code Setup Wizard. Click "Next" to continue.

Read the license agreement, and if you agree, select the checkbox to accept the agreement. Click "Next" to proceed.

Choose the destination folder where you want to install VS Code. The default location is usually fine for most users. Click "Next" to continue.

You will be presented with several additional tasks that you can configure during installation:

* Create a desktop icon: Adds a VS Code icon to your desktop.
* Add "Open with Code" action to Windows Explorer file context menu: Allows you to right-click on files and folders and open them with VS Code directly.
* Add "Open with Code" action to Windows Explorer directory context menu: Similar to the above, but for directories.
* Register Code as an editor for supported file types: Makes VS Code the default editor for file types it supports.
* Add to PATH: This allows you to run VS Code from the command line by typing code.
* Select the tasks you prefer and click "Next"

Click the "Install" button to begin the installation process. The installer will copy the necessary files to your computer.

Once the installation is complete, you will see a completion screen. You can choose to launch Visual Studio Code immediately by checking the "Launch Visual Studio Code" box. Click "Finish" to close the setup wizard.

1. First-time Setup

Basic Configuration

* Update VS Code: Ensure you have the latest version of VS Code.
* Theme and Appearance: Set a theme that suits your preference.
* Font Settings: Adjust the font family and size for better readability.
* Tab Settings: Configure tab settings such as tab size and whether to use spaces or tabs.
* Auto Save: Enable auto save to automatically save your changes.

Important Settings

1. Settings Sync:

Sync your settings across multiple devices by going to File > Preferences > Settings Sync and enabling it. This requires a GitHub or Microsoft account.

1. Integrated Terminal:

Customize the integrated terminal by going to Settings and searching for Terminal Integrated Font Family and Terminal Integrated Font Size to adjust the appearance of the terminal.

1. Code Snippets:

Create custom code snippets to speed up coding. Go to File > Preferences > User Snippets and select the language you want to create snippets for.

1. Linting and Formatting:

Ensure linting and formatting tools are configured to run on save. In Settings, search for Editor: Format On Save and Editor: Lint On Save.

1. Workspace Settings:

Customize settings for specific projects by using workspace settings. Go to File > Preferences > Settings and switch to the Workspace tab.

1. User Interface Overview

1. Activity Bar

Purpose: The Activity Bar is located vertically on the far left side of the VS Code window. It provides quick access to different views and functionalities within VS Code.

Components:

* Explorer: Allows you to navigate and manage files and folders within your project directory.
* Search: Provides search functionality across files and folders.
* Source Control: Integrates with version control systems like Git, displaying changes, commits, and branches.
* Run and Debug: Enables running and debugging applications directly from VS Code.
* Extensions: Manages VS Code extensions, allowing you to browse, install, and manage extensions for additional functionalities.

2. Side Bar

Purpose: The Side Bar is located next to the Activity Bar and contains additional views and panels related to your current project or workspace.

Components:

* Explorer: Shows the file and folder structure of your project directory, enabling navigation and file management operations.
* Search: Provides a dedicated view for search results across files and folders.
* Source Control: Displays information and actions related to version control, such as changes, commits, and branches.
* Extensions: Lists installed extensions and allows you to browse the marketplace for new ones.
* Debug: Provides debugging tools and views for managing breakpoints, call stacks, and variables during debugging sessions.

3. Editor Group

Purpose: The Editor Group is the central area of VS Code where you open and work with files and code.

Functionality:

* Tabbed Interface: Each file or editor you open appears as a tab within the Editor Group.
* Split View: Allows you to split the Editor Group vertically or horizontally to view and work with multiple files simultaneously.
* Maximize and Minimize: You can maximize an editor tab to occupy the entire Editor Group or minimize it to return to the tabbed view.

4. Status Bar

Purpose: The Status Bar is located at the bottom of the VS Code window and provides information and quick access to various features and settings.

Information and Controls:

* Language Mode: Displays the current programming language mode of the active file.
* Line and Column Numbers: Shows the current cursor position within the active file.
* Encoding: Indicates the file encoding format (e.g., UTF-8).
* End of Line (EOL) Character: Displays whether the file ends with a specific newline character (e.g., LF or CRLF).
* Indentation: Shows the current indentation type used in the active file.
* Spaces or Tabs: Indicates whether spaces or tabs are used for indentation.
* Notification Area: Provides notifications and quick actions, such as changing the file format or selecting a language mode.

Customization and Extensions

Customization: VS Code’s UI is highly customizable. You can adjust layouts, hide or show components, and configure themes and color schemes to suit your preferences.

Extensions: VS Code’s functionality can be extended through a vast ecosystem of extensions available in the marketplace, allowing you to add new features, language support, themes, and more.

1. Command Palette

The Command Palette in Visual Studio Code (VS Code) is a powerful tool that allows you to execute various commands and operations quickly without needing to navigate through menus or remember specific keyboard shortcuts.

Accessing the Command Palette

1. Keyboard Shortcut: Press Ctrl+Shift+P (Windows/Linux) or Cmd+Shift+P (Mac). This opens the Command Palette at the top center of the editor.
2. Menu Option: Click on View in the menu bar, then select Command Palette...

Examples of Common Tasks with the Command Palette

* File and Workspace Operations
* Editing and Navigation
* Version Control (Git)
* Extensions Management
* Settings and Preferences
* Debugging
* Tasks and Runners

1. Extensions in VS Code

Role of Extensions in VS Code

Extensions in Visual Studio Code (VS Code) play a crucial role in extending its functionality beyond the core editor features. They allow users to customize their development environment, add support for various programming languages, integrate with external tools, enhance productivity, and more.

Finding Extensions:

1. Extensions Marketplace:

The primary source for VS Code extensions is the Visual Studio Code Marketplace. Here, you can search for extensions by name, category, or functionality.

1. Built-in Extensions Gallery:

In VS Code, you can access the Extensions view by clicking on the Extensions icon in the Activity Bar (or pressing Ctrl+Shift+X). Here, you can search for and browse extensions directly within the editor.

Installing Extensions:

1. From Marketplace:

Click on the extension you want to install in the Marketplace, then click on the "Install" button.

Alternatively, you can install extensions from within VS Code by searching for them in the Extensions view and clicking on the "Install" button next to the extension.

1. Using Command Palette:

Open the Command Palette (Ctrl+Shift+P), type Extensions: Install Extensions, press Enter, then search for and select the extensions you want to install.

Managing Extensions:

1. Disabling and Enabling:

In the Extensions view, installed extensions can be disabled or enabled as needed by toggling the corresponding switch.

1. Updating Extensions:

VS Code automatically checks for updates to installed extensions. You can manually update extensions by clicking on the "Update" button in the Extensions view.

1. Uninstalling Extensions:

To uninstall an extension, go to the Extensions view, find the extension you want to remove, and click on the gear icon (⋮) next to it. Select "Uninstall" from the dropdown menu.

**Essential Extensions for Web Development**

* HTML/CSS/JavaScript Support
* Code Formatting and Linting
* Version Control (Git)
* Debugging
* Task Runners