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 (https://code.visualstudio.com/).

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

2. Run the Installer:

- Locate the downloaded file (e.g., `VSCodeSetup-x64-1.60.0.exe`) and double-click to run it.

- Follow the installation wizard steps. Accept the license agreement and choose the installation location.

3. Select Additional Tasks:

- During installation, you may be prompted to select additional tasks:

- Add "Open with Code" action to the Windows Explorer file context menu.

- Add "Open with Code" action to the Windows Explorer directory context menu.

- Register Code as an editor for supported file types.

- Add to PATH (useful for running `code` command in the terminal).

4. Complete Installation:

- Click "Install" to begin the installation process.

- Once the installation is complete, click "Finish" to launch VS Code.

Prerequisites:

- Operating System: Windows 11.

- Administrator Privileges: Required to install software.

- Internet Connection: Needed to download the installer and extensions.

First-time Setup

Initial Configurations and Settings:

1. Theme Selection:

- Go to `File > Preferences > Color Theme` and choose a theme that suits your preference.

2. Font Size and Family:

- Go to `File > Preferences > Settings`

- Search for "Font Size" and "Font Family" to adjust these settings.

3. Extensions:

- Essential extensions for web development:

- Live Server: Provides a local development server with live reload feature.

- Prettier - Code Formatter: Ensures consistent code formatting.

- ESLint: Integrates ESLint into VS Code to catch JavaScript errors and enforce coding standards.

- HTML CSS Support: Provides IntelliSense for HTML class names defined in CSS files.

User Interface Overview

Main Components of the VS Code User Interface:

1. Activity Bar:

- Located on the far left side of the window.

- Contains icons for different views (Explorer, Search, Source Control, Run, Extensions).

- Allows switching between different activities quickly.

2. Side Bar:

- Located next to the Activity Bar.

- Displays different views and functions depending on the selected activity (e.g., file explorer, search results).

3. Editor Group:

- Main area where files are opened and edited.

- Supports multiple tabs and split views for side-by-side editing.

4. Status Bar:

- Located at the bottom of the window.

- Displays information such as current line and column number, selected language mode, and Git branch status.

Command Palette

Command Palette:

- Accessing the Command Palette:

- Press `Ctrl+Shift+P` or `F1`.

- Common Tasks:

- Search and run any command (e.g., `Toggle Terminal`, `Format Document`).

- Access settings and preferences (`Preferences: Open Settings`).

- Manage extensions (`Extensions: Install Extensions`).

Extensions in VS Code

Role of Extensions:

- Enhance Functionality:

- Extensions add features and capabilities to VS Code, tailored to specific needs (e.g., programming languages, frameworks, tools).

Finding, Installing, and Managing Extensions:

1. Finding Extensions:

- Click the Extensions icon in the Activity Bar.

- Browse or search for extensions in the Marketplace.

2. Installing Extensions:

- Click the "Install" button next to the desired extension.

3. Managing Extensions:

- View installed extensions, disable, enable, or uninstall as needed.

Essential Extensions for Web Development:

- Live Server

- Prettier - Code Formatter

- ESLint

- HTML CSS Support

Integrated Terminal

Opening and Using the Integrated Terminal:

1. Open Terminal:

- Use `Ctrl+` (backtick) or go to `View > Terminal`.

2. Advantages:

- Integrated with the editor, allowing seamless switching between code and terminal.

- Supports multiple terminals, split views, and different shells (e.g., PowerShell, Command Prompt, Git Bash).

File and Folder Management

Creating, Opening, and Managing Files and Folders:

1. Creating:

- Right-click in the Explorer view and select "New File" or "New Folder".

- Use `Ctrl+N` for a new file.

2. Opening:

- Use `File > Open File` or `File > Open Folder`.

- Drag and drop files or folders into the editor.

3. Navigating:

- Use the Explorer view to navigate the file structure.

- Use `Ctrl+P` to quickly open files by name.

Settings and Preferences

Customizing Settings:

- Accessing Settings:

- Go to `File > Preferences > Settings`

- Changing Theme:

- Search for "Color Theme" and choose a preferred theme.

- Adjusting Font Size:

- Search for "Font Size" and set the desired value.

- Changing Keybindings:

- Go to `File > Preferences > Keyboard Shortcuts`

Debugging in VS Code

Setting Up and Starting Debugging:

1. Open the Debug View:

- Click the Run icon in the Activity Bar.

2. Configure Debugger:

- Click "create a launch.json file" to configure debugger settings for your project.

3. Starting Debugging:

- Set breakpoints by clicking in the gutter next to the line numbers.

- Click the green play button in the debug view to start debugging.

Key Debugging Features:

- Breakpoints: Pause program execution at specific lines.

- Watch: Monitor variable values during execution.

- Call Stack: View the call stack and navigate through functions.

- Debug Console: Execute commands and evaluate expressions.

Using Source Control

Integrating Git with VS Code:

1. Initializing a Repository:

- Open the Source Control view by clicking the Source Control icon.

- Click "Initialize Repository".

2. Making Commits:

- Stage changes by clicking the plus icon next to the files.

- Enter a commit message and click the checkmark icon to commit.

3. Pushing Changes to GitHub:

- Click the ellipsis (...) in the Source Control view and select "Push".

- Ensure the remote repository is set up (GitHub integration may require authentication).