Installation of VS Code on Windows 11

1. Download VS Code:

- Go to the [Visual Studio Code website](https://code.visualstudio.com/).

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

2. Run the Installer:

- Open the downloaded file (`VSCodeSetup.exe`).

- Follow the installation wizard:

- Accept the license agreement.

- Choose the installation location (default is usually fine).

- Select additional tasks (e.g., create a desktop icon, add to PATH).

- Click “Install” to start the installation.

- Once finished, click “Finish” to launch VS Code.

3. Prerequisites:

- Administrative rights to install software.

- (Optional) Git for version control ([download Git](https://git-scm.com/)).

First-time Setup

1. Settings Sync:

- Sign in with a Microsoft or GitHub account to sync settings across devices.

2. Theme and Appearance:

- Go to `File > Preferences > Color Theme` to select a preferred theme.

- Adjust font size in `File > Preferences > Settings` and search for `editor.fontSize`.

3. Extensions:

- Install essential extensions:

- Prettier (code formatter)

- ESLint (linting for JavaScript)

- Live Server (local development server)

- Python (support for Python language)

- Access the Extensions view by clicking the Extensions icon in the Activity Bar or pressing `Ctrl+Shift+X`.

4. Workspace Settings:

- Configure workspace-specific settings in `.vscode/settings.json`.

User Interface Overview

1. Activity Bar:

- Located on the left side.

- Provides access to views like Explorer, Search, Source Control, Run & Debug, and Extensions.

2. Side Bar:

- Displays different views (Explorer, Search, Source Control, etc.) based on the selected Activity Bar icon.

3. Editor Group:

- Central 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.

- Shows information about the opened file, like language mode, Git branch, errors and warnings, and more.

Command Palette

Accessing:

- Press `Ctrl+Shift+P` or `F1` to open the Command Palette.

Common Tasks:

- Changing settings (`Preferences: Open Settings`)

- Installing extensions (`Extensions: Install Extensions`)

- Running tasks (`Tasks: Run Task`)

- Viewing keyboard shortcuts (`Preferences: Open Keyboard Shortcuts`)

Extensions in VS Code

Finding and Installing:

- Click on the Extensions icon in the Activity Bar or press `Ctrl+Shift+X`.

- Search for desired extensions and click `Install`.

Managing Extensions:

- Disable or uninstall extensions from the Extensions view.

- Update extensions via the gear icon next to the installed extension.

Essential Extensions for Web Development:

Prettier: Code formatter.

ESLint: JavaScript linter.

Live Server: Launch a development local server.

IntelliSense for CSS/HTML: Enhanced HTML/CSS editing support.

Bracket Pair Colorizer: Colorize matching brackets.

Integrated Terminal

Opening Terminal:

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

Advantages:

- Integrated within the editor for a seamless workflow.

- Supports multiple terminals and split views.

- Shares the same directory context as the opened project.

File and Folder Management

Creating and Opening Files

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

- Double-click to open files or drag and drop to manage them.

Navigating:

- Use the `Explorer` in the Side Bar.

- Quick Open (`Ctrl+P`) to quickly navigate to files.

- Breadcrumb navigation at the top of the editor for easy directory traversal.

Settings and Preferences

Finding Settings:

- Go to `File > Preferences > Settings` or press `Ctrl+,`.

Customizing:

- Change themes via `File > Preferences > Color Theme`.

- Adjust font size under `Text Editor > Font > Font Size`.

- Customize keybindings in `File > Preferences > Keyboard Shortcuts`.

Debugging in VS Code

1. Setup:

- Open the Debug view by clicking the Run icon in the Activity Bar or pressing `Ctrl+Shift+D`.

- Create a launch configuration file (`launch.json`) if not automatically generated.

2. Start Debugging:

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

- Click the green play button in the Debug view or press `F5` to start debugging.

3. Key Features:

- Variable inspection, call stack navigation, watch expressions, and interactive debugging console.

Using Source Control

1. Integrating Git:

- Open the Source Control view by clicking the Git icon in the Activity Bar or pressing `Ctrl+Shift+G`.

- If your project is not already a Git repository, initialize it by clicking on `Initialize Repository`.

2. Making Commits:

- Stage changes by clicking the `+` next to the file in the Source Control view.

- Enter a commit message in the input box and click the checkmark to commit.

3. Pushing Changes:

- Ensure you are signed into GitHub or other Git service.

- Click the `...` in the Source Control view and select `Push`.