1. Describe the steps to download and install Visual Studio Code on Windows 11 operating system. Include any prerequisites that might be needed.

Answer

Open your web browser and navigate the Visual Studio Code website

On the website click download for windows, this will download the VS Code installer

Once the download is complete, locate the VSCode Setup file in your Downloads folder then double-click the installer file to run it.

During installation agree to the license agreement then choose the destination folder where you want to install VS Code

Click install to begin the installation

When you open Visual Studio Code for the first time, it may prompt you to install extensions based on your development needs. Follow the prompts to customize your setup.

1. After installing VS Code, what initial configurations and settings should be adjusted for an optimal coding environment? Mention any important settings or extensions

Answer

Theme and appearance – choose a comfortable color theme and adjust the font size and family for better readability

Editor settings – enable auto save to automatic save your files, configure the tab size spaces

Extensions – download all the recommended extensions and also the extensions that are recommended for certain programming languages

Terminal integration – customize the integrated terminal to your preferred shell

Version control – Ensure git is properly configured and enable git auto fetch to keep your branches up to date

1. Explain the main components of the VS Code user interface. Identify and describe the purpose of the Activity Bar, Side Bar, Editor Group, and Status Bar.

Answer

Activity Bar - It provides quick access to different views and tools within VS Code which allows users to switch between different activities quickly. Features include;

1. Search
2. Source control
3. Explorer
4. Run and debug
5. Extensions

Editor Interface - is the central part of the VS Code where you edit your code and enables one to have multiple editor groups open simultaneously. Features include;

1. Tabs
2. Split editor
3. Minimap
4. Different view – displays differences between files

Status Bar – is located at the bottom of the VS Code interface. It provides information and shortcuts related to the current context or activity. Features include;

* **Current File Information**: Displays information about the current file, such as line and column numbers, file encoding, and file type.
* **Git Branch and Status**: Shows the current Git branch and the status of your repository
* **Errors and Warnings**: Indicates the number of errors and warnings in your code.
* **Language Mode**: Displays the programming language of the current file and allows you to change the language mode.
* **End-of-Line Sequence**: Shows the current end-of-line sequence used in the file.
* **Line Endings and Indentation**: Allows you to quickly change the line endings and indentation settings for the current file.
* **Notifications**: Displays notifications and messages from VS Code or extensions.

4. What is the Command Palette in VS Code, and how can it be accessed? Provide examples of common tasks that can be performed using the Command Palette.

Answer

Command Palette is a feature that allows you to access and execute various commands quickly without navigating the menus and it can be accessed by pressing ‘F1’ or ‘Ctrl+Shift+P’. Examples include;

* Type ‘Open File’ to open the file picker.
* Type ‘Git: Commit’ to commit changes.
* Type ‘Git: Pull’ to pull changes from the remote repository.
* Type ‘Git: Push’ to push changes to the remote repository.
* Type Tasks: Run Task to run a predefined task.
* Type Tasks: Configure Task to create or edit tasks.

1. Discuss the role of extensions in VS Code. How can users find, install, and manage extensions? Provide examples of essential extensions for web development.

Answer

Finding and Installing extensions – Extensions are found in **Extension Marketplace**: VS Code has an extensive marketplace accessible through the Extensions view in the activity bar where users can browse thousands of extensions categorized by functionality, language support, and can be able to install by clicking on the install button.

Managing Extensions - In VS Code, the Extensions view (Ctrl+Shift+X) lists all installed extensions. From here, users can enable or disable, uninstall, update, and configure extensions.

Examples - Live Server:

Purpose: Launches a local development server with live reload feature for static and dynamic pages and for rapid testing and development of web pages without manually refreshing the browser.

1. Describe how to open and use the integrated terminal in VS Code. What are the advantages of using the integrated terminal compared to an external terminal

Answer

Opening the Terminal – By using the menu navigate to View > Terminal and if you have multiple terminals open, you can switch between them by clicking on the terminal tabs at the top or by using the keyboard shortcut Ctrl+``

Advantages;

1. Integrate terminal commands with VS Code tasks for automating build processes or running scripts directly from the editor
2. Debug terminal output directly within VS Code is useful when troubleshooting or inspecting output from build processes or server logs.
3. It is easy to manage multiple projects simultaneously by having a separate terminal instance for each workspace, keeping their contexts isolated.
4. Explain how to create, open, and manage files and folders in VS Code. How can users navigate between different files and directories efficiently?

Answer

By using the menu navigate to File and then click to New text file, this is to create a new text of your choice using any language of choice

By using the Explorer bar there is an option of opening a new folder where your text files will be stored

In managing of files, one is able to rename, delete or copy/move files or folders between different locations within the Explorer view to move them.

File Navigation Shortcuts:

Go to File: Press Ctrl+P to open the Quick Open dialog and type the name of the file you want to open.

Switch Between Tabs: Use Ctrl+Tab to cycle through open editor tabs.

Explorer View Navigation:

Expand/Collapse Folders: Use the arrow icons next to folder names in the Explorer view to expand or collapse directory structures.

Breadcrumb Navigation: Click on items in the breadcrumb navigation bar at the top of the editor to quickly navigate up and down directory levels.

Workspace Navigation:

Go to Symbol: Use Ctrl+Shift+O to open the Go to Symbol dialog, where you can quickly navigate to functions, classes, or symbols within the current file.

Navigate Back and Forward: Use Alt+Left (Back) and Alt+Right (Forward) arrows to navigate through recently visited files.

Command Palette:

Use Ctrl+Shift+P to open the Command Palette, where you can execute various commands related to file and folder management, such as creating new files or opening specific files.

1. Where can users find and customize settings in VS Code? Provide examples of how to change the theme, font size, and key bindings.

Answer

By accessing settings via menu and quick open the settings. VS Code provides two types of settings; the user interface settings and the workspace settings.

**Customizing Settings**

1. Changing the Theme

Go to Settings (Ctrl+, or Cmd+,).

In the search bar, type "Color Theme".

Click on the dropdown under "Color Theme" to select from installed themes.

1. Changing Font Size

Go to Settings (Ctrl+, or Cmd+,).

In the search bar, type "Font Size".

Adjust the "Editor: Font Size" slider to change the font size.

1. Changing Key Binding

Go to Settings (Ctrl+, or Cmd+,).

In the search bar, type "Key bindings".

Click on "Open Keyboard Shortcuts (JSON)" to edit key bindings in JSON format.

1. Outline the steps to set up and start debugging a simple program in VS Code. What are some key debugging features available in VS Code

Answer

1. Install Required Extensions:

Ensure you have the necessary extensions installed for your programming language. For example, for Python, you might need the "Python" extension.

1. Open Your Project:

Open VS Code and open the folder or workspace containing your project.

1. Create or Open a Debug Configuration:

Click on the Debug icon in the Activity Bar on the side (or use Ctrl+Shift+D).

Click on the gear icon to open launch. Json and create or select a configuration for your project.

1. Navigate to the file where you want to debug.

Click in the gutter (the area where line numbers are shown) next to the line where you want to set a breakpoint. A red dot will appear indicating the breakpoint.

1. Start Debugging:

Press F5 or click the green play button next to the configuration in the Debug view to start debugging.

VS Code will launch your program in debugging mode and pause at the breakpoints you have set.

**Key Debugging Features in VS Code**

Breakpoints - used to pause the execution of your code at specific points to inspect variables and state.

Variable Inspection: Hover over variables to see their current values, or view them in the Variables panel.

Call Stack – used to navigate through the call stack to understand the flow of your program and inspect function calls.

Debug Console -Executes commands or evaluate expressions in the Debug Console during debugging.

Watch Expressions - Monitors the values of specific variables by adding them as watch expressions.

Conditional Breakpoints - Sets breakpoints that only trigger when certain conditions are met.

Multi-session Debugging - Debugs multiple sessions concurrently, such as debugging a client and server simultaneously.

Integrated Terminal - Uses the integrated terminal for debugging tasks that require command-line interaction.

1. How can users integrate Git with VS Code for version control? Describe the process of initializing a repository, making commits, and pushing changes to GitHub.

Answer

To integrate Git with VS Code for version control, follow these steps:

1) Install the Git extension for VS Code. This extension provides Git integration and allows you to manage your repositories directly from the VS Code interface.

2) Open your VS Code workspace and open the command palette by pressing `Ctrl + Shift + P` or by clicking the "Command Palette" icon in the left sidebar.

3) Type "Git: Initialize Repository" and press enter to initialize a new Git repository in your workspace. This will create a `.git` directory in your project directory and begin tracking changes to your files.

4) To stage changes to your repository, navigate to the Git Changes panel by clicking the "Source Control" icon in the left sidebar or by pressing `Ctrl + Shift + G`. In this panel, you can see a list of the changes made to your files since the last commit.

5) To stage a change, click on the checkbox next to the file or folder you want to stage. You can also use the "Stage All" button at the top of the panel to stage all changes.

6) Once you have staged your changes, you can commit them to your repository by clicking the "Commit" button in the Git Changes panel. This will open a commit message editor where you can enter a description of the changes you are committing.

7) After you have entered your commit message and clicked the "Commit" button, your changes will be saved in your local repository.

8) To push your changes to a remote repository, such as a GitHub repository, you need to add the remote repository as a remote in your local repository. To do this, open the command palette and type "Git: Remote - Add" and press enter. Enter the URL of the remote repository and click the "Add" button.

9) Once you have added the remote repository, you can push your changes to it by clicking the "Push" button in the Git Changes panel or by running the command "Git: Push" from the command palette.