JOSHUA ONESMO MWILENGA

SE-Assignment-5

Email: mwilenga20@gmail.com

1. Installation for Visual Studio Code (VS code).

The following are the main pre-requisite for installing VS code in any computer.

a) Operating System
   A computer must have an installed operating system.
   For example windows, Linux or Mac OS.
b) System Requirements
   - ✓ Processor of at least 1.6 GHz or faster one.
   - ✓ RAM: 4GB or more.
   - ✓ Hard Disk: minimum of 200MB of available disk for installation.
   - ✓ Display: 1024*768 display resolution or higher.
c) Internet Connection, for downloading the installation package from official website.

The following are the steps to follow for installing VS code in a window Operating system

i. Visit the official website to download: Visual Studio Code - Code Editing. Redefined, then click on the download button for Windows.
ii. Run Installer: Once the download is complete, run the installer (VSCodeSetup.exe). and follow the prompts in the installation wizard.
iii. Launching VS Code:
   - ❖ After installation, you can launch VS Code from the Start Menu or by searching for "Visual Studio Code".
iv. Optional - Install Extensions:
   - ❖ Open VS Code and navigate to the Extensions view or use the keyboard shortcut (Ctrl+Shift+X).
   - ❖ Search for and install any extensions you need for your development work.

2. First Setup: after installing VS code in the system the following are the important settings and configuration for optimum coding environment
   a) Setting the theme
      Go to file then preference then color theme then choose the theme that best suits your preference. (For example Dark+ or Light+).
   b) Font and Font Size
      Go to file > preferences > settings > search for “Font Family” and font size to set your preferred font and size
   c) Auto Save
      In settings set for auto save to reduce the risk of losing your codes through manual saving.

Extensions:

a. Programing Language Support
   Install language-specific extensions like Python, JavaScripts (ES6), C++, SQL tools and Java.
b. Version Control
   - ➢ GitLens for enhanced Git capabilities.
   - ➢ Git Graph for visualizing your Git repository.
c. Debugging
   Install language –specific debuggers like Python and javaScript debugger.

3. User Interface Overview

The Visual Studio Code (VS Code) user interface is designed to be intuitive and efficient for developers. Here are the main components:

a. Activity Bar

The Activity Bar is located on the far left side of the window. It provides access to different views, such as:

- ❖ Explorer: Shows your project's files and folders.
- ❖ Search: Allows you to search across your files.
- ❖ Source Control: Integrates with version control systems like Git.
- ❖ Run and Debug: Manages debugging sessions.
- ❖ Extensions: Allows you to manage your extensions.

b. 2. Side Bar

The Side Bar is next to the Activity Bar and changes content depending on the selected view from the Activity Bar. Common views include:

- ❖ Explorer: Displays the file and folder structure of your workspace.
- ❖ Search: Provides a search input and displays search results.
- ❖ Source Control: Shows changes, branches, and other Git-related information.
- ❖ Run and Debug: Displays debug configurations and controls.
- ❖ Extensions: Lists installed extensions and recommendations.

c. Editor Group

The Editor Group is the central part of the interface where you write your code. You can open multiple files in tabs and split the editor into multiple groups for side-by-side editing.

d. Status Bar

The Status Bar is located at the bottom of the window. It provides information about the current workspace, such as:

- ❖ Branch: Current Git branch.
- ❖ Language Mode: Current programming language mode.
- ❖ Encoding: File encoding.
- ❖ Line/Column: Current cursor position.
- ❖ Errors and Warnings: Count of errors and warnings in the file.

4. Command Palette

   Meaning: Is a powerful feature that allows the user of VS code to access and excute various commands and functions within the editor quickly.
   It's essentially a quick access tool for nearly all of VS Code's functionality, providing an efficient way to navigate and control the editor without using the mouse or menu bars.

   Command palette in VS code can be accessed via keyboard shortcut (Ctrl+Shift+P) on window OS. It allows the user to run commands, search for files, and access settings without using the mouse.

   <u>Examples of common tasks that can be performed using command palette.</u>

   i. Execute a Command

      - ❖ Open the Command Palette and start typing the name of the command you want to run. For example, type `format document` to format the current document.

   ii. Access Settings
      - ❖ Type `settings` to quickly navigate to different settings pages.

   iii. Install Extensions

      - ❖ Type `install extensions` to open the Extensions view and search for new extensions to install.

   iv. Open Files and Symbols

      - ❖ Use `Ctrl+P` on window OS to quickly open files by name.
      - ❖ Use `Ctrl+T` on window OS to search and navigate to symbols in your workspace.

5. Extensions in VS code.
   Extensions play a crucial role in Visual Studio Code (VS Code), significantly enhancing its functionality and allowing developers to tailor the editor to their specific needs.

   The role of extension in VS code

   i. Language support
      Extension in vs code help the user to install different programing languages for the specific needs and use such as Python and JavaScripts.

ii. Debugging

Extensions can add debugging capabilities for different languages and frameworks. They allow you to set breakpoints, inspect variables, and control the execution flow.

- Python: python debugging
- JavaScripts Debugger: Debug JavaScript and TypeScript code.

iii. Source Control Integration

Extensions integrate various version control systems directly into VS Code, providing features like diffing, committing, branching, and merging.

- GitLens: Supercharges the Git capabilities built into VS Code.
- Git Graph: Visualize your Git repository and perform Git actions.

iv. Code Formatting and Linting

Extensions help maintain code quality by providing formatting and linting tools that automatically format your code and highlight potential issues.

- ESLin**t**: Integrates ESLint into VS Code.
- Prettier: An opinionated code formatter that supports many languages.

v. Remote Development

Extensions enable remote development capabilities, allowing you to work on code running on remote machines or containers as if they were local.

- Remote - SSH: Connect to and work with remote SSH servers.
- Remote - WSL: Develop within the Windows Subsystem for Linux.
- Remote - Containers: Work with Docker containers

vi. Framework and Library Support

Extensions provide support for popular frameworks and libraries, offering specific tools and commands to make development easier.

- Angular: Support for Angular development.
- React Native Tools: Debug and develop React Native applications.

Managing Extensions

- Installing: Extensions can be installed from the Extensions view in VS Code (`Ctrl+Shift+X` or `Cmd+Shift+X`).
- Updating: Extensions can be updated through the Extensions view when new versions are available.
- Disabling/Uninstalling**:** Extensions can be disabled or uninstalled if no longer needed.

Exmples of Extensions

- Python Debugger
- SQlite
- GitLens
- Liveshare

6. Integrated Terminal

   The integrated terminal in Visual Studio Code (VS Code) is a powerful tool that allows you to run command-line tasks directly within the editor, without switching to a separate terminal application. Here's how to open and use the integrated terminal in VS Code:

Opening the Integrated Terminal

- Using the Menu**:**
  - Navigate to the menu bar at the top of the window.
  - Click on `View` and then select `Terminal`.
  - Alternatively, you can go to `Terminal > New Terminal`.
- Using Keyboard Shortcuts**:**
  - On Windows/Linux: Press `Ctrl+` (backtick key).
  - On macOS: Press `Cmd+` (backtick key).

Using the Integrated Terminal

Once the terminal is open, it appears at the bottom of the VS Code window. You can perform various tasks, including:

Basic Commands

- Run Commands**:** Type any command as you would in a regular terminal. For example, `ls` (or `dir` on Windows) to list directory contents, `cd` to change directories, etc.
- Execute Scripts: Run scripts for your projects, like `npm start` for Node.js applications, `python script.py` for Python scripts, etc.

Using the integrated terminal in Visual Studio Code (VS Code) offers several key benefits:

a) Convenience: Perform all tasks within VS Code, reducing the need to switch between applications.
b) Context Awareness: Automatically sets the working directory to your project folder, and integrates with VS Code tasks.
c) Workflow Integration: Errors and warnings link directly to the editor, and it works seamlessly with debugging features.
d) Customization: Use different shells and customize settings like font size and profiles.

e) Multiple Terminals: Easily manage and split multiple terminal instances.
f) Integrated Features: Access commands through the Command Palette and integrate with many extensions.
g) Consistent Appearance: Follows VS Code's theme and settings, and can be synced across environments.
h) Efficiency: Reduces the need for external terminals, saving system resources.

7. File and Folder Management.

- Managing files and folders in Visual Studio Code (VS Code) involves straightforward actions directly within the editor:

    - ❖ Creating: Easily create new files and folders using the Explorer or keyboard shortcuts (`Ctrl+N` for files, `Ctrl+Shift+N` for folders).
    - ❖ Opening: Open files by double-clicking in the Explorer, using Quick Open (`Ctrl+P`), or via the menu (`File > Open Folder...`).
    - ❖ Renaming and Deleting: Rename or delete files/folders by right-clicking in the Explorer or using keyboard shortcuts (`F2` to rename, `Delete` to delete).
    - ❖ Moving and Copying: Move or copy files/folders by dragging within the Explorer or using `Ctrl+C`, `Ctrl+X`, `Ctrl+V` (Windows/Linux) or `Cmd+C`, `Cmd+X`, `Cmd+V` (macOS).
    - ❖ Additional Tips: Search within files (`Ctrl+Shift+F`), collapse/expand folders in the Explorer, and use context menu options for specific actions.

Therefore managing files and folders directly in VS Code integrates seamlessly with coding tasks, enhancing productivity by keeping everything within a unified development environment.

- To navigate efficiently between files and directories in Visual Studio Code (VS Code):

    - ➢ Use the File Explorer:
        - o Navigate and organize files using the Explorer view in the Activity Bar.
    - ➢ Quick Navigation:
        - o Use `Ctrl+P` (Windows/Linux) or `Cmd+P` (macOS) for Quick Open to jump to specific files by name.
    - ➢ Tab Management:
        - o Switch between open tabs with `Ctrl+Tab` (Windows/Linux) or `Cmd+Tab` (macOS).
    - ➢ Keyboard Shortcuts:
        - o Use `F12` to go to definitions and `Ctrl+Shift+O` (Windows/Linux) or `Cmd+Shift+O` (macOS) for symbol navigation.
    - ➢ Command Palette:
        - o Access file and directory commands via `Ctrl+Shift+P` (Windows/Linux) or `Cmd+Shift+P` (macOS).
    - ➢ History Navigation:

    - Navigate back and forward with `Alt+Left Arrow` (Windows/Linux) or `Cmd+Left Arrow` (macOS) and `Alt+Right Arrow` (Windows/Linux) or `Cmd+Right Arrow` (macOS).
- Extensions:
    - Consider using extensions like "Bookmarks" for marking and quick navigation.

8. Settings and Preferences
   Users can find and customize settings in VS Code by accessing the "Settings" menu located under the gear icon in the Activity Bar on the side of the window.

- Examples of how to change the theme, font size, and keybindings in Visual Studio Code (VS Code):

    I. Change Theme:
        - Go to `File` > `Preferences` > `Color Theme`, then select your preferred theme from the list.
    II. Adjust Font Size:
        - Navigate to `File` > `Preferences` > `Settings`.
        - Search for `editor.fontSize` and adjust the value to change the font size.
    III. Customize Keybindings:
        - Open `File` > `Preferences` > `Keyboard Shortcuts` to view and customize keybindings.
        - Use the search bar to find specific commands and assign new keybindings as desired.

9. Debugging in VS code.

To set up and start debugging a simple program in Visual Studio Code (VS Code):

- Install Required Extensions:
    - Ensure relevant language extensions (e.g., for Python, JavaScript) are installed for debugging support.
- Open Your Project:
    - Open your project folder in VS Code.
- Set Breakpoints:
    - Navigate to the file containing your code.
    - Click in the gutter next to the line number to set a breakpoint.
- Start Debugging:
    - Press `F5` or go to `Run` > `Start Debugging` to launch the debugger.
    - Select the appropriate environment or configuration if prompted (e.g., Node.js for JavaScript).
- Interact with Debugging Tools:
    - Use the debug toolbar to step through code (`F10` for step over, `F11` for step into), inspect variables, and monitor call stacks.
- Review Output:

- Check the Debug Console for output, errors, and debugging information.

By following these steps, you can effectively set up and debug your code directly within VS Code, enhancing your development workflow.

- Key debugging features available in Visual Studio Code (VS Code) include:

  - Breakpoints: Set breakpoints to pause execution at specific lines of code.
  - Step-through Debugging: Step over (`F10`), step into (`F11`), and step out (`Shift+F11`) of code execution.
  - Watch Expressions: Monitor the value of variables and expressions in real-time.
  - Call Stack: View the current execution path and navigate through function calls.
  - Debug Console: Interact with the program during debugging sessions, execute commands, and view output directly.
  - Conditional Breakpoints: Set breakpoints that only trigger when specific conditions are met.
  - Variable Inspection: Hover over variables to see their current values and properties.

These features help developers efficiently diagnose and resolve issues in their code while debugging in VS Code.

10. Using Source Control

To integrate Git with VS Code for version control:

i. Install Git: Install Git on your system if not already installed.
ii. Open Your Project: Open your project folder in VS Code.
iii. Initialize Git: Use the integrated terminal (`Ctrl+``) to initialize Git (`git init`) in your project directory.
iv. Stage and Commit Changes: Use the Source Control view (Ctrl+Shift+G) to stage files and commit changes with commit messages.
v. Push and Pull: Use Git commands (`git push`, `git pull`) in the terminal to sync changes with remote repositories.

The process of initializing a repository, making commits, and pushing changes to GitHub.:

a. Initialize a Repository:
   - Open your project in VS Code.
   - Open the integrated terminal (`Ctrl+``) and run` git init` to initialize a Git repository.
b. Make Commits:
   - Stage files for commit using the Source Control view (`Ctrl+Shift+G`).
   - Enter commit messages and commit changes.
c. Push Changes to GitHub:
   - Set up a remote repository on GitHub.
   - Use `git remote add origin <remote_repository_URL>` to add the remote.

- Push changes to GitHub using `git push -u origin main` (replace `main` with your branch name if different).

This process sets up version control, commits changes, and syncs them with a remote GitHub repository using Git commands in VS Code's integrated terminal.

References

1. Visual Studio Code Documentation**:**
   - Official documentation covering all aspects of VS Code, including setup, features, customization, and extensions.
   - Website: VS Code Documentation
2. Git Documentation**:**
   - Comprehensive documentation on Git commands, workflows, and integration with tools like VS Code.
   - Website: Git Documentation
3. Debugging in VS Code**:**
   - VS Code's debugging documentation, detailing setup, usage of breakpoints, watch expressions, and more.
   - Website: Debugging in VS Code
4. GitHub Guides**:**
   - Guides and tutorials on using Git with GitHub, including repository setup, committing changes, and syncing with remote repositories.
   - Website: GitHub Guides
5. Stack Overflow**:**
   - Community-driven platform where you can find answers to specific questions or troubleshooting issues related to development tools like VS Code and Git.
   - Website: Stack Overflow