Introduction

Microsoft created Visual Studio coding (VS Code), a potent, open-source, and free coding editor. It has several features, such as debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git, and it supports many programming languages. Its extensive extension marketplace allows you to further personalize and improve your coding experience to meet your unique requirements.

Installing Visual Studio Code on a Windows 11 computer is easy and just requires a few quick steps. In this assignment we discuss the steps to take when installing such a wonderful and popular development coding editor. Below are the steps that will help you ensure you have everything you need to get started coding quickly by guiding you through the download, installation, and setup of Visual Studio Code on your Windows 11 computer.

**1.0 Installation**

**Requirements:**

* **Hardware**

Visual Studio Code is a small download (< 200 MB) and has a disk footprint of < 500 MB. VS Code is lightweight and should easily run on today's hardware. It can therefore work on a computer with.

* 1. GHz or faster processor

1 GB of RAM

* **Platforms**

VS Code is supported on the following platforms:

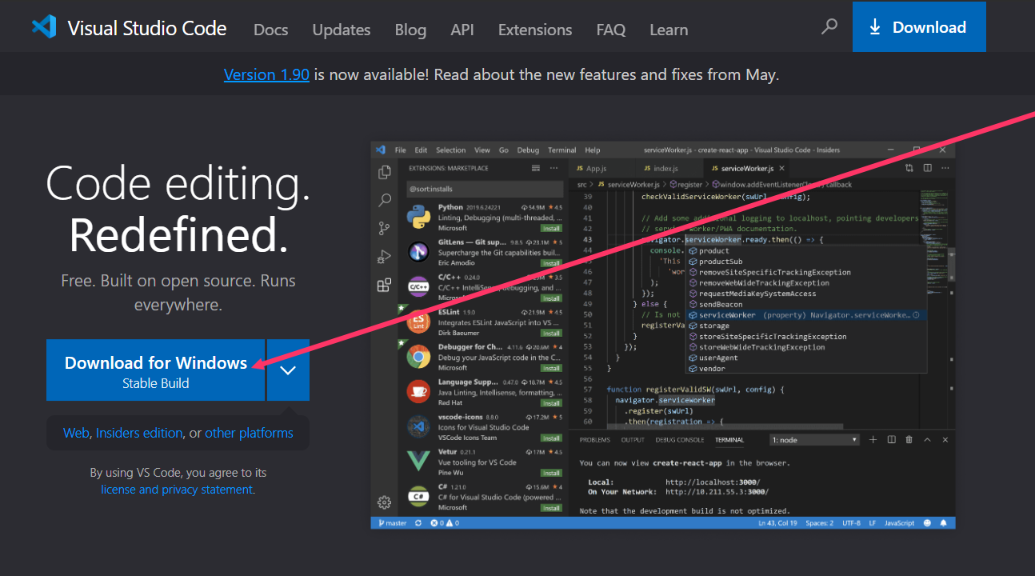
1. Windows 10 and 11 (64-bit)
2. macOS versions with Apple security update support. This is typically the latest release and the two previous versions.
3. Linux (Debian): Ubuntu Desktop 20.04, Debian 10
4. Linux (Red Hat): Red Hat Enterprise Linux 8, Fedora 36

**Setup and Installation:**

1. Download

Navigate to the Visual Studio Code website <https://code.visualstudio.com/> by opening your web browser.

The "Download for Windows" button should be clicked in order to download the Visual Studio Code (VSCodeUserSetup-x64-1.90.2.exe at a time of working on this document). See image below



1. Run and Install

Once the download is complete, locate the downloaded file (typically in your "Downloads" folder) and double-click it to run the installer.

In the setup wizard, review and accept the license agreement, then click "Next".

Choose the destination folder where you want Visual Studio Code to be installed. By default, it will install in C:\Program Files\Microsoft VS Code. Click "Next" to proceed.

Select additional tasks (optional):

* Create a desktop icon.
* Add "Open with Code" action to Windows Explorer file context menu.
* Add "Open with Code" action to Windows Explorer directory context menu.
* Register Code as an editor for supported file types.
* Add to PATH (this makes it possible to use code command in the terminal).

Click "Next" after selecting your preferences.

Click "Install" to begin the installation process, wait until the installation is finished. This could require many minutes. After the installation is complete, you can check the "Launch Visual studio Code" checkbox to start Visual Studio Code right away.

Select "Finish" to close the setup dialogue box.

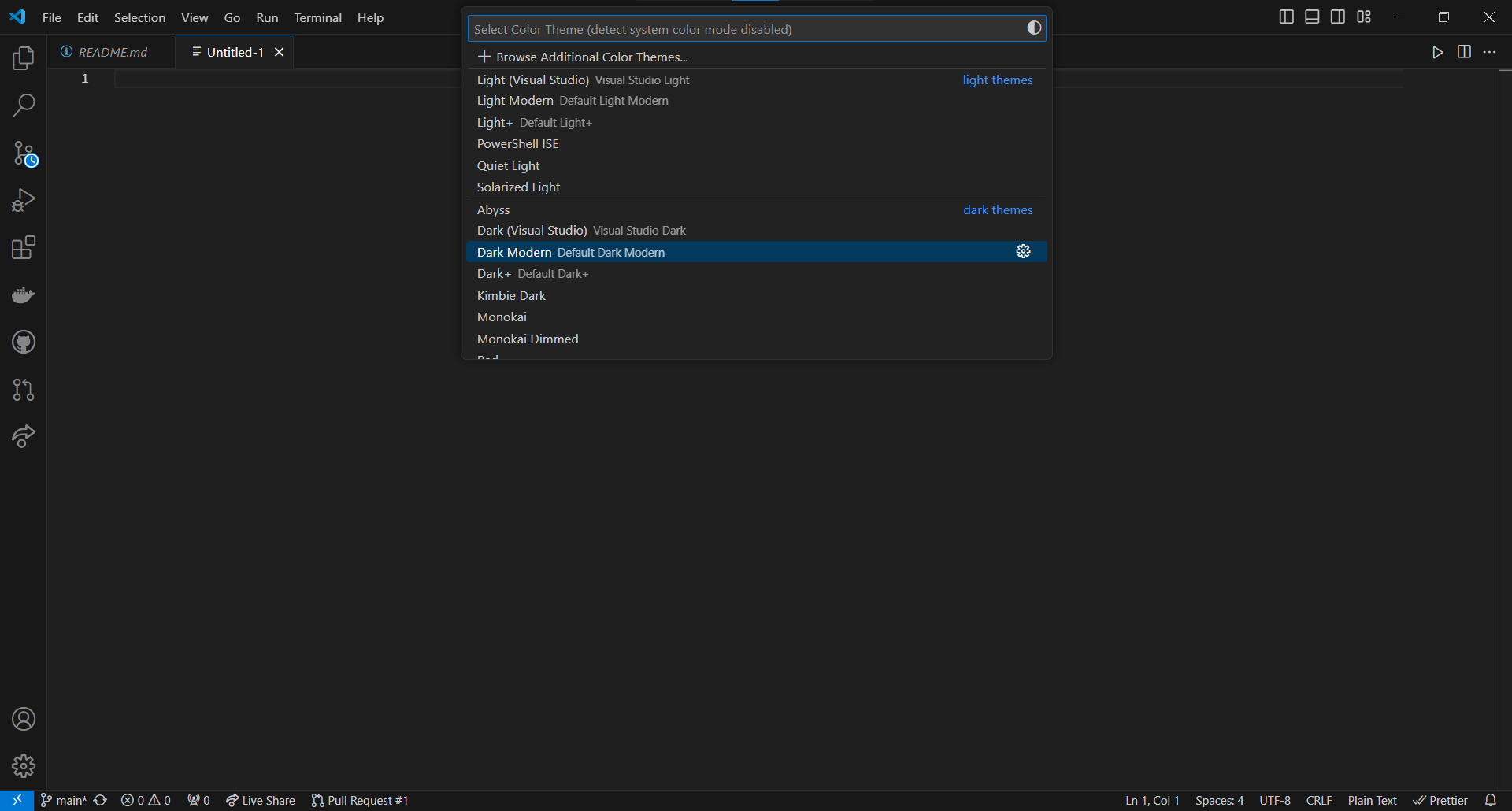
**2.0 Initial Configurations and Settings for Visual Studio Code**

To maximise your coding environment, you might wish to make a few configuration and setting changes after installing Visual Studio Code. The following crucial configurations and suggested add-ons will improve your programming experience:

2.1 Settings

1. Themes

To select a theme that you like, go to File > Preferences > Theme > Color Theme or use the keyboard shortcut Ctrl+K Ctrl+T. The default dark theme, "Dark+," the default light theme, "Light+," and third-party themes like "Dracula" or "Monokai" are among the most popular themes. The snapshot below shows how the Theme settings look like

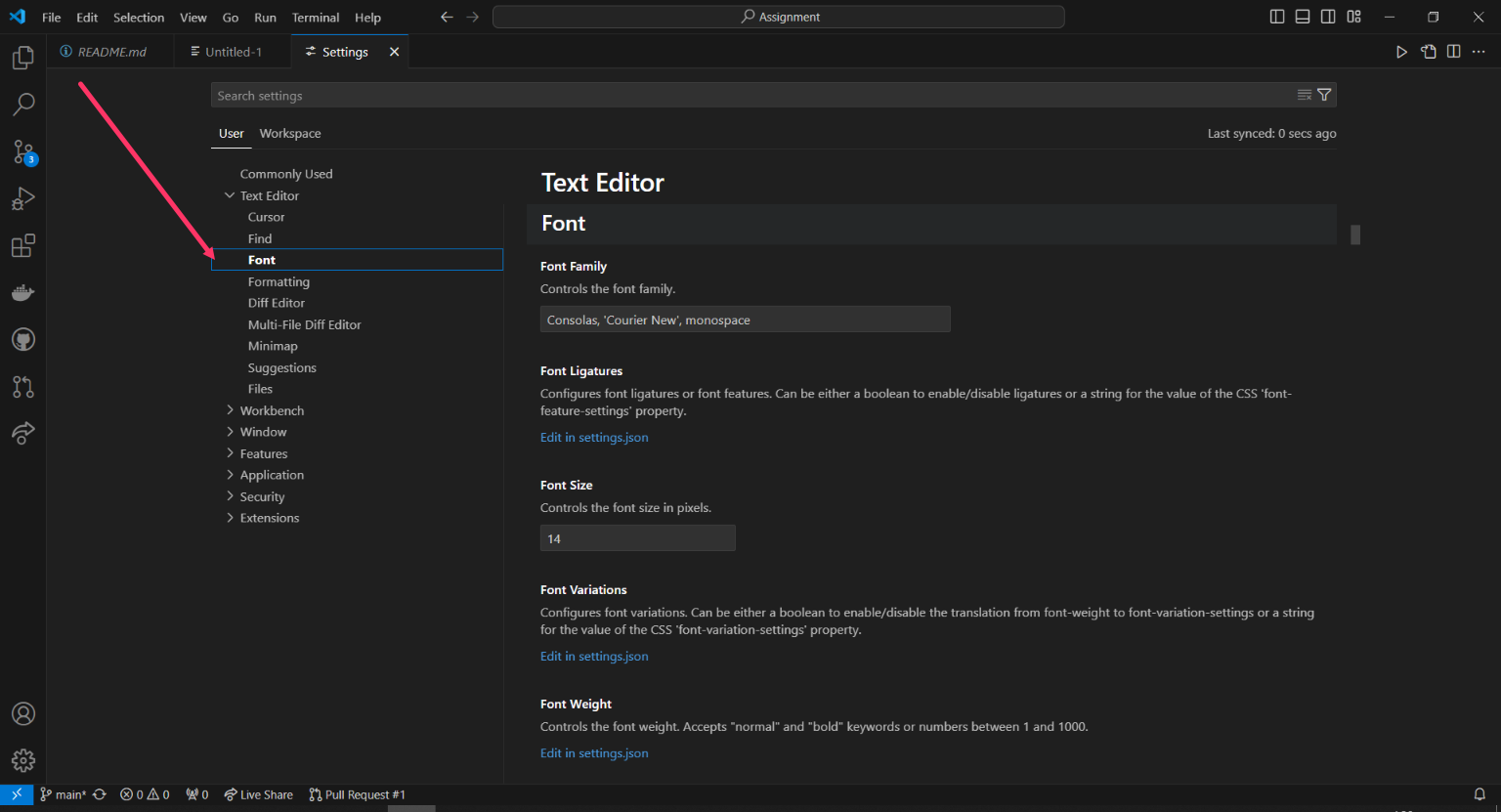


1. Font and Font Size

To view the settings, either choose File > Preferences > Settings or press Ctrl+.

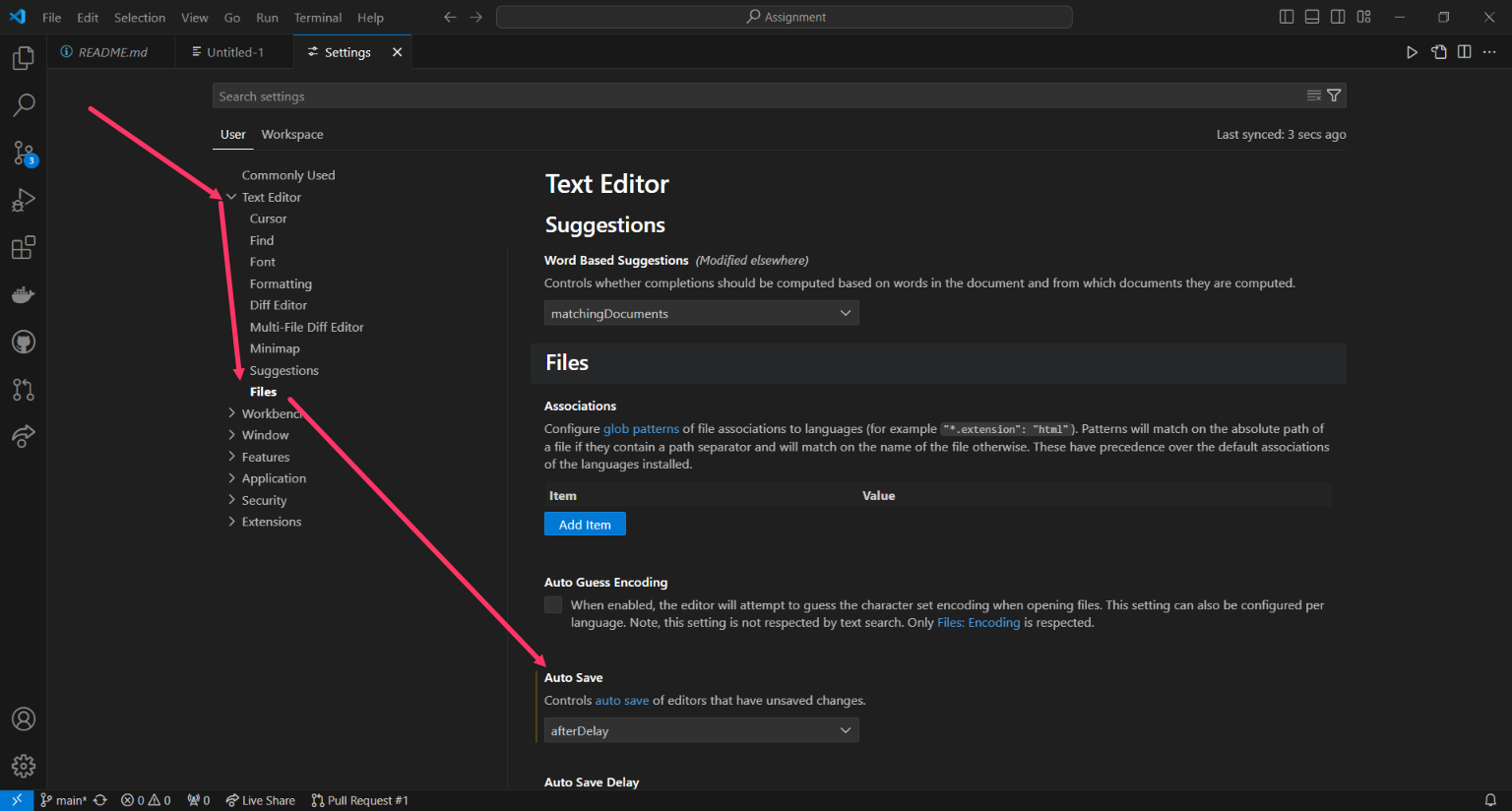
Click on Text Editor to select your favourite font (Fira Code, Consolas, for example).

To suit your tastes, change the font size to, say, 14.



1. Auto Save

Auto Save is a valuable feature that safeguards your work, enhances productivity, and supports seamless collaboration. So in the settings, navigate to Text Editor, click on files and the scroll down to look for search for Auto Save and set it to afterDelay or onWindowChange to automatically save your files.



3.0 User Interface Overview:

The user interface of Visual Studio Code (VS Code) is meant to be both customisable and intuitive. The Editor Group, Status Bar, Side Bar, and Activity Bar are the interface's primary elements. Here is a summary of every element:

3.0.1 The Activity

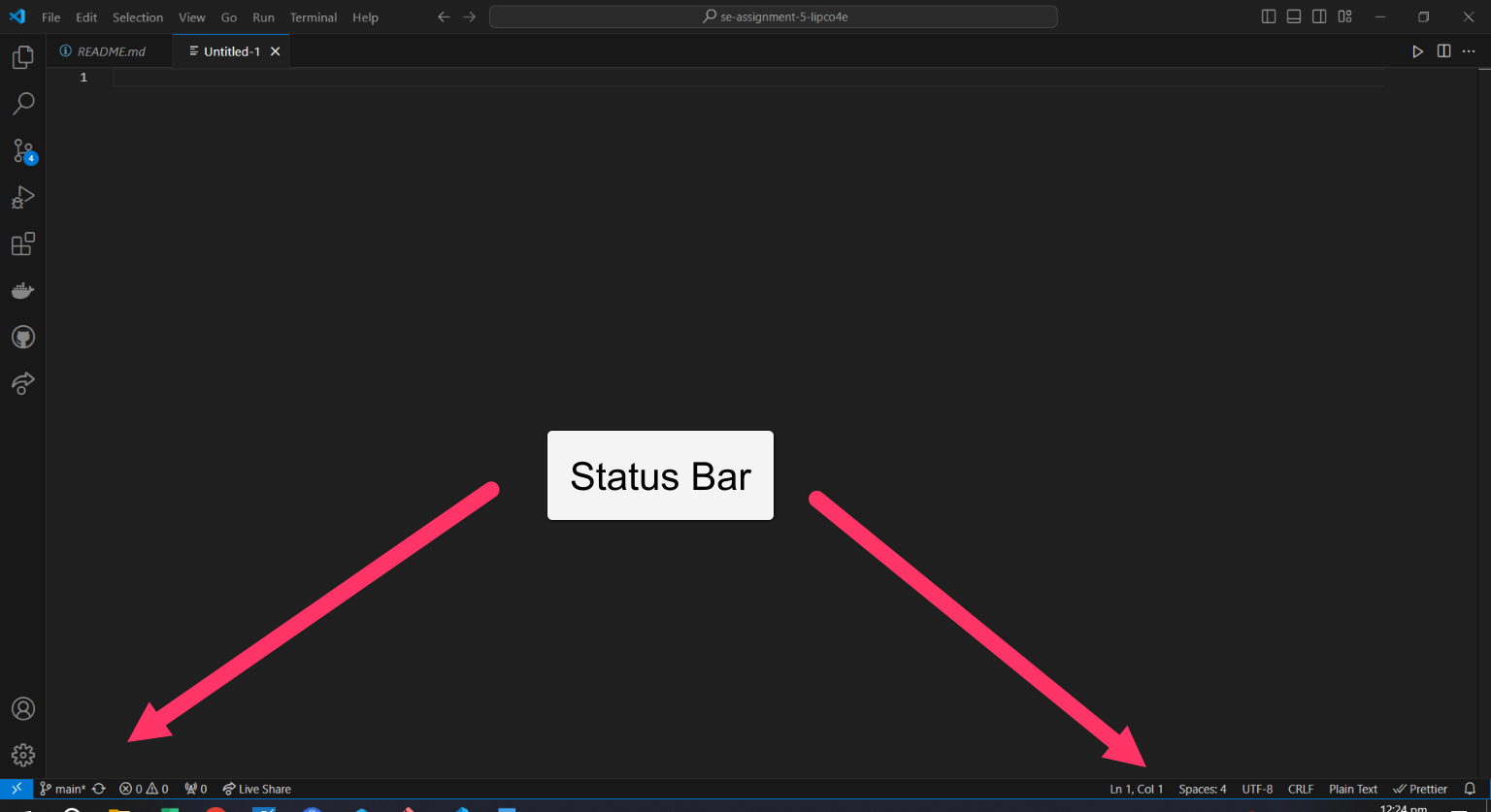
The Activity Bar is a core navigation surface in VS Code which enables rapid access to various views and features. Extensions can contribute View Containers to the Activity Bar that appear as Activity Bar Items. Users can drag the item to other locations like the Panel to customize their layout.

A screenshot of a computer

Description automatically generated

3.0.2 Status Bar

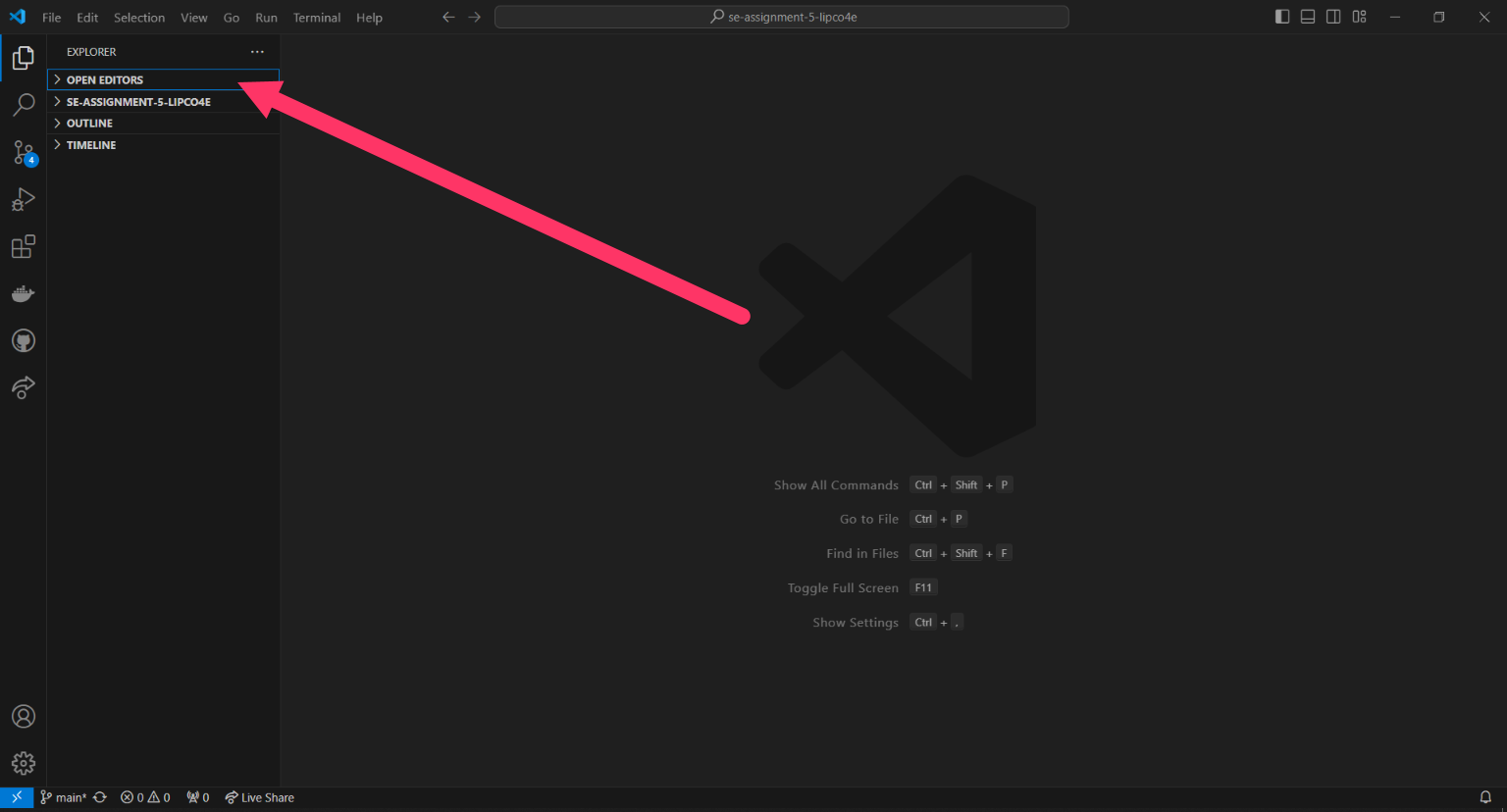
The Status Bar, which is located at the base of the VS Code workbench, shows actions and data related to your workspace. The items are arranged into Primary (left) and Secondary (right) groups. Secondary or contextual items (language, spacing, feedback) go on the right, and those that pertain to the entire workspace (status, problems/warnings, sync) go on the left. Since additional expansions contribute to the same region, keep the number of items added to a minimum.



* + 1. Editors Group

A new editor zone, or edit group, is formed when you divide an editor (using the divide Editor or Open to the Side commands). This edit group can hold many objects. Editor groups can be opened in any number of ways, both vertically and horizontally, side by side.

The Open Editors section at the top of the Explorer view (toggle... > Open Editors in the Explorer view) provides a clear view of these.



**4.0 The Command Palette**

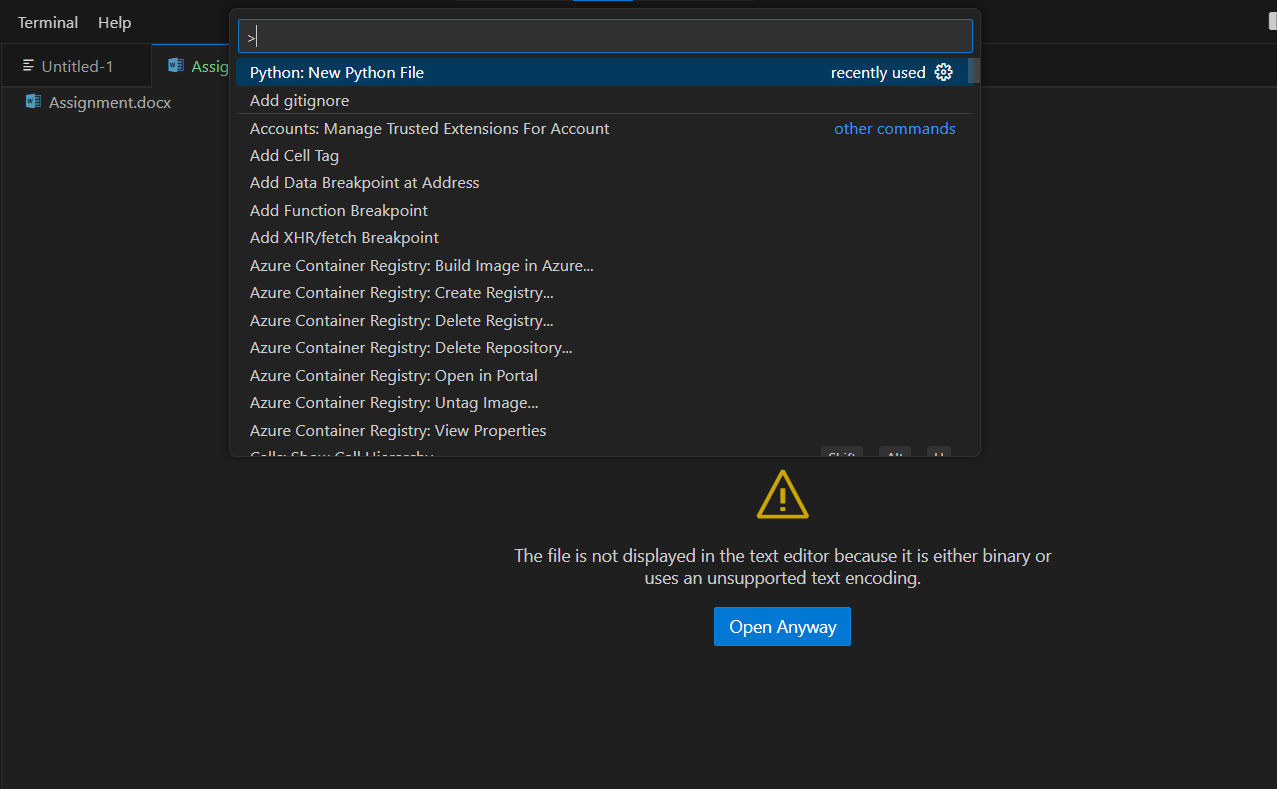
Using the keyboard to access VS Code is equally convenient. The Command Palette can be accessed by pressing Ctrl+Shift+P, which is the most crucial key combination to remember. You may access all of VS Code's features from this point on, including keyboard shortcuts for the most used tasks.

Numerous commands are accessible through the Command Palette. With the same interactive window, you can search for symbols, open files, run editor commands, and get a brief summary of a file. Here are some few tips to using the command palette:

* Ctrl+P enables you to navigate to any file or symbol by typing its name
* Ctrl+Tab cycles you through the last set of files opened
* Ctrl+Shift+P brings you directly to the editor commands
* Ctrl+Shift+O enables you to navigate to a specific symbol in a file
* Ctrl+G enables you to navigate to a specific line in a file

Type ? in the input field to get a list of available commands that you can run from the Command Palette.

The image below shows the command palette once you use this Ctrl+Shift+P command.



5.0 Extensions

Visual Studio Code (VS Code) extensions are essential for improving and personalising the development experience. They enable users to enhance productivity, support more programming languages, introduce new features, and incorporate other tools and services straight into the editor.

These are only a few of the pre-installed features of Visual Studio Code. To enhance your development process, you can add languages, debuggers, and tools to your installation using VS Code extensions. Because of VS Code's extensive extensibility paradigm, extension developers can add functionality by directly connecting to the VS Code UI and utilising the same APIs as VS Code.

5.0.1 Finding Extensions

From within Visual Studio Code, extensions can be browsed and installed. Clicking the Extensions icon in the Activity Bar beside VS Code or using the View: Extensions command (Ctrl+Shift+X) will open the Extensions view.

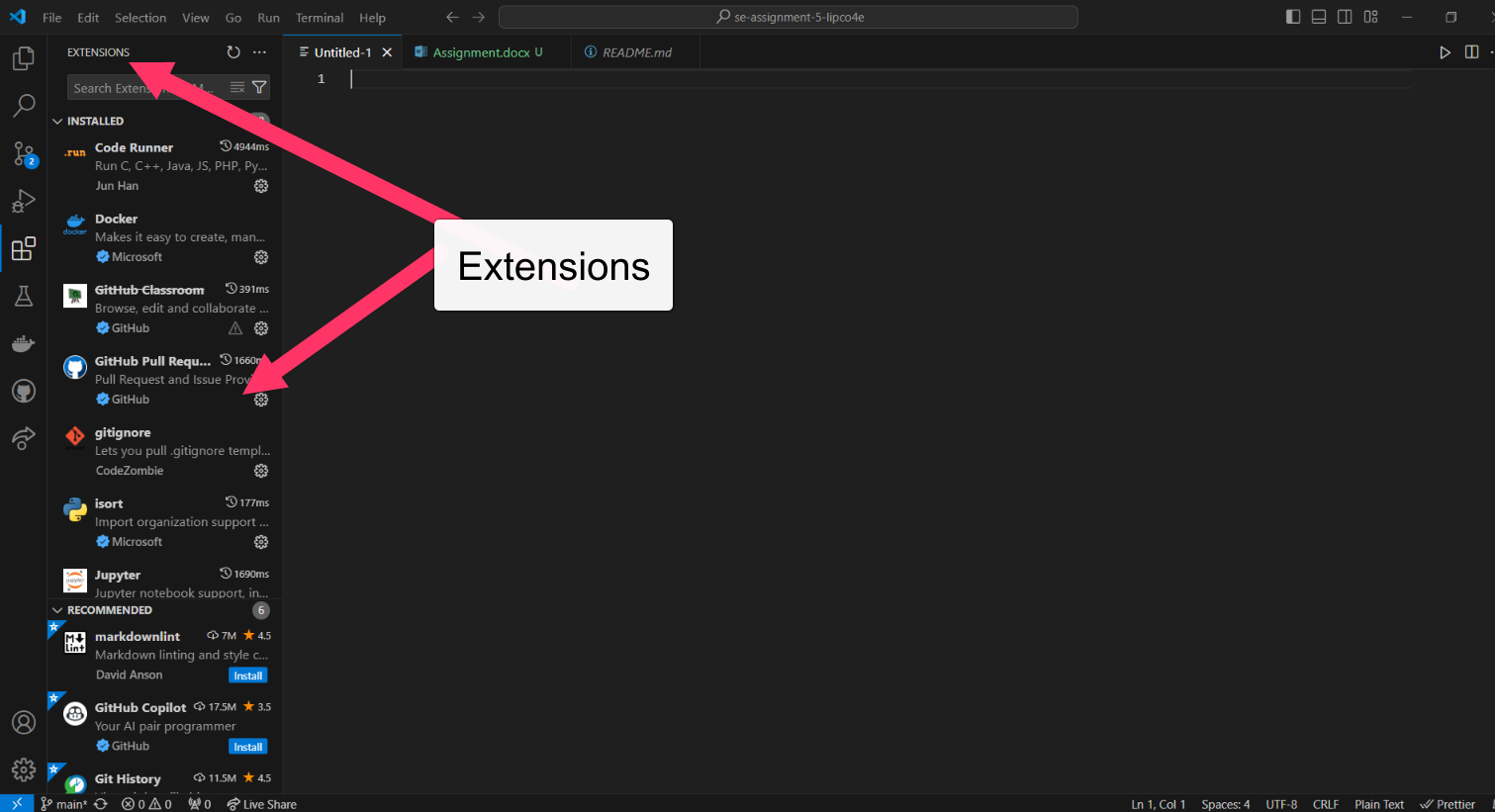
5.0.2 Installing Extensions

Click the Install button to begin installing an extension. The Install button will turn into the Manage gear button after the installation is finished.

Let us install the Code Spell Checker plugin, for instance. With the help of this addon, you can locate and correct spelling mistakes across your codebase with a rudimentary spell checker.

To narrow down the Marketplace options to extensions that have "Code Spell Checker" in the title or metadata, insert the term "Code Spell Checker" into the search box in the Extensions window (Ctrl+Shift+X). The Code Spell Checker extension ought to be visible in the list. Check the steps below undertaken

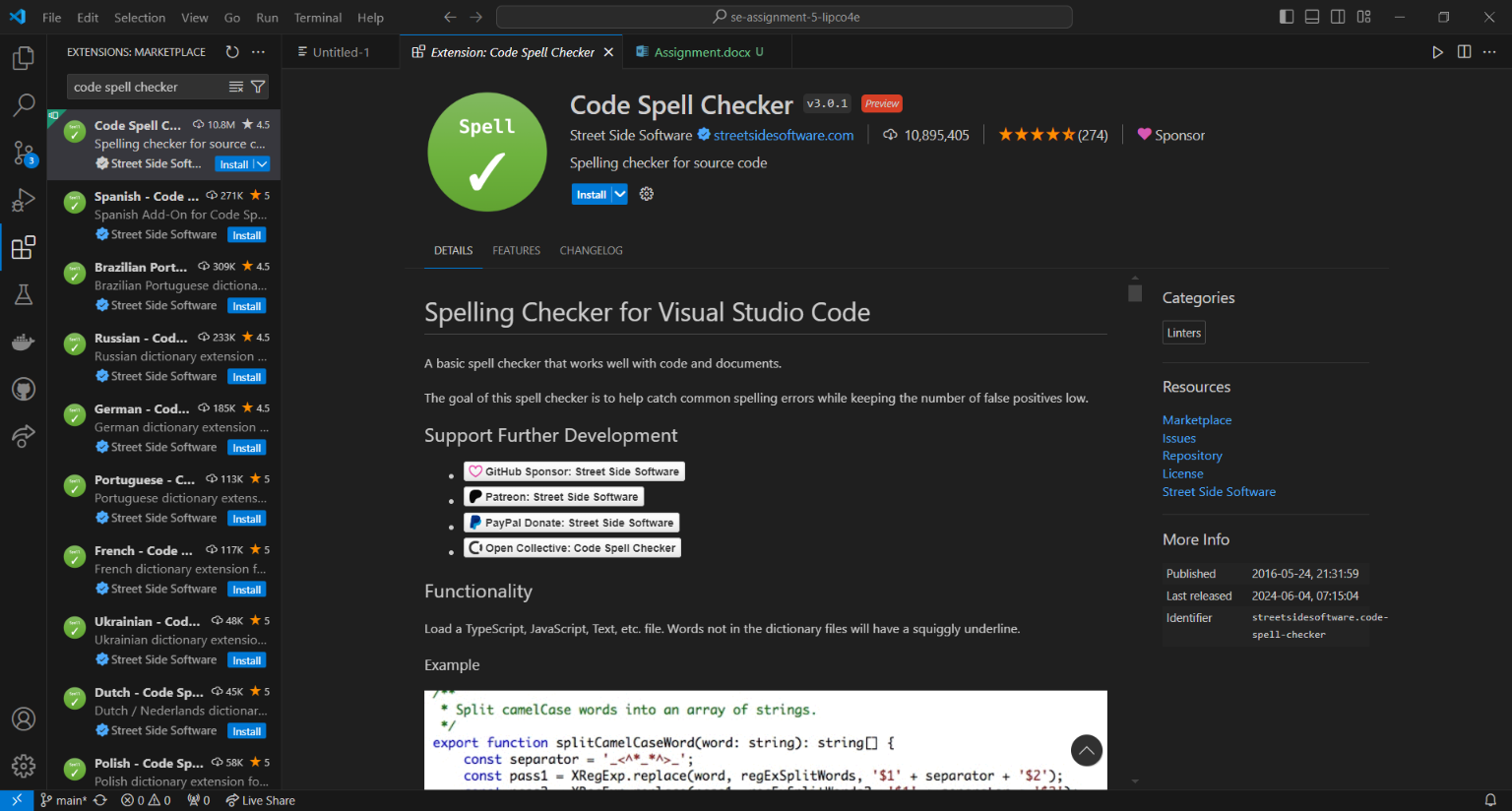
Step 1 Use this command Ctrl+Shift+X



Step 2 Search for Code Spell CheckerA screen shot of a computer

Description automatically generated

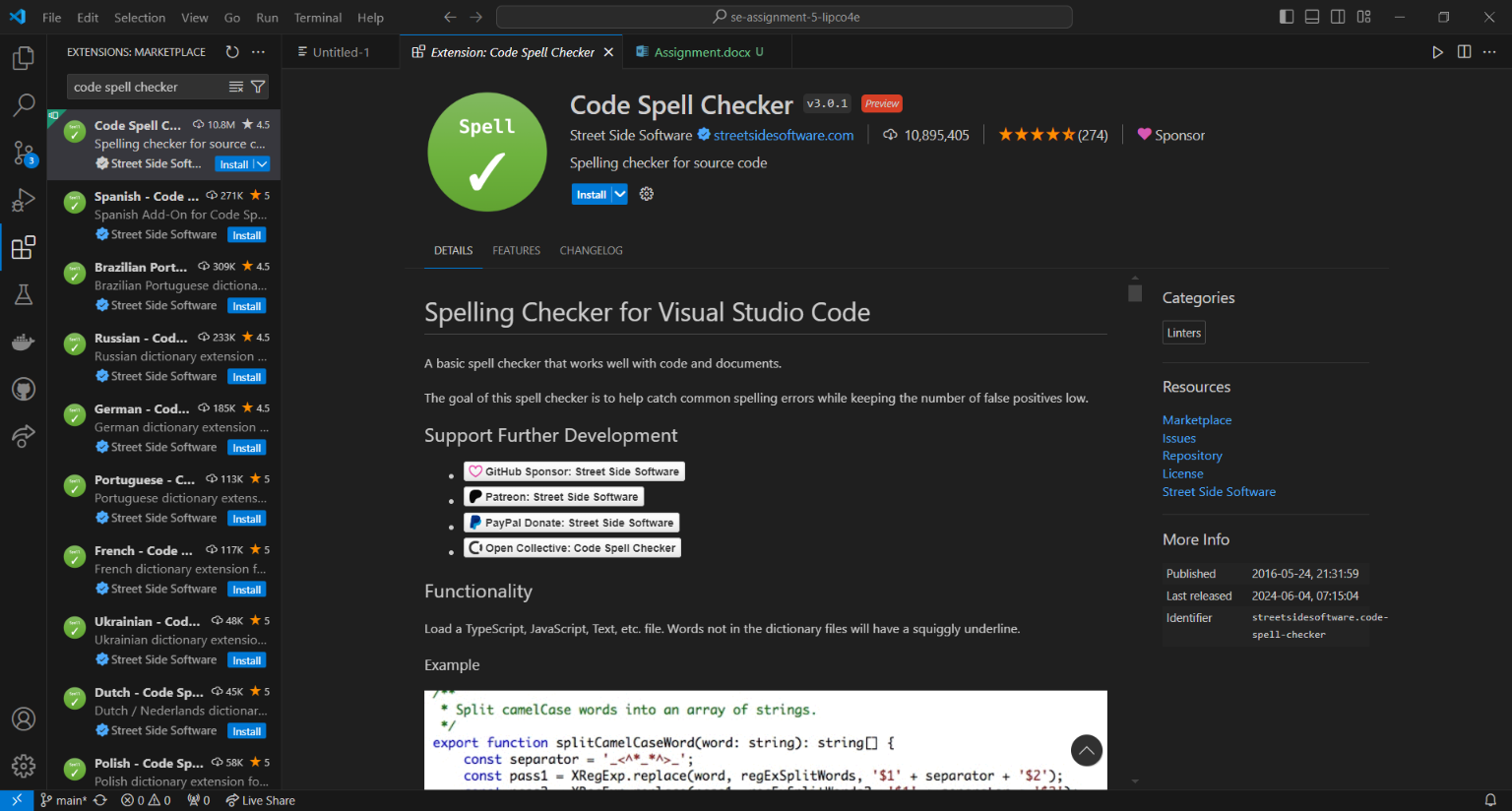
Step 3 Select and Click on install



Step 4 The last step shows that the extension is installed and shows a disable, unistall and switch to pro-mode buttons

A screenshot of a computer

Description automatically generated



6.0 Integrated Terminal

The integrated terminal in Visual Studio Code supports different shells, including PowerShell, Command Prompt, and Bash. You can easily switch between different shells based on your preference or the specific requirements of your project.

With the integrated terminal, you can execute commands directly from the editor without the need to switch between windows or leave your workspace. This allows for a seamless workflow and increased productivity.

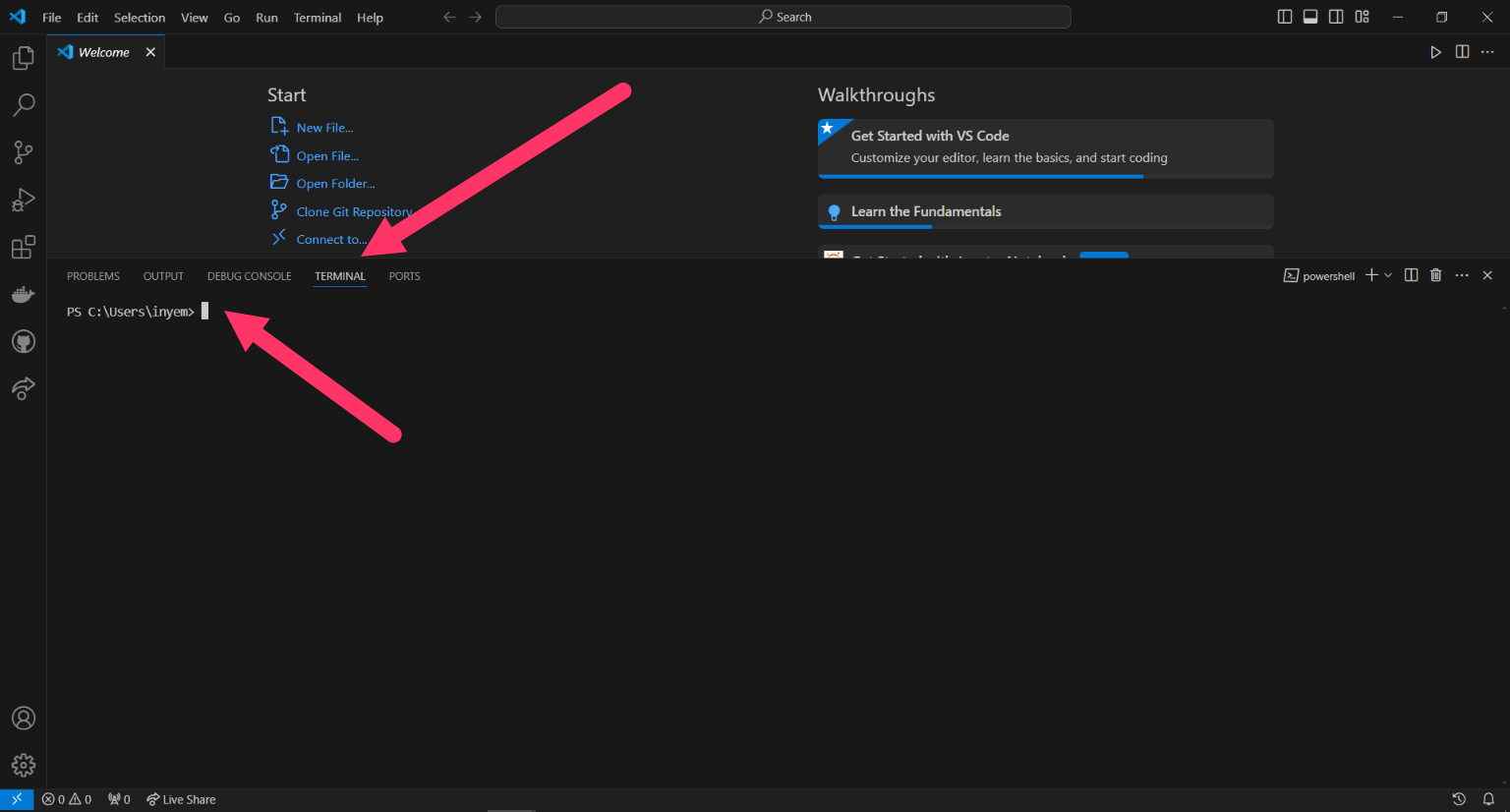
Furthermore, the terminal in Visual Studio Code supports customization options such as changing the default shell, configuring environment variables, and setting up different profiles for varying tasks.

Overall, the integrated terminal in Visual Studio Code provides a convenient and efficient way to interact with your workspace and execute various commands, making it easier to manage your projects and perform tasks without leaving the editor.

You can open a terminal as follows:

* From the menu, use the Terminal > New Terminal or View > Terminal menu commands.
* From the Command Palette (Ctrl+Shift+P), use the View: Toggle Terminal command.
* In the Explorer, you can use the Open in Integrated Terminal context menu command to open a new terminal from a folder.
* To toggle the terminal panel, use the Ctrl+` keyboard shortcut.
* To create a new terminal, use the Ctrl+Shift+` keyboard shortcut.

VS Code's terminal has additional functionality called shell integration that tracks where commands are run with decorations on the left of a command and in the scrollbar:



7.0 File and Folder Management

To make the development process more efficient, Visual Studio Code (VS Code) offers a user-friendly interface for creating, accessing, and organizing files and folders. This detailed guide will help you explore and complete these tasks quickly.

Creating a New File or Folder:

1. Open VS Code.

2. Click on the Explorer icon on the left sidebar (or press Ctrl+Shift+E).

3. To create a new file, right-click on the desired folder in the Explorer panel and select "New File" from the context menu. Alternatively, press Ctrl+N.

4. To create a new folder, right-click on the desired parent folder in the Explorer panel, and select "New Folder" from the context menu. Alternatively, press Ctrl+Shift+N.

5. Enter a name for the file or folder and press Enter.

Opening a File or Folder:

1. Open VS Code.

2. Click on the Explorer icon on the left sidebar (or press Ctrl+Shift+E).

3. To open a file, double-click on the file in the Explorer panel. Alternatively, you can use the "File" menu and select "Open File" or use the keyboard shortcut Ctrl+O. Locate the file in the file picker and click "Open".

4. To open a folder, use the "File" menu and select "Open Folder" or use the keyboard shortcut Ctrl+K Ctrl+O. Locate the folder in the file picker and click "Open".

Managing Files and Folders:

1. Renaming a file or folder:

- Right-click on the file or folder in the Explorer panel.

- Select "Rename" from the context menu.

- Enter the new name and press Enter.

2. Deleting a file or folder:

- Right-click on the file or folder in the Explorer panel.

- Select "Delete" from the context menu.

- Confirm the deletion in the prompt.

3. Copying, cutting, and pasting files or folders:

- Right-click on the file or folder in the Explorer panel.

- Select "Copy" or "Cut" from the context menu.

- Right-click on the destination folder in the Explorer panel.

- Select "Paste" from the context menu.

4. Moving a file or folder:

- Drag and drop the file or folder to the desired location in the Explorer panel.

Efficient Navigation:

1. Use the "Explorer" panel to navigate through the file system and open files or folders.

2. Keyboard shortcuts can help you navigate faster:

- Ctrl+1 (Windows/Linux) or Command+1 (Mac) switches focus to the Explorer panel.

- Ctrl+2 (Windows/Linux) or Command+2 (Mac) switches focus to the Editor panel.

- Ctrl+Tab (Windows/Linux) or Command+Tab (Mac) allows you to cycle through open files.

- Ctrl+W (Windows/Linux) or Command+W (Mac) closes the current file.

With these instructions, you should be able to easily create, open, and manage files and folders in Visual Studio Code.

8.0 Settings and Preferences

Visual Studio Code (VS Code) offers users a range of customizability options to tailor their development environment to their specific preferences. These settings allow users to personalize various aspects of their experience, such as modifying themes, adjusting font sizes, and reconfiguring keybindings. The flexibility provided by VS Code empowers developers to create a workspace that suits their individual needs and facilitates a more efficient and enjoyable coding experience.

To access the settings in the software, users can navigate to the File menu and select Preferences, then choose Settings (or Code > Preferences > Settings on macOS). Alternatively, they can use the shortcut Ctrl+, (or Cmd+, on macOS) to open the settings UI. For more advanced customization options, users can click on the {} icon in the top right corner of the settings UI to access the settings JSON file. Another way to open the settings JSON file is by opening the Command Palette with Ctrl+Shift+P (or Cmd+Shift+P on macOS) and typing "Preferences: Open Settings (JSON)".

To change the theme of the software, users have two options. They can use the Command Palette by pressing Ctrl+Shift+P (or Cmd+Shift+P on macOS), typing "Preferences: Color Theme," and selecting the desired theme. Alternatively, users can use the settings UI by opening it and typing "theme" in the search bar. Under the Appearance section, they can click on Color Theme and select a theme from the dropdown list.

Changing the font size can also be done through the settings UI or the settings JSON file. In the settings UI, users can open it, type "font size" in the search bar, locate Font Size under Text Editor, and adjust the value to their preferred font size. In the settings JSON file, users can add or update the line "editor.fontSize": 14 (replace 14 with the desired font size).

To change keybindings, users can use either the Keyboard Shortcuts UI or the Keybindings JSON file. In the Keyboard Shortcuts UI, users can open it by going to the File menu and selecting Preferences > Keyboard Shortcuts (or Code > Preferences > Keyboard Shortcuts on macOS). They can also use the shortcut Ctrl+K Ctrl+S. After searching for the desired command, they can click the pencil icon next to it and assign a new key combination. In the Keybindings JSON file, users can open the Command Palette with Ctrl+Shift+P (or Cmd+Shift+P on macOS), type "Preferences: Open Keyboard Shortcuts (JSON)," and add or modify keybindings using the provided syntax. For example, to change the keybinding for saving all files to Ctrl+Alt+S, they can add the following entry:

{

"key": "ctrl+alt+s",

"command": "workbench.action.files.saveAll"

}

9.0 Debugging

Visual Studio Code (VS Code) boasts exceptional debugging support as one of its prominent attributes. The integrated debugger within VS Code effectively streamlines the iterative process of editing, compiling, and debugging code, thereby enhancing productivity and efficiency.

To set up and initiate the debugging process in Visual Studio Code (VS Code), adhere to the subsequent steps. While this guide focuses on a basic Node.js application, the fundamental procedure remains comparable for other languages that have suitable extensions.

Steps to Set Up Debugging

9.0.1 Install VS Code

To set up debugging, the first step is to install Visual Studio Code (VS Code) if it is not already installed. Then, you need to install the necessary extension for your programming language. This could be an extension for Node.js, Python, Java, or any other language you are working with.

9.0.1 Open and Create Project Folder

Once you have VS Code installed and the relevant extension installed, you can proceed to open VS Code and create a new folder for your project or open an existing project. Within this project folder, you can create a simple application to test and debug.

For example, if you are working with Node.js, you can create a new file called "app.js" and place it in your project folder. In this file, you can write some code to test and debug.

console.log("Hello, World!");

Setting up the debugging environment involves configuring breakpoints, running the debugger, and analyzing the code execution. This allows you to pause the execution of your code at specific points, examine variables and their values, and step through the code to understand how it behaves.

By following these steps, you can effectively set up and use debugging in your development process, which can help troubleshoot and solve issues in your code more efficiently.

9.0.3 Initialize a Debug Config

To open the Command Palette, you can use the key combination Ctrl+Shift+P (or Cmd+Shift+P on macOS). Once the Command Palette is open, type "Debug: Open launch.json" and select it. If you do not already have a launch.json file, Visual Studio Code will prompt you to create one. After selecting it, you will be asked to choose the environment for your application, such as Node.js. Upon selection, a launch.json file will be created in the .vscode folder with a default configuration. For a basic Node.js application, the launch.json file may appear as follows.

{

"version": "0.2.0",

"configurations": [

{

"type": "node",

"request": "launch",

"name": "Launch Program",

"program": "${workspaceFolder}/app.js"

}

]

}

9.0.4 Set Breakpoints

To initiate the debugging process, it is necessary to open the relevant file, such as "app.js". Locate the specific line where you intend to set a breakpoint either by clicking on the gutter to the left of the line number or by placing the cursor on the desired line and pressing F9.

9.0.5 Start Debugging

To access the Run and Debug view, you can either click on the Run icon in the Activity Bar or use the shortcut Ctrl+Shift+D. Once in the view, choose the specific configuration you wish to utilize, such as "Launch Program." To initiate the debugging process, click the green play button or press the F5 key.

10.0 Using Source Control

To integrate Git with Visual Studio Code (VS Code) for version control, the process involves opening the project folder in VS Code and initializing a Git repository. This can be done either through the Source Control view or the integrated terminal (Ctrl+``). Once the repository is initialized, changes to files can be staged by clicking the + icon next to modified files in the Source Control view. These changes can then be committed with a descriptive message.

For pushing changes to GitHub, it is necessary to set up a remote repository by adding its URL in the terminal using the command "git remote add origin <repository\_url>". Once the remote repository is set up, commits can be pushed using the command "git push -u origin master".

VS Code's seamless integration with Git, combined with extensions like GitHub's for managing pull requests and issues, streamlines the entire version control process within the development environment. This integrated setup allows for efficient collaboration and tracking of project changes, minimizing the need for context switching.

References

https://code.visualstudio.com/docs/terminal/basics

<https://code.visualstudio.com/docs/supporting/requirements>

<https://code.visualstudio.com/docs/getstarted/userinterface>

<https://code.visualstudio.com/api/ux-guidelines/activity-bar>

<https://code.visualstudio.com/docs/editor/extension-marketplace>

<https://www.freecodecamp.org/news/best-vscode-extensions/>