SOFTWARE ENGINEERING DAY 2 ASSIGNMENT 2

NAME: EMMANUEL CLINTON ODHIAMBO

SE-Assignment-5: Installation and Navigation of Visual Studio Code (VS Code)

INTRODUCTION

The primary objective of this assignment is to provide a comprehensive guide to the installation, setup, and navigation of Visual Studio Code (VS Code) on the Windows 11 operating system. This document aims to equip users with the necessary knowledge to effectively utilize VS Code for software development tasks.

The scope of this assignment covers:

- Step-by-step instructions for downloading and installing VS Code on Windows 11.
- Initial configurations and settings adjustments post-installation to optimize the coding environment.
- Detailed explanation of the main components of the VS Code user interface and their functionalities.
- Exploration of essential tools such as the Command Palette, Integrated Terminal, and File Management features within VS Code.
- Guidance on how to utilize extensions for enhancing productivity, particularly in web development.
- Instructions on debugging capabilities and integration with Git for version control.

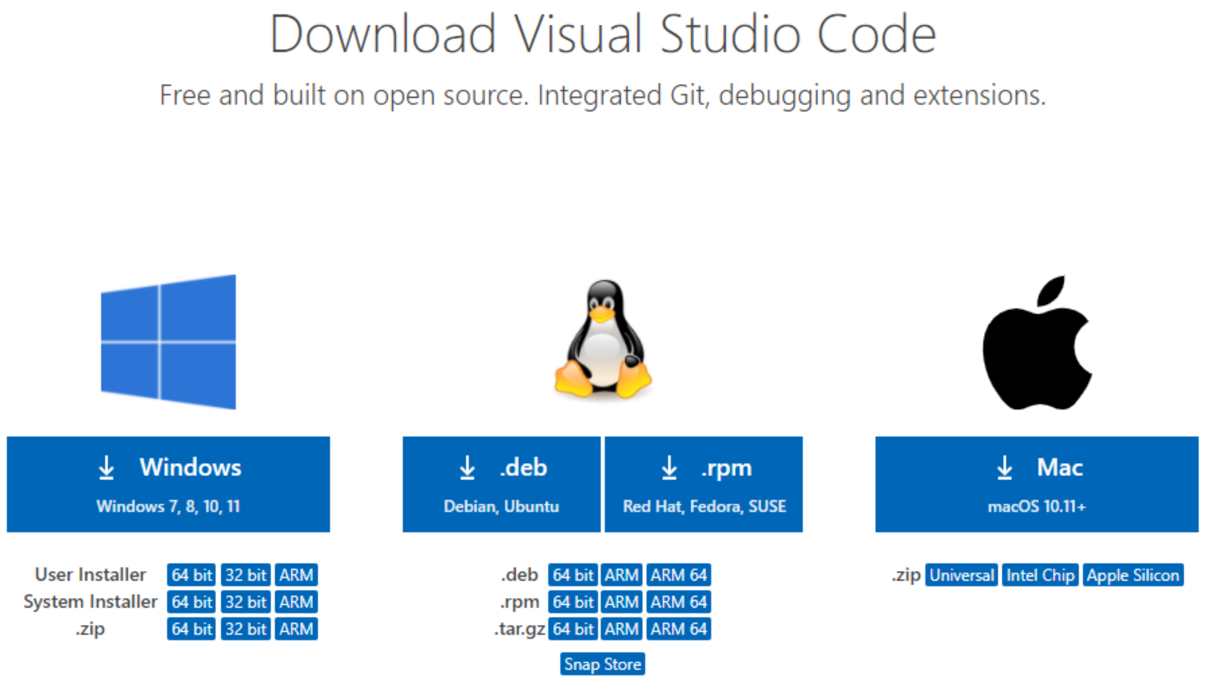
By following this structured approach, users will gain a solid foundation in using VS Code effectively for their programming needs on the Windows 11 platform.

INSTALLATION OF VS CODE

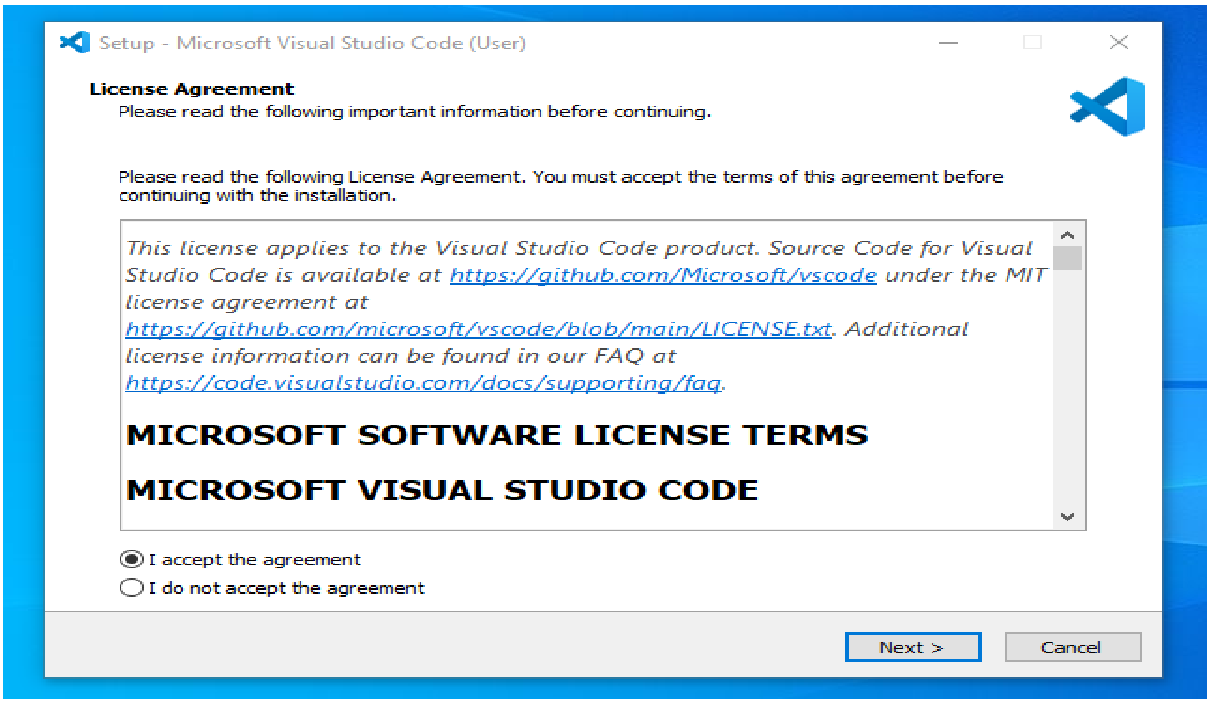
In this chapter, we are going to list all the steps required to Install VS Code on Windows in a detailed format.

1. Visit the Official Website of the Visual Studio Code using any web browser like Google Chrome, Microsoft Edge, etc.

2. Press the "Download for Windows" button on the website to start the download of the Visual Studio Code Application.


Download Visual Studio Code
Free and built on open source. Integrated Git, debugging and extensions.
Windows
Windows 7, 8, 10, 11
.deb
Debian, Ubuntu
.rpm
Red Hat, Fedora, SUSE
Mac
macOS 10.11+
User Installer 64 bit 32 bit ARM
System Installer 64 bit 32 bit ARM
.zip 64 bit 32 bit ARM
.deb 64 bit ARM ARM 64
.rpm 64 bit ARM ARM 64
.tar.gz 64 bit ARM ARM 64
Snap Store
.zip Universal Intel Chip Apple Silicon

3. When the download finishes, then the Visual Studio Code Icon appears in the downloads folder.
4. Click on the Installer icon to start the installation process of the Visual Studio Code.
5. After the Installer opens, it will ask you to accept the terms and conditions of the Visual Studio Code. Click on I accept the agreement and then click the Next button.



Setup - Microsoft Visual Studio Code (User)

License Agreement
Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

This license applies to the Visual Studio Code product. Source Code for Visual Studio Code is available at https://github.com/Microsoft/vscode under the MIT license agreement at https://github.com/microsoft/vscode/blob/main/LICENSE.txt. Additional license information can be found in our FAQ at https://code.visualstudio.com/docs/supporting/faq.

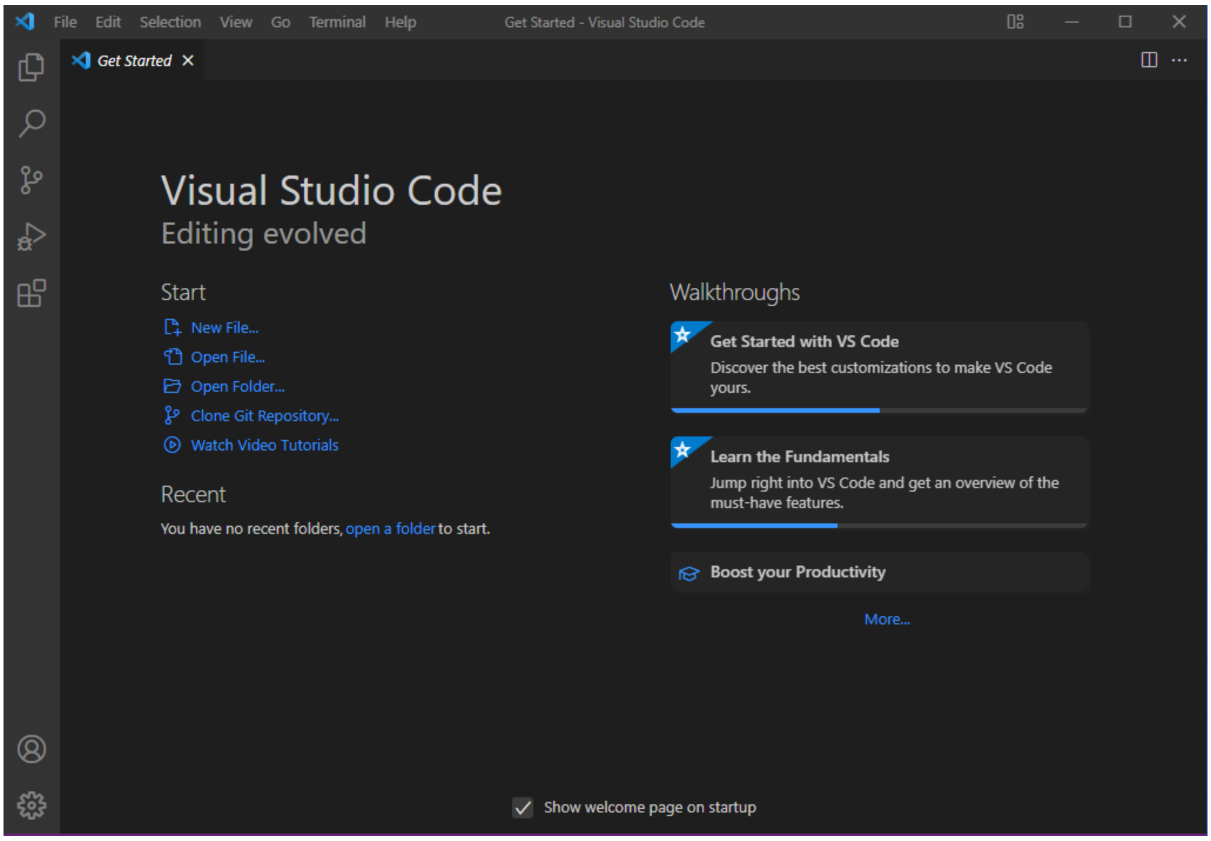
MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT VISUAL STUDIO CODE

(●) I accept the agreement
( ) I do not accept the agreement

6. Choose the location data for running the Visual Studio Code. It will then ask you to browse the location. Then click on the Next button.
7. Then it will ask to begin the installation setup. Click on the Install button.
8. After clicking on Install, it will take about 1 minute to install the Visual Studio Code on your device.

9. After the Installation setup for Visual Studio Code is finished, it will show a window like this below. Tick the "Launch Visual Studio Code" checkbox and then click Next.
10. After the previous step, the Visual Studio Code window opens successfully. Now you can create a new file in the Visual Studio Code window and choose a language of yours to begin your programming journey!



File Edit Selection View Go Terminal Help
Get Started - Visual Studio Code
Get Started

Visual Studio Code
Editing evolved

Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Watch Video Tutorials

Recent
You have no recent folders, open a folder to start.

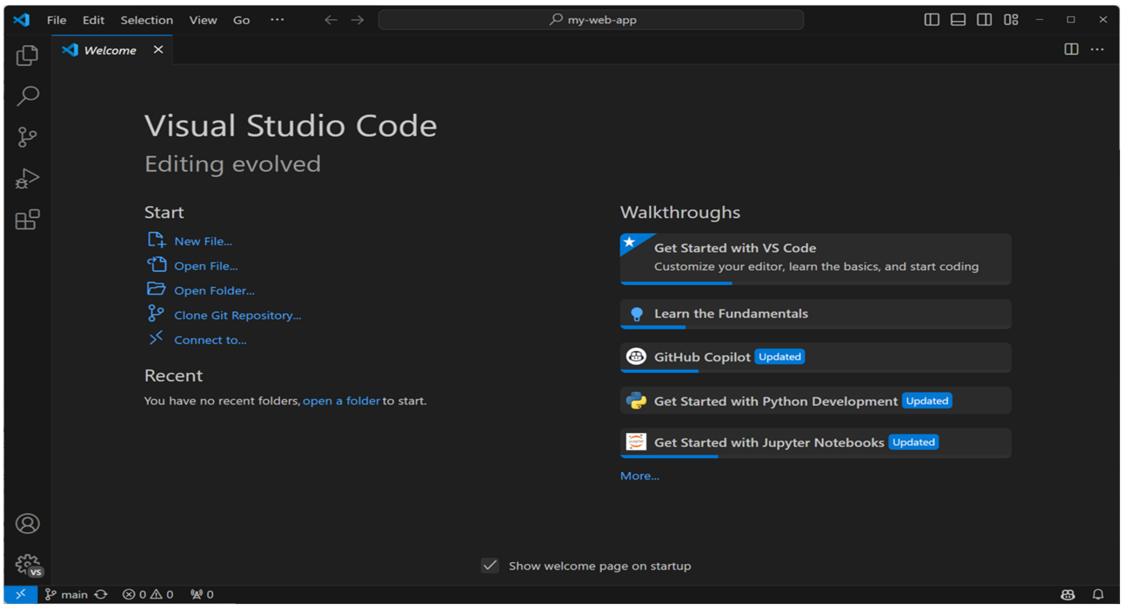
Walkthroughs
Get Started with VS Code
Discover the best customizations to make VS Code yours.
Learn the Fundamentals
Jump right into VS Code and get an overview of the must-have features.
Boost your Productivity
More...

Show welcome page on startup

FIRST-TIME SETUP OF VS CODE

The best way of exploring VS Code hands-on is to open the Welcome page and then pick a Walkthrough for a self-guided tour through the setup steps, features, and deeper customizations that VS Code offers. As you discover and learn, the walkthroughs track your progress.

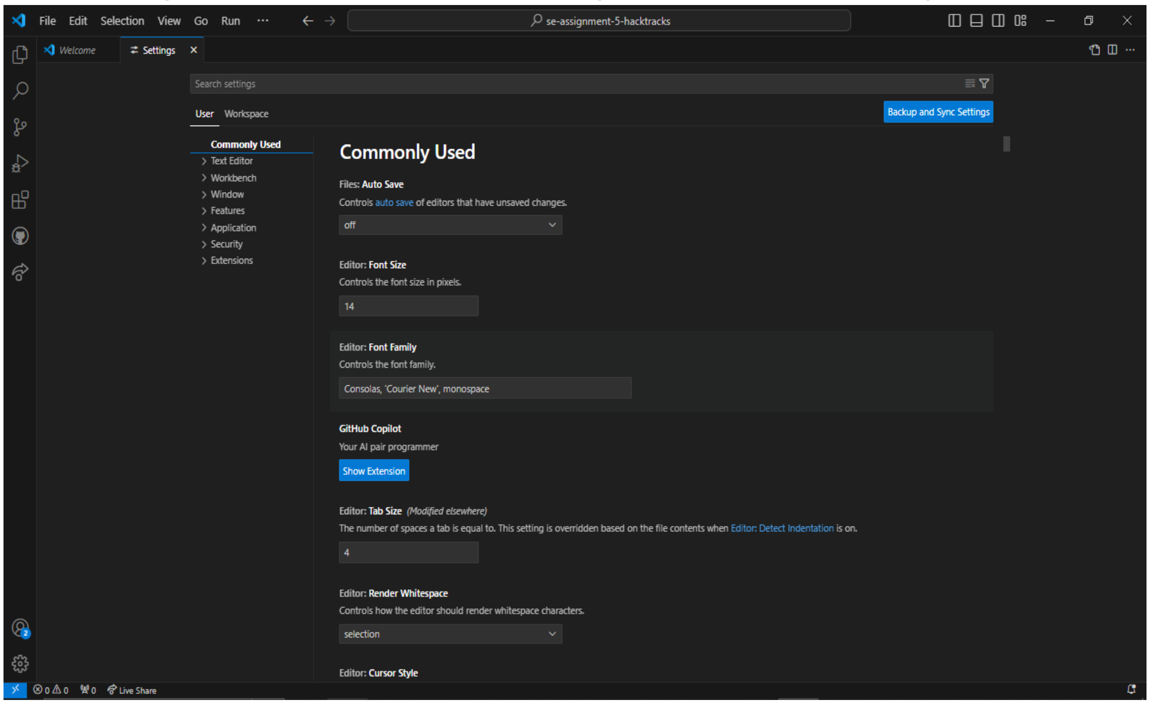
- Open the **Welcome page** from the **Help** > **Welcome menu** or use the **Help: Welcome** command from the **Command Palette (Ctrl+Shift+P).**


File Edit Selection View Go ···
my-web-app
Welcome
Visual Studio Code
Editing evolved
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Connect to...
Recent
You have no recent folders, open a folder to start.
Walkthroughs
Get Started with VS Code
Customize your editor, learn the basics, and start coding
Learn the Fundamentals
GitHub Copilot Updated
Get Started with Python Development Updated
Get Started with Jupyter Notebooks Updated
More...
Show welcome page on startup
main ⊗ 0 ⚠ 0 0

INITIAL CONFIGURATIONS AFTER INSTALLATION

Configuration of Settings for Optimal Coding Environment

- Navigate to `**File** -> **Preferences** -> **Settings**` to access VS Code settings.



Commonly Used

Files: **Auto Save**
Controls auto save of editors that have unsaved changes.
off

Editor: **Font Size**
Controls the font size in pixels.
14

Editor: **Font Family**
Controls the font family.
Consolas, 'Courier New', monospace

GitHub Copilot
Your AI pair programmer
Show Extension

Editor: **Tab Size** *(Modified elsewhere)*
The number of spaces a tab is equal to. This setting is overridden based on the file contents when Editor: Detect Indentation is on.
4

Editor: **Render Whitespace**
Controls how the editor should render whitespace characters.
selection

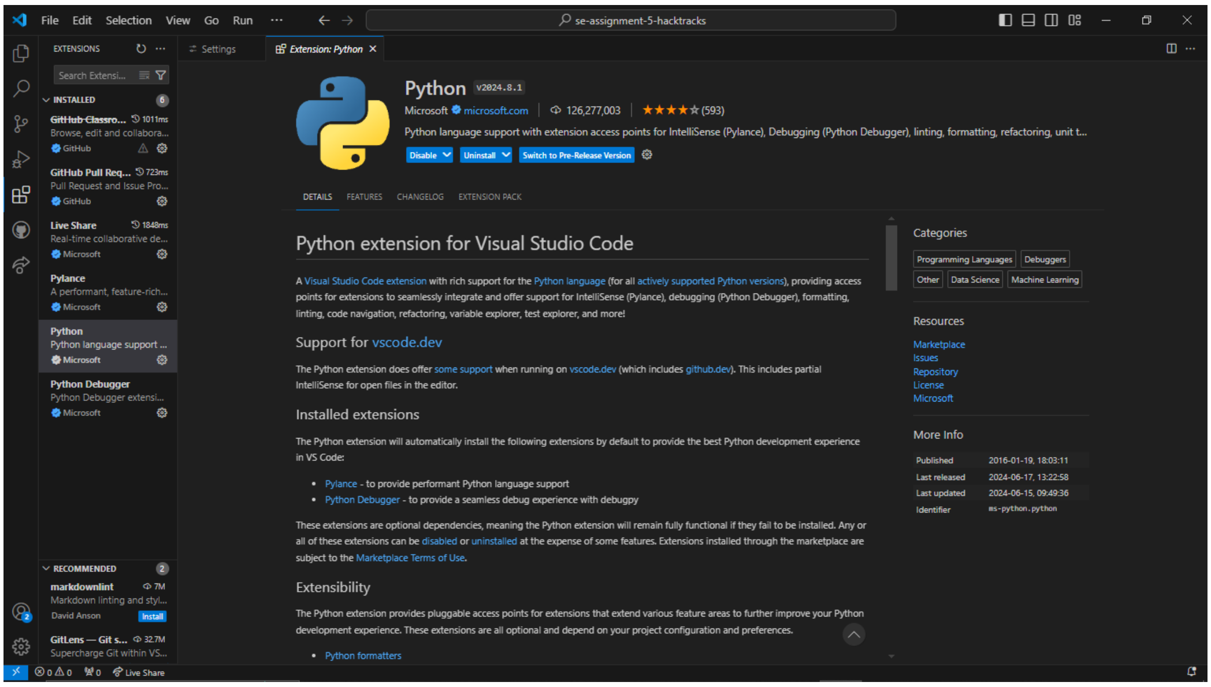
Editor: **Cursor Style**

Customize settings such as:

- **Theme**: Choose a **theme** (e.g., Dark+, Light+) for your preferred coding environment (`*workbench.colorTheme*`).
- **Font Size**: Adjust the editor font size to your comfort (`*editor.fontSize*`).
- **Indentation**: Set tab size and spaces (`*editor.tabSize*`, `*editor.insertSpaces*`) for consistent coding style.
- **Language-specific Settings**: Configure settings specific to languages or file types.

Installation of Essential Extensions:

- Click on the **Extensions** view icon in the **Activity** Bar (or use `*Ctrl+Shift+X*`).
- Search for and install essential extensions:
  Example Extensions:
  - Python Extension: The Python extension does offer some support when running on vscode.dev (which includes github.dev). This includes partial IntelliSense for open files in the editor.



Python v2024.8.1

Microsoft microsoft.com | 126,277,003 | ★★★★☆ (593)

Python language support with extension access points for IntelliSense (Pylance), Debugging (Python Debugger), linting, formatting, refactoring, unit t...

Python extension for Visual Studio Code

A Visual Studio Code extension with rich support for the Python language (for all actively supported Python versions), providing access points for extensions to seamlessly integrate and offer support for IntelliSense (Pylance), debugging (Python Debugger), formatting, linting, code navigation, refactoring, variable explorer, test explorer, and more!

Support for vscode.dev

The Python extension does offer some support when running on vscode.dev (which includes github.dev). This includes partial IntelliSense for open files in the editor.

Installed extensions

The Python extension will automatically install the following extensions by default to provide the best Python development experience in VS Code:

- Pylance - to provide performant Python language support
- Python Debugger - to provide a seamless debug experience with debugpy

These extensions are optional dependencies, meaning the Python extension will remain fully functional if they fail to be installed. Any or all of these extensions can be disabled or uninstalled at the expense of some features. Extensions installed through the marketplace are subject to the Marketplace Terms of Use.

Extensibility

The Python extension provides pluggable access points for extensions that extend various feature areas to further improve your Python development experience. These extensions are all optional and depend on your project configuration and preferences.

- Python formatters

Categories

Programming Languages | Debuggers | Other | Data Science | Machine Learning

Resources

Marketplace
Issues
Repository
License
Microsoft

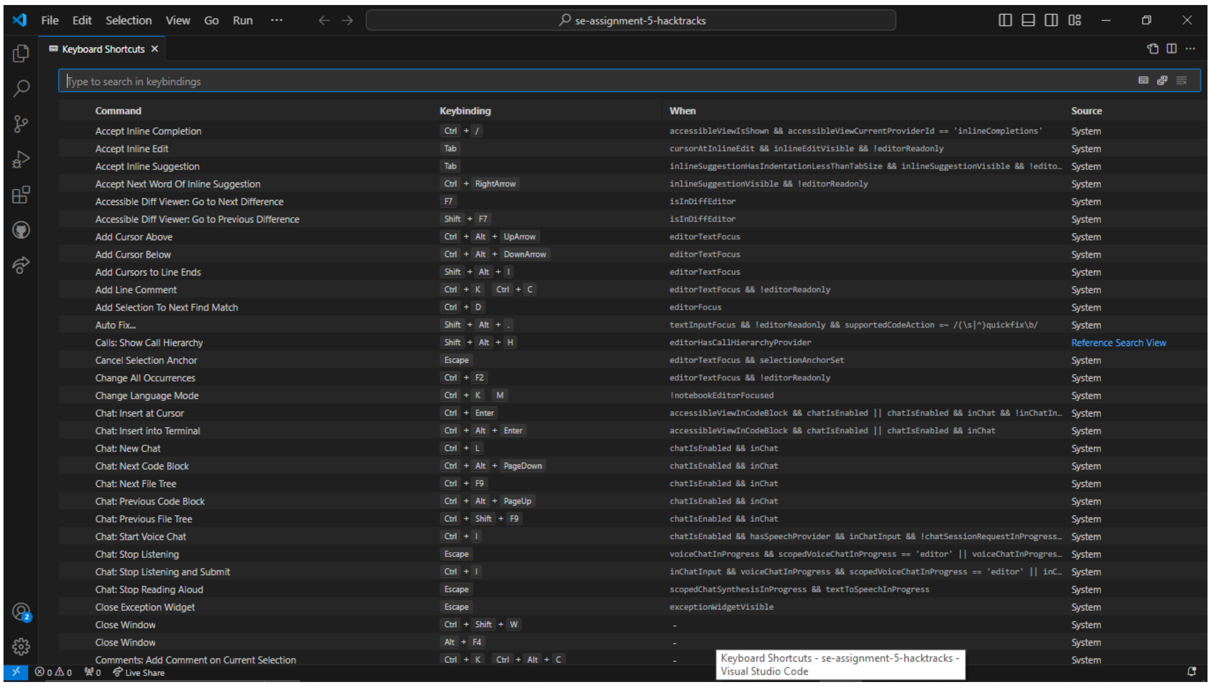
More Info

Published	2016-01-19, 18:03:11
Last released	2024-06-17, 13:22:58
Last updated	2024-06-15, 09:49:36
Identifier	ms-python.python

  - Live Server: For live preview and local server setup in web development.
  - ESLint (for JavaScript/TypeScript) or relevant linters for code quality.
  - GitLens: Enhances Git integration with advanced features.
  - Debugger for Chrome: Facilitates debugging JavaScript code in Chrome.
  - Code Spell Checker: Helps in identifying spelling errors in your comments and strings.

CUSTOMIZATION OF KEYBINDINGS IF NEEDED:

- Modify default keybindings via **File** -> **Preferences** -> **Keyboard Shortcuts** (or *Ctrl+K Ctrl+S*).


File
Edit
Selection
View
Go
Run
se-assignment-5-hacktracks
Keyboard Shortcuts
Type to search in keybindings
0 0
0
Live Share
Keyboard Shortcuts - se-assignment-5-hacktracks - Visual Studio Code

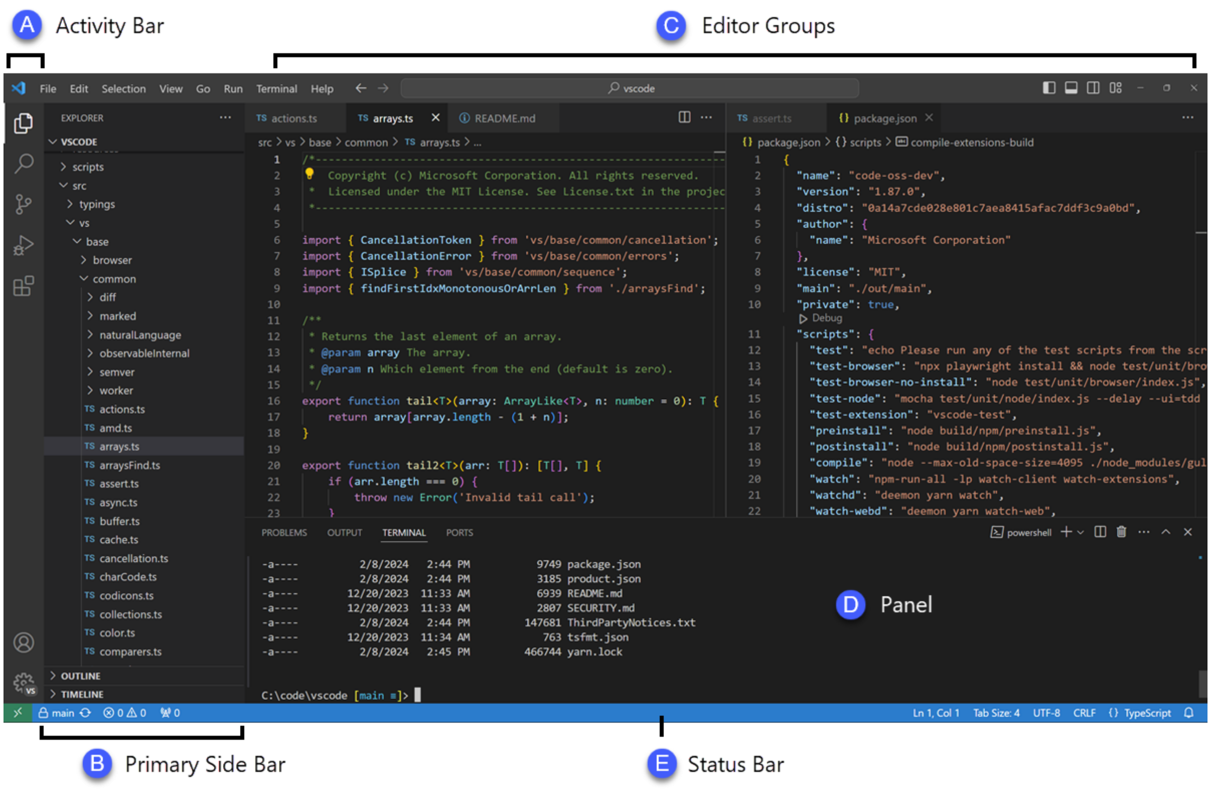
Command	Keybinding	When	Source
Accept Inline Completion	Ctrl + /	accessibleViewIsShown && accessibleViewCurrentProviderId == 'inlineCompletions'	System
Accept Inline Edit	Tab	cursorAtInlineEdit && inlineEditVisible && !editorReadonly	System
Accept Inline Suggestion	Tab	inlineSuggestionHasIndentationLessThanTabSize && inlineSuggestionVisible && !edito…	System
Accept Next Word Of Inline Suggestion	Ctrl + RightArrow	inlineSuggestionVisible && !editorReadonly	System
Accessible Diff Viewer: Go to Next Difference	F7	isInDiffEditor	System
Accessible Diff Viewer: Go to Previous Difference	Shift + F7	isInDiffEditor	System
Add Cursor Above	Ctrl + Alt + UpArrow	editorTextFocus	System
Add Cursor Below	Ctrl + Alt + DownArrow	editorTextFocus	System
Add Cursors to Line Ends	Shift + Alt + I	editorTextFocus	System
Add Line Comment	Ctrl + K Ctrl + C	editorTextFocus && !editorReadonly	System
Add Selection To Next Find Match	Ctrl + D	editorFocus	System
Auto Fix...	Shift + Alt + .	textInputFocus && !editorReadonly && supportedCodeAction =~ /(\s|^)quickfix\b/	System
Calls: Show Call Hierarchy	Shift + Alt + H	editorHasCallHierarchyProvider	Reference Search View
Cancel Selection Anchor	Escape	editorTextFocus && selectionAnchorSet	System
Change All Occurrences	Ctrl + F2	editorTextFocus && !editorReadonly	System
Change Language Mode	Ctrl + K M	!notebookEditorFocused	System
Chat: Insert at Cursor	Ctrl + Enter	accessibleViewInCodeBlock && chatIsEnabled || chatIsEnabled && inChat && !inChatIn…	System
Chat: Insert into Terminal	Ctrl + Alt + Enter	accessibleViewInCodeBlock && chatIsEnabled || chatIsEnabled && inChat	System
Chat: New Chat	Ctrl + L	chatIsEnabled && inChat	System
Chat: Next Code Block	Ctrl + Alt + PageDown	chatIsEnabled && inChat	System
Chat: Next File Tree	Ctrl + F9	chatIsEnabled && inChat	System
Chat: Previous Code Block	Ctrl + Alt + PageUp	chatIsEnabled && inChat	System
Chat: Previous File Tree	Ctrl + Shift + F9	chatIsEnabled && inChat	System
Chat: Start Voice Chat	Ctrl + I	chatIsEnabled && hasSpeechProvider && inChatInput && !chatSessionRequestInProgress…	System
Chat: Stop Listening	Escape	voiceChatInProgress && scopedVoiceChatInProgress == 'editor' || voiceChatInProgres…	System
Chat: Stop Listening and Submit	Ctrl + I	inChatInput && voiceChatInProgress && scopedVoiceChatInProgress == 'editor' || inC…	System
Chat: Stop Reading Aloud	Escape	scopedChatSynthesisInProgress && textToSpeechInProgress	System
Close Exception Widget	Escape	exceptionWidgetVisible	System
Close Window	Ctrl + Shift + W	-	System
Close Window	Alt + F4	-	System
Comments: Add Comment on Current Selection	Ctrl + K Ctrl + Alt + C	-	System

- Define custom keybindings for specific actions or extensions to streamline your workflow.
- Export or import keybindings for consistency across different machines (`keybindings.json`).

By configuring these settings, installing essential extensions, and customizing keybindings, you'll tailor VS Code to meet your specific coding preferences and productivity needs from the outset. This initial setup lays the foundation for a productive coding environment tailored to your workflow.

USER INTERFACE OVERVIEW

At its heart, Visual Studio Code is a code editor. Like many other code editors, VS Code adopts a common user interface and layout of an explorer on the left, showing all of the files and folders you have access to, and an editor on the right, showing the content of the files you have opened.



```
/*---------------------------------------------------------------
 *  Copyright (c) Microsoft Corporation. All rights reserved.
 *  Licensed under the MIT License. See License.txt in the projec
 *---------------------------------------------------------------

import { CancellationToken } from 'vs/base/common/cancellation';
import { CancellationError } from 'vs/base/common/errors';
import { ISplice } from 'vs/base/common/sequence';
import { findFirstIdxMonotonousOrArrLen } from './arraysFind';

/**
 * Returns the last element of an array.
 * @param array The array.
 * @param n Which element from the end (default is zero).
 */
export function tail<T>(array: ArrayLike<T>, n: number = 0): T {
    return array[array.length - (1 + n)];
}

export function tail2<T>(arr: T[]): [T[], T] {
    if (arr.length === 0) {
        throw new Error('Invalid tail call');
    }
```

```
{
  "name": "code-oss-dev",
  "version": "1.87.0",
  "distro": "0a14a7cde028e801c7aea8415afac7ddf3c9a0bd",
  "author": {
    "name": "Microsoft Corporation"
  },
  "license": "MIT",
  "main": "./out/main",
  "private": true,
  "scripts": {
    "test": "echo Please run any of the test scripts from the scr
    "test-browser": "npx playwright install && node test/unit/bro
    "test-browser-no-install": "node test/unit/browser/index.js",
    "test-node": "mocha test/unit/node/index.js --delay --ui=tdd
    "test-extension": "vscode-test",
    "preinstall": "node build/npm/preinstall.js",
    "postinstall": "node build/npm/postinstall.js",
    "compile": "node --max-old-space-size=4095 ./node_modules/gul
    "watch": "npm-run-all -lp watch-client watch-extensions",
    "watchd": "deemon yarn watch",
    "watch-webd": "deemon yarn watch-web",
```

```
-a----          2/8/2024   2:44 PM           9749 package.json
-a----          2/8/2024   2:44 PM           3185 product.json
-a----        12/20/2023  11:33 AM           6939 README.md
-a----        12/20/2023  11:33 AM           2807 SECURITY.md
-a----          2/8/2024   2:44 PM         147681 ThirdPartyNotices.txt
-a----        12/20/2023  11:34 AM            763 tsfmt.json
-a----          2/8/2024   2:45 PM         466744 yarn.lock

C:\code\vscode [main ≡]>
```

VS Code comes with a simple and intuitive layout that maximizes the space provided for the editor, while leaving ample room to browse and access the full context of your folder or project. The user interface is divided into five main areas:

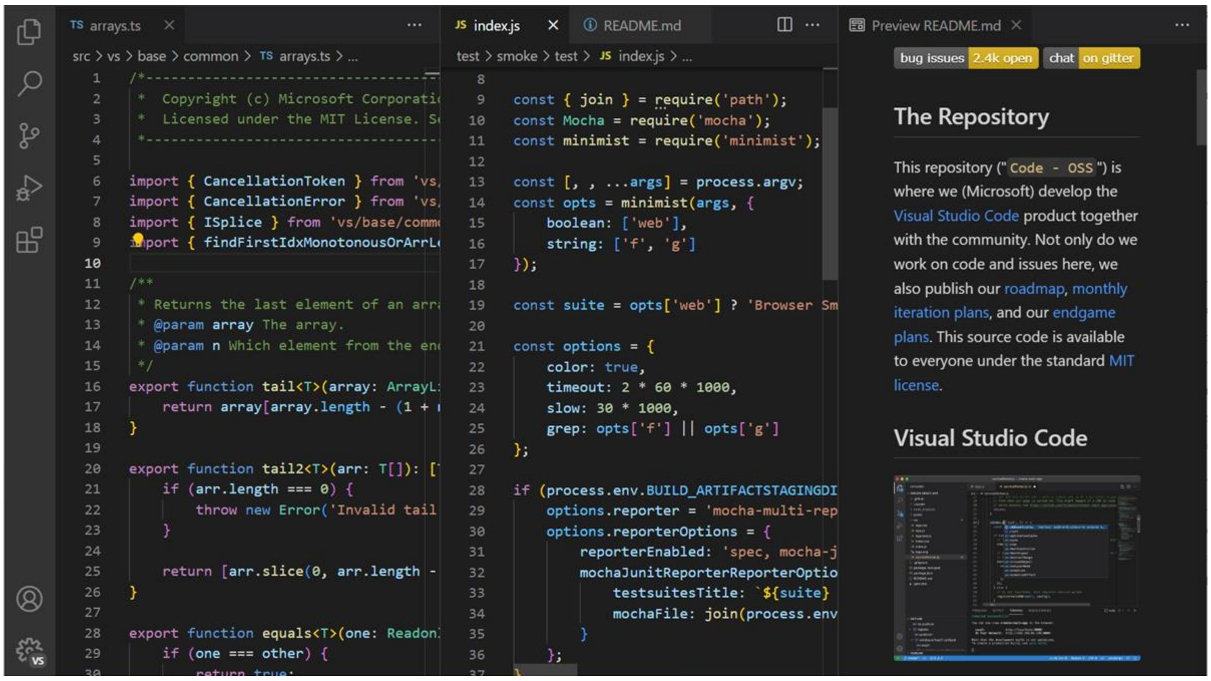
- **Editor** - The main area to edit your files. You can open as many editors as you like side by side vertically and horizontally.
- **Primary Side Bar** - Contains different views like the Explorer to assist you while working on your project.
- **Status Bar** - Information about the opened project and the files you edit.
- **Activity Bar** - Located on the far left-hand side. Lets you switch between views and gives you additional context-specific indicators, like the number of outgoing changes when Git is enabled. You can change the position of the Activity Bar.
- **Panel** - An additional space for views below the editor region. By default, it contains output, debug information, errors and warnings, and an integrated terminal. The Panel can also be moved to the left or right for more vertical space.

Each time you start VS Code, it opens up in the same state it was in when you last closed it. The folder, layout, and opened files are preserved. Open files in each editor are displayed with tabbed headers (Tabs) at the top of the editor region. To learn more about tabbed headers, see the Tabs section.

<u>Side by Side Editing</u>

You can open as many editors as you like side by side vertically and horizontally. If you already have an editor open, there are multiple ways of opening another editor to the side:

- **Alt** and select a file in the Explorer view.
- **Ctrl** + \ to split the active editor into two.
- Open to the Side (**Ctrl** + **Enter**) from the Explorer context menu on a file.
- Select the Split Editor button in the upper right of an editor.
- Drag and drop a file to any side of the editor region.
- Press **Ctrl** + **Enter** in the Quick Open (**Ctrl** + **P**) file list.


TS arrays.ts
src > vs > base > common > TS arrays.ts > ...
JS index.js
README.md
test > smoke > test > JS index.js > ...
Preview README.md
bug issues 2.4k open
chat on gitter

```
/*--------------------------------
 *  Copyright (c) Microsoft Corporati
 *  Licensed under the MIT License. S
 *--------------------------------

import { CancellationToken } from 'vs,
import { CancellationError } from 'vs,
import { ISplice } from 'vs/base/comm
import { findFirstIdxMonotonousOrArrL

/**
 * Returns the last element of an arr
 * @param array The array.
 * @param n Which element from the en
 */
export function tail<T>(array: ArrayL
	return array[array.length - (1 +
}

export function tail2<T>(arr: T[]): [
	if (arr.length === 0) {
		throw new Error('Invalid tail
	}

	return [arr.slice(0, arr.length -
}

export function equals<T>(one: Readon
	if (one === other) {
```

```

const { join } = require('path');
const Mocha = require('mocha');
const minimist = require('minimist');

const [, , ...args] = process.argv;
const opts = minimist(args, {
	boolean: ['web'],
	string: ['f', 'g']
});

const suite = opts['web'] ? 'Browser Sm

const options = {
	color: true,
	timeout: 2 * 60 * 1000,
	slow: 30 * 1000,
	grep: opts['f'] || opts['g']
};

if (process.env.BUILD_ARTIFACTSTAGINGDI
	options.reporter = 'mocha-multi-rep
	options.reporterOptions = {
		reporterEnabled: 'spec, mocha-j
		mochaJunitReporterReporterOptio
			testsuitesTitle: `${suite}
			mochaFile: join(process.env
		}
	};
```

The Repository

This repository ("`Code - OSS`") is where we (Microsoft) develop the Visual Studio Code product together with the community. Not only do we work on code and issues here, we also publish our roadmap, monthly iteration plans, and our endgame plans. This source code is available to everyone under the standard MIT license.

Visual Studio Code

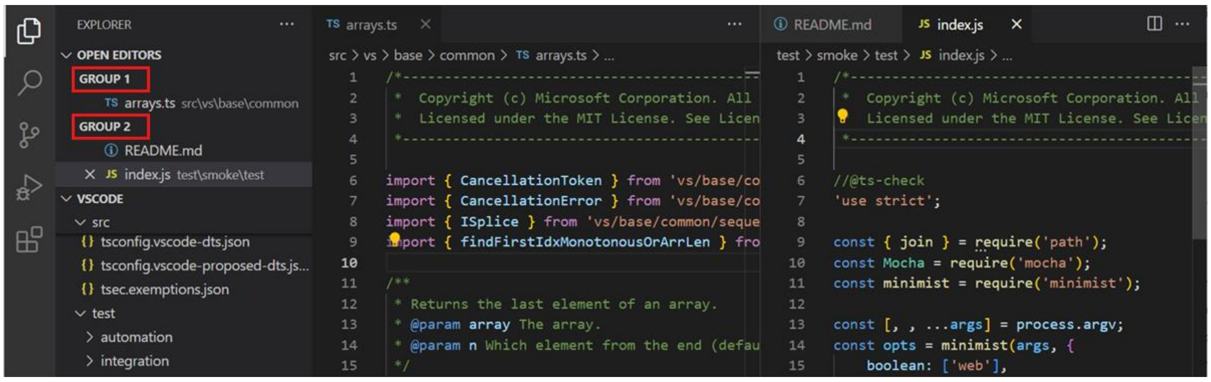
When you open another file, the editor that is active will display the content of that file. By default, editors open to the right-hand side of the active one. You can change this behavior with the *workbench.editor.openSideBySideDirection* setting to open new editors to the bottom of the active one instead.

Editor Groups

When you split an editor (using the **Split Editor** or **Open to the Side** commands), a new editor region (edit group) is created which can hold a group of items. You can open as many editor groups as you like side by side vertically and horizontally.

You can see these clearly in the Open Editors section at the top of the Explorer view

- toggle ... > **Open Editors** in the Explorer view.

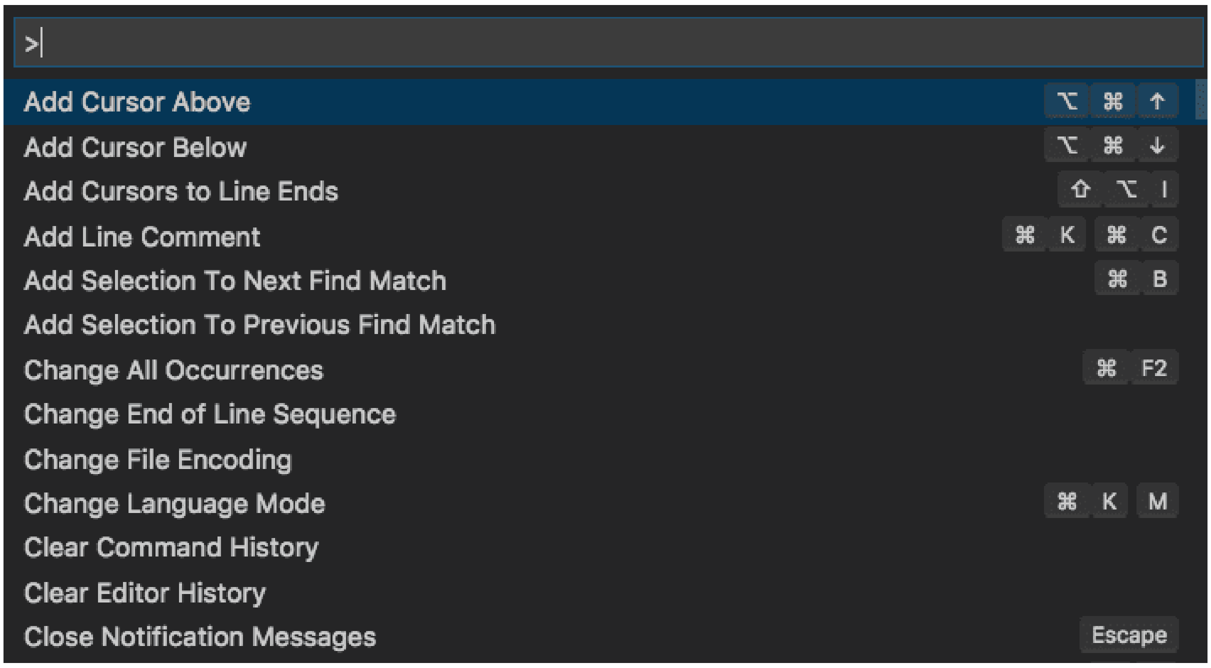


```
/*------------------------------------------
 *  Copyright (c) Microsoft Corporation. All
 *  Licensed under the MIT License. See Licen
 *-----------------------------------------

import { CancellationToken } from 'vs/base/co
import { CancellationError } from 'vs/base/co
import { ISplice } from 'vs/base/common/seque
import { findFirstIdxMonotonousOrArrLen } fro

/**
 * Returns the last element of an array.
 * @param array The array.
 * @param n Which element from the end (defau
 */
```

```
/*------------------------------------------
 *  Copyright (c) Microsoft Corporation. All
 *  Licensed under the MIT License. See Licen
 *-----------------------------------------

//@ts-check
'use strict';

const { join } = require('path');
const Mocha = require('mocha');
const minimist = require('minimist');

const [, , ...args] = process.argv;
const opts = minimist(args, {
    boolean: ['web'],
```

Command Palette

VS Code is equally accessible from the keyboard. The most important key combination to know is ***Ctrl+ Shift+ P***, which brings up the Command Palette. From here, you have access to all functionality within VS Code, including keyboard shortcuts for the most common operations.

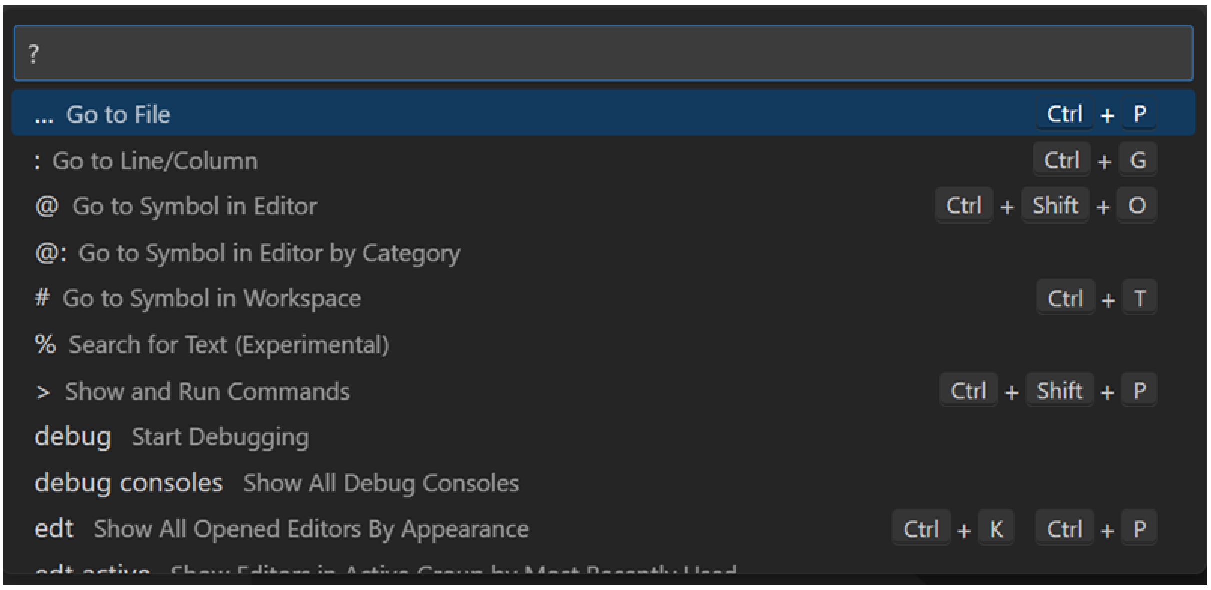


```
>|
Add Cursor Above                          ⌥ ⌘ ↑
Add Cursor Below                          ⌥ ⌘ ↓
Add Cursors to Line Ends                  ⇧ ⌥ I
Add Line Comment                          ⌘ K ⌘ C
Add Selection To Next Find Match          ⌘ B
Add Selection To Previous Find Match
Change All Occurrences                    ⌘ F2
Change End of Line Sequence
Change File Encoding
Change Language Mode                      ⌘ K M
Clear Command History
Clear Editor History
Close Notification Messages               Escape
```

The Command Palette provides access to many commands. You can run editor commands, open files, search for symbols, and see a quick outline of a file, all using the same interactive window. Here are a few tips:

- **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.



```
?
... Go to File                                   Ctrl + P
: Go to Line/Column                              Ctrl + G
@ Go to Symbol in Editor                         Ctrl + Shift + O
@: Go to Symbol in Editor by Category
# Go to Symbol in Workspace                      Ctrl + T
% Search for Text (Experimental)
> Show and Run Commands                          Ctrl + Shift + P
debug  Start Debugging
debug consoles  Show All Debug Consoles
edt  Show All Opened Editors By Appearance       Ctrl + K  Ctrl + P
```

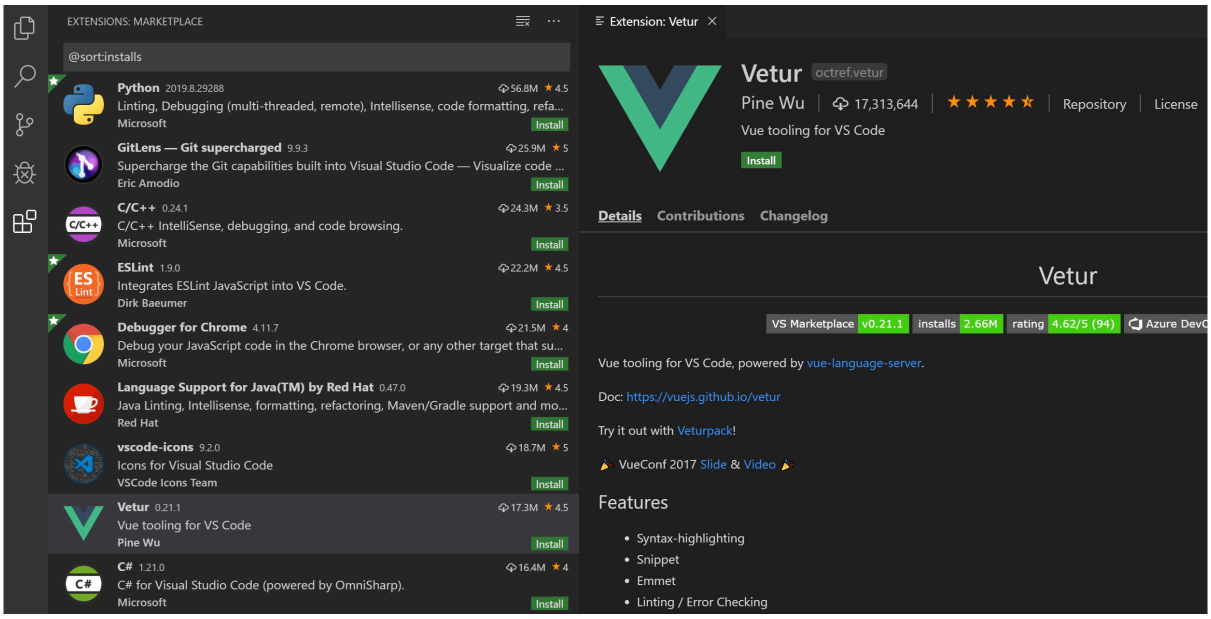
EXTENSIONS IN VS CODE

Increase the power of Visual Studio Code through Extensions. The features that Visual Studio Code includes out-of-the-box are just the start. VS Code extensions let you add languages, debuggers, and tools to your installation to support your development workflow. VS Code's rich extensibility model lets extension authors plug directly into the VS Code UI and contribute functionality through the same APIs used by VS Code. This article explains how to find, install, and manage VS Code extensions from the Visual Studio Code Marketplace.

Browse for extensions

You can browse and install extensions from within VS Code. Bring up the Extensions view by clicking on the Extensions icon in the Activity Bar on the side of VS Code or the View: Extensions command (**Ctrl + Shift + X**).

This will show you a list of the most popular VS Code extensions on the VS Code Marketplace.



EXTENSIONS: MARKETPLACE

@sort:installs

Python 2019.8.29288 56.8M ★ 4.5
Linting, Debugging (multi-threaded, remote), Intellisense, code formatting, refa...
Microsoft
Install

GitLens — Git supercharged 9.9.3 25.9M ★ 5
Supercharge the Git capabilities built into Visual Studio Code — Visualize code ...
Eric Amodio
Install

C/C++ 0.24.1 24.3M ★ 3.5
C/C++ IntelliSense, debugging, and code browsing.
Microsoft
Install

ESLint 1.9.0 22.2M ★ 4.5
Integrates ESLint JavaScript into VS Code.
Dirk Baeumer
Install

Debugger for Chrome 4.11.7 21.5M ★ 4
Debug your JavaScript code in the Chrome browser, or any other target that su...
Microsoft
Install

Language Support for Java(TM) by Red Hat 0.47.0 19.3M ★ 4.5
Java Linting, Intellisense, formatting, refactoring, Maven/Gradle support and mo...
Red Hat
Install

vscode-icons 9.2.0 18.7M ★ 5
Icons for Visual Studio Code
VSCode Icons Team
Install

Vetur 0.21.1 17.3M ★ 4.5
Vue tooling for VS Code
Pine Wu
Install

C# 1.21.0 16.4M ★ 4
C# for Visual Studio Code (powered by OmniSharp).
Microsoft
Install

Extension: Vetur

Vetur octref.vetur
Pine Wu | 17,313,644 | ★★★★★ | Repository | License
Vue tooling for VS Code
Install

Details Contributions Changelog

Vetur

VS Marketplace v0.21.1 installs 2.66M rating 4.62/5 (94) Azure DevO

Vue tooling for VS Code, powered by vue-language-server.

Doc: https://vuejs.github.io/vetur

Try it out with Veturpack!

VueConf 2017 Slide & Video

Features

- Syntax-highlighting
- Snippet
- Emmet
- Linting / Error Checking

Each extension in the list includes a brief description, the publisher, the download count, and a five star rating. You can select the extension item to display the extension's details page where you can learn more.

Install an extension

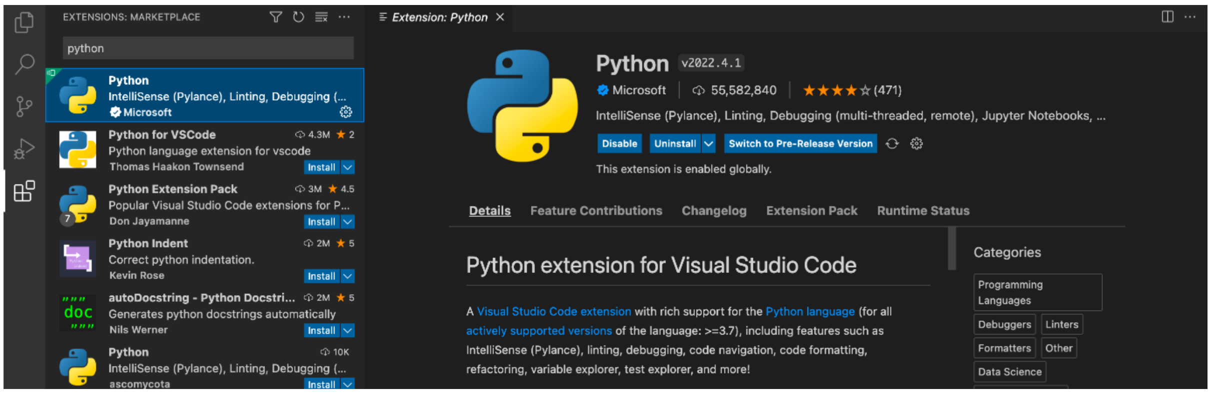
To install an extension, select the Install button. Once the installation is complete, the Install button will change to the Manage gear button

If an extension doesn't provide the functionality you want, you can always **Uninstall** the extension from the Manage button context menu.

Search for an extension

You can clear the Search box at the top of the Extensions view and type in the name of the extension, tool, or programming language you're looking for.

For example, typing 'python' will bring up a list of Python language extensions:



Python
IntelliSense (Pylance), Linting, Debugging (...
Microsoft

Python for VSCode 4.3M ★ 2
Python language extension for vscode
Thomas Haakon Townsend

Python Extension Pack 3M ★ 4.5
Popular Visual Studio Code extensions for P...
Don Jayamanne

Python Indent 2M ★ 5
Correct python indentation.
Kevin Rose

autoDocstring - Python Docstri... 2M ★ 5
Generates python docstrings automatically
Nils Werner

Python 10K
IntelliSense (Pylance), Linting, Debugging (...
ascomycota

Python v2022.4.1
Microsoft | 55,582,840 | ★★★★☆ (471)
IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Jupyter Notebooks, ...
This extension is enabled globally.

Python extension for Visual Studio Code

A Visual Studio Code extension with rich support for the Python language (for all actively supported versions of the language: >=3.7), including features such as IntelliSense (Pylance), linting, debugging, code navigation, code formatting, refactoring, variable explorer, test explorer, and more!

Manage extensions

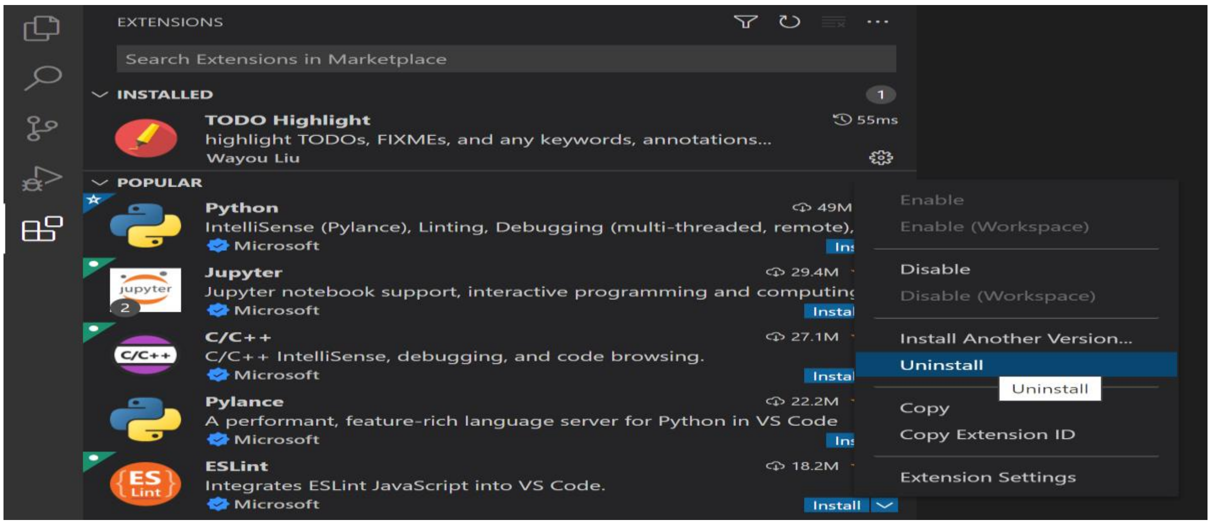
VS Code makes it easy to manage your extensions. You can install, disable, update, and uninstall extensions through the Extensions view, the Command Palette (commands have the Extensions: prefix) or command-line switches.

List installed extensions

By default, the Extensions view will show the extensions you currently have installed, and all extensions that are recommended for you. You can use the Extensions: Focus on Installed View command, available in the Command Palette (**Ctrl + Shift + P**) or in the More Actions (...) dropdown menu > Views > Installed, to clear any text in the search box and show the list of all installed extensions, which includes those that have been disabled.

Uninstall an extension

To uninstall an extension, select the Manage gear button at the right of an extension entry and then choose Uninstall from the dropdown menu. This will uninstall the extension and prompt you to restart the extension host (Restart Extensions).



TODO Highlight 55ms
highlight TODOs, FIXMEs, and any keywords, annotations...
Wayou Liu

Python 49M
IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote),
Microsoft

Jupyter 29.4M
Jupyter notebook support, interactive programming and computing
Microsoft

C/C++ 27.1M
C/C++ IntelliSense, debugging, and code browsing.
Microsoft

Pylance 22.2M
A performant, feature-rich language server for Python in VS Code
Microsoft

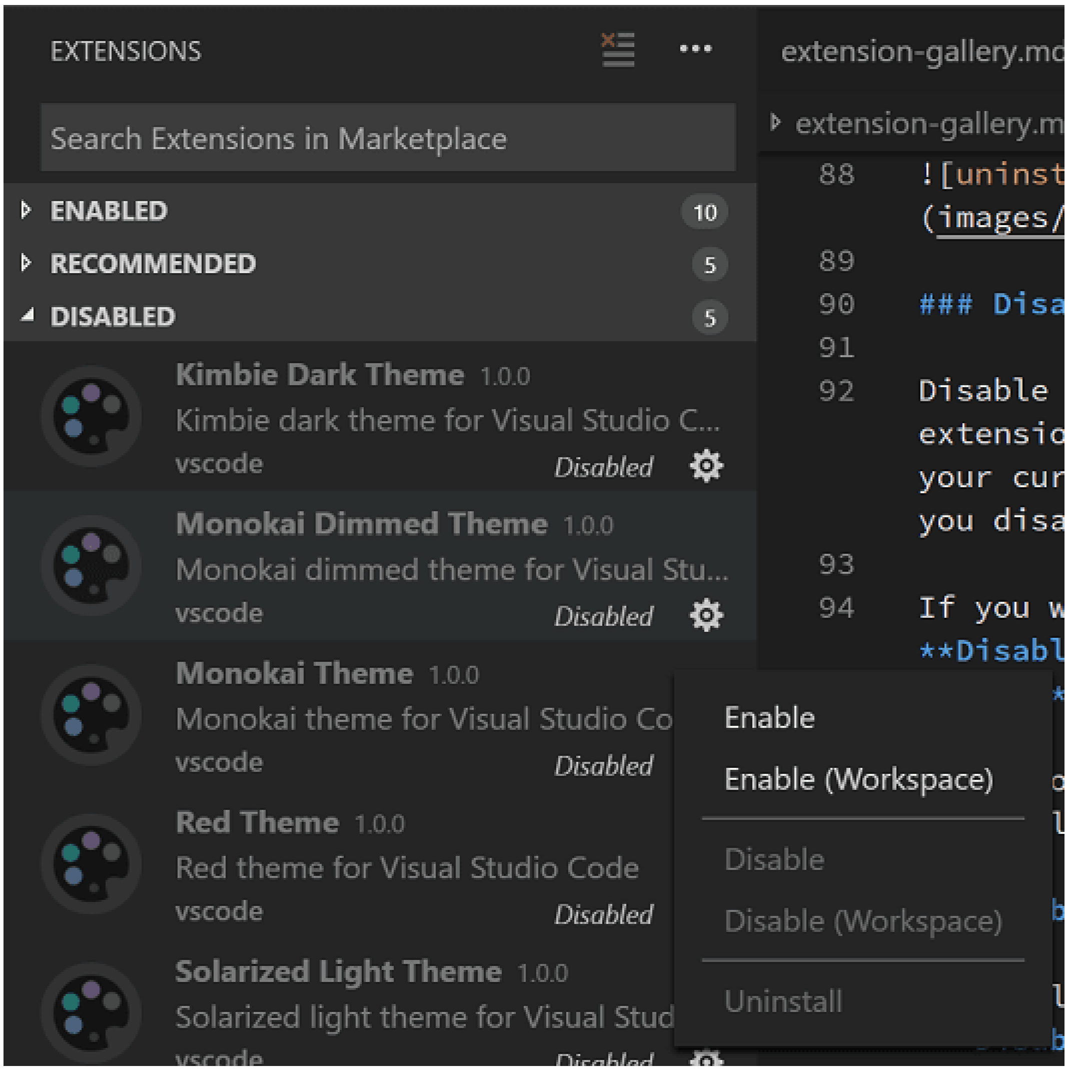
ESLint 18.2M
Integrates ESLint JavaScript into VS Code.
Microsoft

Disable an extension

If you don't want to permanently remove an extension, you can instead temporarily disable the extension by clicking the gear button at the right of an extension entry. You can disable an extension globally or just for your current Workspace. You will be prompted to restart the extension host (Restart Extensions) after you disable an extension.
If you want to quickly disable all installed extensions, there is a Disable All Installed Extensions command in the Command Palette and More Actions (...) dropdown menu. Extensions remain disabled for all VS Code sessions until you re-enable them.

Enable an extension

Similarly if you have disabled an extension (it will be in the Disabled section of the list and marked Disabled), you can re-enable it with the Enable or Enable (Workspace) commands in the dropdown menu.



When setting up Visual Studio Code (VS Code) for web development on Windows 11, installing the right extensions can significantly enhance your productivity and streamline your workflow. Here are some essential extensions along with their functionalities:

EXTENSION	**FUNCTIONALITY**	**BENEFITS**
Live Server	Launches a local development server with live reload capability	Allows you to see changes in your web application instantly without manually refreshing the browser
ESLint	JavaScript linter that helps identify and fix code errors, enforce coding styles, and improve code quality.	Ensures consistency and best practices in JavaScript code, reducing potential bugs and maintaining code readability.
Debugger for Chrome	Allows debugging of JavaScript code running in the Google Chrome browser directly from VS Code.	Streamlines the debugging process by providing a seamless integration between your code editor and the browser
Prettier - Code Formatter	Code formatter that automatically formats your code according to predefined rules.	Maintains consistent code style across your project, reducing formatting-related issues and improving readability.
Auto Close Tag	Automatically closes HTML/XML tags when you type </.	Speeds up HTML/XML coding by reducing the need for manual tag closing, improving efficiency.
CSS Peek	Allows peeking into CSS definitions from HTML files or inline CSS without switching between files.	Enhances CSS editing by providing quick access to style definitions, improving productivity.
Path Intellisense	Autocompletes file paths in your code.	Reduces errors and time spent manually typing or searching for file paths, improving efficiency.
GitLens	Enhances Git integration with advanced features such as commit history, blame information, and inline changes.	Provides comprehensive insights into your Git repositories directly within VS Code, facilitating better version control management.
Jest	Integration with Jest, a JavaScript testing framework.	Enables running and debugging Jest tests directly from VS Code, enhancing the testing process for JavaScript projects.
Color Highlight	Highlights colors (hex, rgb, etc.) in your code as actual color previews.	Makes it easier to identify and edit colors directly within your code, improving design and styling workflows.

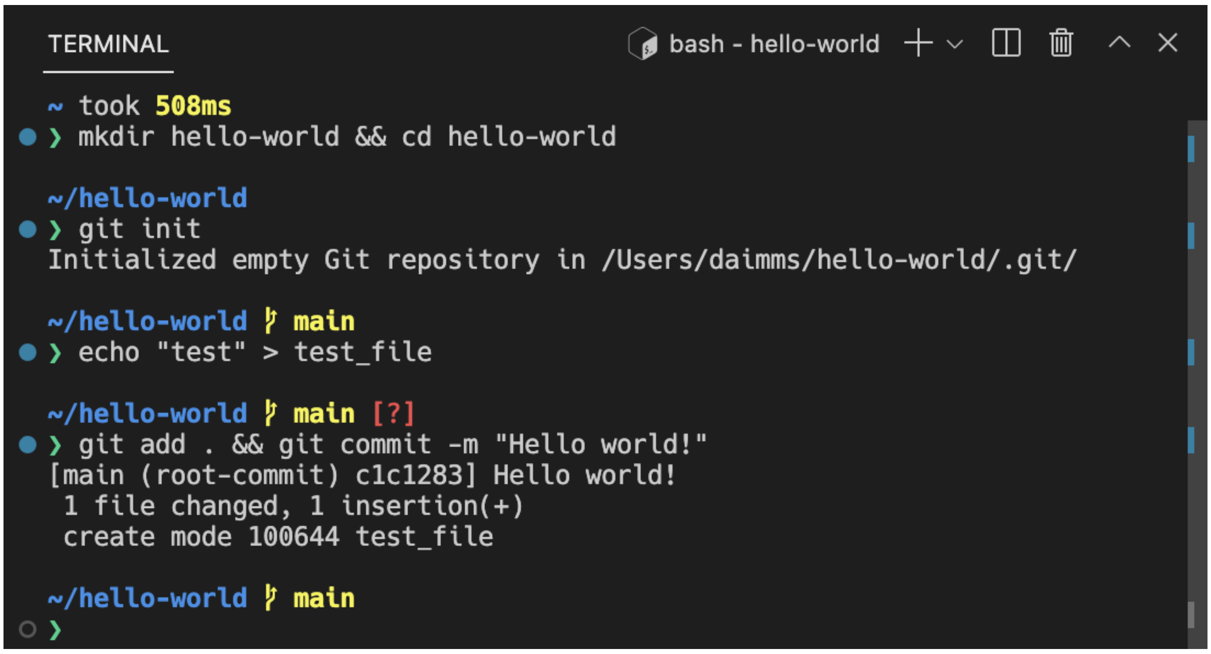
INTEGRATED TERMINAL

Visual Studio Code includes a full featured integrated terminal that starts at the root of your workspace. It provides integration with the editor to support features like links and error detection. The integrated terminal can run commands such as mkdir and git just like a standalone terminal.

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 **menu** 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:



```
~ took 508ms
❯ mkdir hello-world && cd hello-world

~/hello-world
❯ git init
Initialized empty Git repository in /Users/daimms/hello-world/.git/

~/hello-world  main
❯ echo "test" > test_file

~/hello-world  main [?]
❯ git add . && git commit -m "Hello world!"
[main (root-commit) c1c1283] Hello world!
 1 file changed, 1 insertion(+)
 create mode 100644 test_file

~/hello-world  main
❯
```

Using the integrated terminal in Visual Studio Code (VS Code) on Windows 11 offers several advantages over using an external terminal application:

- **Seamless Integration**: The integrated terminal is seamlessly integrated within the VS Code interface. It shares the same window as your code editor, reducing the need to switch between multiple applications or windows.
- **Contextual Awareness**: The integrated terminal automatically opens at the root of your current workspace. This contextual awareness allows for quicker navigation and execution of commands related to your project without needing to manually navigate to the project directory.
- **Direct Interaction with Code**: You can interact directly with files and folders in your project using commands executed in the integrated terminal. For example, you can run build commands, start servers, or execute scripts without leaving the VS Code environment.

- **Customization and Configuration**: VS Code allows you to customize the integrated terminal with different shell environments (e.g., Command Prompt, PowerShell, Bash) according to your preference or project requirements. You can also customize terminal colors, font sizes, and other settings to suit your needs.
- **Enhanced Productivity**: Since the terminal is integrated, you can use VS Code's powerful features such as IntelliSense for code completion, syntax highlighting, and error checking while interacting with the terminal. This integration helps in maintaining productivity by reducing context switching and keeping focus within the same environment.
- **Debugging Support**: Integrated terminals can be directly used for debugging processes. For instance, if your project involves running server processes or scripts that produce debug output, having the terminal integrated allows you to view and interact with debug messages in real-time alongside your code editor.
- **Workspace Management**: VS Code manages multiple terminal instances per workspace, allowing you to have different terminals open for different tasks or projects simultaneously. This flexibility in workspace management enhances organization and multitasking capabilities.
- **Cross-Platform Consistency**: Whether you're developing on Windows 11, macOS, or Linux, the integrated terminal functionality in VS Code provides a consistent experience across different operating systems. This consistency simplifies development workflows when working on projects across diverse environments.

GIT FOR SOURCE CONTROL IN VS CODE

Git and GitHub are the tools you need! And with Visual Studio Code, you can set up and use them in a snap. Even if you're a beginner, VS Code's user-friendly interface guides you through common Git actions like pushing and pulling code, creating and merging branches, and committing code changes.

Set up Git in VS Code

To use Git and GitHub in VS Code, first make sure you have Git installed on your computer. If Git is missing, the Source Control view shows instructions on how to install it. Make sure to restart VS Code afterwards.

Additionally you can sign into VS Code with your GitHub account in the Accounts menu in the lower right of the Activity bar to enable additional features like Settings Sync, but also cloning and publishing repositories from GitHub.

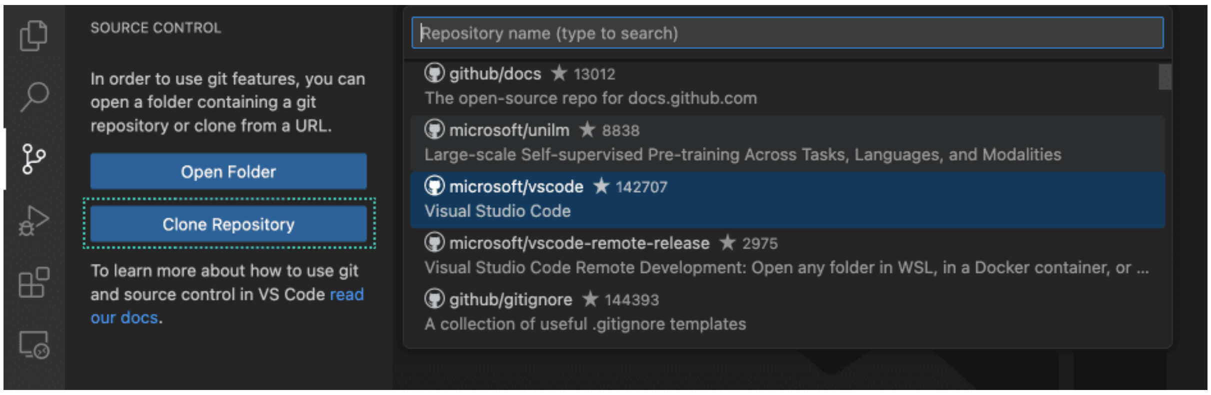
Open a Git repository

VS Code provides several ways to get started in a Git repository, from local to remote cloud-powered environments like GitHub Codespaces.

Clone a repository locally

To clone a repository, run the **Git: Clone** command in the Command Palette (**Ctrl + Shift + P**), or select the Clone Repository button in the Source Control view.

If you clone from GitHub, VS Code prompts you to authenticate with GitHub. Then, select a repository from the list to clone to your machine. The list contains both public and private repositories.


SOURCE CONTROL

In order to use git features, you can open a folder containing a git repository or clone from a URL.

To learn more about how to use git and source control in VS Code read our docs.

github/docs ★ 13012
The open-source repo for docs.github.com

microsoft/unilm ★ 8838
Large-scale Self-supervised Pre-training Across Tasks, Languages, and Modalities

microsoft/vscode ★ 142707
Visual Studio Code

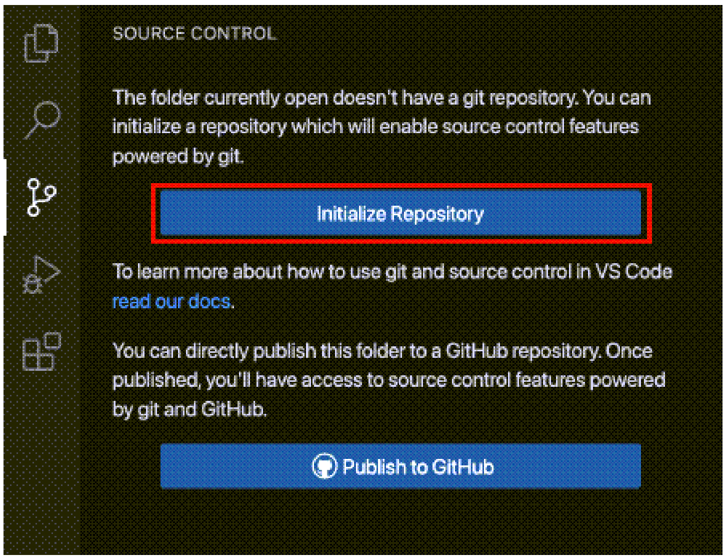
microsoft/vscode-remote-release ★ 2975
Visual Studio Code Remote Development: Open any folder in WSL, in a Docker container, or ...

github/gitignore ★ 144393
A collection of useful .gitignore templates

Initialize a repository in a local folder

To initialize a new local Git repository:

- Pick an existing or new folder on your computer and open it in VS Code.
- In the Source Control view, select the Initialize Repository button.
- This creates a new Git repository in the current folder, allowing you to start tracking code changes.


SOURCE CONTROL

The folder currently open doesn't have a git repository. You can initialize a repository which will enable source control features powered by git.

To learn more about how to use git and source control in VS Code read our docs.

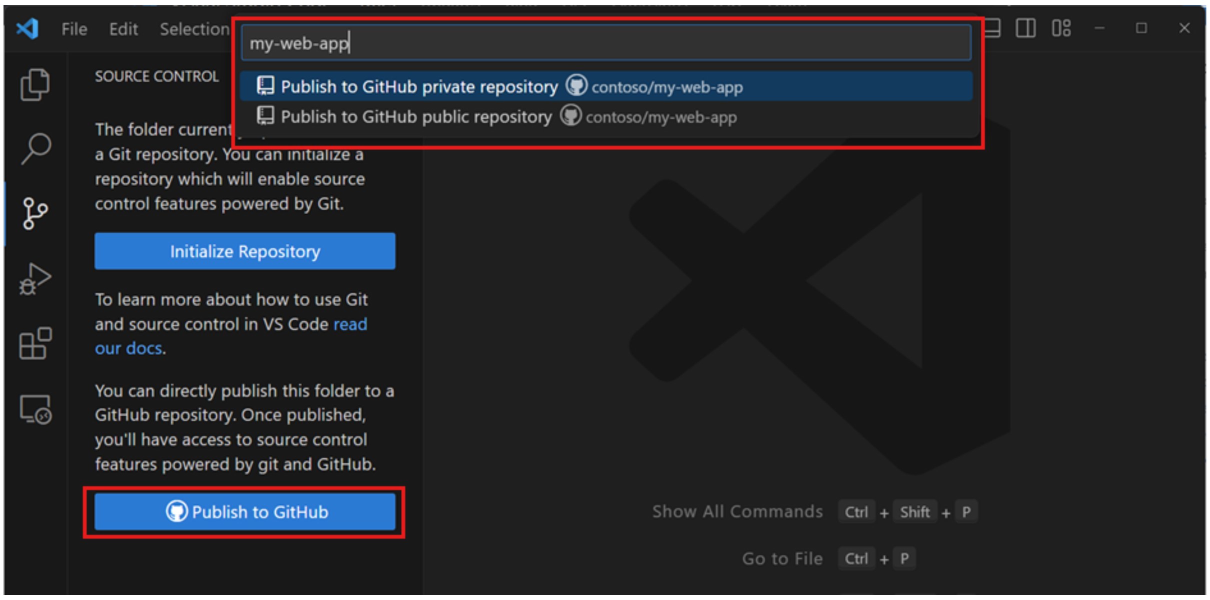
You can directly publish this folder to a GitHub repository. Once published, you'll have access to source control features powered by git and GitHub.

- This action is equivalent to running **git init** on the command-line.

Publish local repository to GitHub

You can also initialize and local repository and publish it directly to GitHub. This creates a new repository on your GitHub account, and pushes your local code changes to the remote repository. Having your source code on a remote repository is a great way to back up your code, collaborate with others, and automate your workflow with GitHub Actions.

Use the Publish to GitHub command button in the Source Control view. You can then choose a name and description for the repository, and whether to make it public or private.


File Edit Selection
my-web-app
Publish to GitHub private repository contoso/my-web-app
Publish to GitHub public repository contoso/my-web-app
SOURCE CONTROL
Initialize Repository
Publish to GitHub
Show All Commands Ctrl + Shift + P
Go to File Ctrl + P

The folder current... a Git repository. You can initialize a repository which will enable source control features powered by Git.

To learn more about how to use Git and source control in VS Code read our docs.

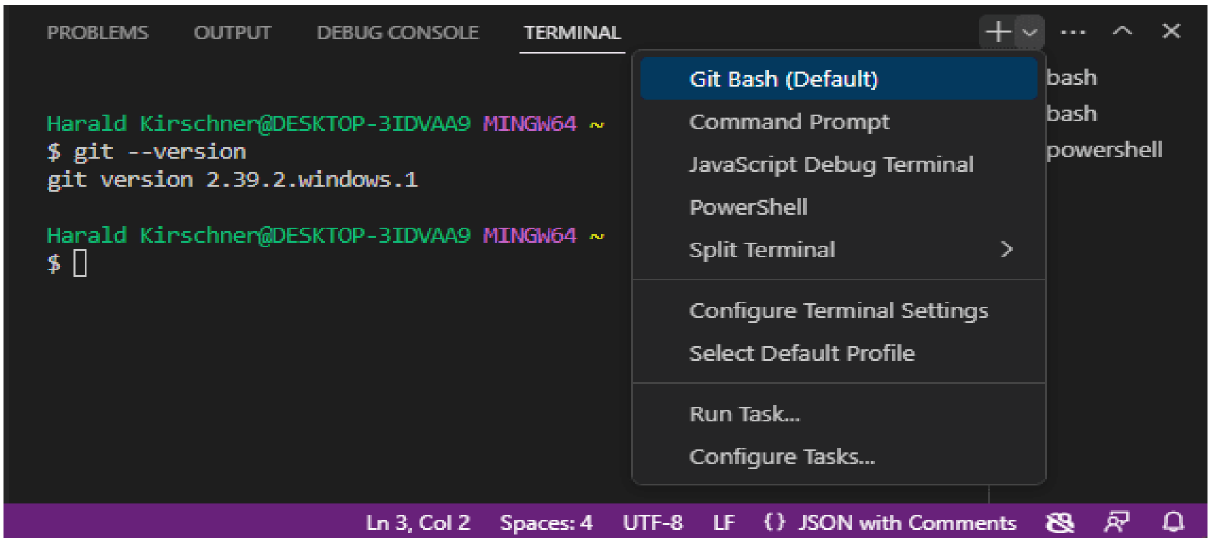
You can directly publish this folder to a GitHub repository. Once published, you'll have access to source control features powered by git and GitHub.

Using Git in the built-in terminal

As all Git state is kept in the local repository, you can easily switch between VS Code's UI, the built-in terminal, or external tools like GitHub Desktop. You can also set up VS Code as your default Git editor, allowing you to use VS Code to edit commit messages and other Git-related files.

Git Bash on Windows

Git Bash is a popular shell environment for Windows that provides a Unix-like command-line interface for working with Git and other command-line tools. Visual Studio Code's integrated terminal supports Git Bash as a shell, allowing you to seamlessly integrate Git Bash into your development workflow. Installing Git on your Windows machine also installs Git Bash, if it wasn't deselected during the installation steps.


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Git Bash (Default)
Command Prompt
JavaScript Debug Terminal
PowerShell
Split Terminal
Configure Terminal Settings
Select Default Profile
Run Task...
Configure Tasks...
bash
bash
powershell
Ln 3, Col 2 Spaces: 4 UTF-8 LF {} JSON with Comments

```
Harald Kirschner@DESKTOP-3IDVAA9 MINGW64 ~
$ git --version
git version 2.39.2.windows.1

Harald Kirschner@DESKTOP-3IDVAA9 MINGW64 ~
$ 
```

Start by opening View > Terminal (Ctrl+`). Select the dropdown arrow next to the + icon in the terminal panel to pick a new shell to open. If Git Bash is installed, it's shown in the list of terminals and shells. You can toggle between different terminals and shells in the Terminal sidebar. With Git Bash configured in Visual Studio Code, you can now use all of your favorite Git commands directly from the terminal in your code editor.

If you want to set Git Bash as your default shell, open the Terminal dropdown (next to the + icon) and select Select Default Profile. This opens a list of available shells, including Git Bash. Selecting Git Bash sets it as your default shell, and all future terminals will be opened with Git Bash.

Initialization of a repository, making commits, and pushing changes to GitHub using Git Bash.

- Create a local directory using the following command:

```
$ mkdir test

$ cd test
```

- The next step is to initialize the directory:

```
$ git init
```

- Go to the folder where "test" is created and create a text document named "demo." Open "demo" and put any content, like "Hello PLP." Save and close the file.
- Enter the Git bash interface and type in the following command to check the status:

```
$ git status
```

- Add the "demo" to the current directory using the following command:

```
$ git add demo.txt
```

- Next, make a commit using the following command:

```
$ git commit -m "committing a text file"
```

- Link the Git to a Github Account:

```
$ git config --global user.username
```

- Open your Github account and create a new repository with the name "test_demo" and click on "Create repository." This is the remote repository. Next, copy the link of "test_demo."
- Go back to Git bash and link the remote and local repository using the following command:

```
$ git remote add origin <link>
```

Here, <link> is the link copied in the previous step.

REFERENCES

Introduction to git in visual studio code. (2021, November 3). Retrieved from Visual Studio Code - Code Editing. Redefined: https://code.visualstudio.com/docs/sourcecontrol/intro-to-git

Visual studio code themes. (2021, November 3). Retrieved from Visual Studio Code - Code Editing. Redefined: https://code.visualstudio.com/docs/getstarted/themes

Visual studio code user interface. (2021, November 2021 3). Retrieved from Visual Studio Code - Code Editing. Redefined: https://code.visualstudio.com/docs/getstarted/userinterface