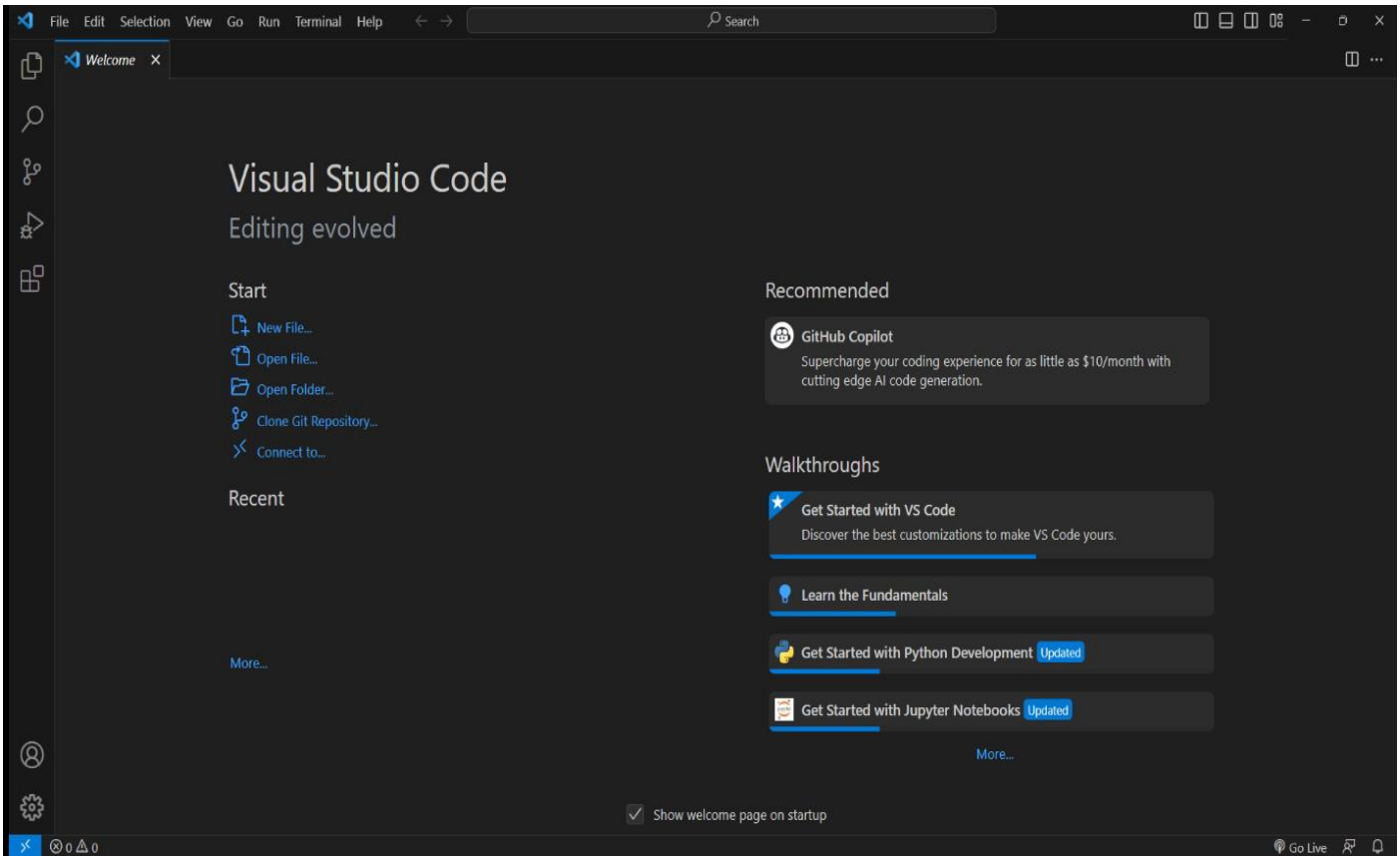
VISUAL CODE STUDIO

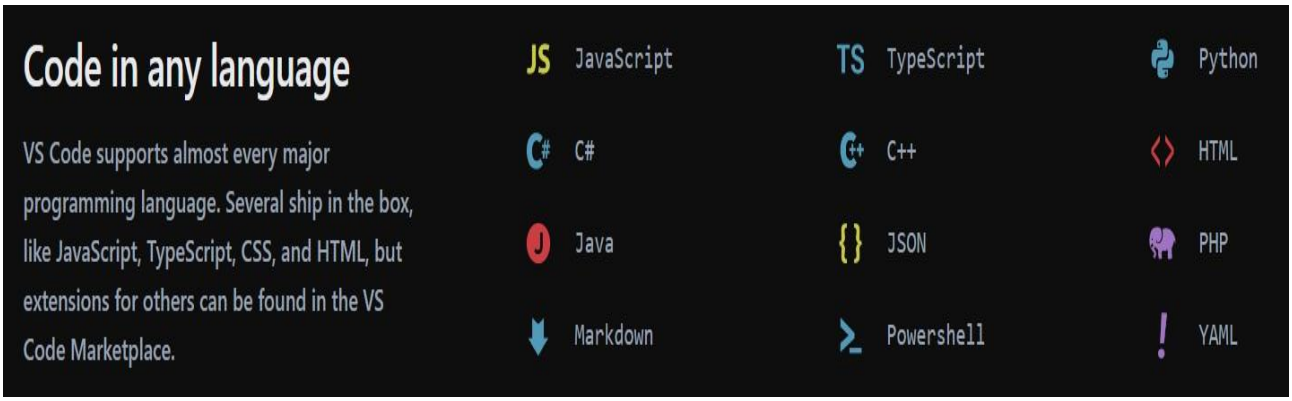
Visual Studio Code is a free, lightweight but powerful source code editor that runs on your desktop and on the web and is available for Windows, macOS, Linux, and Raspberry Pi OS. It comes with built-in support for JavaScript, TypeScript, and Node.js and has a rich ecosystem of extensions for other programming languages (such as C++, C#, Java, Python, PHP, and Go), runtimes (such as .NET and Unity), environments (such as Docker and Kubernetes), and clouds (such as Amazon Web Services, Microsoft Azure, and Google Cloud Platform).

Aside from the whole idea of being lightweight and starting quickly, Visual Studio Code has IntelliSense code completion for variables, methods, and imported modules; graphical debugging; linting, multi-cursor editing, parameter hints, and other powerful editing features; snazzy code navigation and refactoring; and built-in source code control including Git support. Much of this was adapted from Visual Studio technology.



Code in any Language

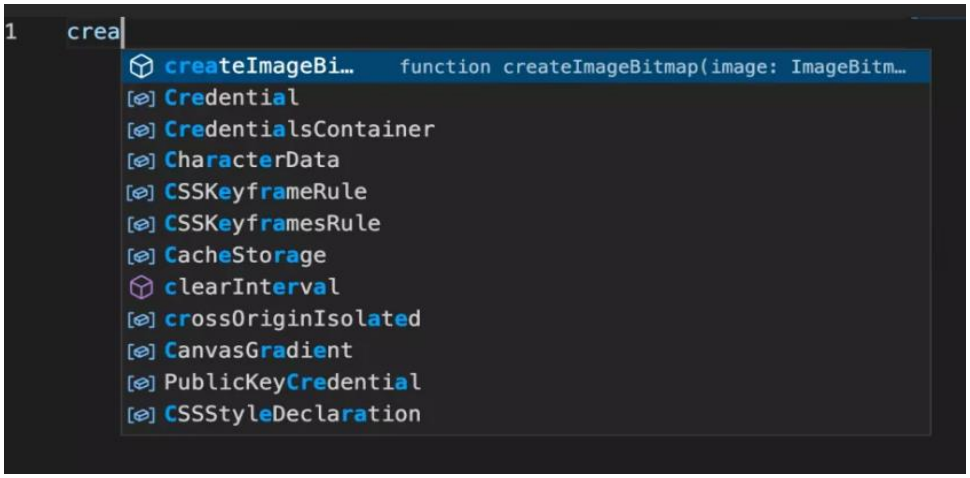
VS Code supports many programming languages and supports a ton of frameworks. This means you can use VS Code to write code in Markdown, HTML, JavaScript, or any other framework you choose



Features of vs code

Visual Studio Code (VS Code) is a highly popular source-code editor developed by Microsoft. It's known for its versatility, extensibility, and efficiency. Here are some of its key features:

1. **Cross-Platform**: VS Code is available on Windows, macOS, and Linux, ensuring a consistent experience across different operating systems.
2. **IntelliSense:** VS Code provides intelligent code completion suggestions, syntax highlighting, and code navigation features, helping developers write code more efficiently.



3. **Built-in Git Integration**: Git commands are seamlessly integrated into VS Code, allowing developers to work with version control directly within the editor.
4. **Extensions**: VS Code has a rich ecosystem of extensions developed by the community, providing additional functionalities such as language support, debugging tools, and

themes. These extensions enhance the editor's capabilities and cater to various programming languages and frameworks.

5. **Debugger**: VS Code includes a built-in debugger that supports debugging for various programming languages. It enables developers to debug their code directly within the editor, inspect variables, set breakpoints, and step through code execution.
6. **Customizable UI**: The user interface of VS Code is highly customizable, allowing users to adjust themes, layout, and keyboard shortcuts according to their preferences.
7. **Integrated Terminal:** VS Code includes an integrated terminal, eliminating the need for developers to switch between the editor and external terminals for executing commands.
8. **Task Automation**: VS Code supports task automation through its built-in task runner and integration with build systems such as Grunt and Gulp. This enables developers to automate repetitive tasks and streamline their workflow.
9. **Extension Marketplace**: VS Code features a centralized marketplace where users can discover and install extensions directly from the editor. The marketplace offers a wide range of extensions categorized by functionality, popularity, and relevance.
10. **Live Share**: VS Code's Live Share extension enables real-time collaboration between developers, allowing them to edit and debug code together, regardless of their physical location.

Visual Studio Code extensions

A quick search of the Visual Studio Code Marketplace yields roughly 38,000 results, supporting hundreds of programming languages. You can manage extensions from the Marketplace, from the Extensions sidebar in VS Code, and from the VS Code Command Palette.

The top extension, for Python, had nearly 60 million installs when I checked in July 2022. In addition to support for Python 3.7+ coding, debugging, and refactoring, the Python extension will automatically install the Pylance (IntelliSense) and Jupyter (notebook) extensions.

Here's a step-by-step guide on how to use GitHub with Visual Studio Code (VS Code), including setting up a repository, making changes, committing, and pushing those changes to GitHub:

Step 1: Install Visual Studio Code and Git

Download and install Visual Studio Code from the official website.

Install Git from the official website, if you haven't already. During installation, ensure you select the option to add Git to your PATH.

Step 2: Configure Git

Open a terminal (Command Prompt on Windows, Terminal on macOS/Linux) and set up your Git username and email address:

Copy code

git config --global user.name "Your Name"

git config --global user.email your.email@example.com

Step 3: Create a GitHub Repository

Sign in to your GitHub account and click on the "+" icon in the top-right corner, then select "New repository".

Enter a name for your repository, choose visibility (public or private), and add a description if desired. Optionally, initialize the repository with a README file, .gitignore, or license.

Click "Create repository".

Step 4: Clone the Repository in VS Code

Open Visual Studio Code.

Open the Command Palette (Ctrl+Shift+P or Cmd+Shift+P on macOS) and type "Git: Clone" and press Enter.

Enter the URL of the repository you created on GitHub and choose a local directory for cloning.

Once cloned, VS Code will open the repository in a new window.

Step 5: Make Changes

Open a file in the repository that you want to edit.

Make changes to the file as needed using the VS Code editor.

Step 6: Stage and Commit Changes

Open the Source Control view in VS Code by clicking on the source control icon in the activity bar on the left side.

You'll see a list of changed files. Click on the "+" icon next to each file you want to stage for commit. Alternatively, you can stage all changes by clicking the "+" icon at the top.

Enter a commit message in the text box at the top of the Source Control view.

Click the checkmark icon to commit your changes.

Step 7: Push Changes to GitHub

After committing your changes, click on the three dots (...) at the top-right of the Source Control view and select "Push".

If prompted, sign in to your GitHub account.

Once authenticated, VS Code will push your changes to the remote repository on GitHub.

Step 8: Verify Changes on GitHub

Go to your repository on GitHub in your web browser.

You should see the changes you made reflected in the files in the repository.

Shortcut keys

Visual Studio Code (VS Code) offers a plethora of keyboard shortcuts to streamline your coding workflow and increase productivity. Here's a list of some essential shortcut keys categorized by functionality:

File Management:

Ctrl + N (Windows/Linux) / Cmd + N (Mac): New File

Ctrl + O (Windows/Linux) / Cmd + O (Mac): Open File

Ctrl + S (Windows/Linux) / Cmd + S (Mac): Save File

Ctrl + Shift + S (Windows/Linux) / Cmd + Shift + S (Mac): Save As

Ctrl + W (Windows/Linux) / Cmd + W (Mac): Close File

Ctrl + Shift + T (Windows/Linux) / Cmd + Shift + T (Mac): Reopen Closed File

Editing:

Ctrl + X (Windows/Linux) / Cmd + X (Mac): Cut

Ctrl + C (Windows/Linux) / Cmd + C (Mac): Copy

Ctrl + V (Windows/Linux) / Cmd + V (Mac): Paste

Ctrl + Z (Windows/Linux) / Cmd + Z (Mac): Undo

Ctrl + Y (Windows/Linux) / Cmd + Y (Mac): Redo

Ctrl + D (Windows/Linux) / Cmd + D (Mac): Add Next Occurrence

Ctrl + Shift + L (Windows/Linux) / Cmd + Shift + L (Mac): Select All Occurrences

Navigation:

Ctrl + P (Windows/Linux) / Cmd + P (Mac): Quick Open

Ctrl + Tab (Windows/Linux) / Cmd + Tab (Mac): Switch Editor

Ctrl + G (Windows/Linux) / Cmd + G (Mac): Go to Line

Ctrl + Shift + E (Windows/Linux) / Cmd + Shift + E (Mac): Show Explorer

Ctrl + Shift + F (Windows/Linux) / Cmd + Shift + F (Mac): Find in Files

Code Manipulation:

Alt + ↑/↓ (Windows/Linux) / Option + ↑/↓ (Mac): Move Line Up/Down

Alt + Shift + ↑/↓ (Windows/Linux) / Option + Shift + ↑/↓ (Mac): Copy Line Up/Down

Alt + Shift + F (Windows/Linux) / Option + Shift + F (Mac): Format Document

Terminal:

Ctrl + (Windows/Linux) / Cmd + (Mac): Toggle Terminal

Ctrl + Shift + (Windows/Linux) / Cmd + Shift + (Mac): Create New Terminal

Debugging:

F5: Start Debugging

F9: Toggle Breakpoint

F10: Step Over

F11: Step Into

Shift + F11: Step Out

Source Control:

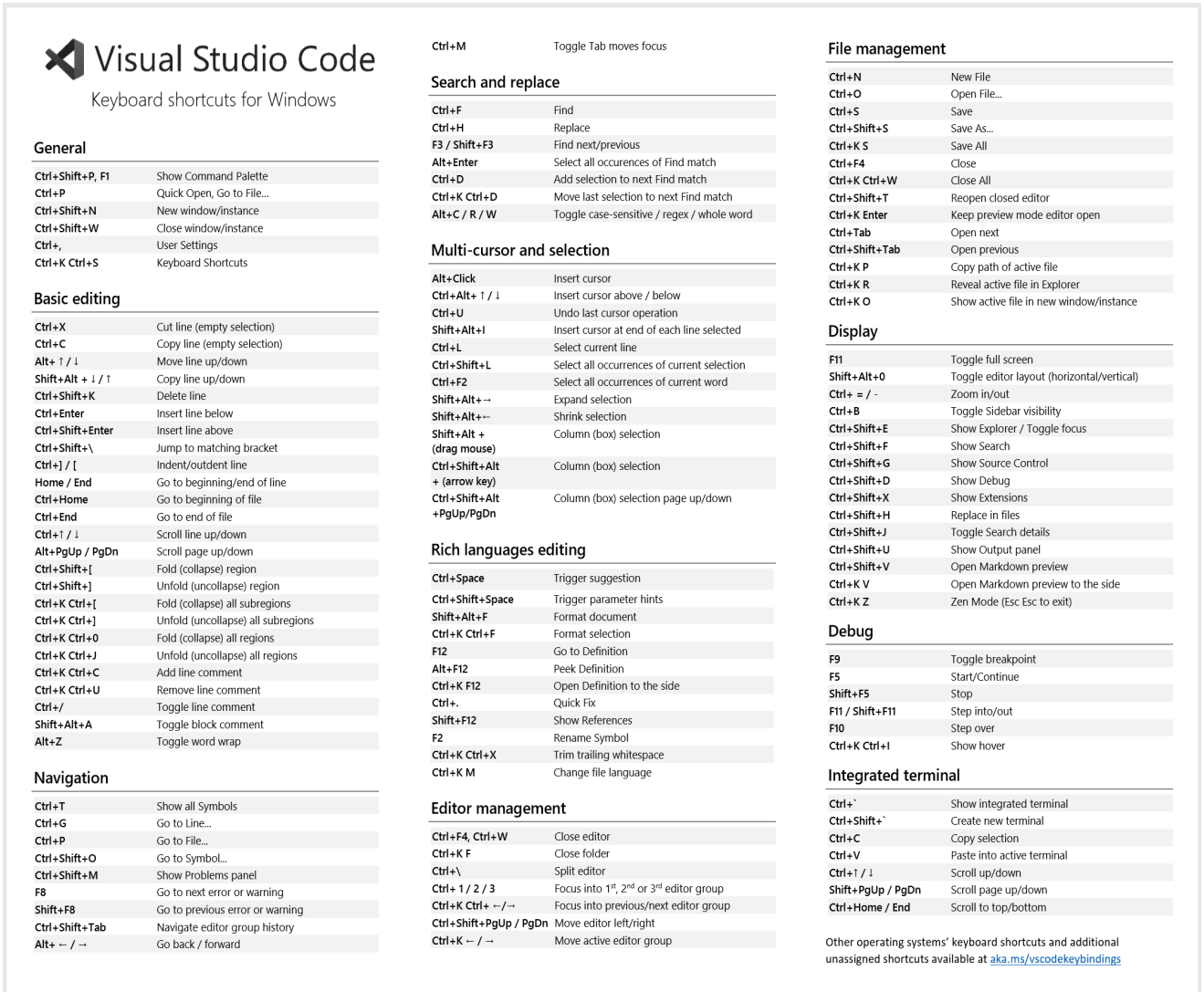
Ctrl + Shift + G (Windows/Linux) / Cmd + Shift + G (Mac): Toggle Source Control

Ctrl + Enter (Windows/Linux) / Cmd + Enter (Mac): Commit

Extensions:

Ctrl + Shift + X (Windows/Linux) / Cmd + Shift + X (Mac): Extensions View

Ctrl + Shift + P (Windows/Linux) / Cmd + Shift + P (Mac): Show All Commands



Visual Studio Code

Keyboard shortcuts for Windows

General

Ctrl+Shift+P, F1	Show Command Palette
Ctrl+P	Quick Open, Go to File...
Ctrl+Shift+N	New window/instance
Ctrl+Shift+W	Close window/instance
Ctrl+,	User Settings
Ctrl+K Ctrl+S	Keyboard Shortcuts

Basic editing

Ctrl+X	Cut line (empty selection)
Ctrl+C	Copy line (empty selection)
Alt+ ↑ / ↓	Move line up/down
Shift+Alt + ↓ / ↑	Copy line up/down
Ctrl+Shift+K	Delete line
Ctrl+Enter	Insert line below
Ctrl+Shift+Enter	Insert line above
Ctrl+Shift+\	Jump to matching bracket
Ctrl+] / [	Indent/outdent line
Home / End	Go to beginning/end of line
Ctrl+Home	Go to beginning of file
Ctrl+End	Go to end of file
Ctrl+↑ / ↓	Scroll line up/down
Alt+PgUp / PgDn	Scroll page up/down
Ctrl+Shift+[	Fold (collapse) region
Ctrl+Shift+]	Unfold (uncollapse) region
Ctrl+K Ctrl+[	Fold (collapse) all subregions
Ctrl+K Ctrl+]	Unfold (uncollapse) all subregions
Ctrl+K Ctrl+0	Fold (collapse) all regions
Ctrl+K Ctrl+J	Unfold (uncollapse) all regions
Ctrl+K Ctrl+C	Add line comment
Ctrl+K Ctrl+U	Remove line comment
Ctrl+/	Toggle line comment
Shift+Alt+A	Toggle block comment
Alt+Z	Toggle word wrap

Navigation

Ctrl+T	Show all Symbols
Ctrl+G	Go to Line...
Ctrl+P	Go to File...
Ctrl+Shift+O	Go to Symbol...
Ctrl+Shift+M	Show Problems panel
F8	Go to next error or warning
Shift+F8	Go to previous error or warning
Ctrl+Shift+Tab	Navigate editor group history
Alt+ ← / →	Go back / forward
Ctrl+M	Toggle Tab moves focus

Search and replace

Ctrl+F	Find
Ctrl+H	Replace
F3 / Shift+F3	Find next/previous
Alt+Enter	Select all occurences of Find match
Ctrl+D	Add selection to next Find match
Ctrl+K Ctrl+D	Move last selection to next Find match
Alt+C / R / W	Toggle case-sensitive / regex / whole word

Multi-cursor and selection

Alt+Click	Insert cursor
Ctrl+Alt+ ↑ / ↓	Insert cursor above / below
Ctrl+U	Undo last cursor operation
Shift+Alt+I	Insert cursor at end of each line selected
Ctrl+L	Select current line
Ctrl+Shift+L	Select all occurrences of current selection
Ctrl+F2	Select all occurrences of current word
Shift+Alt+→	Expand selection
Shift+Alt+←	Shrink selection
Shift+Alt + (drag mouse)	Column (box) selection
Ctrl+Shift+Alt + (arrow key)	Column (box) selection
Ctrl+Shift+Alt +PgUp/PgDn	Column (box) selection page up/down

Rich languages editing

Ctrl+Space	Trigger suggestion
Ctrl+Shift+Space	Trigger parameter hints
Shift+Alt+F	Format document
Ctrl+K Ctrl+F	Format selection
F12	Go to Definition
Alt+F12	Peek Definition
Ctrl+K F12	Open Definition to the side
Ctrl+.	Quick Fix
Shift+F12	Show References
F2	Rename Symbol
Ctrl+K Ctrl+X	Trim trailing whitespace
Ctrl+K M	Change file language

Editor management

Ctrl+F4, Ctrl+W	Close editor
Ctrl+K F	Close folder
Ctrl+\	Split editor
Ctrl+ 1 / 2 / 3	Focus into 1st, 2nd or 3rd editor group
Ctrl+K Ctrl+ ←/→	Focus into previous/next editor group
Ctrl+Shift+PgUp / PgDn	Move editor left/right
Ctrl+K ← / →	Move active editor group

File management

Ctrl+N	New File
Ctrl+O	Open File...
Ctrl+S	Save
Ctrl+Shift+S	Save As...
Ctrl+K S	Save All
Ctrl+F4	Close
Ctrl+K Ctrl+W	Close All
Ctrl+Shift+T	Reopen closed editor
Ctrl+K Enter	Keep preview mode editor open
Ctrl+Tab	Open next
Ctrl+Shift+Tab	Open previous
Ctrl+K P	Copy path of active file
Ctrl+K R	Reveal active file in Explorer
Ctrl+K O	Show active file in new window/instance

Display

F11	Toggle full screen
Shift+Alt+0	Toggle editor layout (horizontal/vertical)
Ctrl+ = / -	Zoom in/out
Ctrl+B	Toggle Sidebar visibility
Ctrl+Shift+E	Show Explorer / Toggle focus
Ctrl+Shift+F	Show Search
Ctrl+Shift+G	Show Source Control
Ctrl+Shift+D	Show Debug
Ctrl+Shift+X	Show Extensions
Ctrl+Shift+H	Replace in files
Ctrl+Shift+J	Toggle Search details
Ctrl+Shift+U	Show Output panel
Ctrl+Shift+V	Open Markdown preview
Ctrl+K V	Open Markdown preview to the side
Ctrl+K Z	Zen Mode (Esc Esc to exit)

Debug

F9	Toggle breakpoint
F5	Start/Continue
Shift+F5	Stop
F11 / Shift+F11	Step into/out
F10	Step over
Ctrl+K Ctrl+I	Show hover

Integrated terminal

Ctrl+`	Show integrated terminal
Ctrl+Shift+`	Create new terminal
Ctrl+C	Copy selection
Ctrl+V	Paste into active terminal
Ctrl+↑ / ↓	Scroll up/down
Shift+PgUp / PgDn	Scroll page up/down
Ctrl+Home / End	Scroll to top/bottom

Other operating systems' keyboard shortcuts and additional unassigned shortcuts available at aka.ms/vscodekeybindings

Breakpoints

A breakpoint in Visual Studio Code (VS Code) is a marker that developers can place in their code to pause the execution of a program at a specific line or condition during debugging. When the execution reaches a breakpoint, the program halts, allowing developers to inspect the current state of variables, evaluate expressions, and analyze the program's behaviour.

Breakpoints are essential tools for pinpointing issues and gaining insights into code behaviour during the debugging process in Visual Studio Code. They empower developers to diagnose problems effectively and iteratively refine their code for optimal performance and functionality.

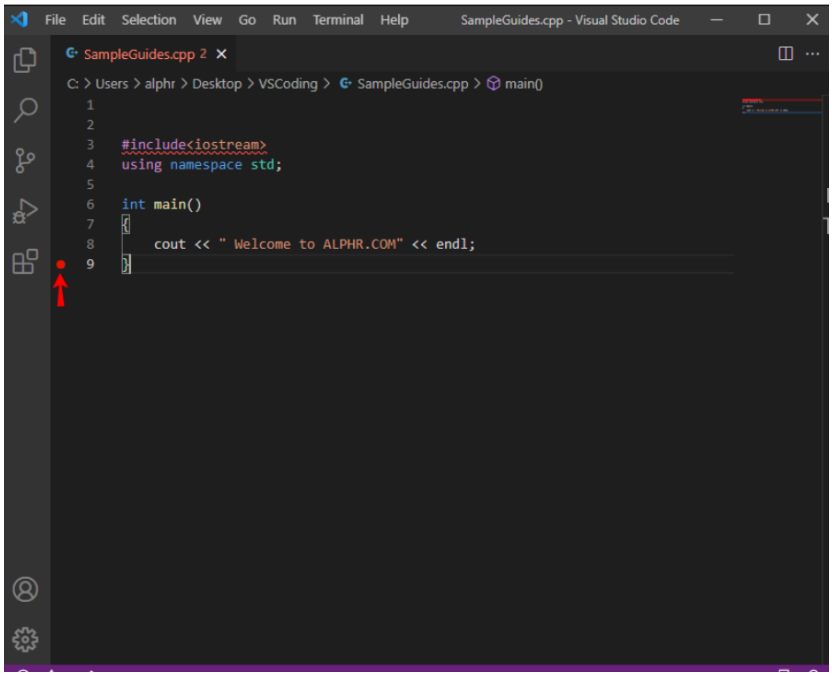
How to Use Breakpoints in VS Code

To set a breakpoint in your source code, take the following steps:

1. Click the left margin or strike the F9 key next to the line you wish to stop.

2. Run the code or press F5 (“Continue”).

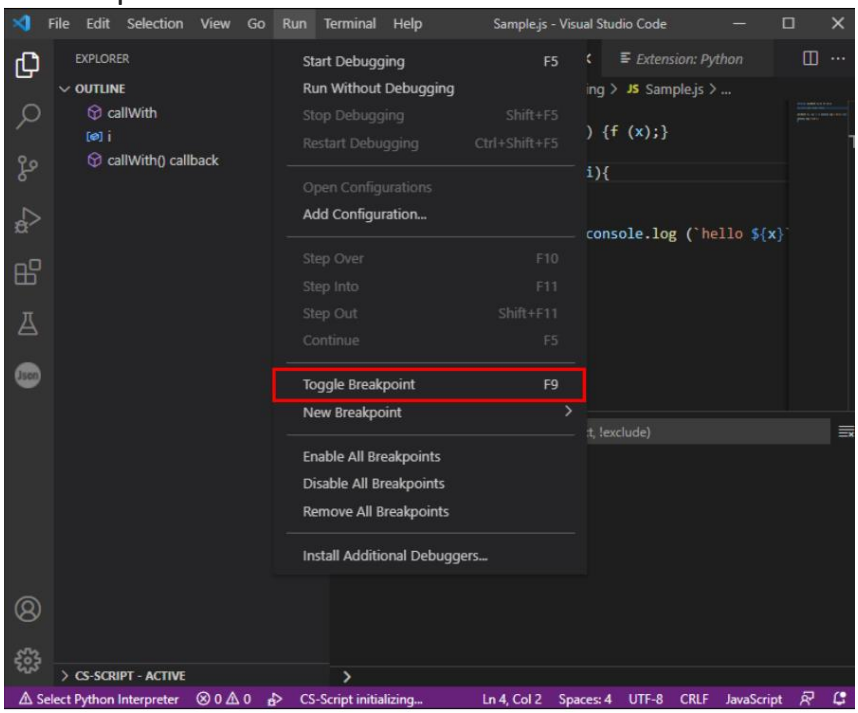
3. Your code will now pause before the marked execution. The breakpoint will appear as a red dot inside your left margin.



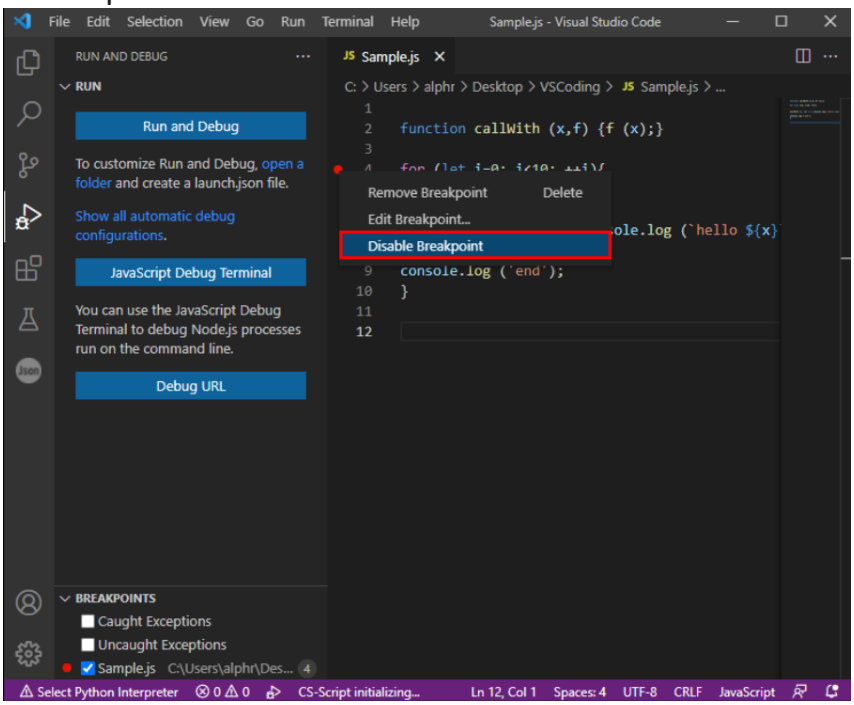
By default, current execution code lines and breakpoints are automatically highlighted for most programming languages, including C#.

Here are a few more notable breakpoints commands:

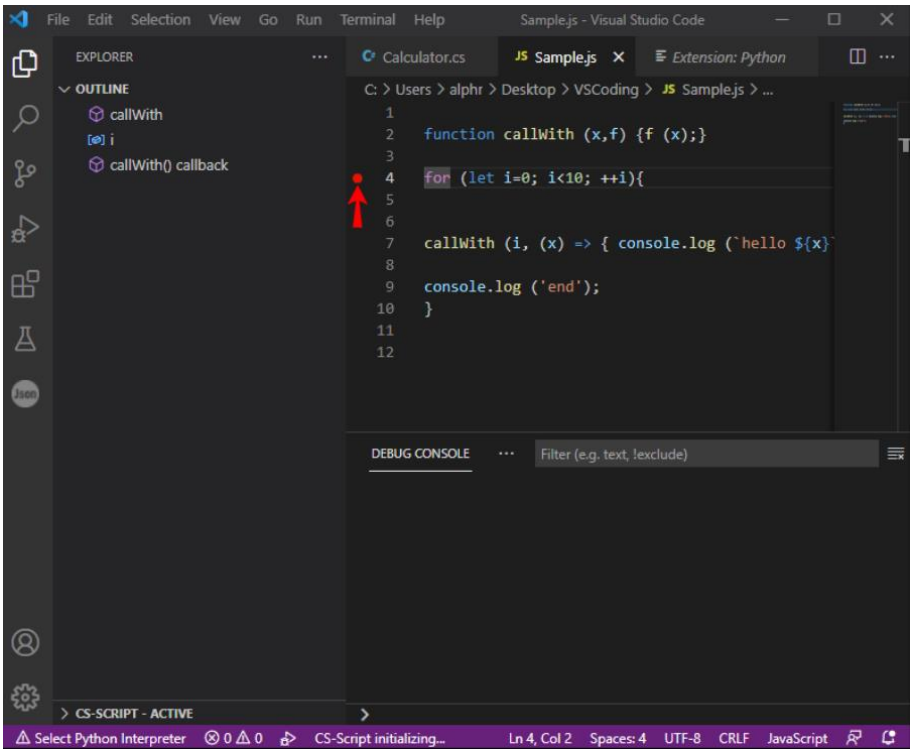
- “Toggle Breakpoint” – Among other things, this command lets you reinsert or delete a breakpoint.



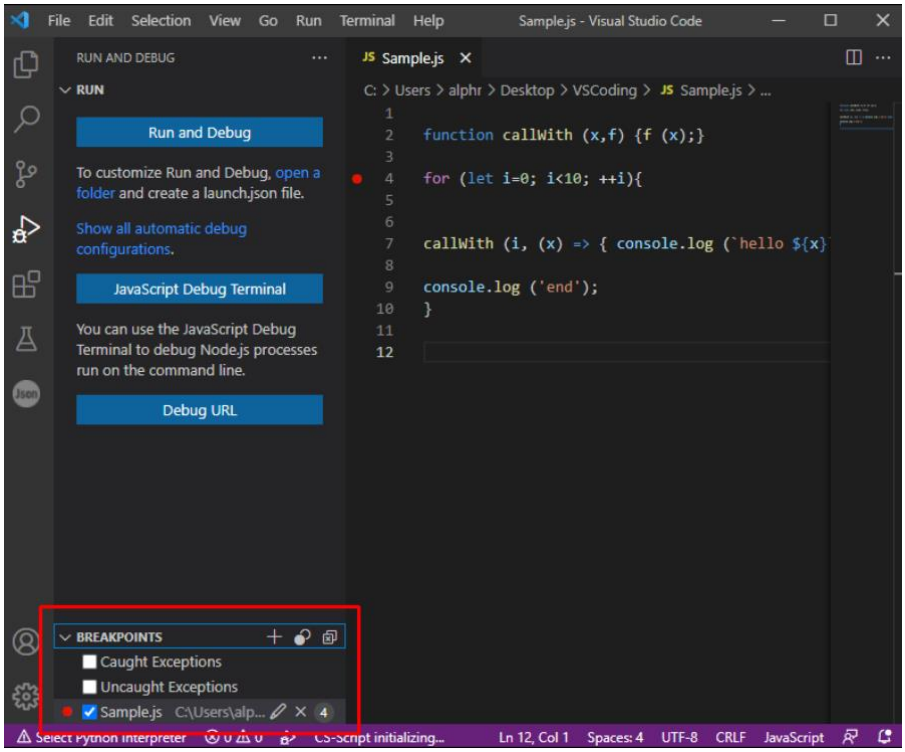
- “Disable Breakpoint” – Disable your breakpoint without deleting it. Such breakpoints are shown as hollow dots in your left margins or your “Breakpoints” window.



- “Enable Breakpoint – This command appears once you hover over a disabled breakpoint and lets you re-activate it.



- “Settings” – The “Settings” section contains numerous commands that let you add, edit, and export your breakpoints. The menu appears once you hover over the breakpoint and press “Settings.”



- “Reapply All Breakpoints” – Return all your breakpoints to the original location. This function is handy if the debug environment misplaces breakpoints within source code that haven’t been executed yet.