



# VS Code Shortcuts CheatSheet

Created by JS Mastery

Visit [jsmastery.pro](https://jsmastery.pro) for more



# 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

# General

**Ctrl+,**

User Settings

**Ctrl+K Ctrl+S**

Keyboard Shortcuts

# Brought to you by JSM

This guide will provide you with useful information and actionable steps, but if you truly want to dominate the competition and secure a high-paying job as a full-stack software developer, [jsmastery.pro](https://jsmastery.pro) is the answer.

Read until the end for more information and **special discounts!** 



# 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

# Basic editing

**Ctrl+Shift+K**

Delete line

**Ctrl+Enter**

Insert line below

**Ctrl+Shift+Enter**

Insert line above

**Ctrl+Shift+ \**

Jump to matching bracket

# Basic editing

**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

# Basic editing

**Ctrl+↑ / ↓**

Scroll line up/down

**Alt+PgUp / PgDn**

Scroll page up/down

**Ctrl+Shift+[**

Fold (collapse) region

**Ctrl+Shift+]**

Unfold (uncollapse) region

# Basic editing

**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

# Basic editing

**Ctrl+K Ctrl+C**

Add line comment

**Ctrl+K Ctrl+U**

Remove line comment

**Ctrl+/**

Toggle line 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...

# Navigation

**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

# Search and replace

**Ctrl+F**

Find

**Ctrl+H**

Replace

**F3 / Shift+F3**

Find next/previous

**Alt+Enter**

Select all occurrences of Find match

# Search and replace

**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 & 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

# Multi-cursor & selection

**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

# Multi-cursor & 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, Ctrl+I**

Trigger suggestion

**Ctrl+Shift+Space**

Trigger parameter hints

**Shift+Alt+F**

Format document

**Ctrl+K Ctrl+F**

Format selection

# Rich languages editing

**F12**

Go to Definition

**Alt+F12**

Peek Definition

**Ctrl+K F12**

Open Definition to the side

**Ctrl+.**

Quick Fix

# Rich languages editing

**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

# Editor management

**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...

# File management

**Ctrl+K S**

Save All

**Ctrl+F4**

Close

**Ctrl+K Ctrl+W**

Close All

**Ctrl+Shift+T**

Reopen closed editor

# File management

**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

# File management

**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

# Display

**Ctrl+Shift+E**

Show Explorer / Toggle focus

**Ctrl+Shift+F**

Show Search

**Ctrl+Shift+G**

Show Source Control

**Ctrl+Shift+D**

Show Debug

# Display

**Ctrl+Shift+X**

Show Extensions

**Ctrl+Shift+H**

Replace in files

**Ctrl+Shift+J**

Toggle Search details

**Ctrl+Shift+U**

Show Output panel

# Display

**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

# Debug

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

# Integrated terminal

**Ctrl+↑ / ↓**

Scroll up/down

**Shift+PgUp / PgDn**

Scroll page up/down

**Ctrl+Home / End**

Scroll to top/bottom

# Other

**Ctrl+M**

Toggle Tab moves focus

**Shift+Alt+A**

Toggle block comment

---

That's all for the VS Code Shortcuts, Now you can find out some fantastic VS Code Extensions

So, let's dive right into it →

BONUS

# Fantastic vs Code Extensions



# **Bracket Pair Colorizer 2**

The VS extension Bracket Pair Colorizer matches corresponding brackets in your code with the same color. This is a great help when you're working with things like nested components, objects, or functions that all have brackets or parentheses.

With this simple extension, it's much easier to find matching pairs and understand your code. The biggest advantage of this extension is improved navigation and accessibility. It also makes it easier for others to read and understand your code.

# **Better Comments**

Better Comments is an extension used for writing human-friendly comments in your code, which is helpful for you and anyone who is reading source-

code (especially for teams). Descriptive, human-friendly comments save so much time for everyone involved.

With this VS extension, you can use the following characters after a double forward slash `//` to add easier commenting:

`*` – for highlighted text

`!` – for errors and warnings

`?` – for queries and questions

`//` – for strikethrough

`TODO` – for to-dos

## Snippets

Snippets are an excellent add-on for productivity.

Snippets is a collection of various extensions for the most commonly used programming languages.

The ES7 React/Redux/GraphQL..... snippet is a popular extension, for example, that allows you to use and create shorthands for things you do over and over again.

React snippets → [ES7 React/Redux/GraphQL/React-Native snippets](#)

## Icons

Icons allows you to create descriptive icons to help differentiate between files and folders. This makes your code more visual, so it's easier to work as teams, return to code after some time, or even just make the experience more fun. For example, you could change the color of a default folder icon using the command palette.

There are a few different Icons extensions : →

- Material Icon Theme
- vscod-icons
- Easy icon theme
- Nomo Dark Icon Theme

## Themes

- One dark pro
- Shades of Purple
- Palenight
- Monokai Pro
- Material Theme

## GitLens

The GitLens extension combines the capabilities of Git with VSCode that allows you to visualize code authorship via Git. GitLens is great for understanding code better, so you can learn who, why, and when

code was changed. It also allows you to explore the history and evolution of a codebase.

It also has many other features to use such as:

- Revision navigation through file history
- A current line blame annotation at the end of the line showing the commit
- A status bar blame annotation for the commit & author who last modified the current line

## Auto-Close Tag

Using this tool is non-negotiable, every developer needs to use this in their vscode toolbox, it will not only catch possible 'stupid' errors but it is very handy, you really don't have to spend your precious energy thinking about the small stuff like making sure you don't forget to close your tag.

# Auto Rename Tag

While VSCode inherently highlights matching tags and immediately adds closing tags whenever you type an opening tag, Auto Rename Tag automatically renames tags that you change.

The extension works for HTML, XML, PHP, and JavaScript, and removes the need to change your tag names twice.

**Note** : a few people in the comments have mentioned that Auto Rename Tag is quite buggy, so proceed with care.

# Path Intellisense

With over a 4.2 Million downloads. Path Intellisense can help with auto-completion for the imports or path

after calling a component, and trust me, it helped me big time.

Why do we have to memorize the file's entire path, it just a waste of resources, when this tool can just do it for you?

## **NPM Intellisense**

It has almost the same functionality as path Intellisense, but they create this one for npm packages.

NPM without a doubt can be a pain not just on how it eats up our memory, but there are just so many in there that's a bit difficult to track.

If you have been working or planning on taking Nodejs, then you'd see how helpful this tool is, importing several packages or modules, and there are like

thousands of them, you really can't just memorize their paths, right?

## Import Cost

I've recently just bumped into this extension, it simply calculates the size of your imports and I just find it useful to be able to decide whether to just import the entire package or simply just import the specific modules.

It helps, and you'll appreciate it when you start working on extensive projects.

## Turbo Console Log

This extension make debugging much easier by automating the operation of writing meaningful log message.

# Todo+

Todo+ helps you manage todo lists with ease, Its powerful, portable, easy to use and customizable..

You can do many amazing things with it such as :

- Custom colors
- Replace default symbols with custom symbols
- Custom special tags
- View todos from a custom activity bar section
- Timer, Timekeeping, Statistics, etc.

# SVG

Its a powerful SVG language support extension.

Almost all the features you need to handle SVG files such as:

- SVG Live Preview and Export PNG
- Minify SVG with SVGO

- SVG Full Auto Completion.
- Fast Color Picker
- Document Symbol tree.

## Code Time

Code Time is an open source plugin for automatic programming metrics and time tracking in VS Code.

Code Time has feature called Flow Mode. It makes it easy to eliminate distractions, mute notifications, and stay focused when you are in flow.

Code Time also shows you a personalized dashboard to summarizes your coding data – such as your code time by project, lines of code, and keystrokes—today, yesterday, last week, and over the last 90 days.

You can also see more advanced data visualizations by clicking more data at software.com in the sidebar.

# Code Spell Checker

Code Spell Checker helps you catch common spelling errors. We often have spelling mistakes in our code when declaring a variable, writing a comment, etc.

Code Spell Checker is a spell checker that works well with camelCase code. Many of us write code in camelCase, so its really helpful.

# REST Client

REST client is a really helpful extensions for backend developers, as it allows you to send HTTP request and view the response directly in Visual Studio Code.

Before this, we had to use softwares like Postman, Insomnia, etc. It has many features, you can learn more about it [here](#).

# Visual Studio IntelliCode

The Visual Studio IntelliCode extension provides AI-assisted development features for Python, TypeScript/JavaScript, and Java developers in Visual Studio Code, with insights based on understanding your code context combined with machine learning.

This extension provides AI-assisted IntelliSense by showing recommended completion items for your code context at the top of the completions list.

# JS Mastery Pro

Looking to advance your career and understand the concepts & technologies that top-shelf employers are looking for?

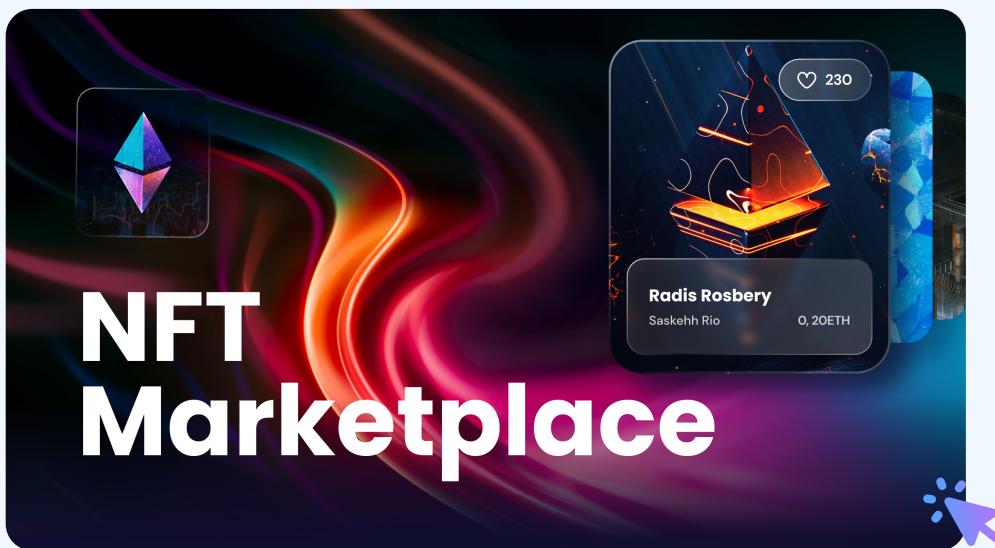
JS Mastery Pro offers two courses that will help you master libraries, tools, and technologies such as React.js, Next.js, Material UI, Solidity, Redux, and many more.

If your goal is to earn a high income while working on projects you love, JS Mastery Pro can help you develop your skills to become a top candidate for lucrative employment and freelance positions.





**Become a React.js master as you create a stunning Netflix clone streaming app to showcase movies, actor bios, and more with advanced AI voice functionality.**



**Leverage Web 3.0 and blockchain technology to build a comprehensive NFT platform where users can discover, create, purchase, & sell non-fungible tokens.**

Plus, if you really want to make a splash and add multiple group projects to your portfolio, join the JSM Masterclass Experience to set yourself above the rest and impress hiring managers.



**Collaborate with other developers on exciting monthly group projects, have your code reviewed by industry experts, and participate in mock interviews and live Q&As. With two masterclass options available, this is the best way to truly launch your programming career and secure the job of your dreams!**

Visit [jsmastery.pro](https://jsmastery.pro) today to get started!

