

Sidebar

Programming · VS Code Keybindings Cheat Sheet

How to change VS Code's keyboard shortcuts in `keybindings.json`, plus the default Windows/Linux shortcuts worth knowing. On macOS, `Ctrl` is usually `Cmd`. Sheet 7 in the VS Code series.

FINDING & CHANGING SHORTCUTS

- `Ctrl+K Ctrl+S` Open the Keyboard Shortcuts editor (search by command name or by pressing the keys)
- `Open Keyboard Shortcuts (JSON)` Button in that editor's title bar: opens your own `keybindings.json`
- `Preferences: Open Default Keyboard Shortcuts (JSON)` Command Palette: read-only list of every default binding
- `Copy Command ID` Right-click a row in the editor: gives the command ID to use in `keybindings.json`
- `Help > Keyboard Shortcuts Reference` Opens the official printable PDF of the defaults
- `Developer: Toggle Keyboard Shortcuts Troubleshooting` Logs each pressed shortcut and the command it ran, in the Output panel

KEYBINDINGS.JSON SYNTAX

- `{ "key": "ctrl+f", "command": "action" }` A keybinding object
- `"command": "-jumpToNextSnippetPlaceHolder"` A command name
- `"command": ""` An empty command unbound to any action
- `"key": "ctrl+k ctrl+w"` A space between keys
- `args` Extra input for a command, e.g. `{ "text": "workbench.action.terminal.selectAll" }`
- `// comments` `keybindings.json` accepts comments

KEY NAMES & WHEN CLAUSES

- `Modifiers` `ctrl`, `shift`, `alt`, and `win` on Windows (`cmd` on macOS, `meta` on Linux)
- `Named keys` `f1` – `f19`, `left`, `pageup`, `tab`, `enter`, `escape`, `space`, `backspace`, `delete`, `numpad0` – `numpad9`
- `editorTextFocus` True when the text editor has focus
- `terminalFocus` True when the integrated terminal has focus
- `!editorReadonly` True when the editor is not read-only
- `editorLangId == python` True only in one language
- `resourceExtname == .md` True for files with that extension
- `&&` `||` `!` `~` AND, OR, NOT, and regular-expression match, for combining conditions

COMMAND PALETTE & NAVIGATION

- `Ctrl+Shift+P / F1` Command Palette: normal
- `Ctrl+P` Quick Open a file by name (type) or recent file
- `Ctrl+G` Go to line
- `Ctrl+Shift+O / Ctrl+T` Go to symbol in current file
- `F12 / Alt+F12 / Shift+F12` Go to definition
- `F2` Rename a symbol everywhere
- `Ctrl+.` Quick fix and code actions
- `Alt+Left / Alt+Right` Go back / forward in current file
- `Ctrl+Shift+\`` Jump to the matching brace

EDITING LINES

- `Alt+Up / Alt+Down` Move the line (or selection) up / down
- `Shift+Alt+Up / Shift+Alt+Down` Copy the line up / down
- `Ctrl+Shift+K` Delete the line
- `Ctrl+Enter / Ctrl+Shift+Enter` Insert a line below / above
- `Ctrl+] / Ctrl+[` Indent / outdent the line
- `Ctrl+/` Toggle line comment
- `Shift+Alt+A` Toggle block comment
- `Ctrl+K Ctrl+X` Trim trailing whitespace

SELECTION & MULTIPLE CURSORS

- `Ctrl+L` Select the current line
- `Shift+Alt+Right / Left` Expand / shrink selection to line
- `Ctrl+D` Add the next occurrence of the selection
- `Ctrl+Shift+L` Select every occurrence of the selection
- `Alt+Click` Add a cursor where you click
- `Ctrl+Alt+Up / Ctrl+Alt+Down` Add a cursor to previous / next line
- `Shift+Alt+drag` Column (box) selection
- `Ctrl+U` Undo the last cursor action

CODE HELP, FORMATTING & FOLDING

<code>Ctrl+Space</code>	Trigger suggestions
<code>Ctrl+Shift+Space</code>	Show parameter hints
<code>Shift+Alt+F</code>	Format the whole document
<code>Ctrl+K Ctrl+F</code>	Format the selection
<code>Ctrl+Shift+[/ Ctrl+Shift+]</code>	Fold / unfold the current region
<code>Ctrl+K Ctrl+0 / Ctrl+K Ctrl+J</code>	Fold all / unfold all
<code>Ctrl+K Ctrl+C / Ctrl+K Ctrl+U</code>	Add / remove line comment

FILES & EDITORS

<code>Ctrl+N / Ctrl+O / Ctrl+K Ctrl+O</code>	New file / open file / open folder
<code>Ctrl+S / Ctrl+K S / Ctrl+Shift+S</code>	Save / save all / save as
<code>Ctrl+W</code>	Close the current editor
<code>Ctrl+K Ctrl+W</code>	Close all editors
<code>Ctrl+Shift+T</code>	Reopen the editor you just closed
<code>Ctrl+Tab</code>	Switch between open editors
<code>Ctrl+K Enter</code>	Keep a preview tab open
<code>Ctrl+K P / Ctrl+K R</code>	Copy the file's path / reveal it in the Explorer

RUN & DEBUG

<code>F5 / Ctrl+F5 / Shift+F5</code>	Start debugging / run without debugging / stop
<code>F9</code>	Toggle a breakpoint on the current line
<code>F10 / F11 / Shift+F11</code>	Step over / step into / step out
<code>Ctrl+Shift+Y</code>	Open the Debug Console

A FEW CUSTOM SHORTCUTS TO COPY (MY SUGGESTIONS)

<code>{ "key": "ctrl+alt+s", "command": "workbench.action.files.saveAll" }</code>	A one-step Save All
<code>{ "key": "ctrl+alt+t", "command": "workbench.action.terminal.toggleTerminal" }</code>	A second way to toggle the terminal
<code>{ "key": "ctrl+alt+f", "command": "editor.action.formatDocument", "when": "editorTextFocus" }</code>	Format with an easier chord
<code>{ "key": "ctrl+shift+k", "command": "-editor.action.deleteLines" }</code>	Frees Ctrl+Shift+K if you want it for something else (match the original <code>when</code> if the removal has no effect)

A SHORTCUT DOES NOTHING? FIND OUT WHY

First look it up in the Keyboard Shortcuts editor (`Ctrl+K Ctrl+S`): a second command bound to the same keys, or a when condition that is false right now, is the usual cause. Then run *Developer: Toggle Keyboard Shortcuts Troubleshooting* and press the keys: the Output panel shows which command VS Code actually ran. In the terminal, many keys go to the shell, see the Terminal cheat sheet.

FIND & REPLACE

<code>Ctrl+F / Ctrl+H</code>	Find / replace in the current file
<code>F3 / Shift+F3</code>	Next / previous match
<code>Alt+Enter</code>	Select all matches of the current match
<code>Alt+C / Alt+W / Alt+R</code>	Match case / word / regular expression
<code>Ctrl+Shift+F / Ctrl+Shift+H</code>	Find / replace in all files
<code>editor.find.loop</code>	Wrap around from the end to the beginning
<code>editor.find.seedSearchStringFromSelection</code>	Use the current selection as the seed for the search
<code>editor.find.autoFindInSelection</code>	Find the next match automatically

LAYOUT, VIEWS & PANELS

<code>Ctrl+B / Ctrl+J</code>	Show or hide the side bar
<code>Ctrl+Shift+E / F / G / D / X</code>	Explorer / Search / Run and Debug / Output / Console
<code>Ctrl+' / Ctrl+Shift+M / Ctrl+Shift+V</code>	Source Explorer / Run and Debug Console / Output / Console
<code>Ctrl+\\ / Ctrl+1 / Ctrl+2</code>	Split the editor
<code>Ctrl+K Z / F11</code>	Zen Mode / full screen
<code>Ctrl+K Ctrl+T</code>	Colour theme picker

KEYBOARD-RELATED SETTINGS

<code>keyboard.dispatch</code>	<code>code</code> (default) or <code>keyCode</code> (for character it types (not the command))
<code>keyboard.mapAltGrToCtrlAlt</code>	<code>true</code> : yes, <code>false</code> : no
<code>workbench.commandPalette.preserveInput</code>	<code>true</code> : yes, <code>false</code> : no
<code>editor.multiCursorModifier</code>	Key for creating multiple cursors

CHORDS: PRESSING CTRL+K THEN WAITING

Shortcuts that begin Ctr1+K are two-step chords. After the first press VS Code shows *(Ctrl+K) was pressed. Waiting for second key of chord...* in the status bar. Press Esc to cancel. Some chords need Ctrl held for the second key too, as in Ctr1+K Ctr1+S; others use a plain second key, as in Ctr1+K Z.