

TOPICS Tips and Tricks ▼

Visual Studio Code Tips and Tricks



(<https://github.com/Microsoft/vscode-docs/blob/master/docs/getstarted/tips-and-tricks.md>)

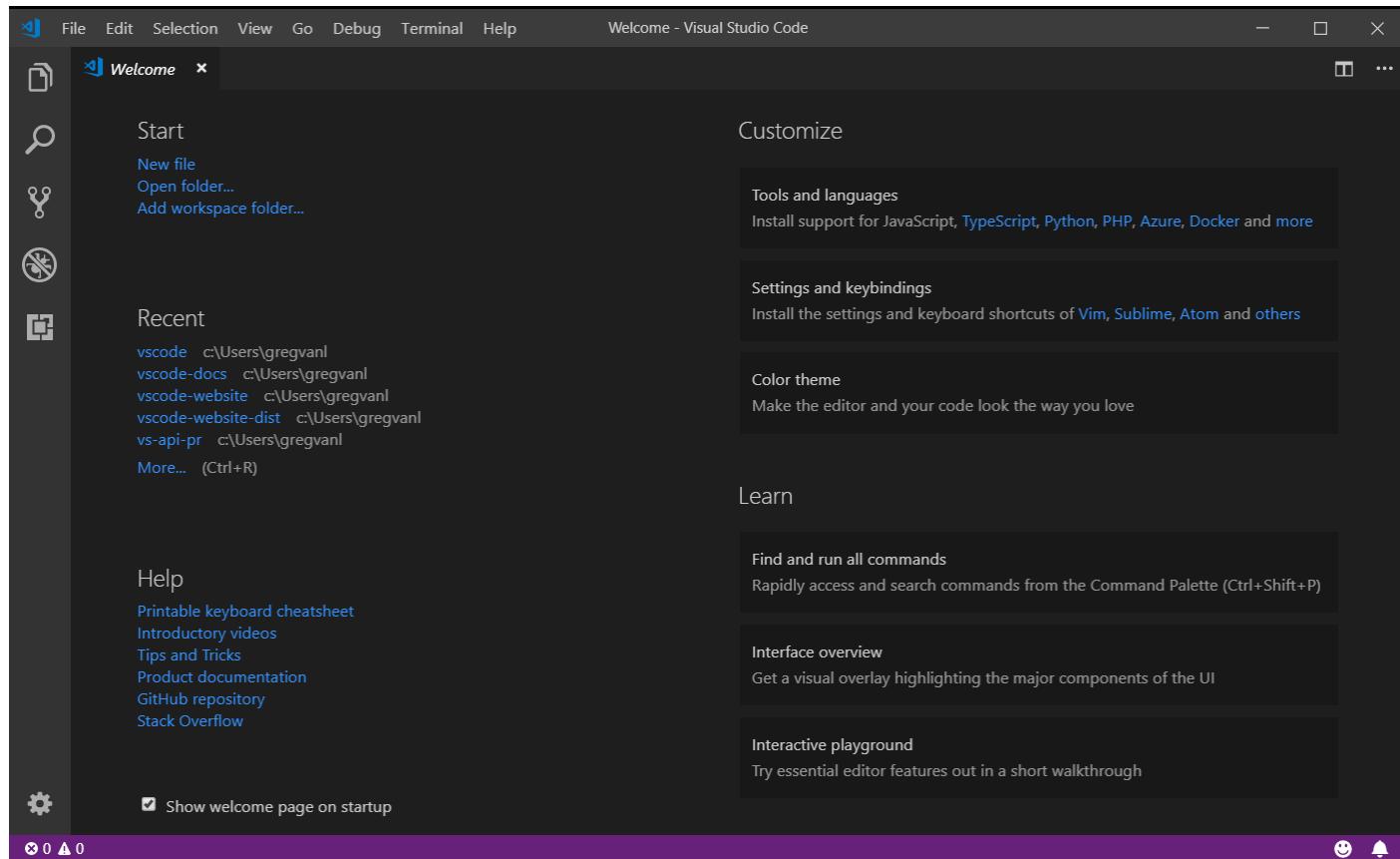
"Tips and Tricks" lets you jump right in and learn how to be productive with Visual Studio Code. You'll become familiar with its powerful editing, code intelligence, and source code control features and learn useful keyboard shortcuts. This topic goes pretty fast and provides a broad overview, so be sure to look at the other in-depth topics in [Getting Started](#) (/docs/getstarted/userinterface) and the [User Guide](#) (/docs/editor/codebasics) to learn more.

If you don't have Visual Studio Code installed, go to the [Download](#) (/download) page. You can find platform specific setup instructions at [Running VS Code on Linux](#) (/docs/setup/linux), [macOS](#) (/docs/setup/mac), and [Windows](#) (/docs/setup/windows).

Basics

Getting started

Open the **Welcome** page to get started with the basics of VS Code. [Help > Welcome](#).



In the bottom right of the **Welcome** page, there is a link to the [Interactive playground](#) where you can interactively try out VS Code's features. [Help > Interactive Playground](#).

Multi-Cursor Editing

Using multiple cursors allows you to edit multiple parts of the document at once, greatly improving your productivity. Try the following actions in the code block below:

1. Box Selection - press any combination of **Ctrl+Shift+Alt+DownArrow**, **Ctrl+Shift+Alt+RightArrow**, **Ctrl+Shift+Alt+UpArrow**, **Ctrl+Shift+Alt+LeftArrow** to select a block of text. You can also press **Shift+Alt** while selecting text with the mouse or drag-select using the middle mouse button.
2. Add a cursor - press **Ctrl+Alt+UpArrow** to add a new cursor above, or **Ctrl+Alt+DownArrow** to add a new cursor below. You can also use your mouse with **Alt+Click** to add a cursor anywhere.
3. Create cursors on all occurrences of a string - select one instance of a string e.g. **background-color** and press **Ctrl+Shift+L**. Now you can replace all instances by simply typing.

That is the tip of the iceberg for multi-cursor editing. Have a look at the selection menu and our handy [keyboard reference guide](#) for additional actions.

```
1  #p1 {background-color: #ff0000;}           /* red in HEX format */
2  #p2 {background-color: hsl(120, 100%, 50%);}  /* green in HSL format */
3  #p3 {background-color: rgba(0, 4, 255, 0.733);} /* blue with alpha channel in RGBA format */
```

CSS Tip: you may have noticed in the example above we also provide color swatches inline for CSS, additionally if you hover over an element such as **#p1** we will show how this is represented in HTML. These swatches also act as color pickers that allow you to easily change a color value. A simple example of some language-specific editor features.

IntelliSense

Visual Studio Code comes with the powerful IntelliSense for JavaScript and TypeScript pre-installed. In the below example, position the text cursor in front of the error underline, right after the dot and press **Ctrl+Space** to invoke IntelliSense. Notice how the suggestion comes from the Request API.

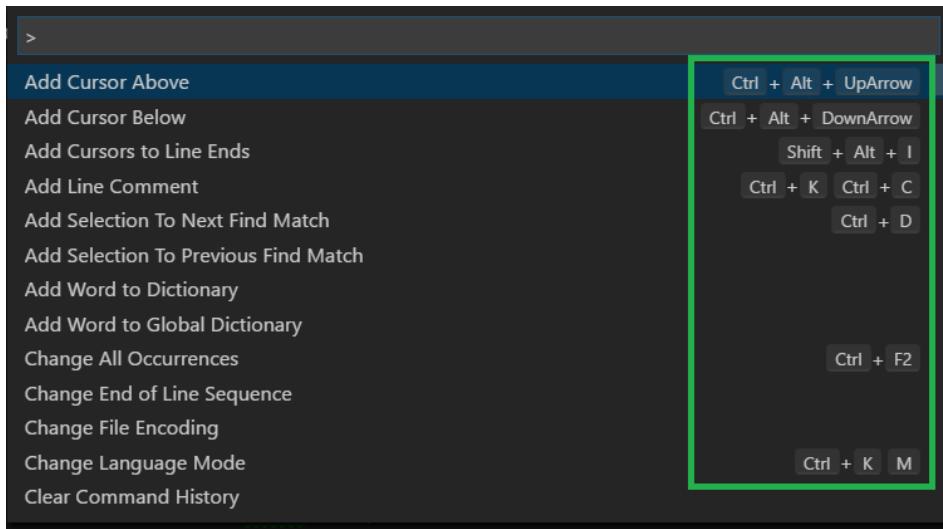
Command Palette

Access all available commands based on your current context.

Keyboard Shortcut: **Ctrl+Shift+P**

Default keyboard shortcuts

All of the commands are in the **Command Palette** with the associated key binding (if it exists). If you forget a keyboard shortcut, use the **Command Palette** to help you out.



Visual Studio Code	
Keyboard shortcuts for macOS	
General	
⌘P, F1	Show Command Palette
⌘P	Quick Open, Go to File...
⌘N	New window-instance
⌘W	Close window-instance
⌘	User Settings
⌘K ⌘S	Keyboard Shortcuts
Basic editing	
⌘X	Cut line (empty selection)
⌘C	Copy line (empty selection)
⌃L / ⌃U	Move line down/up
⌃L / ⌃U	Copy line down/up
⌘K ⌃K	Delete line
⌘Enter / ⌃⌘Enter	Insert line below/above
⌃L / ⌃U	Jump to matching bracket
⌘ / ⌃[Indent/outdent line
Home / End	Go to beginning/end of line
⌃L / ⌃U	Go to beginning/end of file
⌃PgUp / ⌃PgDn	Scroll line up/down
⌘PgUp / ⌃PgDn	Scroll page up/down
⌃[/ ⌃]	Fold/unfold region
⌘K ⌃[/ ⌃K ⌃]	Fold/unfold all subregions
⌘K ⌃[/ ⌃K ⌃J	Fold/unfold all regions
⌘K ⌃C	Add line comment
⌘K ⌃U	Remove line comment
⌃/	Toggle line comment
⌃A	Toggle block comment
⌃Z	Toggle word wrap
Multi-cursor and selection	
⌃ click	Insert cursor
⌃⌃↑	Insert cursor above
⌃⌃↓	Insert cursor below
⌘U	Undo last cursor operation
⌃-d	Insert cursor at end of each line selected
⌃I	Select current line
⌃⌘L	Select all occurrences of current selection
⌃F2	Select all occurrences of current word
⌃⌘- / ↵	Expand / shrink selection
⌃- drag mouse	Column (box) selection
⌃- ⌃- / ↵	Column (box) selection up/down
⌃- ⌃- / ↵	Column (box) selection left/right
⌃- ⌃PgUp	Column (box) selection page up
⌃- ⌃PgDn	Column (box) selection page down
Search and replace	
⌘F	Find
⌃⌘F	Replace
⌃G / ⌃⌘G	Find next/previous
⌃Enter	Select all occurrences of Find match
⌘D	Add selection to next Find match
⌘K ⌃D	Move last selection to next Find match
Rich languages editing	
⌃Space	Trigger suggestion
⌃⌘Space	Trigger parameter hints
⌃F	Format document
⌘K ⌃F	Format selection
F12	Go to Definition
⌃F12	Open Definition to the side
⌘ / ⌃	Quick Fix
⌃F12	Show References
F2	Rename Symbol
⌘K ⌃X	Trim trailing whitespace
⌘K M	Change file language
Navigation	
⌘T	Show all Symbols
⌃G	Go to Line...
⌃P	Go to File...
⌃S	Go to Symbol...
⌃⌘M	Show Problems panel
F8 / ⌃F8	Go to next/previous error or warning
⌃Tab	Navigate editor group history
⌃- / ⌃-	Go back/forward
⌃M	Toggle Tab moves focus
Editor management	
⌘W	Close editor
⌘K F	Close folder
⌘K ⌃	Split editor
⌃F1 / ⌃F2 / ⌃F3	Focus into 1 st , 2 nd , 3 rd editor group
⌘K ⌃← / ⌃K ⌃→	Focus into previous/next editor group
⌘K ⌃← / ⌃K ⌃→	Move editor left/right
⌘K ← / ⌃K ←	Move active editor group
File management	
⌘N	New File
⌘O	Open File...
⌘S	Save
⌃⌘S	Save As...
⌃⌘S	Save All
⌘W	Close
⌘K ⌃W	Close All
Display	
⌃⌘F	Reopen closed editor
⌘K Enter	Keep preview mode editor open
⌃Tab / ⌃⌘Tab	Open next/previous
⌘K P	Copy path of active file
⌘K R	Reveal active file in Explorer
⌘K O	Show active file in new window/instance
Debug	
F9	Toggle breakpoint
F5	Start/Continue
F11 / ⌃F11	Step into/out
F10	Step over
⌃F5	Stop
⌃K ⌃I	Show hover
Integrated terminal	
⌃T	Show integrated terminal
⌃`	Create new terminal
⌘C	Copy selection
unassigned	Paste into active terminal
⌃↑ / ⌃↓	Scroll up/down
PgUp / PgDn	Scroll page up/down
⌃Home / End	Scroll to top/bottom

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

Quick Open

Quickly open files.

Keyboard Shortcut: ⌃Ctrl+P



Tip: Type `?` to view help suggestions.

Navigate between recently opened files

Repeat the **Quick Open** keyboard shortcut to cycle quickly between recently opened files.

Open multiple files from Quick Open

You can open multiple files from **Quick Open** by pressing the Right arrow key. This will open the currently selected file in the background and you can continue selecting files from **Quick Open**.

Command line

VS Code has a powerful command line interface (CLI) which allows you to customize how the editor is launched to support various scenarios.

Make sure the VS Code binary is on your path so you can simply type 'code' to launch VS Code. See the platform specific setup topics if VS Code is added to your environment path during installation (Running VS Code on Linux (/docs/setup/linux), macOS (/docs/setup/mac), Windows (/docs/setup/windows)).

```
# open code with current directory
code .

# open the current directory in the most recently used code window
code -r .

# create a new window
code -n

# change the language
code --locale=es

# open diff editor
code --diff <file1> <file2>

# open file at specific line and column <file:line[:character]>
code --goto package.json:10:5

# see help options
code --help

# disable all extensions
code --disable-extensions .
```

.vscode folder

Workspace specific files are in a `.vscode` folder at the root. For example, `tasks.json` for the Task Runner and `launch.json` for the debugger.

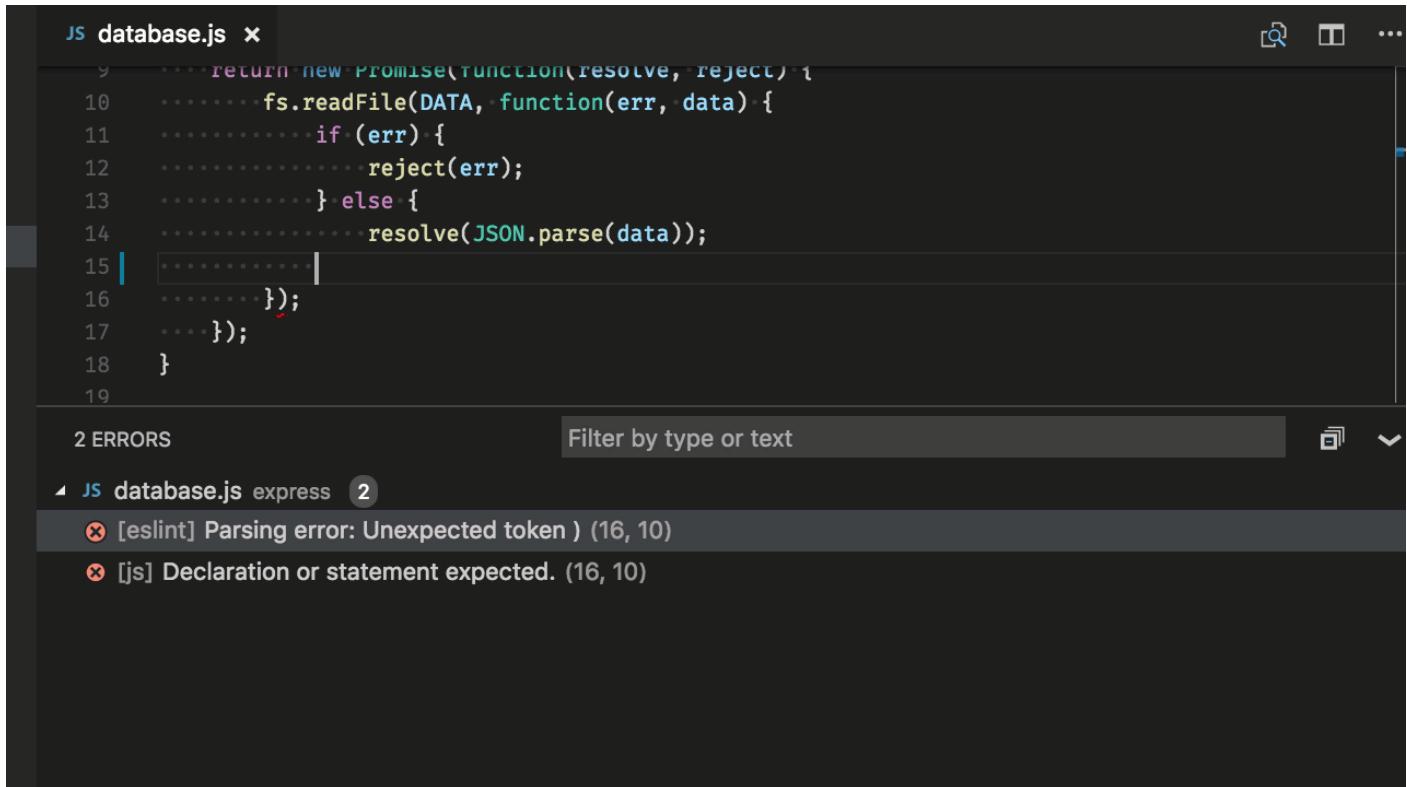
Status Bar

Errors and warnings

Keyboard Shortcut: **Ctrl+Shift+M**

Quickly jump to errors and warnings in the project.

Cycle through errors with F8 or Shift+F8



JS database.js x

```
9     return new Promise(function(resolve, reject) {
10    fs.readFile(DATA, function(err, data) {
11      if (err) {
12        reject(err);
13      } else {
14        resolve(JSON.parse(data));
15      }
16    });
17  });
18}
```

2 ERRORS

Filter by type or text

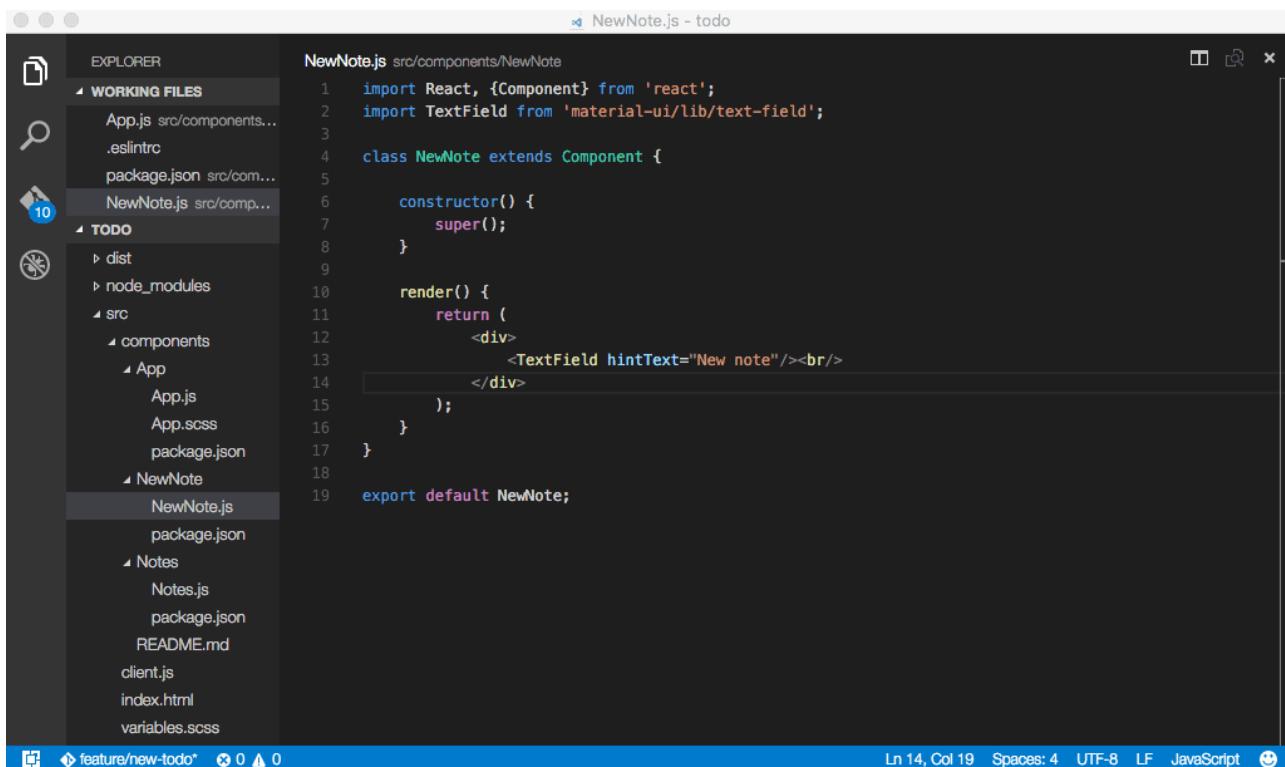
▲ JS database.js express 2

- ✖ [eslint] Parsing error: Unexpected token) (16, 10)
- ✖ [js] Declaration or statement expected. (16, 10)

You can filter problems either by type ('errors', 'warnings') or text matching.

Change language mode

Keyboard Shortcut: Ctrl+K M



EXPLORER

WORKING FILES

- App.js
- .eslintrc
- package.json
- NewNote.js

TODO

- dist
- node_modules
- src
- components
- App
- App.js
- App.scss
- package.json
- NewNote
- NewNote.js
- package.json
- Notes
- Notes.js
- package.json
- README.md
- client.js
- index.html
- variables.scss

NewNote.js src/components/NewNote

```
1 import React, {Component} from 'react';
2 import TextField from 'material-ui/lib/text-field';
3
4 class NewNote extends Component {
5
6   constructor() {
7     super();
8   }
9
10  render() {
11    return (
12      <div>
13        <TextField hintText="New note"/><br/>
14      </div>
15    );
16  }
17}
18
19 export default NewNote;
```

Ln 14, Col 19 Spaces: 4 UTF-8 LF JavaScript

If you want to persist the new language mode for that file type, you can use the **Configure File Association for** command to associate the current file extension with an installed language.

Customization

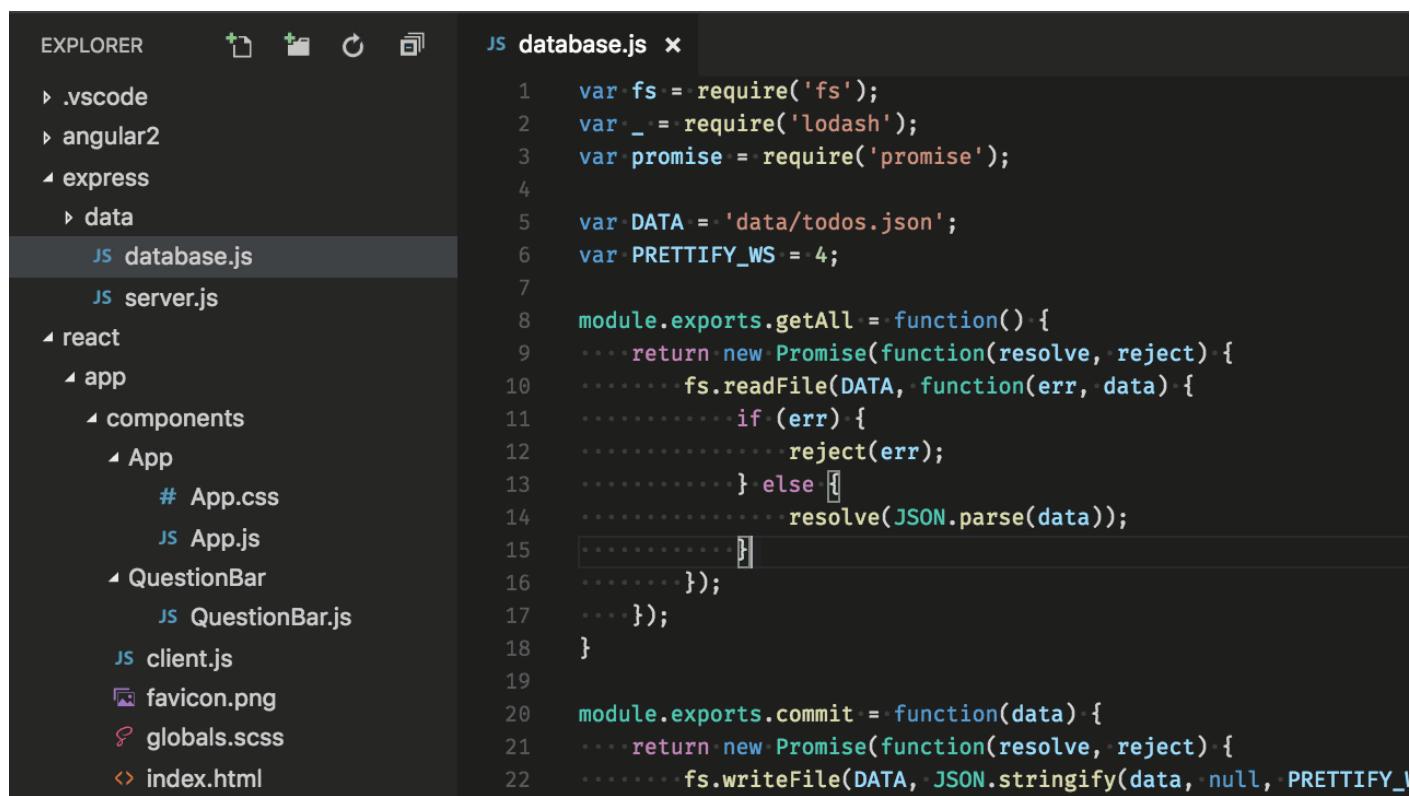
There are many things you can do to customize VS Code.

- Change your theme
- Change your keyboard shortcuts
- Tune your settings
- Add JSON validation
- Create snippets
- Install extensions

Change your theme

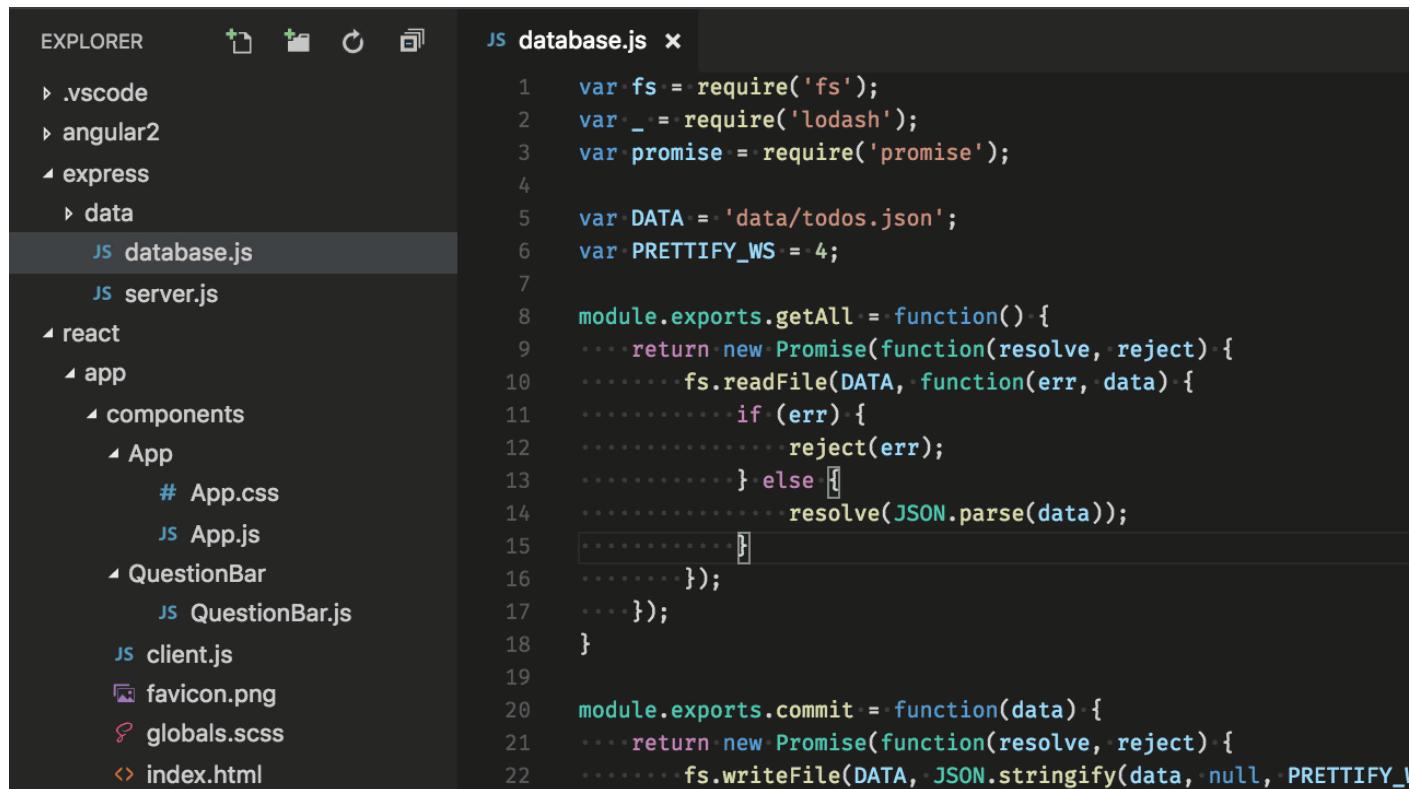
Keyboard Shortcut: **Ctrl+K Ctrl+T**

You can install more themes from the VS Code extension Marketplace (<https://marketplace.visualstudio.com/search?target=VSCode&category=Themes&sortBy=Downloads>).



```
1  var fs = require('fs');
2  var _ = require('lodash');
3  var promise = require('promise');
4
5  var DATA = 'data/todos.json';
6  var PRETTIFY_WS = 4;
7
8  module.exports.getAll = function() {
9      return new Promise(function(resolve, reject) {
10          fs.readFile(DATA, function(err, data) {
11              if (err) {
12                  reject(err);
13              } else {
14                  resolve(JSON.parse(data));
15              }
16          });
17      });
18  }
19
20 module.exports.commit = function(data) {
21     return new Promise(function(resolve, reject) {
22         fs.writeFile(DATA, JSON.stringify(data, null, PRETTIFY_WS));
23     });
24 }
```

Additionally, you can install and change your File Icon themes.



```
1  var fs = require('fs');
2  var _ = require('lodash');
3  var promise = require('promise');
4
5  var DATA = 'data/todos.json';
6  var PRETTIFY_WS = 4;
7
8  module.exports.getAll = function() {
9      return new Promise(function(resolve, reject) {
10          fs.readFile(DATA, function(err, data) {
11              if (err) {
12                  reject(err);
13              } else {
14                  resolve(JSON.parse(data));
15              }
16          });
17      });
18  }
19
20 module.exports.commit = function(data) {
21     return new Promise(function(resolve, reject) {
22         fs.writeFile(DATA, JSON.stringify(data, null, PRETTIFY_WS));
23     });
24 }
```

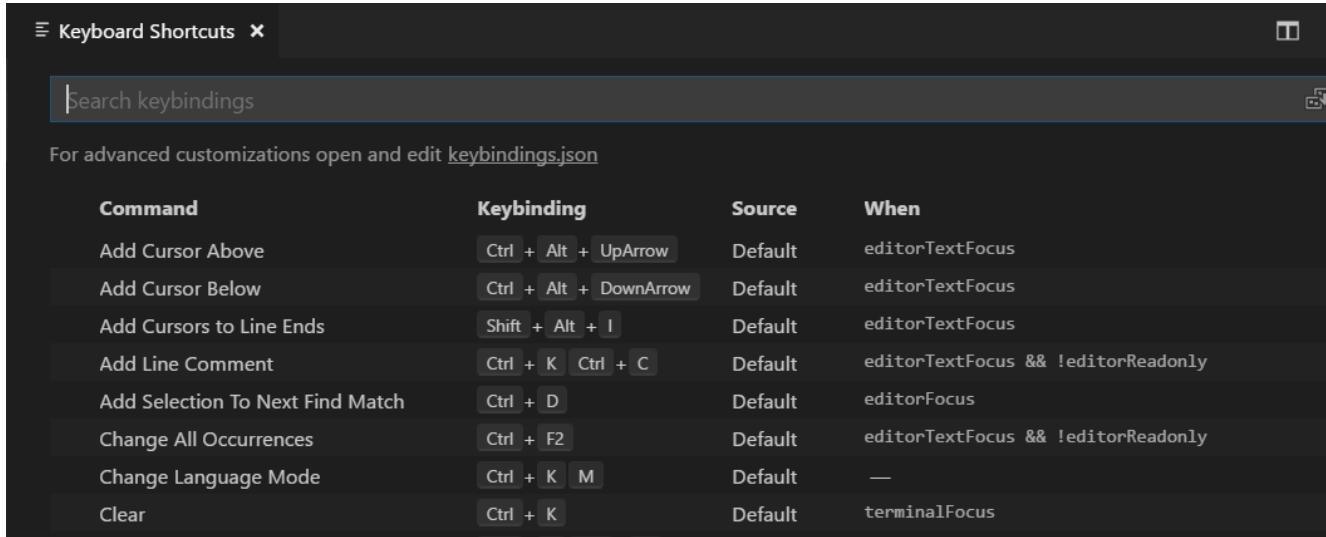
Keymaps

Are you used to keyboard shortcuts from another editor? You can install a Keymap extension that brings the keyboard shortcuts from your favorite editor to VS Code. Go to [Preferences > Keymap Extensions](#) to see the current list on the Marketplace (<https://marketplace.visualstudio.com/search?target=VSCode&category=Keymaps&sortBy=Downloads>). Some of the more popular ones:

- Vim (<https://marketplace.visualstudio.com/items?itemName=vscodevim.vim>)
- Sublime Text Keymap (<https://marketplace.visualstudio.com/items?itemName=ms-vscode.sublime-keybindings>)
- Emacs Keymap (<https://marketplace.visualstudio.com/items?itemName=hiro-sun.vscode-emacs>)
- Atom Keymap (<https://marketplace.visualstudio.com/items?itemName=ms-vscode.atom-keybindings>)
- Eclipse Keymap (<https://marketplace.visualstudio.com/items?itemName=alphabotsec.vscode-eclipse-keybindings>)

Customize your keyboard shortcuts

Keyboard Shortcut: **Ctrl+K** **Ctrl+S**



You can search for shortcuts and add your own keybindings to the `keybindings.json` file.



See more in [Key Bindings for Visual Studio Code](#) (`/docs/getstarted/keybindings`).

Tune your settings

By default VS Code shows the Settings editor, you can find settings listed below in a search bar, but you can still edit the underlying `settings.json` file by using the [Open Settings \(JSON\)](#) command or by changing your default settings editor with the `workbench.settings.editor` setting.

Open User Settings `settings.json`

Keyboard Shortcut: **Ctrl+,**

Format on paste

```
"editor.formatOnPaste": true
```

Change the font size of various UI elements

```
// Main editor
"editor.fontSize": 18,
// Terminal panel
"terminal.integrated.fontSize": 14,
// Output panel
"[Log]": {
  "editor.fontSize": 15
}
```

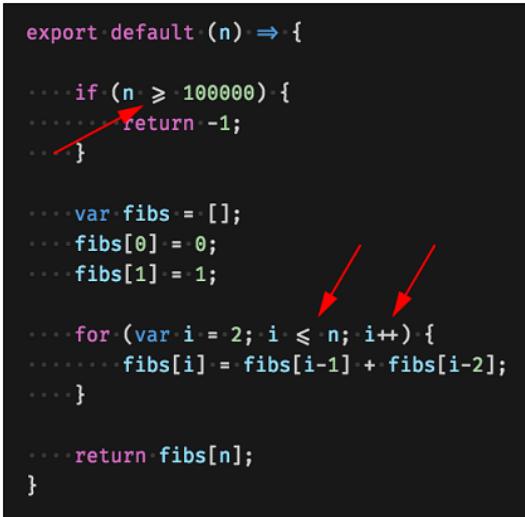
Change the zoom level

```
"window.zoomLevel": 5
```

Font ligatures

```
"editor.fontFamily": "Fira Code",  
"editor.fontLigatures": true
```

Tip: You will need to have a font installed that supports font ligatures. FiraCode (<https://github.com/tonsky/FiraCode>) is a popular font on the VS Code team.



```
export default (n) => {  
  if (n >= 100000) {  
    return -1;  
  }  
  
  var fibs = [];  
  fibs[0] = 0;  
  fibs[1] = 1;  
  
  for (var i = 2; i < n; i++) {  
    fibs[i] = fibs[i-1] + fibs[i-2];  
  }  
  
  return fibs[n];  
}
```

Auto Save

```
"files.autoSave": "afterDelay"
```

You can also toggle Auto Save from the top-level menu with the **File > Auto Save**.

Format on save

```
"editor.formatOnSave": true
```

Change the size of Tab characters

```
"editor.tabSize": 4
```

Spaces or Tabs

```
"editor.insertSpaces": true
```

Render whitespace

```
"editor.renderWhitespace": "all"
```

Ignore files / folders

Removes these files / folders from your editor window.

```
"files.exclude": {  
  "somefolder/": true,  
  "somefile": true  
}
```

Remove these files / folders from search results.

```
"search.exclude": {  
  "someFolder/": true,  
  "somefile": true  
}
```

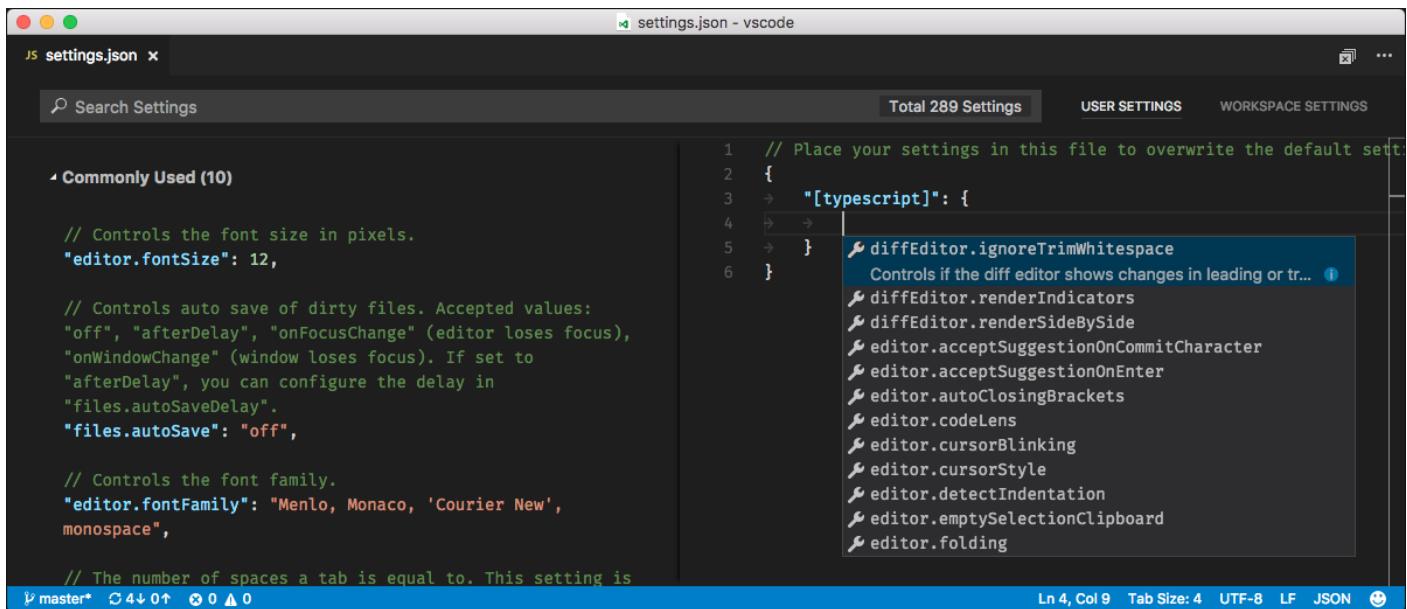
And many, many other customizations ([/docs/getstarted/settings](#)).

Language specific settings

For the settings, which you only want for specific languages, you can scope the settings by the language identifier. You can find a list of commonly used language ids in the [Language Identifiers](#) ([/docs/languages/identifiers](#)) reference.

```
"[languageid)": {  
}
```

Tip: You can also create language specific settings with the [Configure Language Specific Settings](#) command.



A screenshot of the VS Code interface showing the settings.json file. The file contains configuration for the editor, including font size, auto save, and font family. A tooltip is open over the "[typescript]" section, listing various editor settings like ignoreTrimWhitespace, renderIndicators, and folding. The status bar at the bottom shows the file is 289 settings long and is in JSON mode.

```
// Place your settings in this file to overwrite the default settings
{
  "[typescript]": {
    "diffEditor.ignoreTrimWhitespace": true,
    "diffEditor.renderIndicators": true,
    "diffEditor.renderSideBySide": false,
    "editor.acceptSuggestionOnCommitCharacter": true,
    "editor.acceptSuggestionOnEnter": true,
    "editor.autoClosingBrackets": true,
    "editor.codeLens": true,
    "editor.cursorBlinking": true,
    "editor.cursorStyle": "underline",
    "editor.detectIndentation": true,
    "editor.emptySelectionClipboard": true,
    "editor.folding": true
  }
}
```

Add JSON validation

Enabled by default for many file types. Create your own schema and validation in `settings.json`

```
"jsonschemas": [
  {
    "fileMatch": [
      "/bower.json"
    ],
    "url": "http://json.schemastore.org/bower"
  }
]
```

or for a schema defined in your workspace

```
"jsonschemas": [
  {
    "fileMatch": [
      "/foo.json"
    ],
    "url": "./myschema.json"
  }
]
```

or a custom schema

```
"jsonschemas": [
  {
    "fileMatch": [
      "./myconfig"
    ],
    "schema": {
      "type": "object",
      "properties": {
        "name" : {
          "type": "string",
          "description": "The name of the entry"
        }
      }
    }
  }
],
```

See more in the [JSON \(/docs/languages/json\) documentation](#).

Extensions

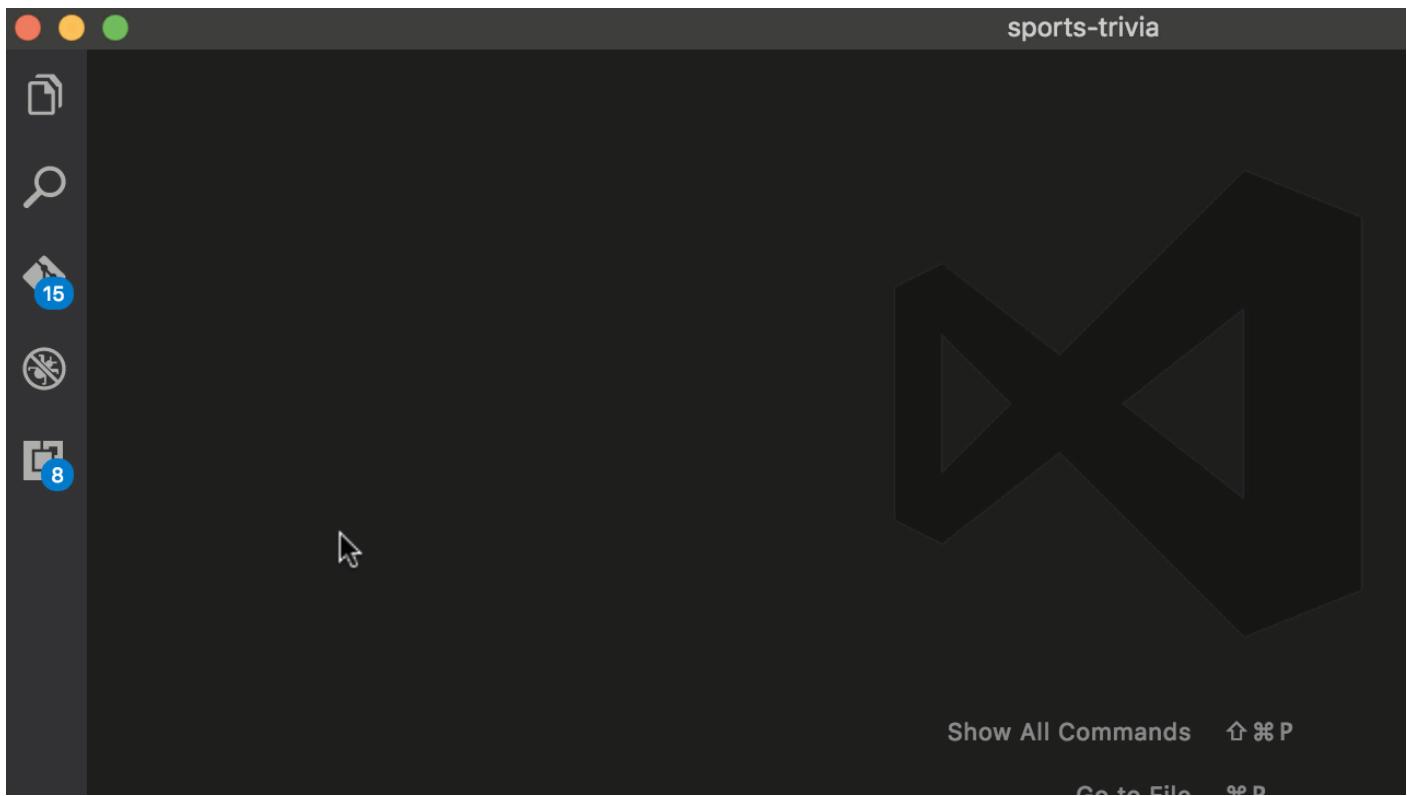
Keyboard Shortcut: **Ctrl+Shift+X**

Find extensions

1. In the VS Code Marketplace (<https://marketplace.visualstudio.com/vscode>).
2. Search inside VS Code in the **Extensions** view.
3. View extension recommendations
4. Community curated extension lists, such as `awesome-vscode` (<https://github.com/viatsko/awesome-vscode>).

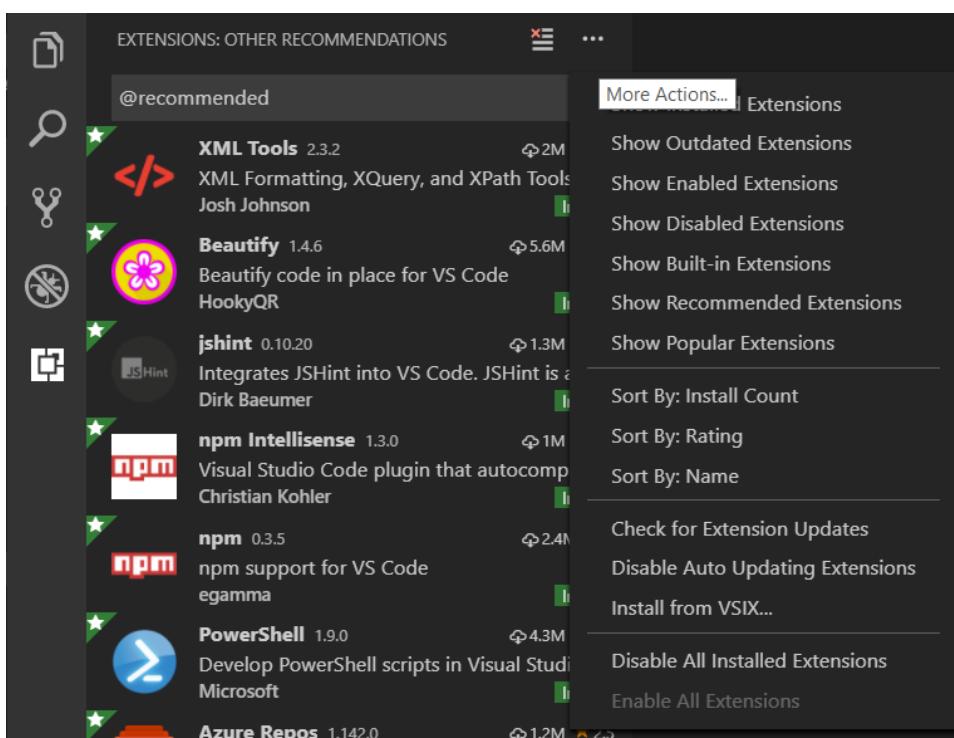
Install extensions

In the **Extensions** view, you can search via the search bar or click the **More Actions (...)** button to filter and sort by install count.



Extension recommendations

In the Extensions view, click **Show Recommended Extensions** in the **More Actions (...)** button menu.



Creating my own extension

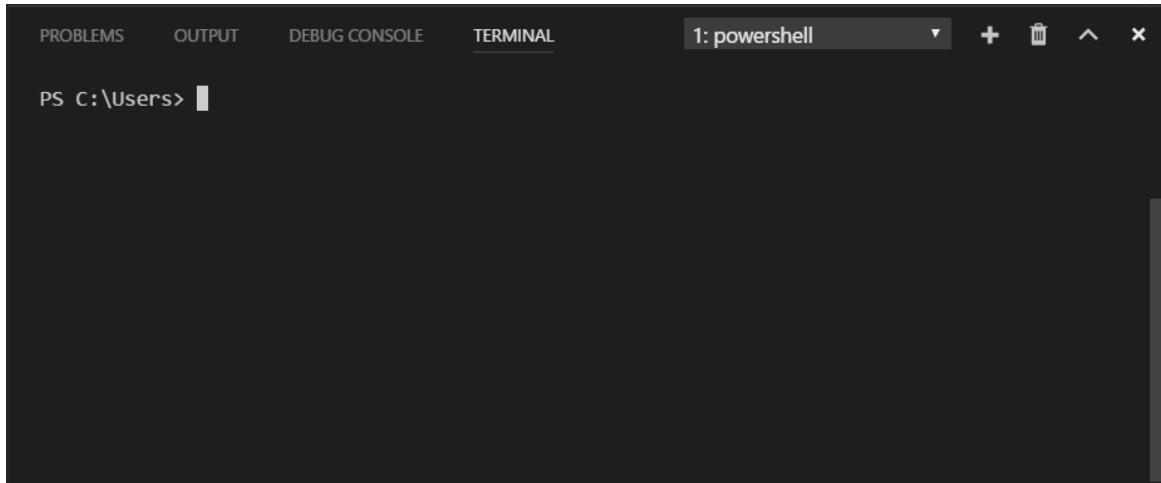
Are you interested in creating your own extension? You can learn how to do this in the Extension API documentation ([/api](#)), specifically check out the documentation on contribution points ([/api/references/contribution-points](#)).

- configuration
- commands
- keybindings
- languages
- debuggers
- grammars
- themes
- snippets
- jsonValidation

Files and folders

Integrated Terminal

Keyboard Shortcut: **Ctrl+`**



Further reading:

- Integrated Terminal ([/docs/editor/integrated-terminal](#)) documentation
- Mastering VS Code's Terminal article (<https://www.growingwiththeweb.com/2017/03/mastering-vscode-terminal.html>)

Auto Save

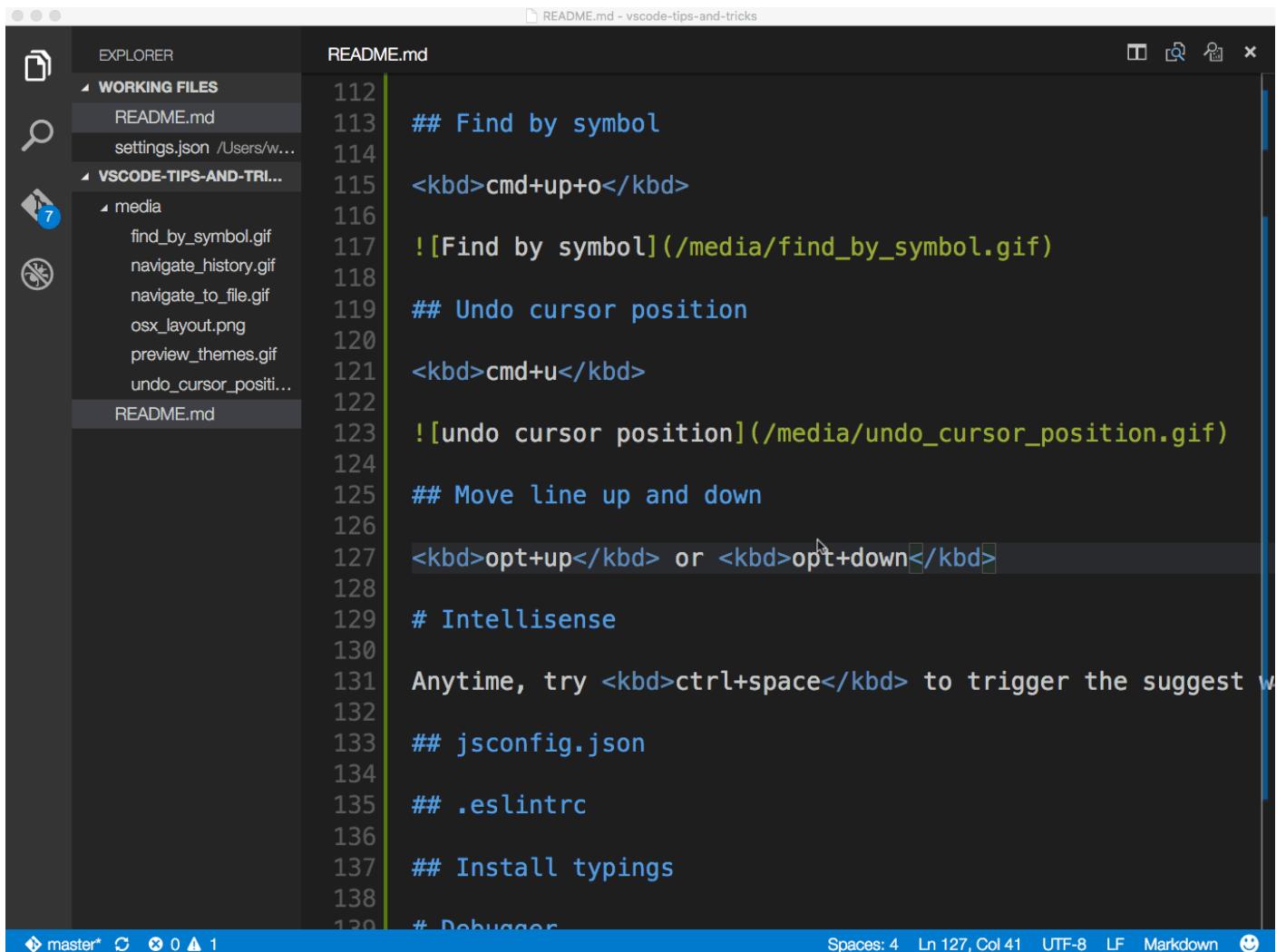
Open User Settings `settings.json` with **Ctrl+,**

```
"files.autoSave": "afterDelay"
```

You can also toggle Auto Save from the top-level menu with the **File > Auto Save**.

Toggle Sidebar

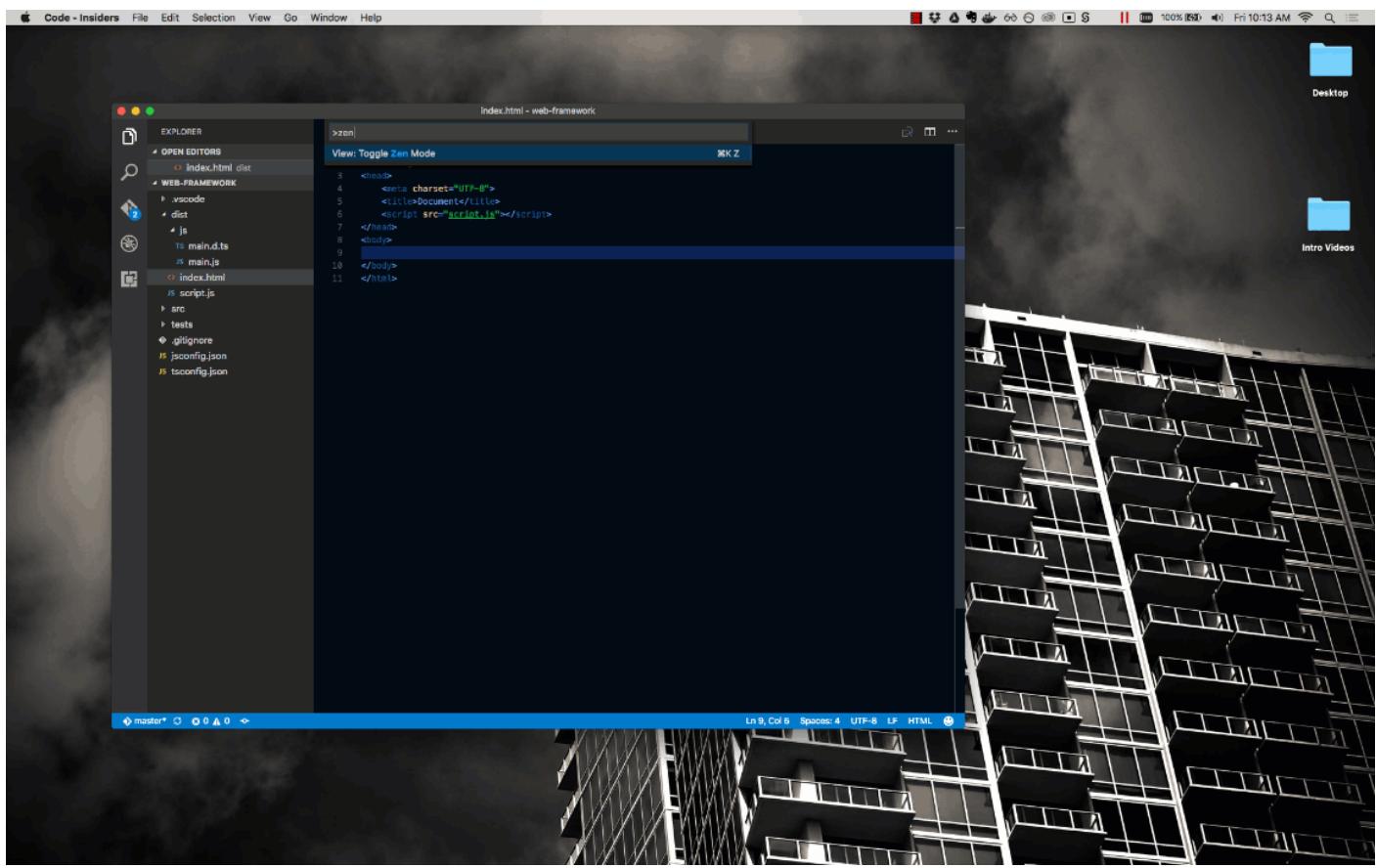
Keyboard Shortcut: **Ctrl+B**



Zen mode

Keyboard Shortcut: **Ctrl+K Z**

Spaces: 4 Ln 127, Col 41 UTF-8 LF Markdown



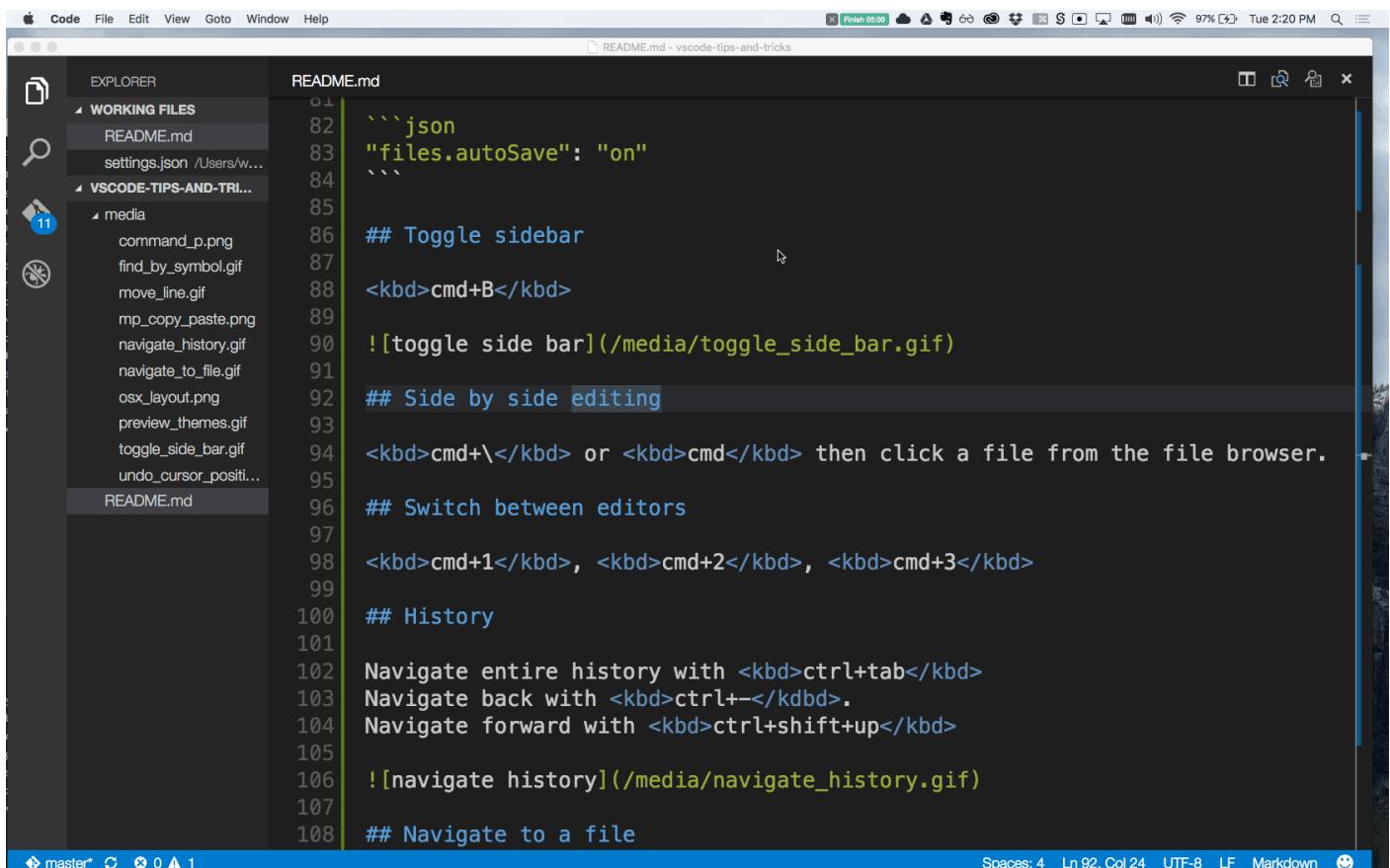
Enter distraction free Zen mode.

Press `Esc` twice to exit Zen Mode.

Side by side editing

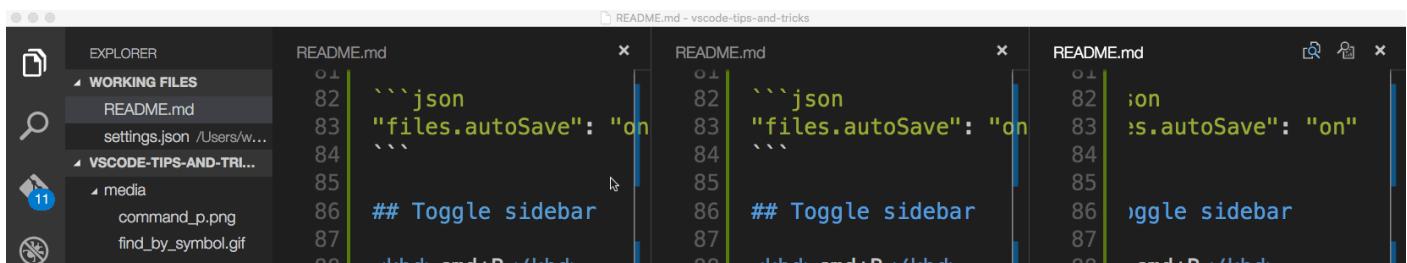
Keyboard Shortcut: `Ctrl+Shift+\`

You can also drag and drop editors to create new editor groups and move editors between groups.



Switch between editors

Keyboard Shortcut: `Ctrl+1`, `Ctrl+2`, `Ctrl+3`



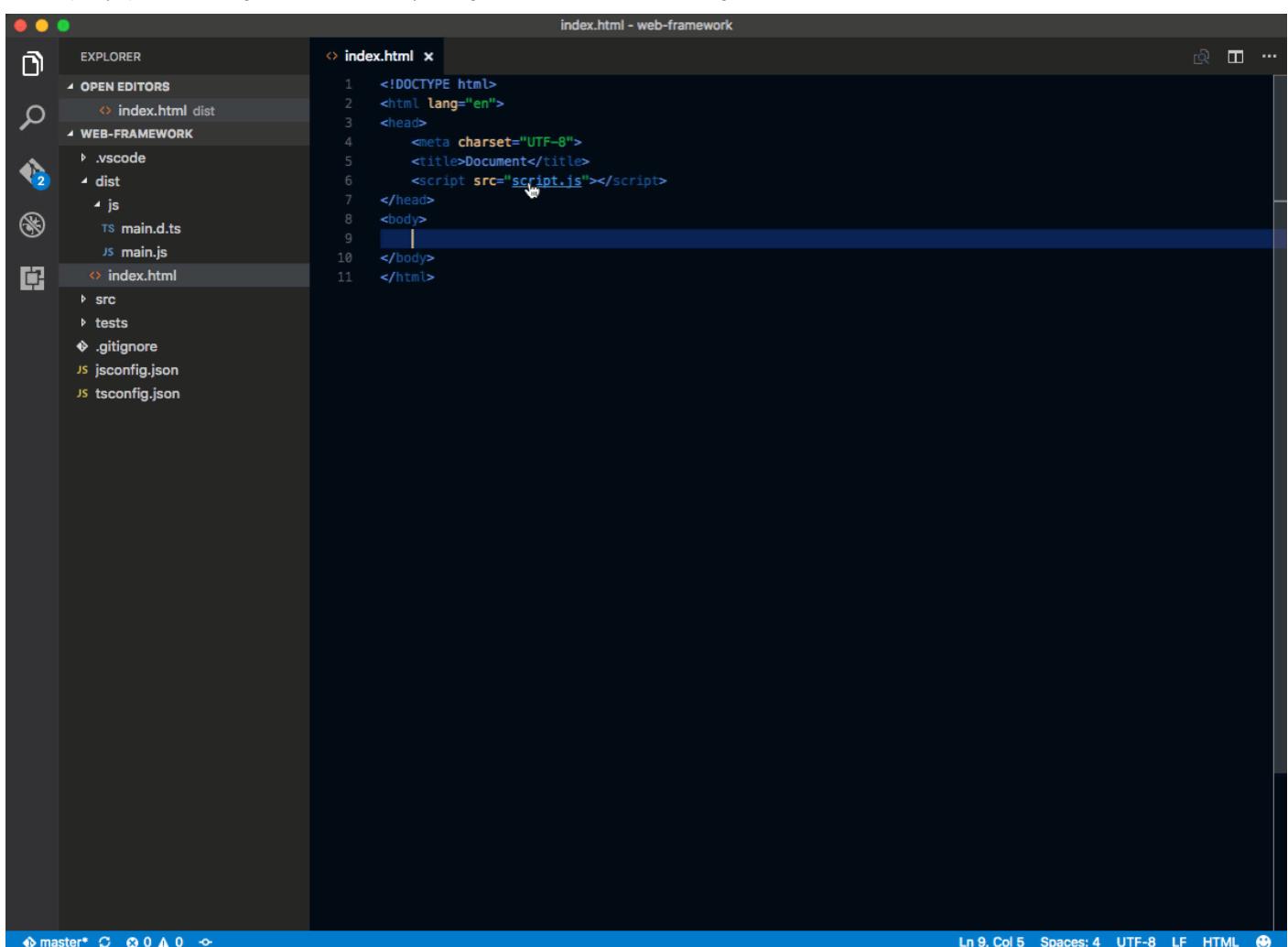
Move to Explorer window

Keyboard Shortcut: **Ctrl+Shift+E**

Create or open a file

Keyboard Shortcut: **Ctrl+click** (**Cmd+click** on macOS)

You can quickly open a file or image or create a new file by moving the cursor to the file link and using **Ctrl+click**.



index.html - web-framework

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Document</title>
6      <script src="script.js"></script>
7  </head>
8  <body>
9
10 </body>
11 </html>
```

Ln 9, Col 5 Spaces: 4 UTF-8 LF HTML

Close the currently opened folder

Keyboard Shortcut: **Ctrl+F4**

Navigation history

Navigate entire history: **Ctrl+Tab**

Navigate back: **Alt+Left**

Navigate forward: **Alt+Right**

```
package.json src/components/TodoList
1  {
2    "name": "todo-list",
3    "version": "0.1.0",
4    "private": true,
5    "main": "./TodoList.js"
6  }
7
```

File associations

Create language associations for files that aren't detected correctly. For example, many configuration files with custom file extensions are actually JSON.

```
"files.associations": {
  ".database": "json"
}
```

Preventing dirty writes

VS Code will show you an error message when you try to save a file that cannot be saved because it has changed on disk. VS Code blocks saving the file to prevent overwriting changes that have been made outside of the editor.

In order to resolve the save conflict, click the **Compare** action in the error message to open a diff editor that will show you the contents of the file on disk (to the left) compared to the contents in VS Code (on the right):

Accept / Revert

```
play.htm (on disk) ⇄ play.htm (in Code - OSS Dev) - Resolve save conflict
```

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>Mankala</title>
6   <meta http-equiv="X-UA-Compatible" content="IE=Edge" />
7   <script type="text/javascript" src="Game.js"></script>
8   <script type="text/javascript" src="Driver.js"></script>
9   <script type="text/javascript" src="Features.js"></script>
10  <script type="text/javascript" src="Position.js"></script>
11  <script type="text/javascript" src="lib/Geometry.js"></script>
12  <script type="text/javascript" src="lib/Base.js"></script>
13 </head>
14 -
15 <body id="bod" onload="Mankala.testBrowser()">
16   <svg height="10px">
```

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>Mankala</title>
6   <meta http-equiv="X-UA-Compatible" content="IE=Edge" />
7   <script type="text/javascript" src="Game.js"></script>
8   <script type="text/javascript" src="Driver.js"></script>
9   <script type="text/javascript" src="Features.js"></script>
10  <script type="text/javascript" src="Position.js"></script>
11  <script type="text/javascript" src="lib/Geometry.js"></script>
12  <script type="text/javascript" src="lib/Base.js"></script>
13 </head>
14 +<change>
15 <body id="bod" onload="Mankala.testBrowser()">
16   <svg height="10px">
```

Use the actions in the editor toolbar to resolve the save conflict. You can either **Accept** your changes and thereby overwriting any changes on disk, or **Revert** to the version on disk. Reverting means that your changes will be lost.

Note: The file will remain dirty and cannot be saved until you pick one of the two actions to resolve the conflict.

Editing hacks

Here is a selection of common features for editing code. If the keyboard shortcuts aren't comfortable for you, consider installing a keymap extension (<https://marketplace.visualstudio.com/search?target=VSCode&category=Keymaps&sortBy=Downloads>) for your old editor.

Tip: You can see recommended keymap extensions in the **Extensions** view with **Ctrl+K Ctrl+M** which filters the search to **@recommended:keymaps**.

Multi cursor selection

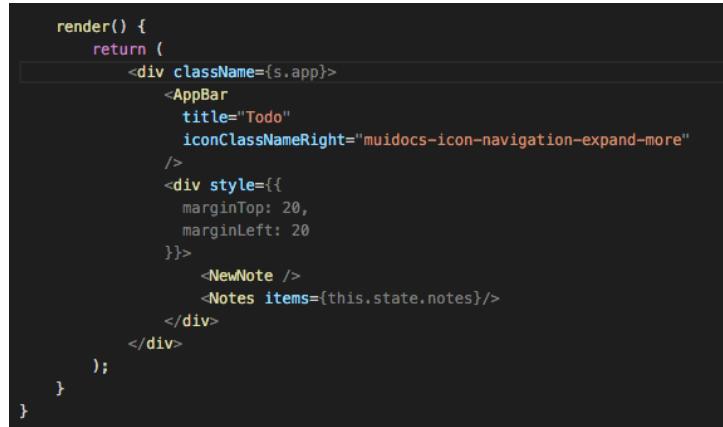
To add cursors at arbitrary positions, select a position with your mouse and use **Alt+Click** (**Option+click** on macOS).

To set cursors above or below the current position use:

Keyboard Shortcut: **Ctrl+Alt+Up** or **Ctrl+Alt+Down**

```
31 .global-message-list.transition {
32   -webkit-transition: top 200ms linear;
33   -ms-transition: top 200ms linear;
34   -moz-transition: top 200ms linear;
35   -khtml-transition: top 200ms linear;
36   -o-transition: top 200ms linear;
37   transition: top 200ms linear;
38 }
```

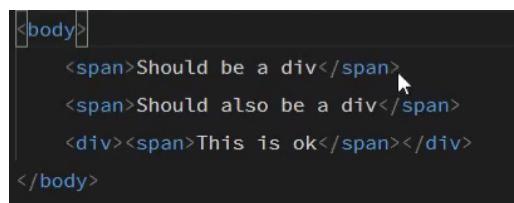
You can add additional cursors to all occurrences of the current selection with `Ctrl+Shift+L`.



```
render() {
  return (
    <div className={s.app}>
      <AppBar
        title="Todo"
        iconClassNameRight="muidocs-icon-navigation-expand-more"
      />
      <div style={{
        marginTop: 20,
        marginLeft: 20
      }}>
        <NewNote />
        <Notes items={this.state.notes}>/>
      </div>
    </div>
  );
}
```

Note: You can also change the modifier to `Ctrl/Cmd` for applying multiple cursors with the `editor.multiCursorModifier` setting ([/docs/getstarted/settings](#)). See [Multi-cursor Modifier](#) ([/docs/editor/codebasics#_multicursor-modifier](#)) for details.

If you do not want to add all occurrences of the current selection, you can use `Ctrl+D` instead. This only selects the next occurrence after the one you selected so you can add selections one by one.



```
<body>
  <span>Should be a div</span>
  <span>Should also be a div</span>
  <div><span>This is ok</span></div>
</body>
```

Column (box) selection

You can select blocks of text by holding `Shift+Alt` (`Shift+Option` on macOS) while you drag your mouse. A separate cursor will be added to the end of each selected line.



Line	Character	Virtual Key Code
208	// Key	
209	';	186
210	// Semicolon	
211	::	186
212	// Equals sign	187
213	'+'	187
214	// Comma	188
215	'<'	188
216	'-'	189
217	// Minus	
218	'_'	189
219	// Underscore	
220	'.'	190
221	// Period	
222	'>'	190
223	'/'	191

You can also use keyboard shortcuts ([/docs/editor/codebasics#_column-box-selection](#)) to trigger column selection.

Fast scrolling

Pressing the `Alt` key enables fast scrolling in the editor and Explorers. By default, fast scrolling uses a 5X speed multiplier but you can control the multiplier with the **Editor: Fast Scroll Sensitivity** (`editor.fastScrollSensitivity`) setting.

Copy line up / down

Keyboard Shortcut: `Shift+Alt+Up` or `Shift+Alt+Down`

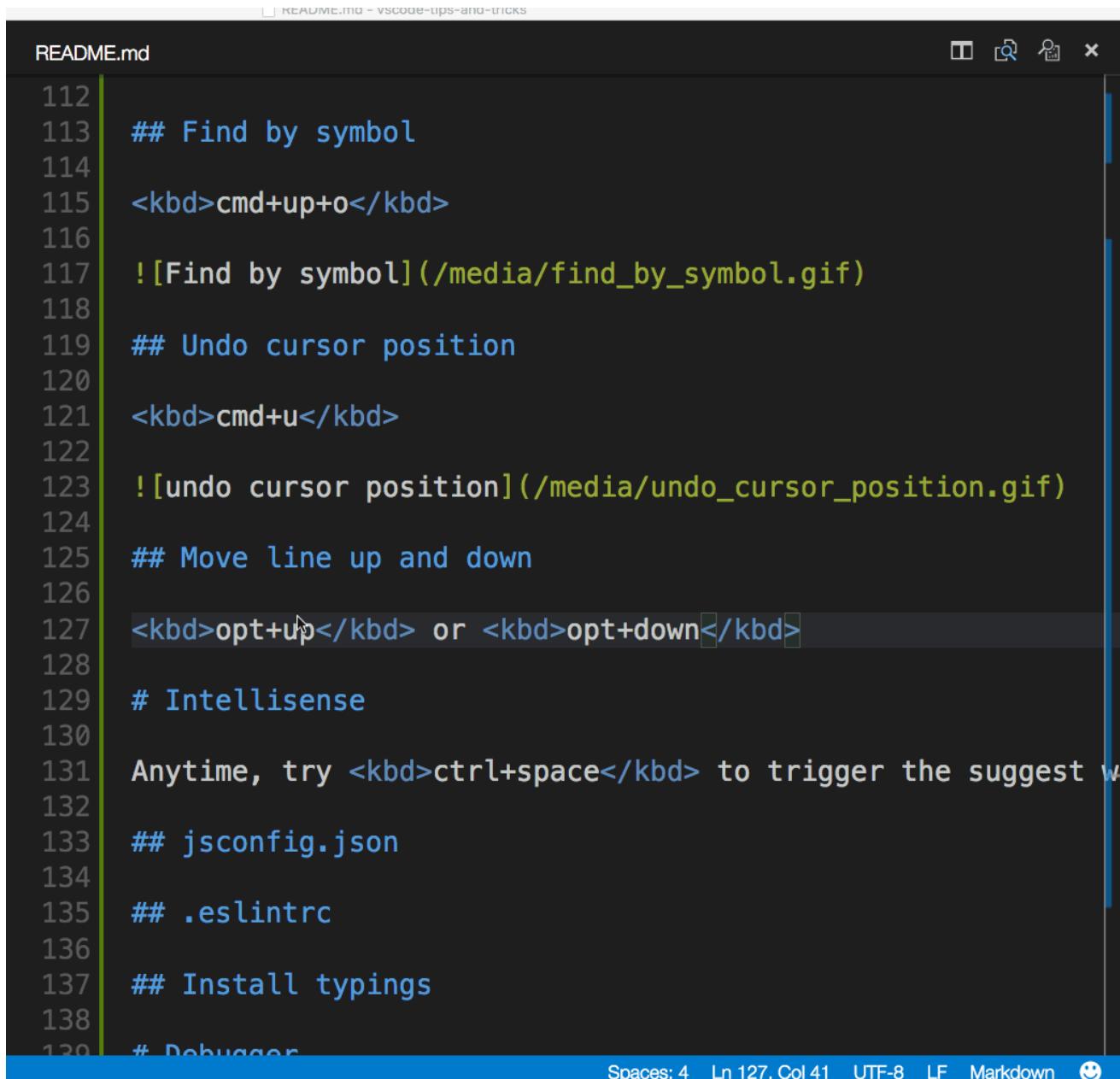
The commands `Copy Line Up/Down` are unbound on Linux because the VS Code default keybindings would conflict with Ubuntu keybindings, see Issue #509 (<https://github.com/Microsoft/vscode/issues/509>). You can still set the commands `editor.action.copyLinesUpAction` and `editor.action.copyLinesDownAction` to your own preferred keyboard shortcuts.

server.js

```
1  var express = require('express');
2  var bodyParser = require('body-parser')
3
4  var database = require('./database');
5
```

Move line up and down

Keyboard Shortcut: Alt+Up or Alt+Down



```
112
113  ## Find by symbol
114
115  <kbd>cmd+up+o</kbd>
116
117  ! [Find by symbol] (/media/find_by_symbol.gif)
118
119  ## Undo cursor position
120
121  <kbd>cmd+u</kbd>
122
123  ! [undo cursor position] (/media/undo_cursor_position.gif)
124
125  ## Move line up and down
126
127  <kbd>opt+up</kbd> or <kbd>opt+down</kbd>
128
129  # Intellisense
130
131  Anytime, try <kbd>ctrl+space</kbd> to trigger the suggest w
132
133  ## jsconfig.json
134
135  ## .eslintrc
136
137  ## Install typings
138
139  # Debugger
```

Spaces: 4 Ln 127, Col 41 UTF-8 LF Markdown 😊

Shrink / expand selection

Keyboard Shortcut: Shift+Alt+Left or Shift+Alt+Right

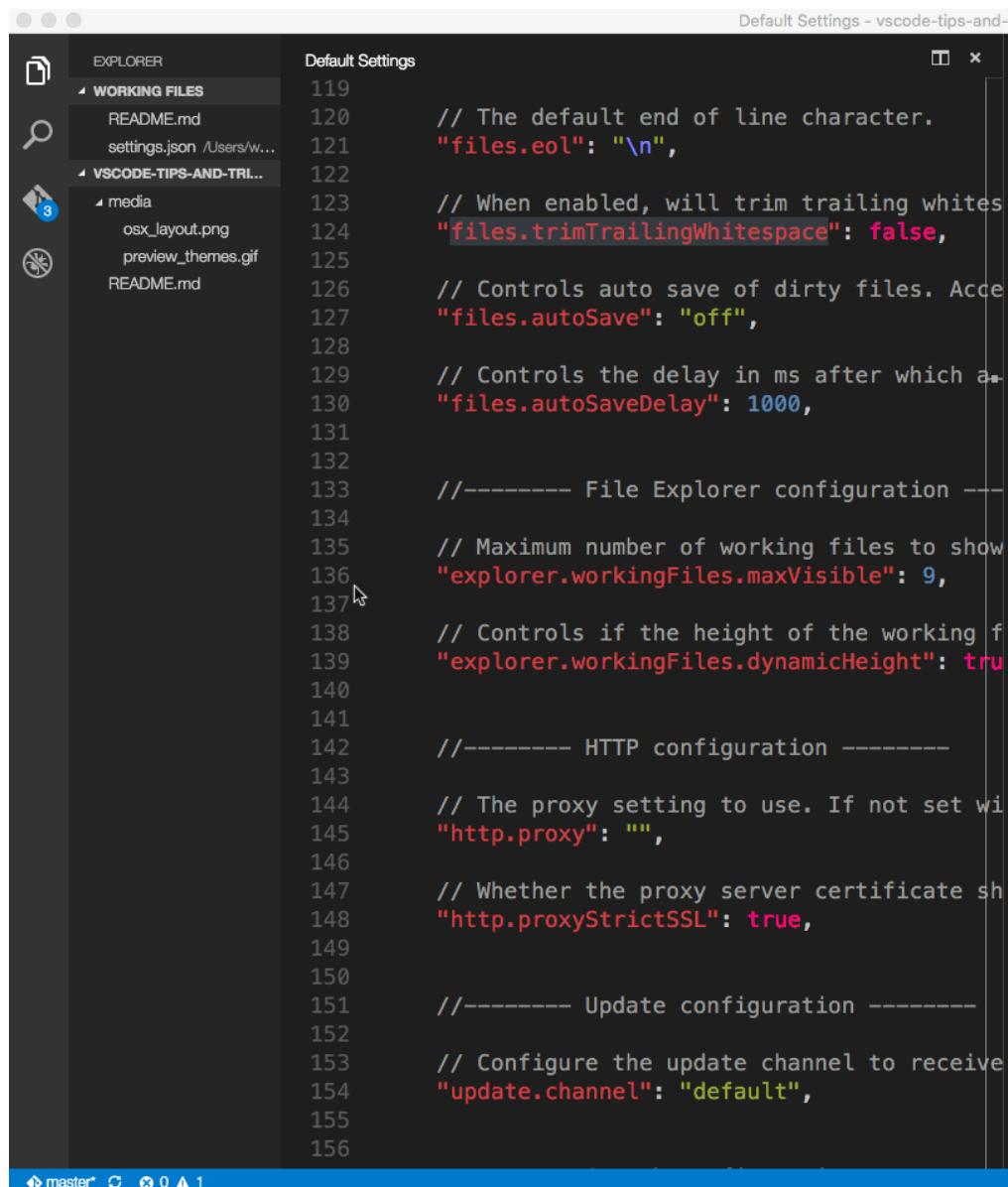


```
constructor() {
  super();
  this.state = {
    notes: [
      {
        text: 'Book flight for build'
      },
      {
        text: 'Write a cool, new app'
      },
      {
        text: 'build a react app'
      }
    ]
  };
}
```

You can learn more in the [Basic Editing \(/docs/editor/codebasics#_shrinkexpand-selection\)](#) documentation.

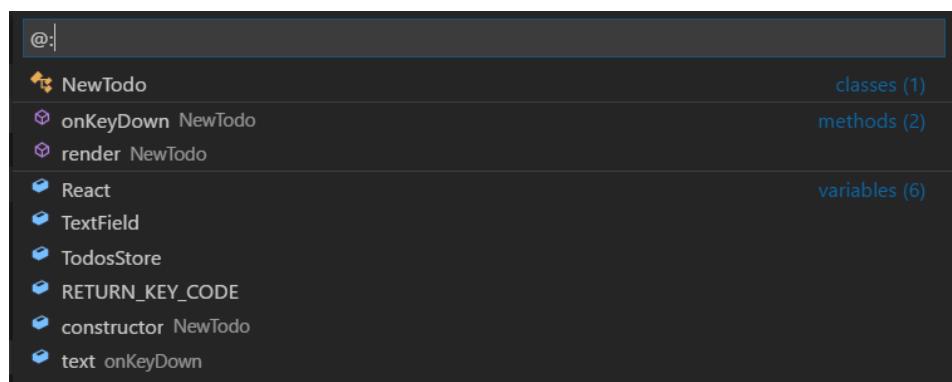
Go to Symbol in File

Keyboard Shortcut: **Ctrl+Shift+T**



```
119 // The default end of line character.
120 "files.eol": "\n",
121
122 // When enabled, will trim trailing whites
123 "files.trimTrailingWhitespace": false,
124
125 // Controls auto save of dirty files. Acce
126 "files.autoSave": "off",
127
128 // Controls the delay in ms after which a
129 "files.autoSaveDelay": 1000,
130
131
132 //----- File Explorer configuration -----
133
134 // Maximum number of working files to show
135 "explorer.workingFiles.maxVisible": 9,
136
137 // Controls if the height of the working f
138 "explorer.workingFiles.dynamicHeight": tru
139
140
141 //----- HTTP configuration -----
142
143 // The proxy setting to use. If not set wi
144 "http.proxy": "",
145
146 // Whether the proxy server certificate sh
147 "http.proxyStrictSSL": true,
148
149
150 //----- Update configuration -----
151
152 // Configure the update channel to receive
153 "update.channel": "default",
154
155
156
```

You can group the symbols by kind by adding a colon, `@:` .



Symbol Kind	Count	Symbol Names
classes	1	NewTodo
methods	2	onKeyDown, render
variables	6	React, TextField, TodosStore, RETURN_KEY_CODE, constructor, text

Go to Symbol in Workspace

Keyboard Shortcut: **Ctrl+T**

```
#todo
  todo add — TodosStore.js
  todo server\server.js
  ↗ Todos app\components\Todos.js
  ↗ todos render — Todos.js
  ↗ todos server\database.js
  ↗ TodoData app\components\TodoData.js
  ↗ TodosStore app\components\TodosStore.js
  ↗ NewTodo app\components\NewTodo.js
```

Navigate to a specific line

Keyboard Shortcut: **Ctrl+G**

Undo cursor position

Keyboard Shortcut: **Ctrl+U**

Trim trailing whitespace

Keyboard Shortcut: **Ctrl+K Ctrl+X**

```
render() {
  return (
    <List>{this.props.items.map(this.createItem)}</List>
  );
}
}
```

Code formatting

Currently selected source code: **Ctrl+K Ctrl+F**

Whole document format: **Shift+Alt+F**

```
render() {
  return (
    <div className={s.app}>
      <AppBar
        title="Notes"
        iconClassNameRight="muidocs-icon-navigation-expand-right"
      />
      <div style={{
        marginTop: 20,
        marginLeft: 20
      }}>
        <NewNote />
        <Notes items={this.state.notes}/>
      </div>
    );
}
```

Code folding

Keyboard Shortcut: **Ctrl+Shift+[** and **Ctrl+Shift+]**

```
class App extends Component {

  constructor() {
    super();
    this.state = {
      notes: [
        {
          text: "go to the grocery store"
        },
        {
          text: 'read medium article about engineering'
        },
        {
          text: 'create build session'
        },
        {
          text: 'fix bug #232'
        }
      ]
    };
  }
}
```

Select current line

Keyboard Shortcut: **Ctrl+L**

Navigate to beginning and end of file

Keyboard Shortcut: **Ctrl+Home** and **Ctrl+End**

Open Markdown preview

In a Markdown file, use

Keyboard Shortcut: **Ctrl+Shift+V**



tips-and-tricks.md Preview tips-and-tricks.md

Visual Studio Code Tips and Tricks

"Tips and Tricks" lets you jump right in and learn how to be productive with Visual Studio Code. You'll become familiar with its powerful editing, code intelligence, and source code control features and learn useful keyboard shortcuts. This topic goes pretty fast and provides a broad overview, so be sure to look at the other in-depth topics in [Getting Started](#) and the [User Guide](#) to learn more.

If you don't have Visual Studio Code installed, go to the [Download](#) page. You can find platform specific setup instructions at [Running VS Code on Linux](#), [macOS](#), and [Windows](#).

Basics

Getting Started

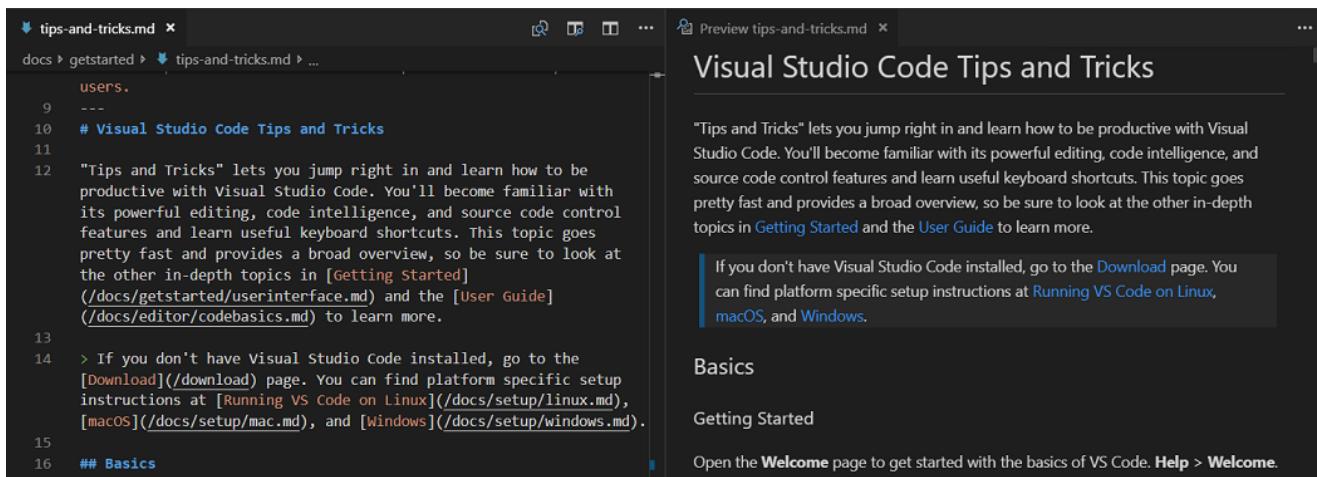
Open the **Welcome** page to get started with the basics of VS Code. **Help > Welcome**.

Side by side Markdown edit and preview

In a Markdown file, use

Keyboard Shortcut: **Ctrl+K V**

The preview and editor will synchronize with your scrolling in either view.



docs > getstarted > tips-and-tricks.md > ...

```
9  ---  
10 # Visual Studio Code Tips and Tricks  
11  
12 "Tips and Tricks" lets you jump right in and learn how to be  
productive with Visual Studio Code. You'll become familiar with  
its powerful editing, code intelligence, and source code control  
features and learn useful keyboard shortcuts. This topic goes  
pretty fast and provides a broad overview, so be sure to look at  
the other in-depth topics in [Getting Started]  
(/docs/getstarted/userinterface.md) and the [User Guide]  
(/docs/editor/codebasics.md) to learn more.  
13  
14 > If you don't have Visual Studio Code installed, go to the  
[Download](/download) page. You can find platform specific setup  
instructions at [Running VS Code on Linux](/docs/setup/linux.md),  
[macOS](/docs/setup/mac.md), and [Windows](/docs/setup/windows.md).  
15  
16 ## Basics
```

Visual Studio Code Tips and Tricks

"Tips and Tricks" lets you jump right in and learn how to be productive with Visual Studio Code. You'll become familiar with its powerful editing, code intelligence, and source code control features and learn useful keyboard shortcuts. This topic goes pretty fast and provides a broad overview, so be sure to look at the other in-depth topics in [Getting Started](#) and the [User Guide](#) to learn more.

If you don't have Visual Studio Code installed, go to the [Download](#) page. You can find platform specific setup instructions at [Running VS Code on Linux](#), [macOS](#), and [Windows](#).

Basics

Getting Started

Open the [Welcome](#) page to get started with the basics of VS Code. [Help > Welcome](#).

IntelliSense

ctrl+Space to trigger the Suggestions widget.

```
18 // Routes  
19  
20 // Retrieve all todos  
21 server.get('/todos', function(req, res, next){  
  database.getAll().then(function(todos){  
    ...  
    ...});  
  });  
25  
26  
27 // Add a new todo  
28 server.post('/todos', function(req, res, next){  
  var todo = req.body;  
  database.add(todo).then(function(todos){  
    res.send(todos);  
    next();  
  });  
33});
```

You can view available methods, parameter hints, short documentation, etc.

Peek

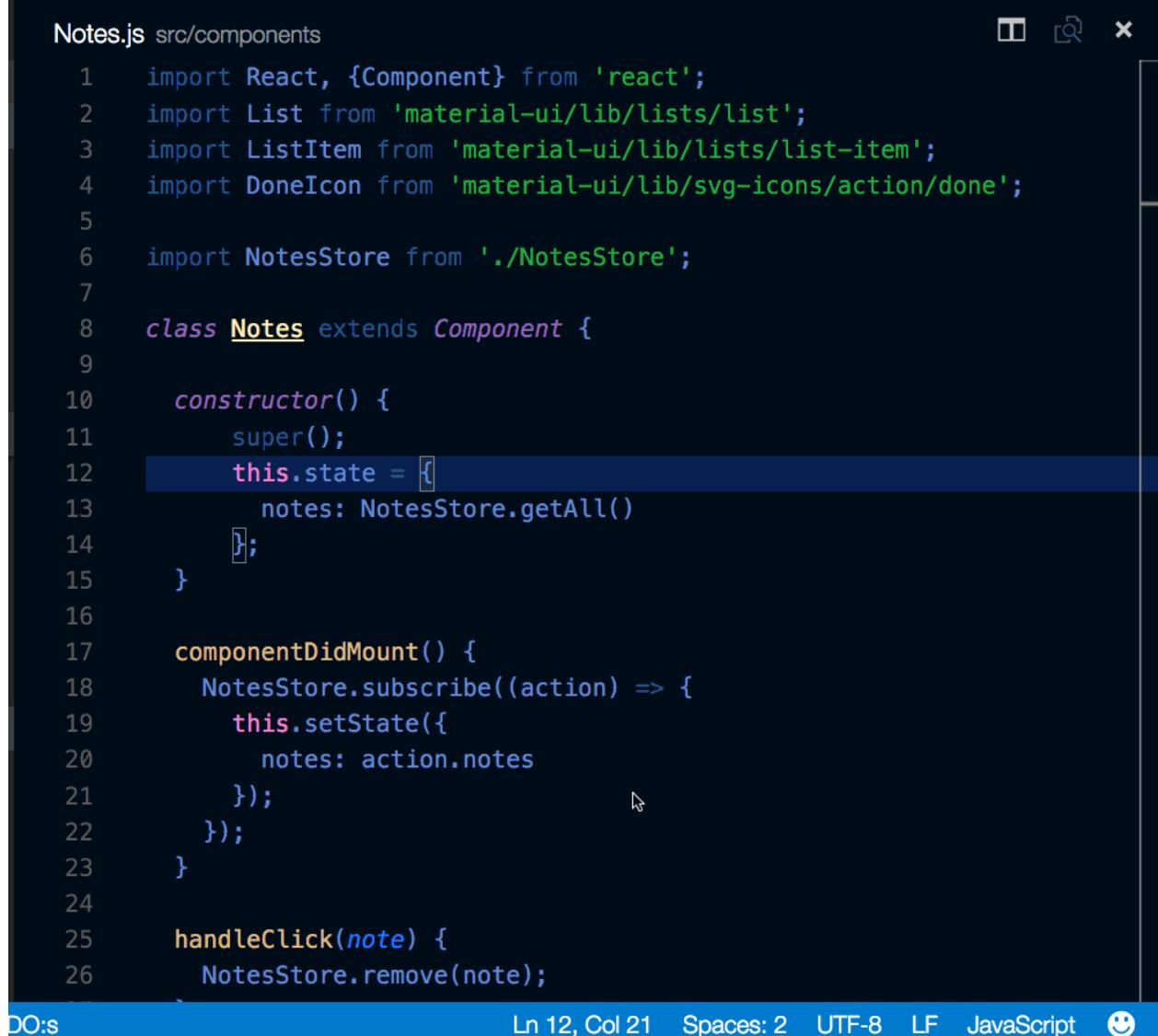
Select a symbol then type Alt+F12. Alternatively, you can use the context menu.

```
22    });  
23  }  
24  
25  handleClick(note) {  
26    NotesStore.remove(note);  
27  }  
28  
29  create(note) {  
30    return (<ListItem onMouseDown={this.handleClick.bind(null, note)}  
31              {note.text}  
32              </ListItem>  
33    );  
34  }  
35  
36  render() {  
37    const todos = this.state.notes.map(this.create.bind(this));  
38    return (  
      <List style={{width: 400}}>
```

Go to Definition

Select a symbol then type `F12`. Alternatively, you can use the context menu or `Ctrl+click` (`Cmd+click` on macOS).

↳ Notes.js - todo



```
1 import React, {Component} from 'react';
2 import List from 'material-ui/lib/lists/list';
3 import ListItem from 'material-ui/lib/lists/list-item';
4 import DoneIcon from 'material-ui/lib/svg-icons/action/done';
5
6 import NotesStore from './NotesStore';
7
8 class Notes extends Component {
9
10    constructor() {
11        super();
12        this.state = [
13            notes: NotesStore.getAll()
14        ];
15    }
16
17    componentDidMount() {
18        NotesStore.subscribe((action) => {
19            this.setState({
20                notes: action.notes
21            });
22        });
23    }
24
25    handleClick(note) {
26        NotesStore.remove(note);
27    }
28}
```

DO:s

Ln 12, Col 21 Spaces: 2 UTF-8 LF JavaScript 😊

You can go back to your previous location with the `Go > Back` command or `Alt+Left`.

You can also see the type definition if you press `Ctrl` (`Cmd` on macOS) when you are hovering over the type.

Peek References

Select a symbol then type `Shift+F12`. Alternatively, you can use the context menu.

```
5
6  import NotesStore from './NotesStore';
7
8  class Todo extends Component {
9
10    constructor() {
11      super();
12      this.state = {
13        notes: NotesStore.getAll()
14      };
15    }
16
17    componentDidMount() {
18      NotesStore.subscribe((action) => {
19        this.setState({
20          notes: action.notes
21        });
22      });
23    }
24
25    handleClick(note) {
26      NotesStore.remove(note);
27    }
28
29    create(note) {
30      return (<ListItem onMouseDown={this.handleClick.bind(null, note)} key={note.id}>
31        {note.text}
32      </ListItem>
33    );
34  }
35}
```

O:s

Ln 8, Col 10 Spaces: 2 UTF-8 LF JavaScript 😊

Find All References view

Select a symbol then type `Shift+Alt+F12` to open the References view showing all your file's symbols in a dedicated view.

Rename Symbol

Select a symbol then type `F2`. Alternatively, you can use the context menu.

```
7
8  class Notes extends Component {  ↵
9
10     constructor() {
11         super();
12         this.state = {
13             notes: NotesStore.getAll()
14         };
15     }
16
17     componentDidMount() {
18         NotesStore.subscribe((action) => {
19             this.setState({
20                 notes: action.notes
21             });
22         });
23     }
24
25     handleClick(note) {
26         NotesStore.remove(note);
27     }
28
29     create(note) {
30         return (<ListItem onMouseDown={this.handleClick.bind(null, note)} key={note.id}
31                 {note.text}
32             </ListItem>
33         );
34     }
35
36     render() {
37         return (
38             <div>
39                 {this.state.notes.map(this.create)}
40             </div>
41         );
42     }
43 }
```

10

Ln 16, Col 1 Spaces: 2 UTF-8 LF JavaScript

Search and modify

Besides searching and replacing expressions, you can also search and reuse parts of what was matched, using regular expressions with capturing groups. Enable regular expressions in the search box by clicking the **Use Regular Expression** `.*` button (Alt+R) and then write a regular expression and use parenthesis to define groups. You can then reuse the content matched in each group by using `$1`, `$2`, etc. in the Replace field.

```
0 - root
1 - bin
2 - daemon
3 - adm
4 - lp
5 - sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/sbin/nologin
```

.eslintrc.json

Install the ESLint extension (<https://marketplace.visualstudio.com/items?itemName=dbaeumer.vscode-eslint>). Configure your linter however you'd like. Consult the ESLint specification (<https://eslint.org/docs/user-guide/configuring>) for details on its linting rules and options.

Here is configuration to use ES6.

```
{  
  "env": {  
    "browser": true,  
    "commonjs": true,  
    "es6": true,  
    "node": true  
  },  
  "parserOptions": {  
    "ecmaVersion": 6,  
    "sourceType": "module",  
    "ecmaFeatures": {  
      "jsx": true,  
      "classes": true,  
      "defaultParams": true  
    }  
  },  
  "rules": {  
    "no-const-assign": 1,  
    "no-extra-semi": 0,  
    "semi": 0,  
    "no-fallthrough": 0,  
    "no-empty": 0,  
    "no-mixed-spaces-and-tabs": 0,  
    "no-redeclare": 0,  
    "no-this-before-super": 1,  
    "no-undef": 1,  
    "no-unreachable": 1,  
    "no-use-before-define": 0,  
    "constructor-super": 1,  
    "curly": 0,  
    "eqeqeq": 0,  
    "func-names": 0,  
    "valid-typeof": 1  
  }  
}
```

package.json

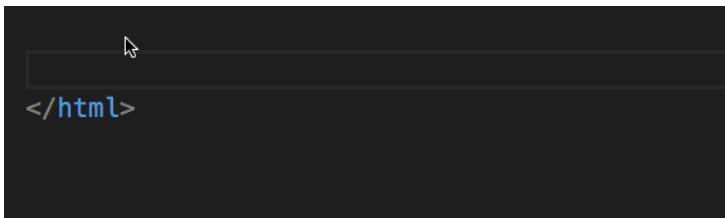
See IntelliSense for your `package.json` file.

package.json

```
1  {  
2    "name": "react-todo",  
3    "version": "1.0.0",  
4    "description": "Simple todo application to demo react and es6",  
5    "main": "webpack.config.js",  
6    "scripts": {  
7      "build": "webpack --config webpack.config.js",  
8      "dev": "webpack-dev-server --config webpack.config.js --op  
9    },  
10   "author": "waderyan",  
11   "license": "ISC",  
12   "devDependencies": {  
13     "babel-core": "^6.5.2",  
14     "babel-loader": "^6.2.3",  
15     "babel-preset-es2015": "^6.5.0",  
16     "babel-preset-react": "^6.5.0",  
17     "babel-preset-stage-0": "^6.5.0",  
18     "html-webpack-plugin": "^2.8.1",  
19     "node-sass": "^3.4.2",  
20     "sass-loader": "^3.1.2",  
21     "style-loader": "^0.13.0",  
22     "webpack": "^1.12.13",  
23     "webpack-dev-server": "^1.14.1"  
24   }  
}
```

Emmet syntax

Support for Emmet syntax ([/docs/editor/emmet](#)).



Snippets

Create custom snippets

File > Preferences > User Snippets (Code > Preferences > User Snippets on macOS), select the language, and create a snippet.

```
"create component": {  
  "prefix": "component",  
  "body": [  
    "class $1 extends React.Component {",  
    "",  
    "\trender() {",  
    "\t\treturn ($2);",  
    "\t}",  
    "",  
    "}"  
  ]  
},
```

See more details in [Creating your own Snippets](#) (/docs/editor/userdefinedsnippets).

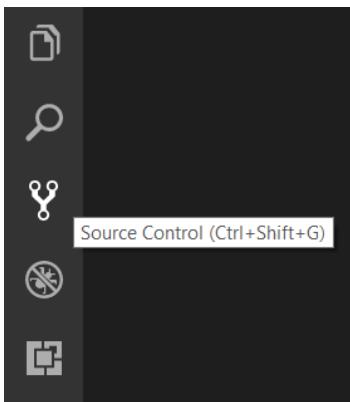
Git integration

Keyboard Shortcut: **Ctrl+Shift+G**

Git integration comes with VS Code "out-of-the-box". You can install other SCM providers from the extension Marketplace. This section describes the Git integration but much of the UI and gestures are shared by other SCM providers.

Diffs

From the **Source Control** view, select the file to diff.



Side by side

Default is side by side diff.

```

172 # Debugging
173
174 # Task Runner
175
176 ## Auto detect tasks
177
178 ## Gulp tasks
179
180 ## Mocha
181
182 # Git Integration
183
184 Excellent integration for entire Git workflow.
185
186 ## Diffs
187
188 ## Branches
189
190 ## Staging
191
192 ## Committing
193
194 ## See Git output
195
196 Sometimes I want to see what my tool is doing. V
197 what git commands are running. This is helpful w
198
199
200
201
202
203
204 Sometimes I want to see what my tool is doing. V
205 what git commands are running. This is helpful w

```

Ln 188, Col 1 Tab Size: 4 UTF-8 LF Markdown ☺

Inline view

Toggle inline view by clicking the More Actions (...) button in the top right and selecting Switch to Inline View.

```

11 11 class App extends Component {
12 12
13 13   constructor() {
14 14     super();
15 15     this.state = {
16 16       todos: [
17 17         {
18 18           text: 'Go to the grocery store'
19 19         },
20 20         {
21 21           text: 'Book flight for build'
22 22         },
23 23         {
24 24           text: 'Book flight for build'
25 25           text: 'Write a cool, new app'
26 26         },
27 27       ],
28 28     };
29 29
30 30     getChildContext(){
31 31       return {

```

If you prefer the inline view, you can set "diffEditor.renderSideBySide": false.

Review pane

Navigate through diffs with `F7` and `Shift+F7`. This will present them in a unified patch format. Lines can be navigated with arrow keys and pressing `Enter` will jump back in the diff editor and the selected line.

```
1/2: @@ -7,10 +7,13 @@
 7      7  Version: 0.5
 8      8  */
 9      9
10     10 +// Here are some inserted lines
11     11 +// with some extra comments
12     12 +...
13     13 (function (global, undefined) {
14     14   'use strict';
15     15   undefinedVariable = {};
16     16   undefinedVariable.prop = 5;
17     17
18     18   function initializeProperties(target, members) {
19     19     var keys = Object.keys(members);
```

Edit pending changes

You can make edits directly in the pending changes of the diff view.

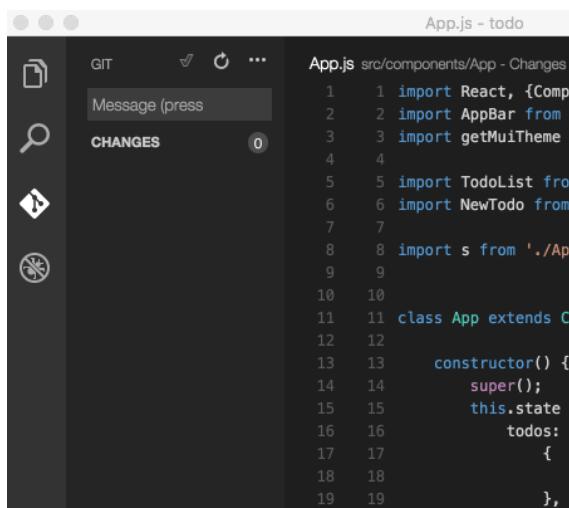
Branches

Easily switch between Git branches via the Status Bar.

```
App.js - todo
GIT
App.js src/components/App - Changes on index
1 1 import React, {Component, PropTypes} from 'react';
2 2 import AppBar from 'material-ui/lib/app-bar';
3 3 import getMuiTheme from 'material-ui/lib/styles/getMuiTheme';
4 4
5 5 import TodoList from '../TodoList';
6 6 import NewTodo from '../NewTodo';
7 7
8 8 import s from './App.scss';
9 9
10 10
11 11 class App extends Component {
12 12
13 13   constructor() {
14 14     super();
15 15     this.state = {
16 16       todos: [
17 17         {
18 18           text: 'Book flight for build'
19 19         },
20 20         {
21 21           text: 'Write a cool, new app'
22 22         },
23 23         {
24 24           text: 'Create a react todo list app'
25 25         }
26 26     ]
27 27   }
28 28 }
29 29
30 30
31 31
32 32
33 33
34 34
35 35
36 36
37 37
38 38
39 39
40 40
41 41
42 42
43 43
44 44
45 45
46 46
47 47
48 48
49 49
50 50
51 51
52 52
53 53
54 54
55 55
56 56
57 57
58 58
59 59
60 60
61 61
62 62
63 63
64 64
65 65
66 66
67 67
68 68
69 69
70 70
71 71
72 72
73 73
74 74
75 75
76 76
77 77
78 78
79 79
80 80
81 81
82 82
83 83
84 84
85 85
86 86
87 87
88 88
89 89
90 90
91 91
92 92
93 93
94 94
95 95
96 96
97 97
98 98
99 99
100 100
101 101
102 102
103 103
104 104
105 105
106 106
107 107
108 108
109 109
110 110
111 111
112 112
113 113
114 114
115 115
116 116
117 117
118 118
119 119
120 120
121 121
122 122
123 123
124 124
125 125
126 126
127 127
128 128
129 129
130 130
131 131
132 132
133 133
134 134
135 135
136 136
137 137
138 138
139 139
140 140
141 141
142 142
143 143
144 144
145 145
146 146
147 147
148 148
149 149
150 150
151 151
152 152
153 153
154 154
155 155
156 156
157 157
158 158
159 159
160 160
161 161
162 162
163 163
164 164
165 165
166 166
167 167
168 168
169 169
170 170
171 171
172 172
173 173
174 174
175 175
176 176
177 177
178 178
179 179
180 180
181 181
182 182
183 183
184 184
185 185
186 186
187 187
188 188
189 189
190 190
191 191
192 192
193 193
194 194
195 195
196 196
197 197
198 198
199 199
200 200
201 201
202 202
203 203
204 204
205 205
206 206
207 207
208 208
209 209
210 210
211 211
212 212
213 213
214 214
215 215
216 216
217 217
218 218
219 219
220 220
221 221
222 222
223 223
224 224
225 225
226 226
227 227
228 228
229 229
230 230
231 231
232 232
233 233
234 234
235 235
236 236
237 237
238 238
239 239
240 240
241 241
242 242
243 243
244 244
245 245
246 246
247 247
248 248
249 249
250 250
251 251
252 252
253 253
254 254
255 255
256 256
257 257
258 258
259 259
260 260
261 261
262 262
263 263
264 264
265 265
266 266
267 267
268 268
269 269
270 270
271 271
272 272
273 273
274 274
275 275
276 276
277 277
278 278
279 279
280 280
281 281
282 282
283 283
284 284
285 285
286 286
287 287
288 288
289 289
290 290
291 291
292 292
293 293
294 294
295 295
296 296
297 297
298 298
299 299
300 300
301 301
302 302
303 303
304 304
305 305
306 306
307 307
308 308
309 309
310 310
311 311
312 312
313 313
314 314
315 315
316 316
317 317
318 318
319 319
320 320
321 321
322 322
323 323
324 324
325 325
326 326
327 327
328 328
329 329
330 330
331 331
332 332
333 333
334 334
335 335
336 336
337 337
338 338
339 339
340 340
341 341
342 342
343 343
344 344
345 345
346 346
347 347
348 348
349 349
350 350
351 351
352 352
353 353
354 354
355 355
356 356
357 357
358 358
359 359
360 360
361 361
362 362
363 363
364 364
365 365
366 366
367 367
368 368
369 369
370 370
371 371
372 372
373 373
374 374
375 375
376 376
377 377
378 378
379 379
380 380
381 381
382 382
383 383
384 384
385 385
386 386
387 387
388 388
389 389
390 390
391 391
392 392
393 393
394 394
395 395
396 396
397 397
398 398
399 399
400 400
401 401
402 402
403 403
404 404
405 405
406 406
407 407
408 408
409 409
410 410
411 411
412 412
413 413
414 414
415 415
416 416
417 417
418 418
419 419
420 420
421 421
422 422
423 423
424 424
425 425
426 426
427 427
428 428
429 429
430 430
431 431
432 432
433 433
434 434
435 435
436 436
437 437
438 438
439 439
440 440
441 441
442 442
443 443
444 444
445 445
446 446
447 447
448 448
449 449
450 450
451 451
452 452
453 453
454 454
455 455
456 456
457 457
458 458
459 459
460 460
461 461
462 462
463 463
464 464
465 465
466 466
467 467
468 468
469 469
470 470
471 471
472 472
473 473
474 474
475 475
476 476
477 477
478 478
479 479
480 480
481 481
482 482
483 483
484 484
485 485
486 486
487 487
488 488
489 489
490 490
491 491
492 492
493 493
494 494
495 495
496 496
497 497
498 498
499 499
500 500
501 501
502 502
503 503
504 504
505 505
506 506
507 507
508 508
509 509
510 510
511 511
512 512
513 513
514 514
515 515
516 516
517 517
518 518
519 519
520 520
521 521
522 522
523 523
524 524
525 525
526 526
527 527
528 528
529 529
530 530
531 531
532 532
533 533
534 534
535 535
536 536
537 537
538 538
539 539
540 540
541 541
542 542
543 543
544 544
545 545
546 546
547 547
548 548
549 549
550 550
551 551
552 552
553 553
554 554
555 555
556 556
557 557
558 558
559 559
560 560
561 561
562 562
563 563
564 564
565 565
566 566
567 567
568 568
569 569
570 570
571 571
572 572
573 573
574 574
575 575
576 576
577 577
578 578
579 579
580 580
581 581
582 582
583 583
584 584
585 585
586 586
587 587
588 588
589 589
590 590
591 591
592 592
593 593
594 594
595 595
596 596
597 597
598 598
599 599
600 600
601 601
602 602
603 603
604 604
605 605
606 606
607 607
608 608
609 609
610 610
611 611
612 612
613 613
614 614
615 615
616 616
617 617
618 618
619 619
620 620
621 621
622 622
623 623
624 624
625 625
626 626
627 627
628 628
629 629
630 630
631 631
632 632
633 633
634 634
635 635
636 636
637 637
638 638
639 639
640 640
641 641
642 642
643 643
644 644
645 645
646 646
647 647
648 648
649 649
650 650
651 651
652 652
653 653
654 654
655 655
656 656
657 657
658 658
659 659
660 660
661 661
662 662
663 663
664 664
665 665
666 666
667 667
668 668
669 669
670 670
671 671
672 672
673 673
674 674
675 675
676 676
677 677
678 678
679 679
680 680
681 681
682 682
683 683
684 684
685 685
686 686
687 687
688 688
689 689
690 690
691 691
692 692
693 693
694 694
695 695
696 696
697 697
698 698
699 699
700 700
701 701
702 702
703 703
704 704
705 705
706 706
707 707
708 708
709 709
710 710
711 711
712 712
713 713
714 714
715 715
716 716
717 717
718 718
719 719
720 720
721 721
722 722
723 723
724 724
725 725
726 726
727 727
728 728
729 729
730 730
731 731
732 732
733 733
734 734
735 735
736 736
737 737
738 738
739 739
740 740
741 741
742 742
743 743
744 744
745 745
746 746
747 747
748 748
749 749
750 750
751 751
752 752
753 753
754 754
755 755
756 756
757 757
758 758
759 759
760 760
761 761
762 762
763 763
764 764
765 765
766 766
767 767
768 768
769 769
770 770
771 771
772 772
773 773
774 774
775 775
776 776
777 777
778 778
779 779
780 780
781 781
782 782
783 783
784 784
785 785
786 786
787 787
788 788
789 789
790 790
791 791
792 792
793 793
794 794
795 795
796 796
797 797
798 798
799 799
800 800
801 801
802 802
803 803
804 804
805 805
806 806
807 807
808 808
809 809
810 810
811 811
812 812
813 813
814 814
815 815
816 816
817 817
818 818
819 819
820 820
821 821
822 822
823 823
824 824
825 825
826 826
827 827
828 828
829 829
830 830
831 831
832 832
833 833
834 834
835 835
836 836
837 837
838 838
839 839
840 840
841 841
842 842
843 843
844 844
845 845
846 846
847 847
848 848
849 849
850 850
851 851
852 852
853 853
854 854
855 855
856 856
857 857
858 858
859 859
860 860
861 861
862 862
863 863
864 864
865 865
866 866
867 867
868 868
869 869
870 870
871 871
872 872
873 873
874 874
875 875
876 876
877 877
878 878
879 879
880 880
881 881
882 882
883 883
884 884
885 885
886 886
887 887
888 888
889 889
890 890
891 891
892 892
893 893
894 894
895 895
896 896
897 897
898 898
899 899
900 900
901 901
902 902
903 903
904 904
905 905
906 906
907 907
908 908
909 909
910 910
911 911
912 912
913 913
914 914
915 915
916 916
917 917
918 918
919 919
920 920
921 921
922 922
923 923
924 924
925 925
926 926
927 927
928 928
929 929
930 930
931 931
932 932
933 933
934 934
935 935
936 936
937 937
938 938
939 939
940 940
941 941
942 942
943 943
944 944
945 945
946 946
947 947
948 948
949 949
950 950
951 951
952 952
953 953
954 954
955 955
956 956
957 957
958 958
959 959
960 960
961 961
962 962
963 963
964 964
965 965
966 966
967 967
968 968
969 969
970 970
971 971
972 972
973 973
974 974
975 975
976 976
977 977
978 978
979 979
980 980
981 981
982 982
983 983
984 984
985 985
986 986
987 987
988 988
989 989
990 990
991 991
992 992
993 993
994 994
995 995
996 996
997 997
998 998
999 999
1000 1000
1001 1001
1002 1002
1003 1003
1004 1004
1005 1005
1006 1006
1007 1007
1008 1008
1009 1009
1010 1010
1011 1011
1012 1012
1013 1013
1014 1014
1015 1015
1016 1016
1017 1017
1018 1018
1019 1019
1020 1020
1021 1021
1022 1022
1023 1023
1024 1024
1025 1025
1026 1026
1027 1027
1028 1028
1029 1029
1030 1030
1031 1031
1032 1032
1033 1033
1034 1034
1035 1035
1036 1036
1037 1037
1038 1038
1039 1039
1040 1040
1041 1041
1042 1042
1043 1043
1044 1044
1045 1045
1046 1046
1047 1047
1048 1048
1049 1049
1050 1050
1051 1051
1052 1052
1053 1053
1054 1054
1055 1055
1056 1056
1057 1057
1058 1058
1059 1059
1060 1060
1061 1061
1062 1062
1063 1063
1064 1064
1065 1065
1066 1066
1067 1067
1068 1068
1069 1069
1070 1070
1071 1071
1072 1072
1073 1073
1074 1074
1075 1075
1076 1076
1077 1077
1078 1078
1079 1079
1080 1080
1081 1081
1082 1082
1083 1083
1084 1084
1085 1085
1086 1086
1087 1087
1088 1088
1089 1089
1090 1090
1091 1091
1092 1092
1093 1093
1094 1094
1095 1095
1096 1096
1097 1097
1098 1098
1099 1099
1100 1100
1101 1101
1102 1102
1103 1103
1104 1104
1105 1105
1106 1106
1107 1107
1108 1108
1109 1109
1110 1110
1111 1111
1112 1112
1113 1113
1114 1114
1115 1115
1116 1116
1117 1117
1118 1118
1119 1119
1120 1120
1121 1121
1122 1122
1123 1123
1124 1124
1125 1125
1126 1126
1127 1127
1128 1128
1129 1129
1130 1130
1131 1131
1132 1132
1133 1133
1134 1134
1135 1135
1136 1136
1137 1137
1138 1138
1139 1139
1140 1140
1141 1141
1142 1142
1143 1143
1144 1144
1145 1145
1146 1146
1147 1147
1148 1148
1149 1149
1150 1150
1151 1151
1152 1152
1153 1153
1154 1154
1155 1155
1156 1156
1157 1157
1158 1158
1159 1159
1160 1160
1161 1161
1162 1162
1163 1163
1164 1164
1165 1165
1166 1166
1167 1167
1168 1168
1169 1169
1170 1170
1171 1171
1172 1172
1173 1173
1174 1174
1175 1175
1176 1176
1177 1177
1178 1178
1179 1179
1180 1180
1181 1181
1182 1182
1183 1183
1184 1184
1185 1185
1186 1186
1187 1187
1188 1188
1189 1189
1190 1190
1191 1191
1192 1192
1193 1193
1194 1194
1195 1195
1196 1196
1197 1197
1198 1198
1199 1199
1200 1200
1201 1201
1202 1202
1203 1203
1204 1204
1205 1205
1206 1206
1207 1207
1208 1208
1209 1209
1210 1210
1211 1211
1212 1212
1213 1213
1214 1214
1215 1215
1216 1216
1217 1217
1218 1218
1219 1219
1220 1220
1221 1221
1222 1222
1223 1223
1224 1224
1225 1225
1226 1226
1227 1227
1228 1228
1229 1229
1230 1230
1231 1231
1232 1232
1233 1233
1234 1234
1235 1235
1236 1236
1237 1237
1238 1238
1239 1239
1240 1240
1241 1241
1242 1242
1243 1243
1244 1244
1245 1245
1246 1246
1247 1247
1248 1248
1249 1249
1250 1250
1251 1251
1252 1252
1253 1253
1254 1254
1255 1255
1256 1256
1257 1257
1258 1258
1259 1259
1260 1260
1261 1261
1262 1262
1263 1263
1264 1264
1265 1265
1266 1266
1267 1267
1268 1268
1269 1269
1270 1270
1271 1271
1272 1272
1273 1273
1274 1274
1275 1275
1276 1
```

Stage a portion of a file by selecting that file (using the arrows) and then choosing **Stage Selected Ranges** from the Command Palette.

Undo last commit



App.js - todo

```
1 import React, {Component} from 'react';
2 import AppBar from 'material-ui/AppBar';
3 import getMuiTheme from 'material-ui/getMuiTheme';
4 import TodoList from './TodoList';
5 import NewTodo from './NewTodo';
6 import s from './App.css';
7
8 class App extends Component {
9   constructor() {
10     super();
11     this.state = {
12       todos: []
13     };
14   }
15
16   render() {
17     return (
18       <div>
19         <AppBar title="To-Do List" theme={getMuiTheme()} />
20         <TodoList todos={this.state.todos} />
21         <NewTodo onAdd={this.addTodo} />
22       </div>
23     );
24   }
25
26   addTodo(text) {
27     const newTodos = this.state.todos.concat([text]);
28     this.setState({ todos: newTodos });
29   }
30
31   renderTodos(todos) {
32     return todos.map((todo, index) =>
33       <div key={index}>
34         <input checked="checked" type="checkbox" />
35         <span>{todo}</span>
36       </div>
37     );
38   }
39
40   render() {
41     return (
42       <div>
43         <h1>To-Do List</h1>
44         <h2>Add a new todo</h2>
45         <input type="text" value="New Todo" />
46         <button>Add</button>
47         <hr />
48         <h3>Todos</h3>
49         <ul>
50           {this.renderTodos(this.state.todos)}
51         </ul>
52       </div>
53     );
54   }
55 }
56
57 export default App;
```

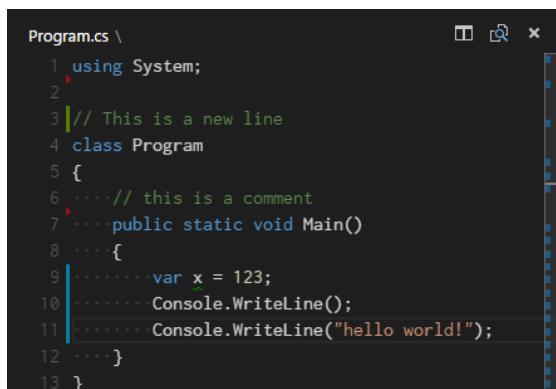
See Git output

VS Code makes it easy to see what Git commands are actually running. This is helpful when learning Git or debugging a difficult source control issue.

Use the **Toggle Output** command (**Ctrl+Shift+U**) and select **Git** in the drop-down.

Gutter indicators

View diff decorations in editor. See documentation ([/docs/editor/versioncontrol#gutter-indicators](#)) for more details.



```
Program.cs \
```

```
1 using System;
2
3 // This is a new line
4 class Program
5 {
6     // this is a comment
7     public static void Main()
8     {
9         var x = 123;
10        Console.WriteLine();
11        Console.WriteLine("Hello world!");
12    }
13 }
```

Resolve merge conflicts

During a merge, go to the **Source Control** view (**Ctrl+Shift+G**) and make changes in the diff view.

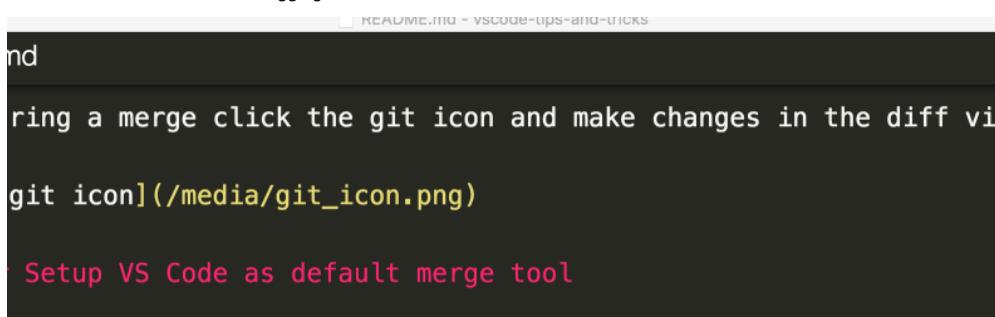
Set VS Code as default merge tool

```
git config --global merge.tool code
```

Debugging

Configure debugger

Open the **Command Palette** (**Ctrl+Shift+P**) and select **Debug: Open launch.json**, which will prompt you to select the environment that matches your project (Node.js, Python, C++, etc). This will generate a `launch.json` file. Node.js support is built-in and other environments require installing the appropriate language extensions. See the debugging documentation ([/docs/editor/debugging](#)) for more details.



```
REALMING - VS CODE TIPS AND TRICKS
```

```
nd
```

```
ring a merge click the git icon and make changes in the diff view
```

```
git icon] (/media/git_icon.png)
```

```
Setup VS Code as default merge tool
```

Breakpoints and stepping through

Place breakpoints next to the line number. Navigate forward with the Debug widget.

The screenshot shows the Visual Studio Code interface with the debugger open. The code editor displays a portion of `server.js` with a yellow highlight on line 20. A red dot and a yellow circle are positioned on line 20, indicating a breakpoint. The code on line 20 is `database.add(todo, function(todos) {`. The line numbers 20 and 21 are highlighted in red, while 22 and 23 are in yellow. The code editor has tabs for `DEBUG`, `Launch`, `Variables`, `Local`, `WATCH`, `CALL STACK`, and `BREAKPOINTS`. The `CALL STACK` shows the current stack trace with three entries: `(anonymous function)` at `server.js:20`, `handle` at `layer.js:95`, and `next` at `route.js:131`. The `BREAKPOINTS` section shows four checkboxes: `All Exceptions` (unchecked), `Uncaught Exceptions` (checked), `server.js:20` (checked), and `server.js:21` (unchecked). The `DEBUG CONSOLE` shows the command `node --debug-brk=35517 server.js` and the message `Debugger listening on port 35517`. The status bar at the bottom shows the file is `master*`, has `0` errors and `0` warnings, and is a `TODO:0` file. The status bar also indicates the code is in `Ln 20, Col 1` with `Spaces: 2`, `UTF-8`, `LF`, and `JavaScript`.

Data inspection

Inspect variables in the **Debug** panels and in the console.

The screenshot shows the Visual Studio Code interface with the following panels visible:

- DEBUG** panel: Shows the current file is `server.js` with the title `node-server`. The **VARIABLES** section shows local variables: `next`, `req`, `res`, and `this`. The `res.send(todos);` line is highlighted in yellow, indicating it is the current line of execution. Breakpoints are marked with red and yellow circles at lines 20 and 21.
- WATCH** panel: Not currently expanded.
- CALL STACK** panel: Shows the call stack: (anonymous function) at `server.js:21`, (anonymous function) at `database.js:21`, and (anonymous function) at `database.js:9`. The text **PAUSED ON BREAKPOINT** is displayed above the stack.
- BREAKPOINTS** panel: Shows the configuration for breakpoints. All Exceptions, Uncaught Exceptions, `server.js:20`, and `server.js:21`.
- DEBUG CONSOLE** panel: Not currently expanded.

Inline values

You can set `"debug.inlineValues": true` to see variable values inline in the debugger. This feature can be expensive and may slow down stepping, so it is disabled by default.

Task runner

Auto detect tasks

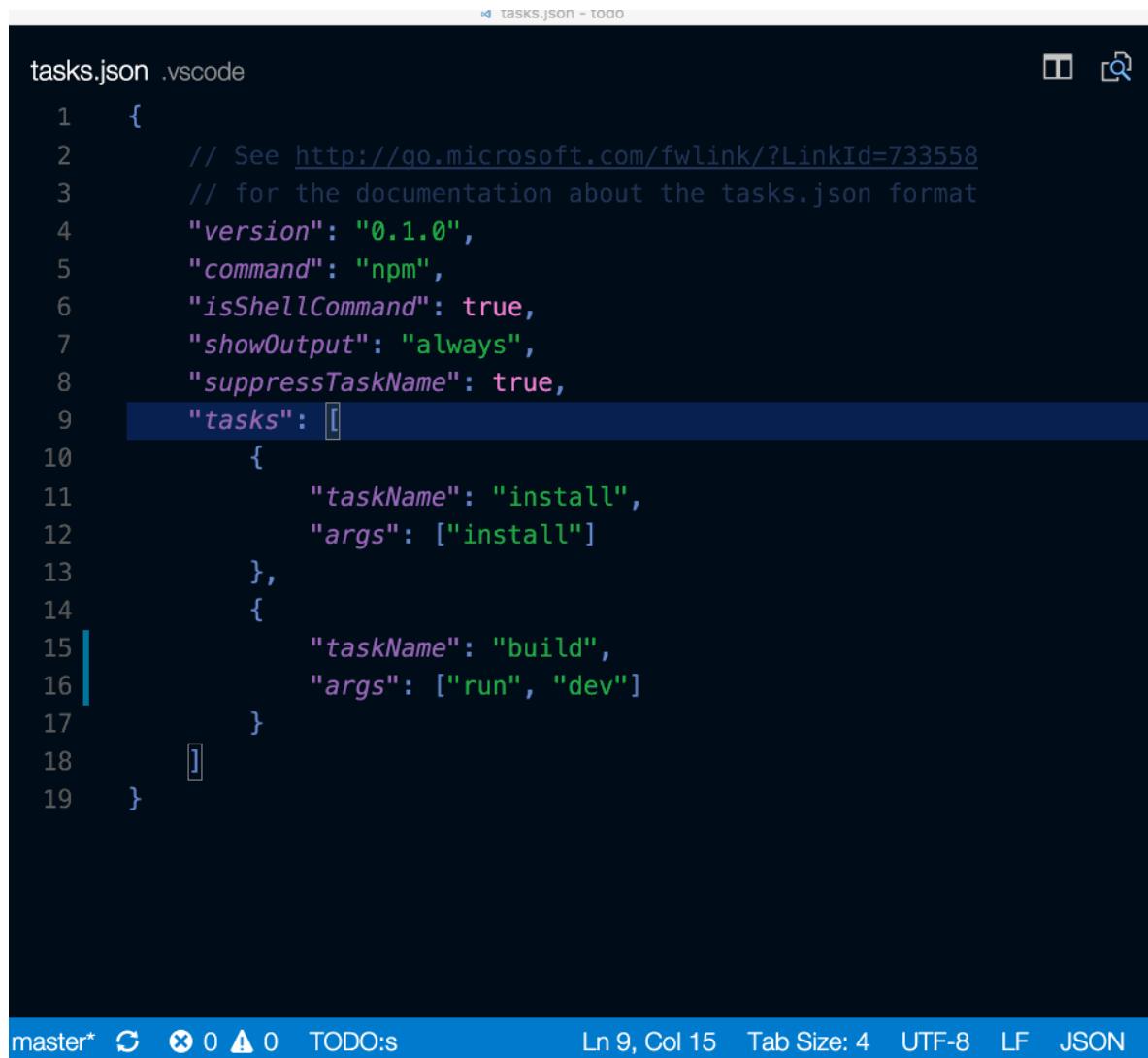
Select **Terminal** from the top-level menu, run the command **Configure Tasks**, then select the type of task you'd like to run. This will generate a `tasks.json` file with content like the following. See the [Tasks \(/docs/editor/tasks\)](#) documentation for more details.

```
{  
  // See https://go.microsoft.com/fwlink/?LinkId=733558  
  // for the documentation about the tasks.json format  
  "version": "2.0.0",  
  "tasks": [  
    {  
      "type": "npm",  
      "script": "install",  
      "group": {  
        "kind": "build",  
        "isDefault": true  
      }  
    }  
  ]  
}
```

There are occasionally issues with auto generation. Check out the documentation for getting things to work properly.

Run tasks from the Terminal menu

Select **Terminal** from the top-level menu, run the command **Run Task**, and select the task you want to run. Terminate the running task by running the command **Terminate Task**



```
tasks.json .vscode  
1  {  
2    // See http://go.microsoft.com/fwlink/?LinkId=733558  
3    // for the documentation about the tasks.json format  
4    "version": "0.1.0",  
5    "command": "npm",  
6    "isShellCommand": true,  
7    "showOutput": "always",  
8    "suppressTaskName": true,  
9    "tasks": [  
10      {  
11        "taskName": "install",  
12        "args": ["install"]  
13      },  
14      {  
15        "taskName": "build",  
16        "args": ["run", "dev"]  
17      }  
18    ]  
19  }
```

Define keyboard shortcuts for tasks

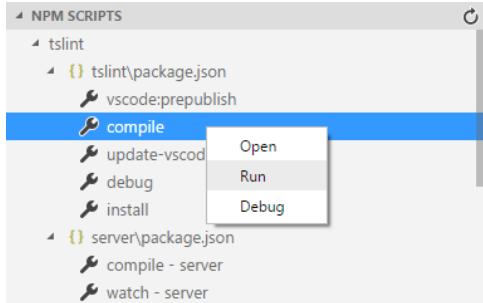
You can define a keyboard shortcut for any task. From the **Command Palette** (`Ctrl+Shift+P`), select **Preferences: Open Keyboard Shortcuts File**, bind the desired shortcut to the `workbench.action.tasks.runTask` command, and define the Task as `args`.

For example, to bind `Ctrl+H` to the `Run tests` task, add the following:

```
{
  "key": "ctrl+h",
  "command": "workbench.action.tasks.runTask",
  "args": "Run tests"
}
```

Run npm scripts as tasks from the explorer

With the setting `npm.enableScriptExplorer`, you can enable an explorer that shows the scripts defined in your workspace.



From the explorer you can open a script in the editor, run it as a task, and launch it with the node debugger (when the script defines a debug option like `--inspect-brk`). The default action on click is to open the script. To run a script on a single click, set `npm.scriptExplorerAction` to "run". Use the setting `npm.exclude` to exclude scripts in `package.json` files contained in particular folders.

Portable mode

VS Code has a Portable mode (/docs/editor/portable) which lets you keep settings and data in the same location as your installation, for example, on a USB drive.

Insiders builds

The Visual Studio Code team uses the Insiders version to test the latest features and bug fixes of VS Code. You can also use the Insiders version by downloading it here (/insiders).

- For Early Adopters - Insiders has the most recent code changes for users and extension authors to try out.
- Frequent Builds - New builds every day with the latest bug fixes and features.
- Side-by-side install - Insiders installs next to the Stable build allowing you to use either independently.

Was this documentation helpful?

7/3/2019

IN THIS ARTICLE

[Basics](#)
[Command line](#)
[Status Bar](#)
[Customization](#)
[Extensions](#)
[Files and folders](#)

{ Editing hacks

[IntelliSense](#)
[Snippets](#)
[Git integration](#)
[Debugging](#)
[Task runner](#)
[Portable mode](#)
[Insiders builds](#)

[Tweet\(](#)

[Subscribe\(\[/feed.xml\]\(https://code.visualstudio.com/_api/Feed.xml\)\)](#)

[Ask questions\(<https://stackoverflow.com/questions/tagged/vscode>\)](#)

[Follow @code\(<https://go.microsoft.com/fwlink/?LinkID=533687>\)](#)

[Request features\(<https://go.microsoft.com/fwlink/?LinkID=533482>\)](#)

[Report issues\(<https://github.com/Microsoft/vscode/issues>\)](#)

[Watch videos\(\[https://www.youtube.com/channel/UCs5Y5_7XK8HLDX0SLNwkd3w\]\(https://www.youtube.com/channel/UCs5Y5_7XK8HLDX0SLNwkd3w\)\)](#)

Hello from Seattle. Follow @code (<https://go.microsoft.com/fwlink/?LinkId=533687>)

[Star](#) 79,485

[Support](#) (https://support.microsoft.com/en-us/getsupport?wf=0&tenant=ClassicCommercial&oaspworkflow=start_1.0.0.0&locale=en-us&supportregion=en-us&pesid=16064&ccsid=636196895839595242)
[Privacy](#) (<https://privacy.microsoft.com/en-us/privacystatement>) [Terms of Use](#) (<https://www.microsoft.com/en-us/legal/intellectualproperty/copyright/default.aspx>) [License](#) (/License)



© 2019 Microsoft