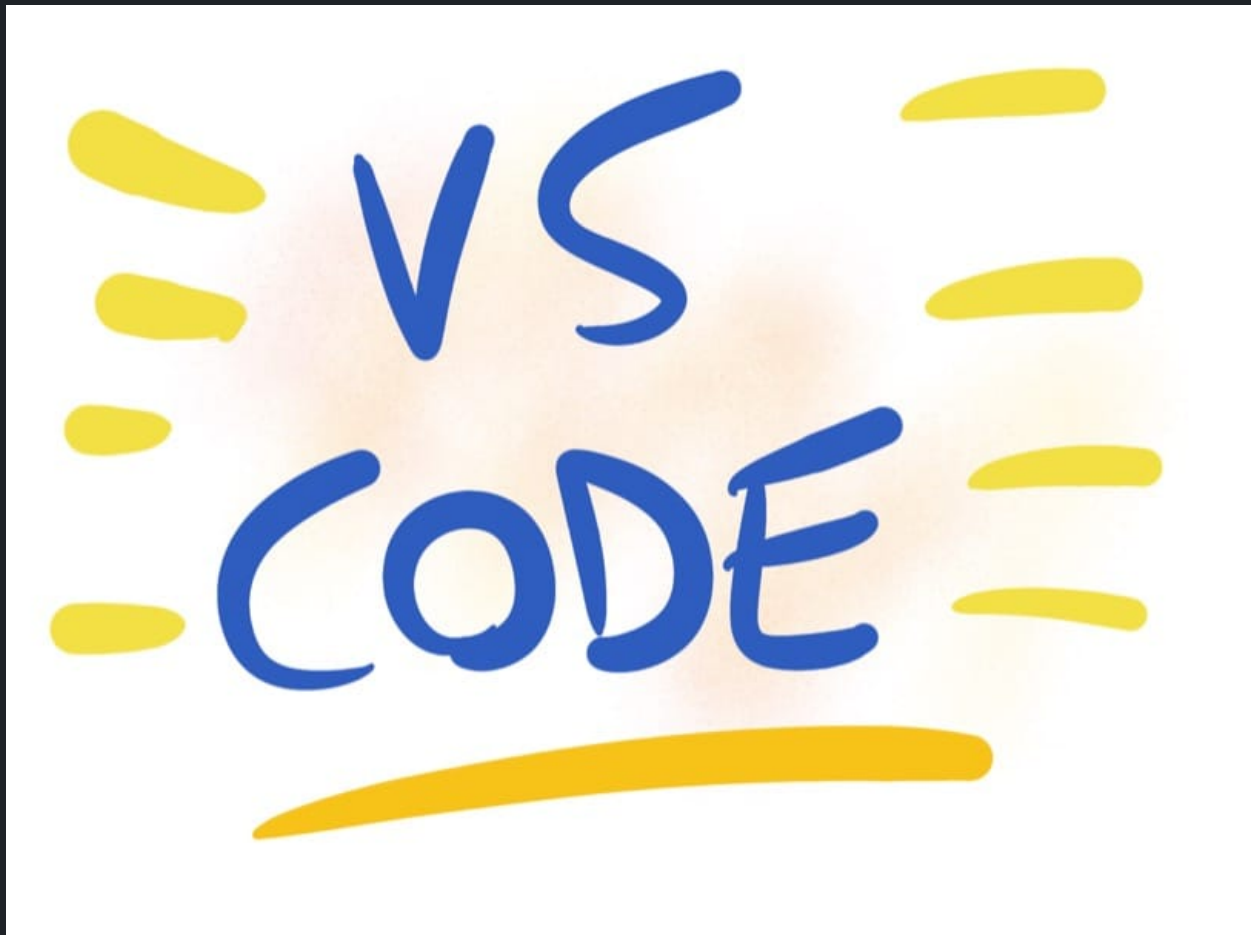


HOW TO USE VISUAL STUDIO CODE

Visual Studio Code, VSCode for friends, is an incredibly powerful editor that's hugely growing in popularity. Find out why, and its main features for developers

PUBLISHED MAY 31, 2018



INDEX

- [Index](#)
- [Introduction](#)
- [Should I switch to VS Code? And why?](#)
- [Getting started](#)
- [Explorer](#)
- [Search](#)
- [Source Control](#)
- [Debugger](#)
- [Extensions](#)
- [The Terminal](#)
- [The Command Palette](#)

- [Themes](#)
- [Customization](#)
- [Nice configuration options](#)
- [The best font for coding](#)
- [Workspaces](#)
- [Editing](#)
 - [IntelliSense](#)
 - [Code Formatting](#)
 - [Errors and warnings](#)
 - [Keyboard shortcuts](#)
 - [Keymaps](#)
- [Code snippets](#)
- [Extensions showcase](#)
- [The VS Code CLI command](#)
- [Solving high usage CPU issues](#)

INTRODUCTION

Since the beginning, editors are a strange beast. Some people defend their editor choice strenuously. In the Unix world you have those `Emacs` vs `vi` “wars”, and I kind of imagine *why* so much time is spent debating the advantages of one versus another.

I used tons of editors and IDEs in the past few years. I can remember TextMate, TextWrangler, Espresso, BBEdit, XCode, Coda, Brackets, Sublime Text, Atom, vim, PHPStorm. The difference between an IDE and an editor is mostly in the feature set, and complexity.

I largely prefer an editor over an IDE, as it’s faster and gets less in the way.

In the last 12 months I’ve been using VS Code, the Open Source editor from Microsoft, and it’s quickly become my favorite editor ever.

SHOULD I SWITCH TO VS CODE? AND WHY?

If you’re looking for suggestions for whether to use it or not, let me say **yes**, you should switch to it from whatever other editor you are using now.

This editor builds on top of decades of editor experience from Microsoft.

The code of the editor is completely Open Source, and there’s no payment required to use it.

It uses [Electron](#) as its base, which enables it to be cross platform and work on Mac, Windows and Linux. It’s built using Node.js, and you can extend it using JavaScript (which makes it a win for all us JavaScript developers).

It’s **fast**, easily the fastest editor I’ve used after Sublime Text.

It has won the enthusiasm of the community: there are thousands of **extensions**, some official, and some made by the community, and it’s [winning surveys](#).

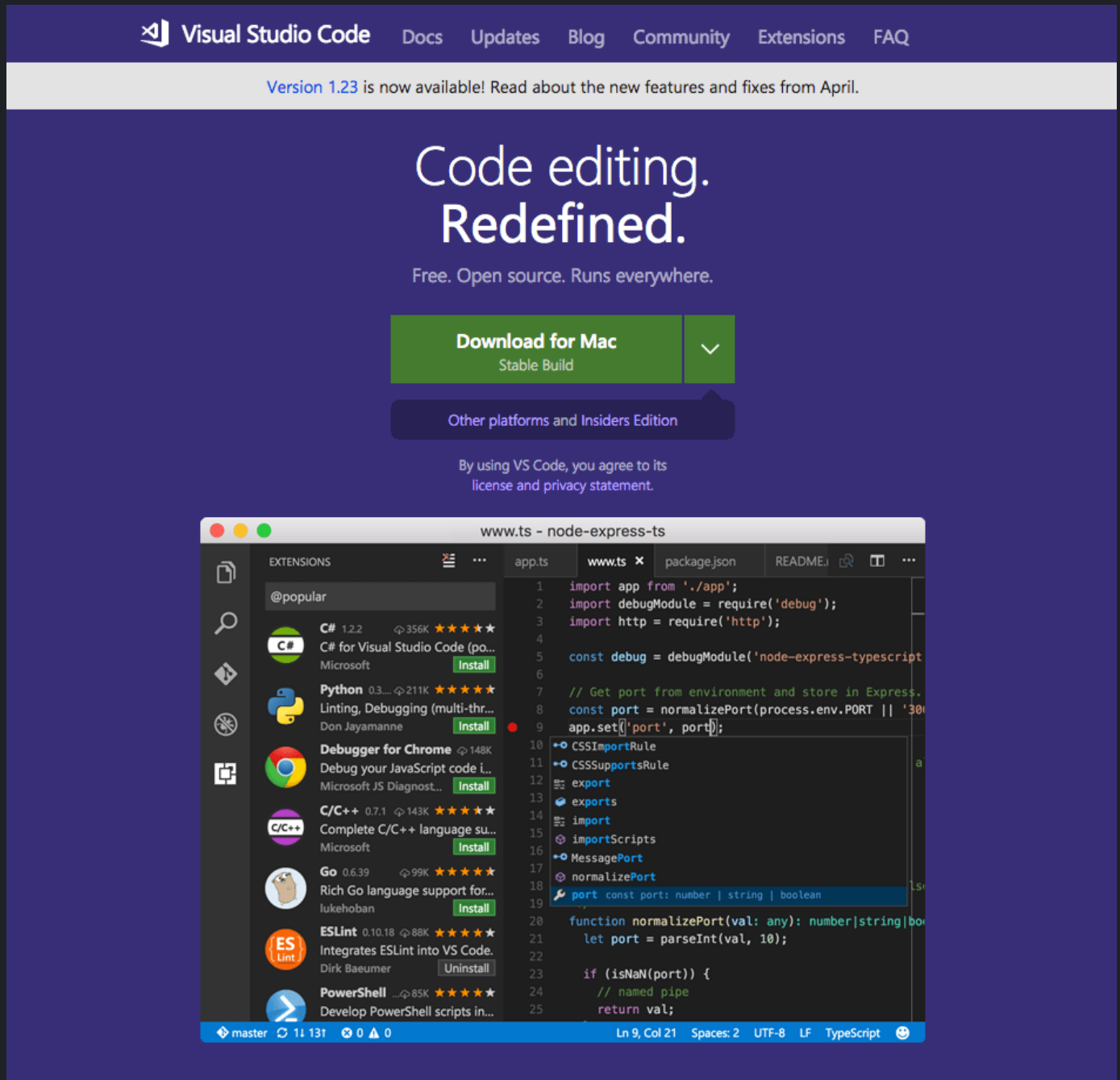
Microsoft releases an update every month. Frequent updates foster innovation and Microsoft is listening to its users, while keeping the platform as stable as possible (I should say I never had an issue with VS Code in 1 year of using it every day almost all

day).

GETTING STARTED

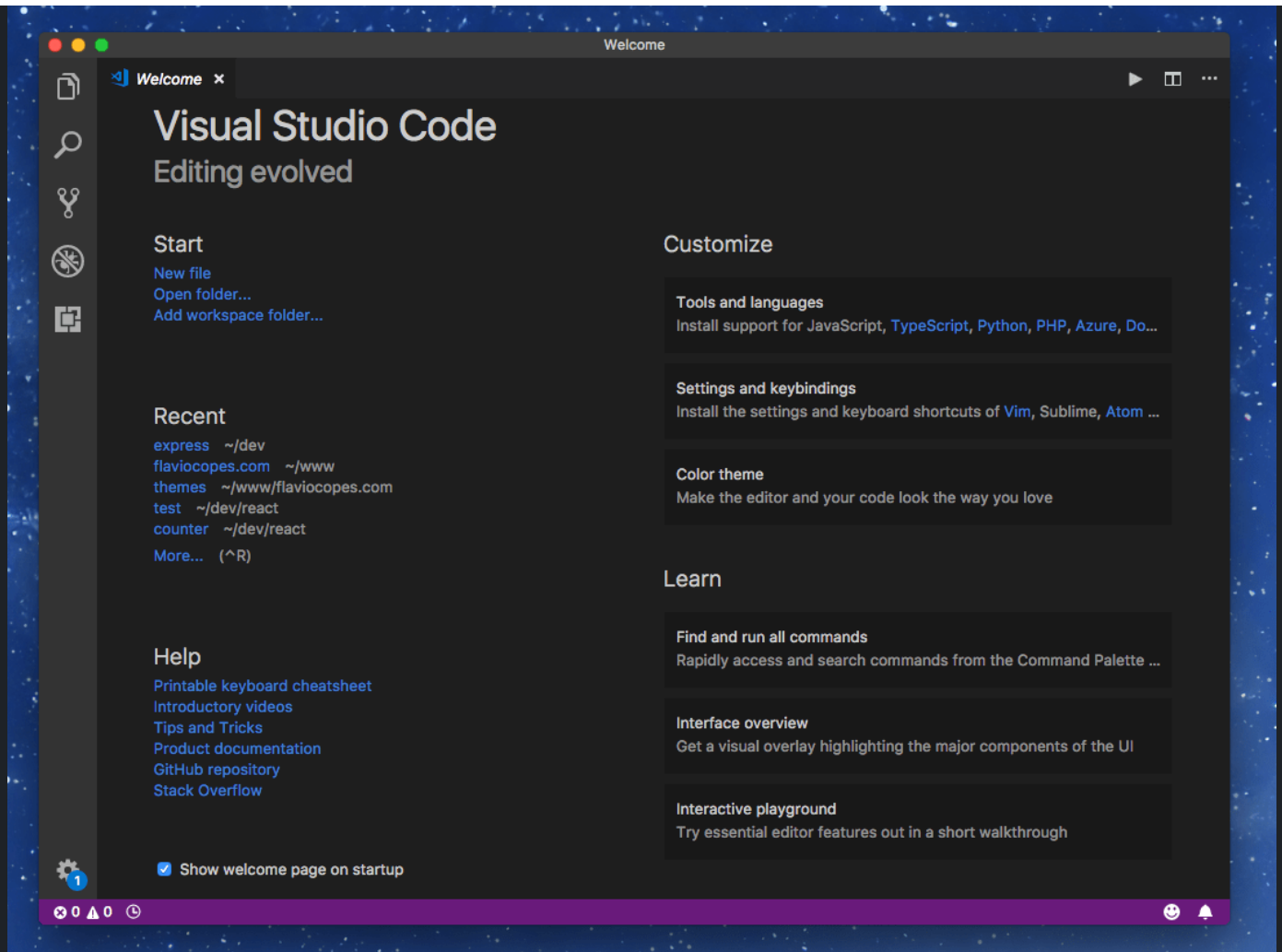
The home page of Visual Studio Code on the internet is <https://code.visualstudio.com/>.

Go to that site to download the latest stable release of the editor.



The installation process depends on the platform, and you should be used to it.

When you start the editor for the first time you will see the welcome screen:

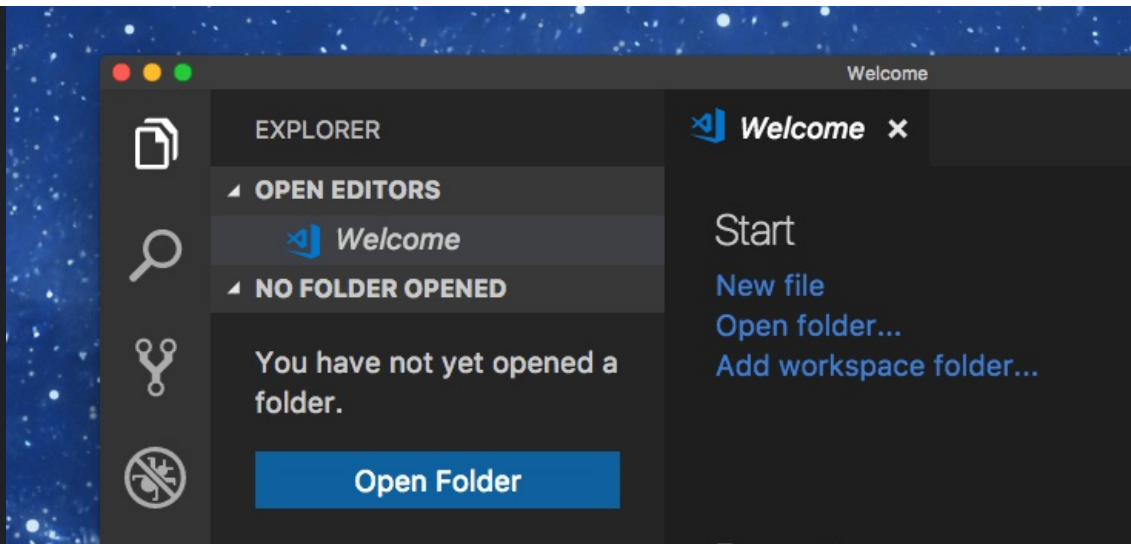


There is a toolbar on the left with 5 icons. That gives access to:

- The File Explorer
- Search
- Source Control
- The Debugger
- The Extensions

EXPLORER

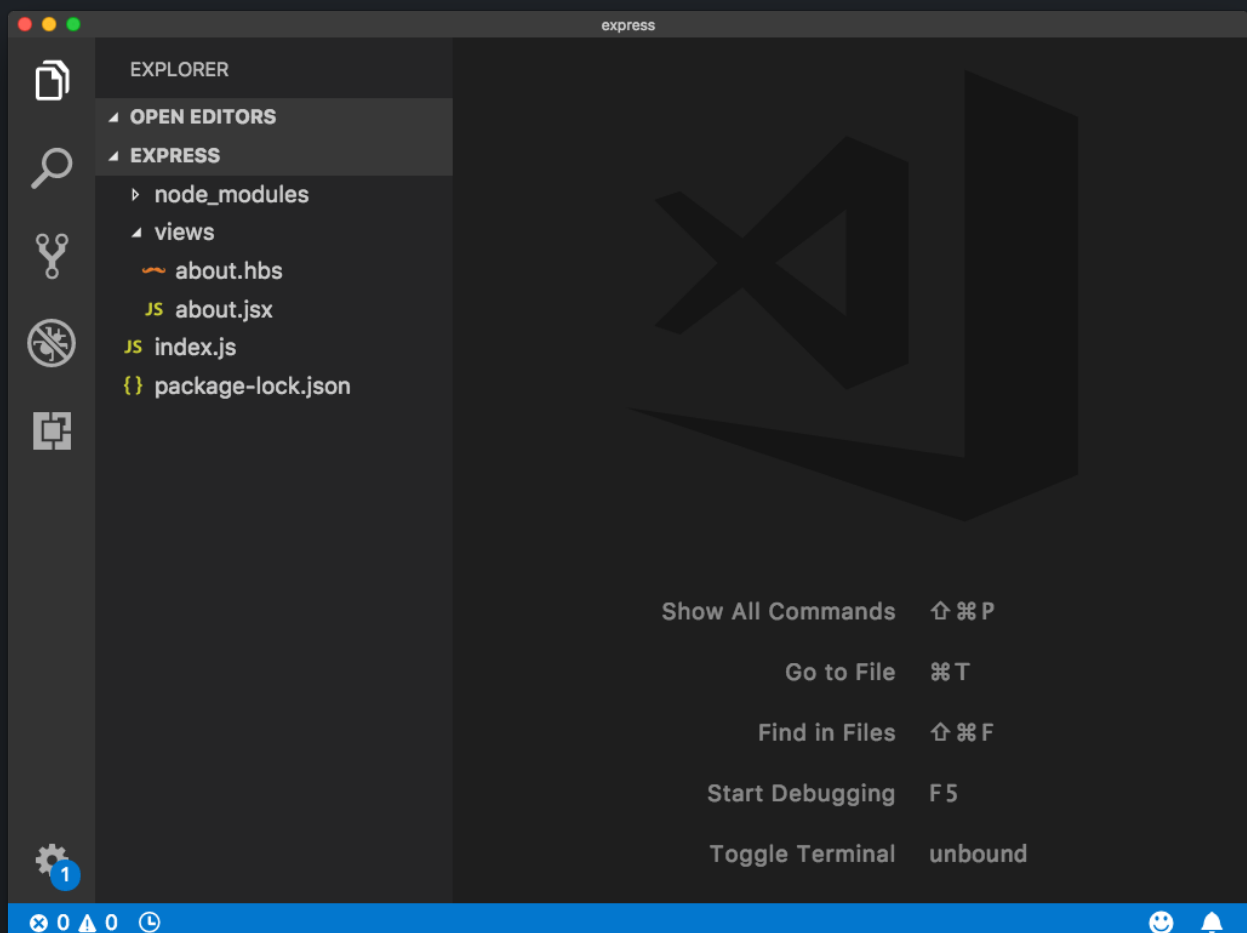
Let's start the exploration with the explorer (pun intended).



Press the “Open Folder” button in the sidebar, or the `Open folder...` link in the Welcome page. Both will trigger the file picker view.

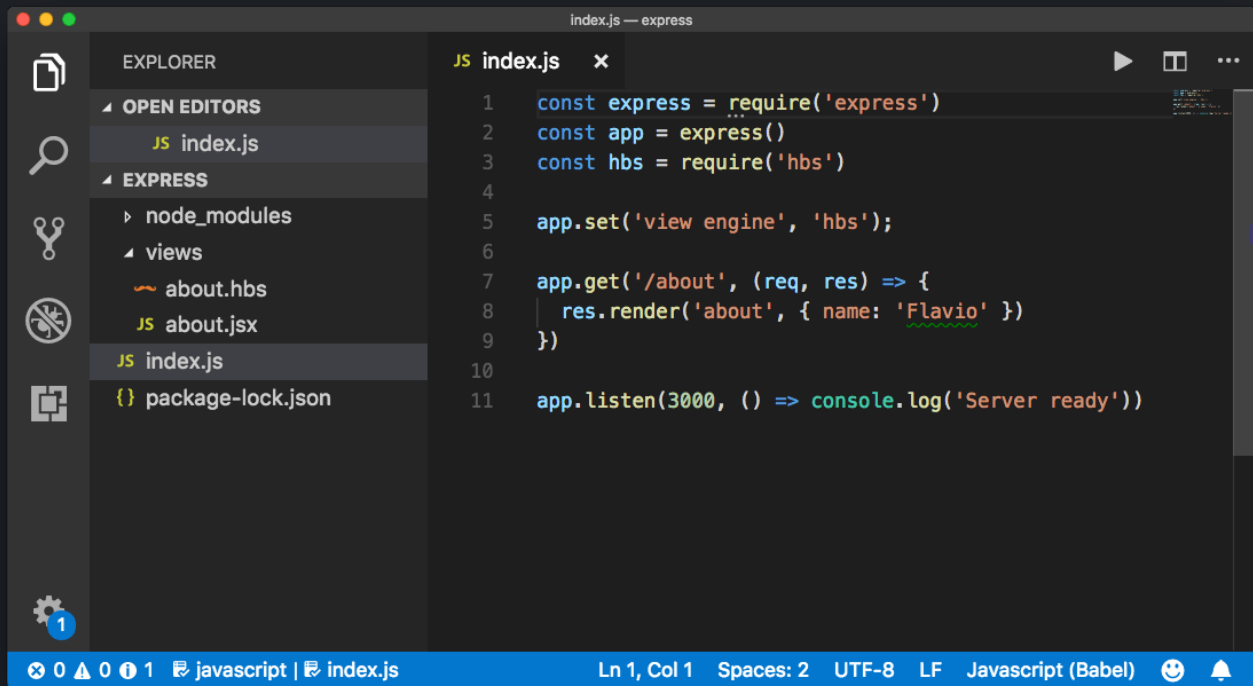
Choose one folder where you have source code, or even just text files, and open it.

VS Code will show that folder content in your view:

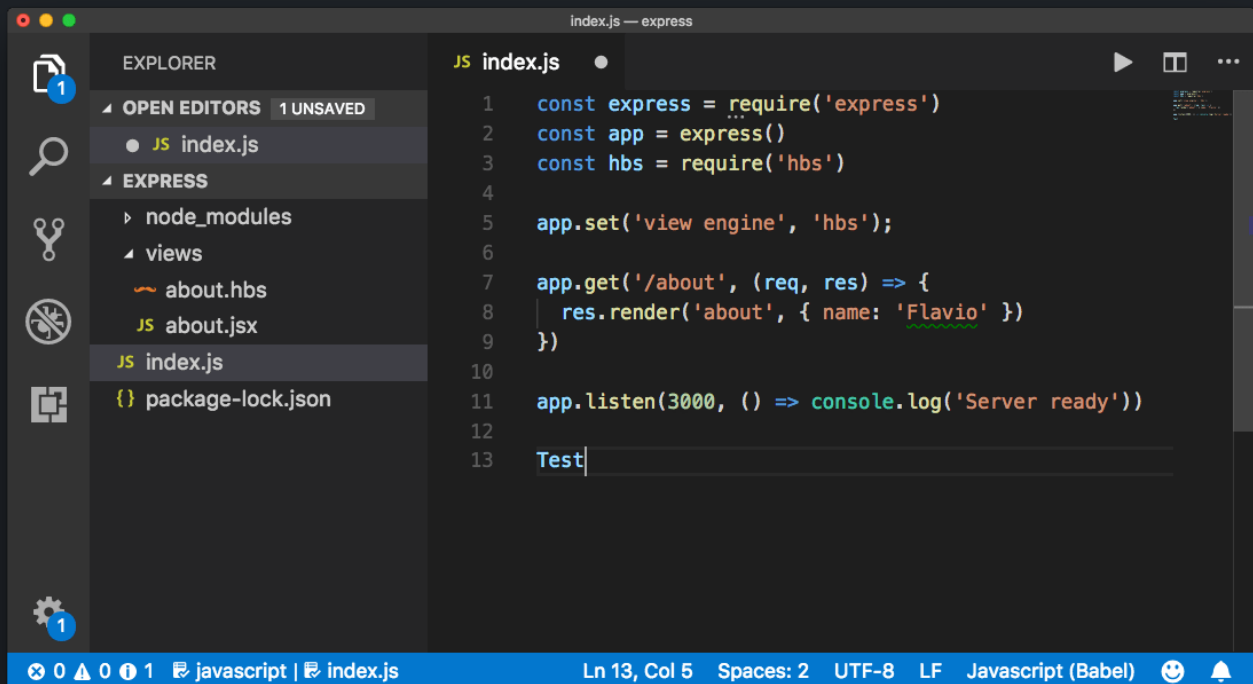


On the right, the empty view shows some commands to perform some quick operations, and their keyboard shortcut.

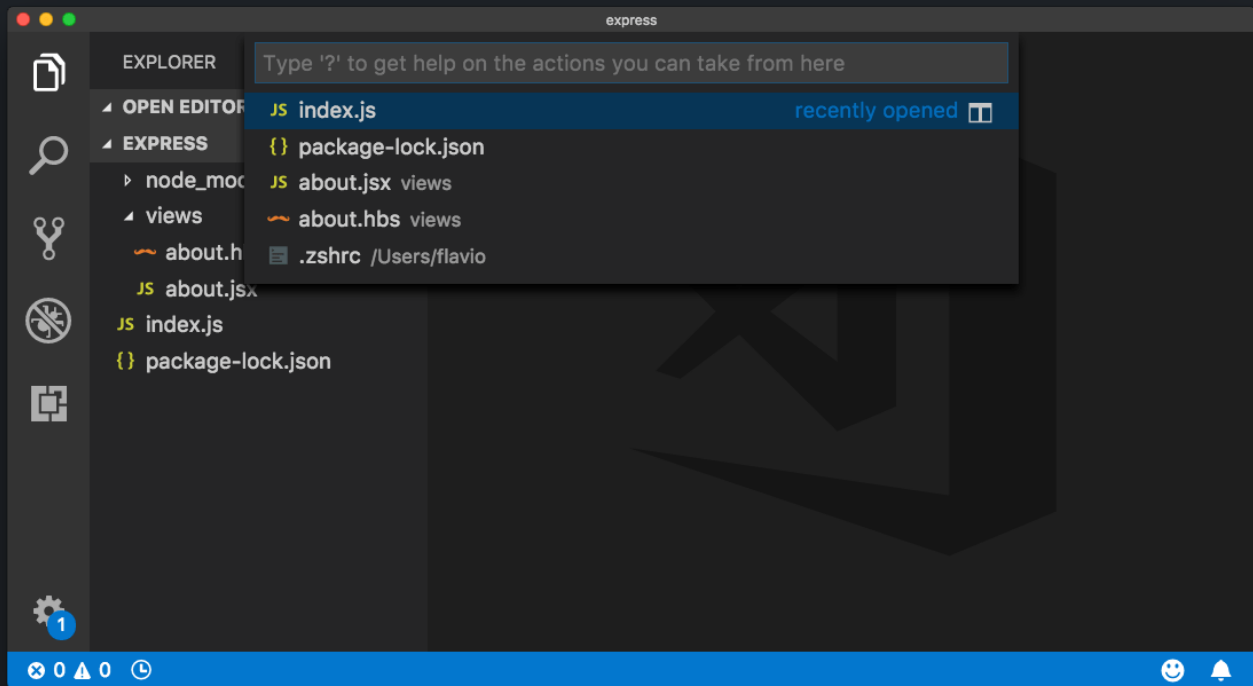
If you select a file on the left, that file will open on the main panel:



and if you start editing it, notice a dot will appear next to the file name in the tab, and in the sidebar as well:



Pressing **CMD+P** will show you a quick file picker to easily move in files on large projects:

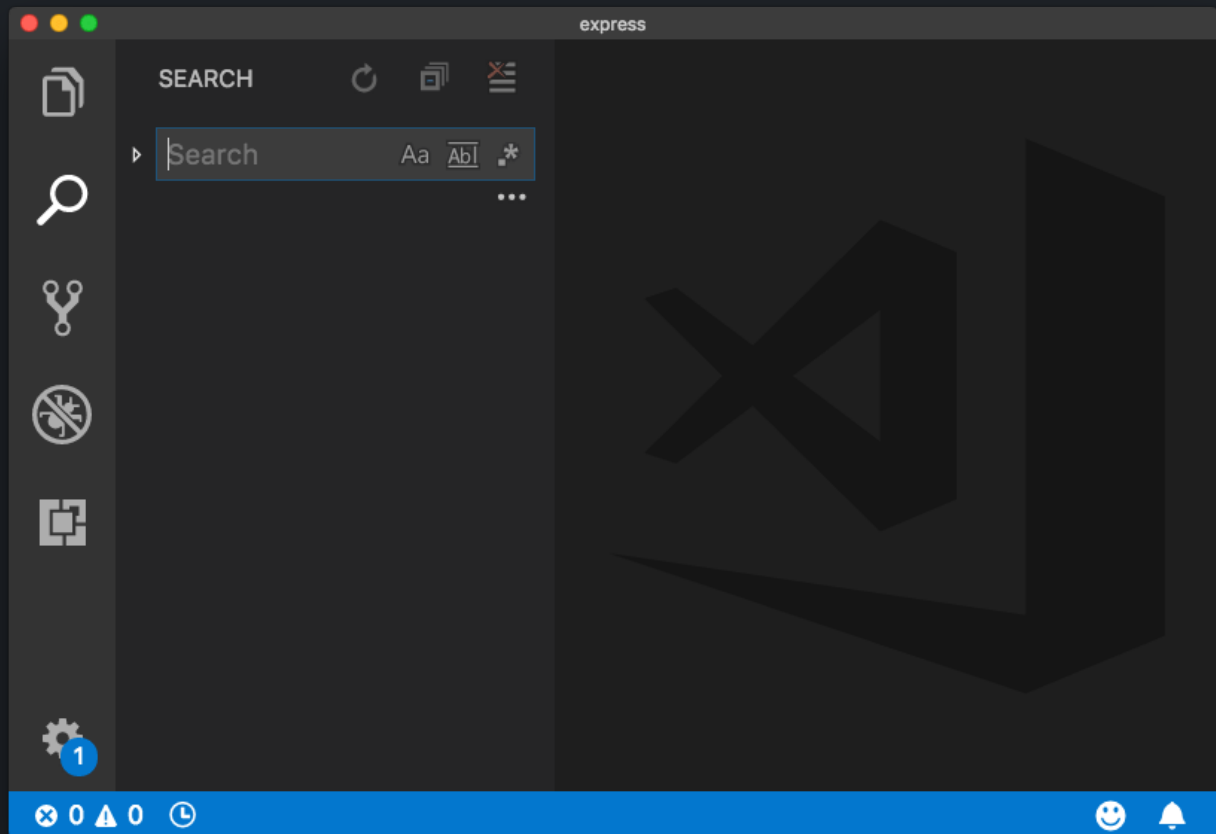


You can hide the sidebar that hosts the file using the shortcut `CMD+B`.

Note: I'm using the Mac keyboard shortcuts. Most of the times, on Windows and Linux you just change CMT to CTRL and it works, but not always. Print your [keyboard shortcuts reference](#).

SEARCH

The second icon in the toolbar is “Search”. Clicking it shows the search interface:

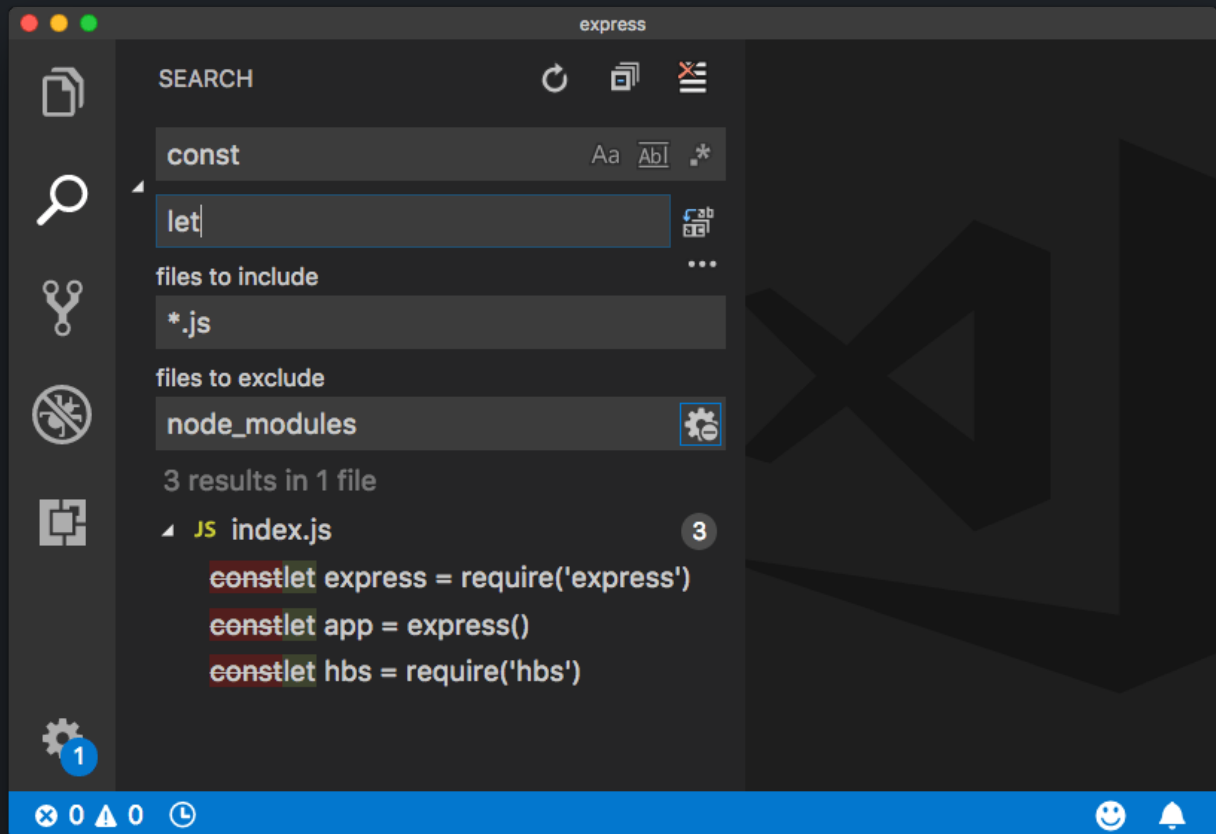


You can click the icons to make the search case sensitive, to match whole words (not substrings), and to use a regular expression for the search string.

To perform the search, press `enter`.

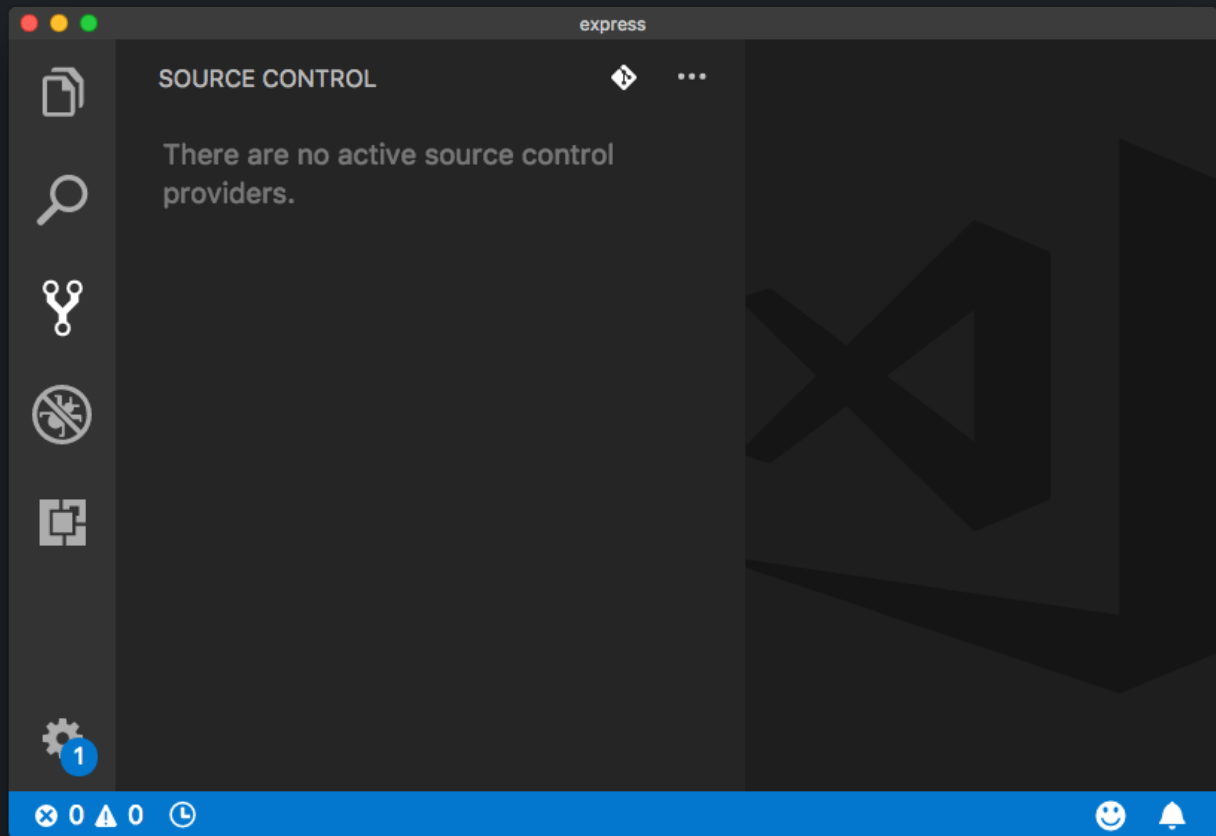
Clicking the `>` symbol on the left enables the search and replace tool.

Clicking the 3 dots shows a panel that lets you just include some specific kind of files, and exclude other files:



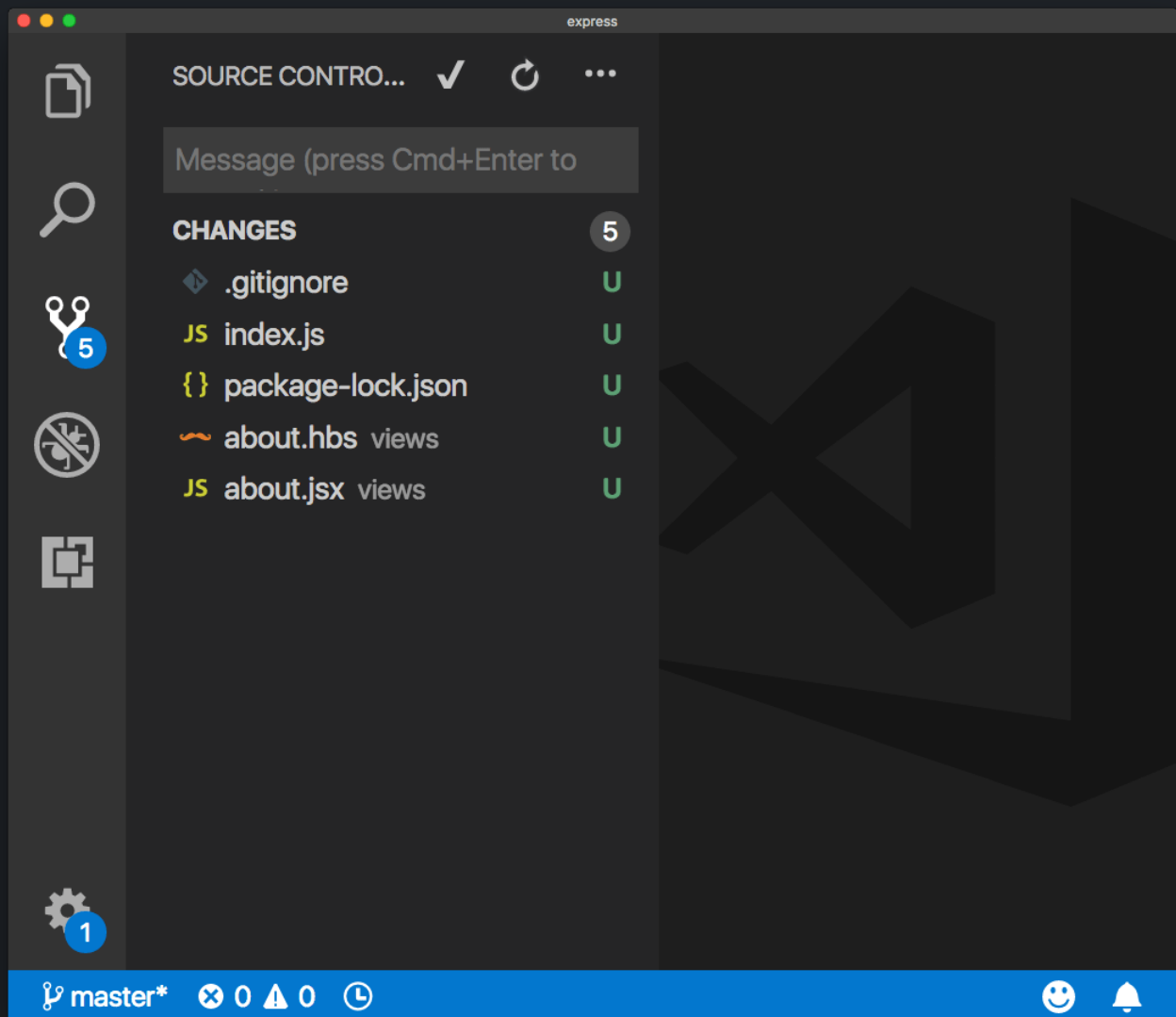
SOURCE CONTROL

The Source Control tab is enabled by clicking the third icon in the toolbar.



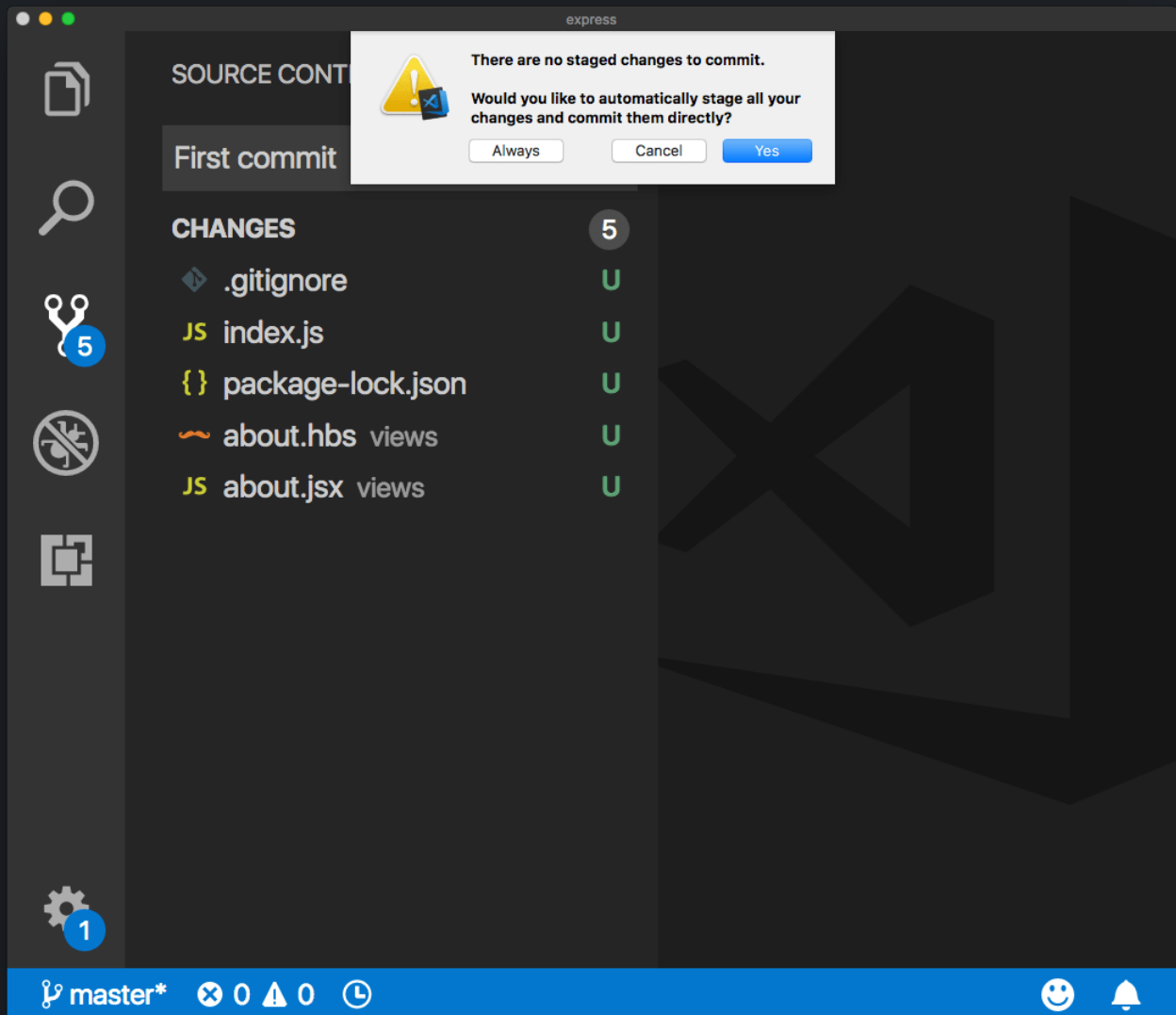
VS Code comes with Git support out of the box. In this case the folder we opened does not have source control initialized.

Clicking the first icon on top, with the Git logo, allows us to initialize the Git repository:



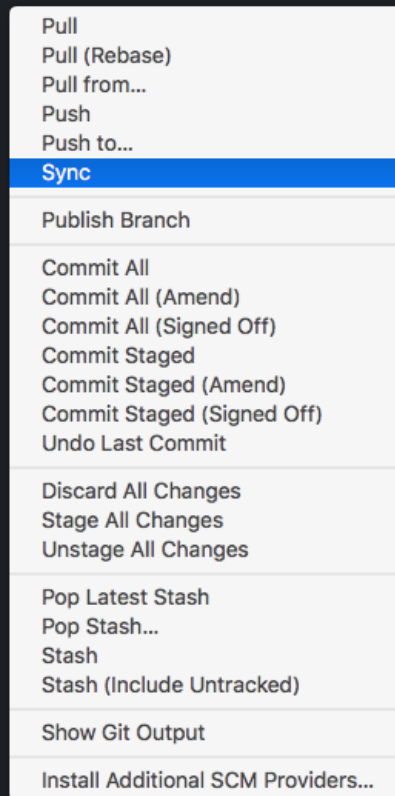
The **U** beside each file means that it's been updated since the last commit (since we never did a commit in the first place, all files are updated).

Create the first commit by writing a text message and pressing **Cmd-Enter**, or clicking the **✓** icon on top.



I usually set this to automatically stage the changes when I commit them.

The 3 dots icon, when clicked, offers lots of options for interacting with Git:

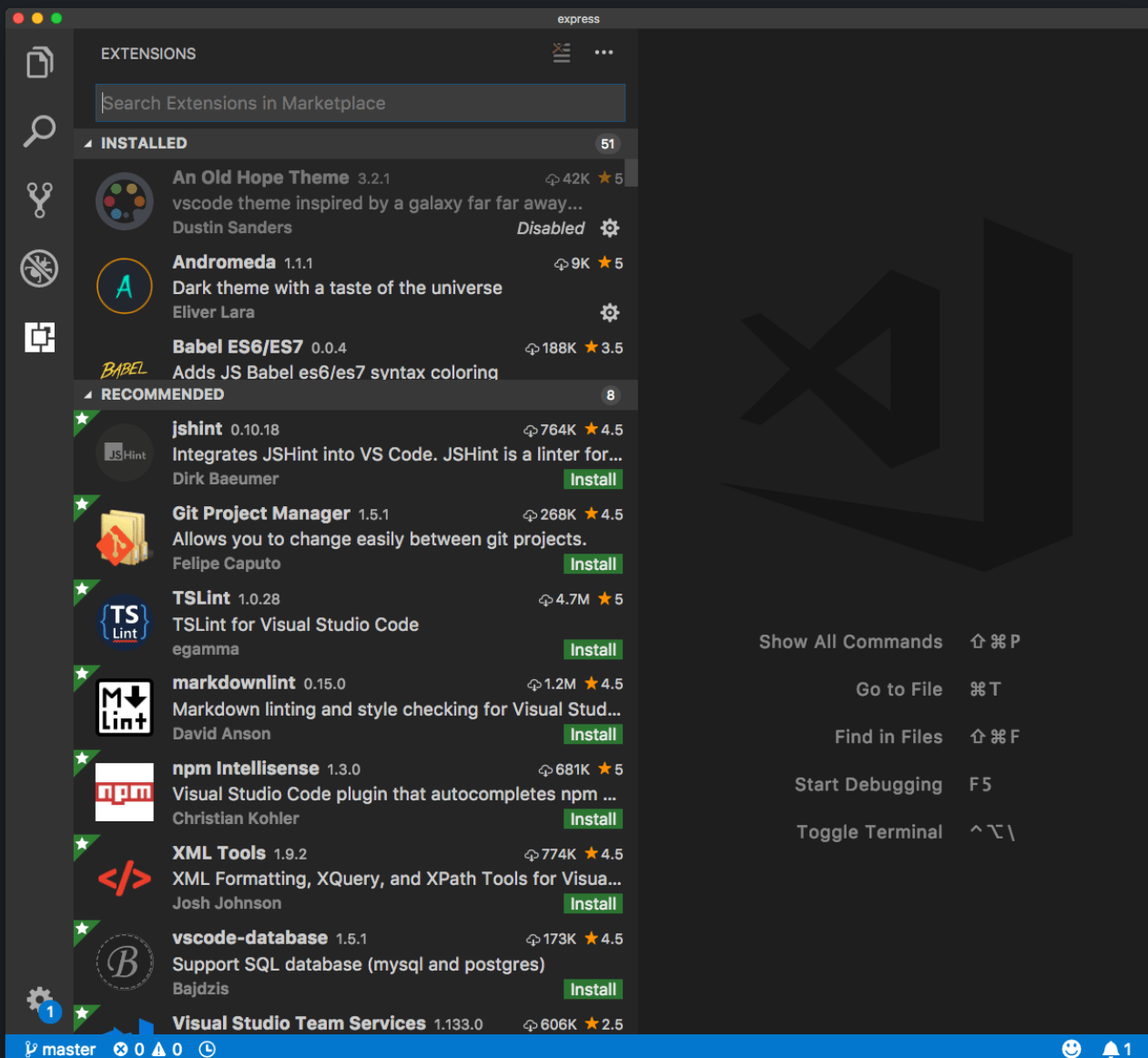


DEBUGGER

The fourth icon in the toolbar opens the JavaScript debugger. This deserves an article on its own. In the meantime check out [the official docs](#).

EXTENSIONS

The fifth icon brings us to extensions.



Extensions are one killer feature of VS Code.

They can provide so much value that you'll surely end up using tons of them.

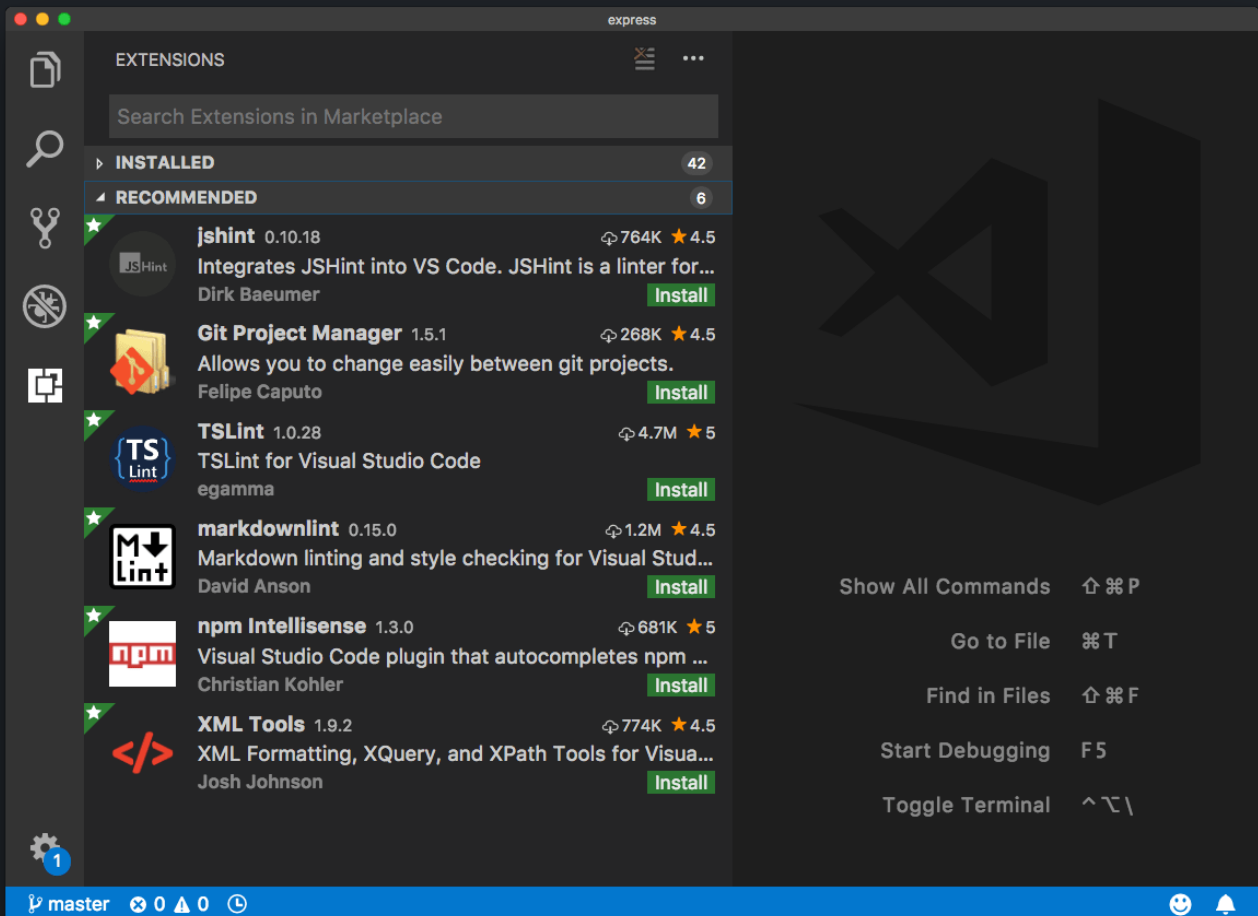
I have lots of extensions installed.

One thing to remember is that every extension you install is going to impact (more or less) the performance of your editor.

You can disable an extension you install, and enable only when you need it.

You can also disable an extension for a specific workspace (we'll talk about workspaces later). For example, you don't want to enable the JavaScript extensions in a Go project.

There is a list of recommended extensions, which include all the most popular tools.



Since I edit lots of markdown files for my blog, VS Code suggests me the `markdownlint` extension, which provides linting and syntax checking for Markdown files.

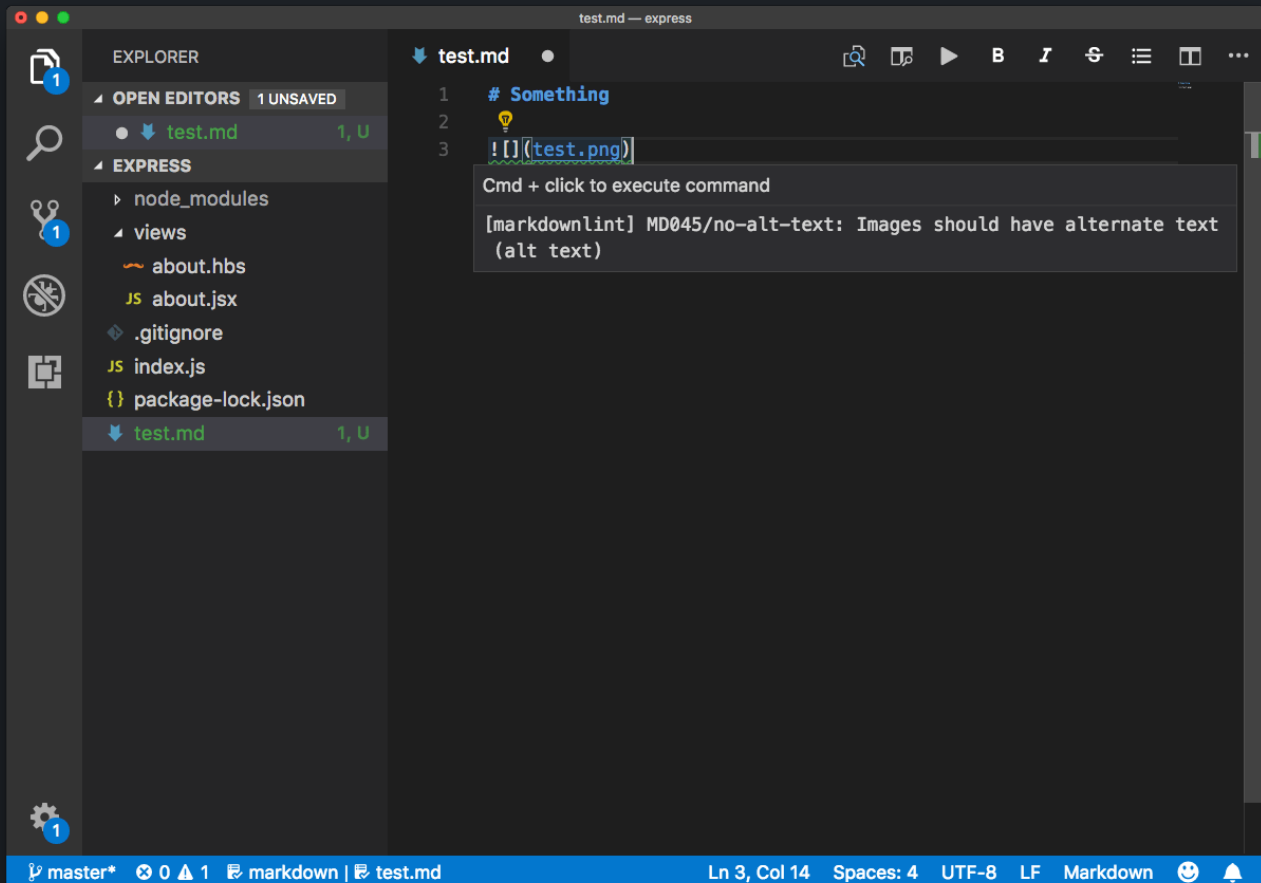
As an example, let's install it.

First, I inspect the number of views. It's 1.2M, so many! And the reviews are positive (4.⁵/5). Clicking the extension name opens the details on the right.

The screenshot shows the Visual Studio Code interface with the Extensions Marketplace open. On the left, a list of recommended extensions is visible, with 'markdownlint' highlighted. The main panel shows the details for 'markdownlint' by David Anson. The extension has a rating of 4.5 stars and 1,269,795 installations. The description states it is for 'Markdown linting and style checking for Visual Studio Code'. The 'Intro' section explains that Markdown is designed to be easy to read, write, and understand, but its flexibility can lead to inconsistency. The 'Install' section provides three steps: 1. Open Visual Studio Code, 2. Press `Ctrl+P` to open the Quick Open dialog, and 3. Type `ext install markdownlint` to find the extension.

Pressing the green Install button starts the installation process, which is straightforward. It does everything for you, and you just need to click the “Reload” button to activate it, which basically reboots the editor window.

Done! Let’s test it by creating a markdown file with an error, like a missing `alt` attribute on an image. It successfully tells us so:

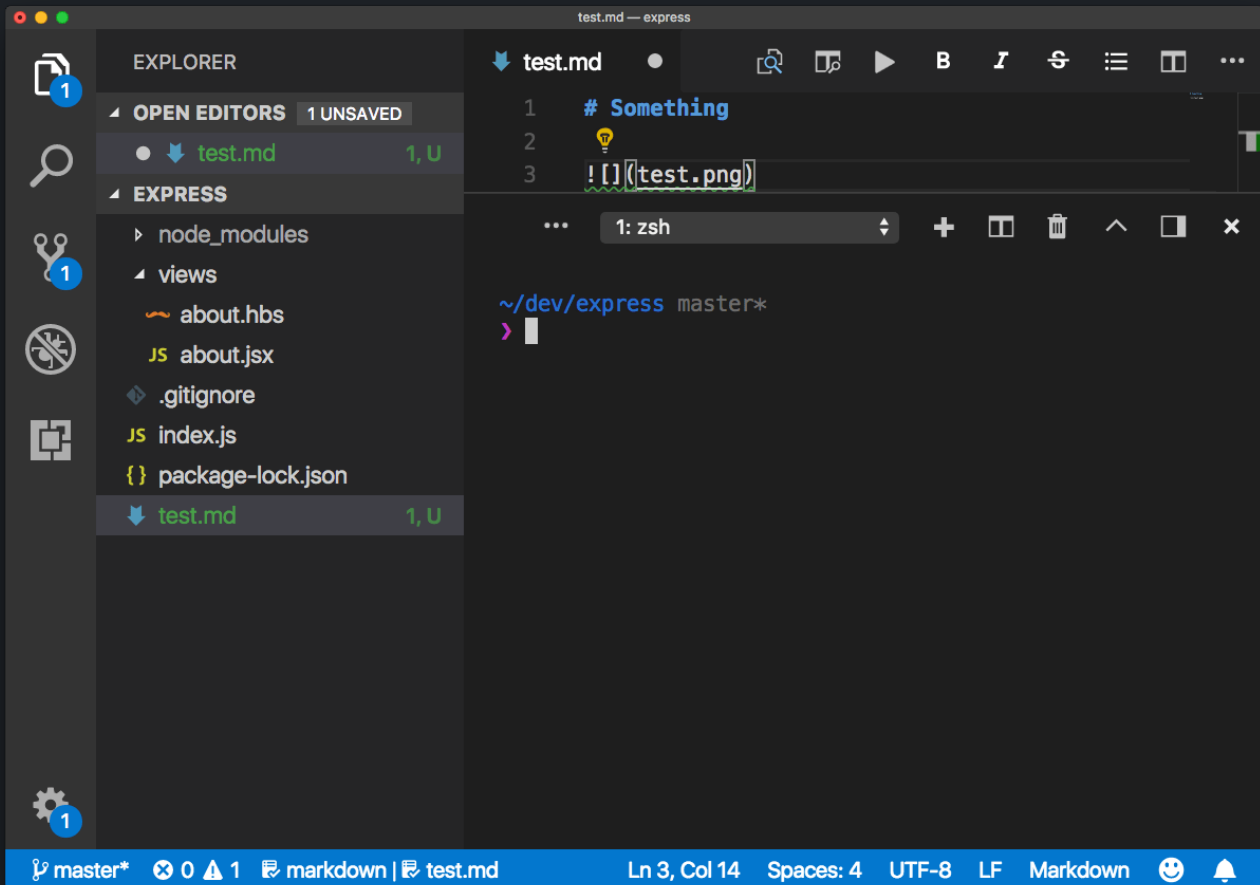


Down below I introduce some popular extensions you don't want to miss, and the ones I use the most.

THE TERMINAL

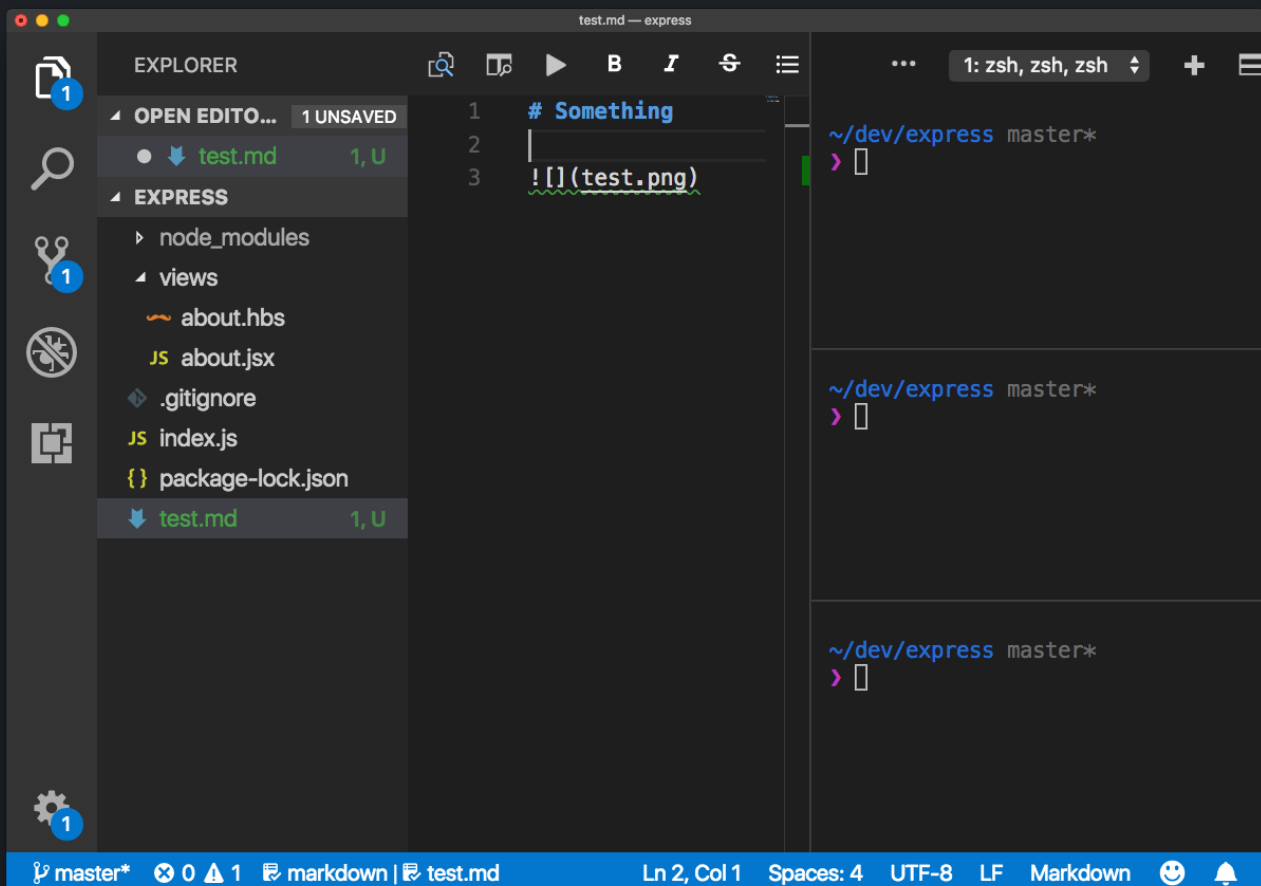
VS Code has an integrated terminal.

You can activate it from the menu `View > Integrated Terminal`, or using `CMD+\` and it will open with your default shell.



This is very convenient because in modern web development you almost always have some `npm` or `yarn` process running in the background.

You can create more than one terminal tab, and show them one next to the other, and also stack them to the right rather than in the bottom of the window:



THE COMMAND PALETTE

The Command Palette is a very powerful tool. You enable it by clicking **View > Command Palette**, or using **CMD+SHIFT+P**

A modal window will appear at the top, offering you various options, depending on which plugins you have installed, and which commands you used last.

Common operations I perform are:

- **Extensions: Install Extensions**
- **Preferences: Color Theme** to change the color theme (I sometimes change from night to day)
- **Format Document**, which formats code automatically
- **Run Code**, which is provided by Code Runner, and executes the highlighted lines of JavaScript

you can activate any of those by starting typing, and the autocomplete functionality will show you the one you want.

Remember when you typed **CMD+P** to see the list of files, before? That's a shortcut to a specific feature of the Command Palette. There are others:

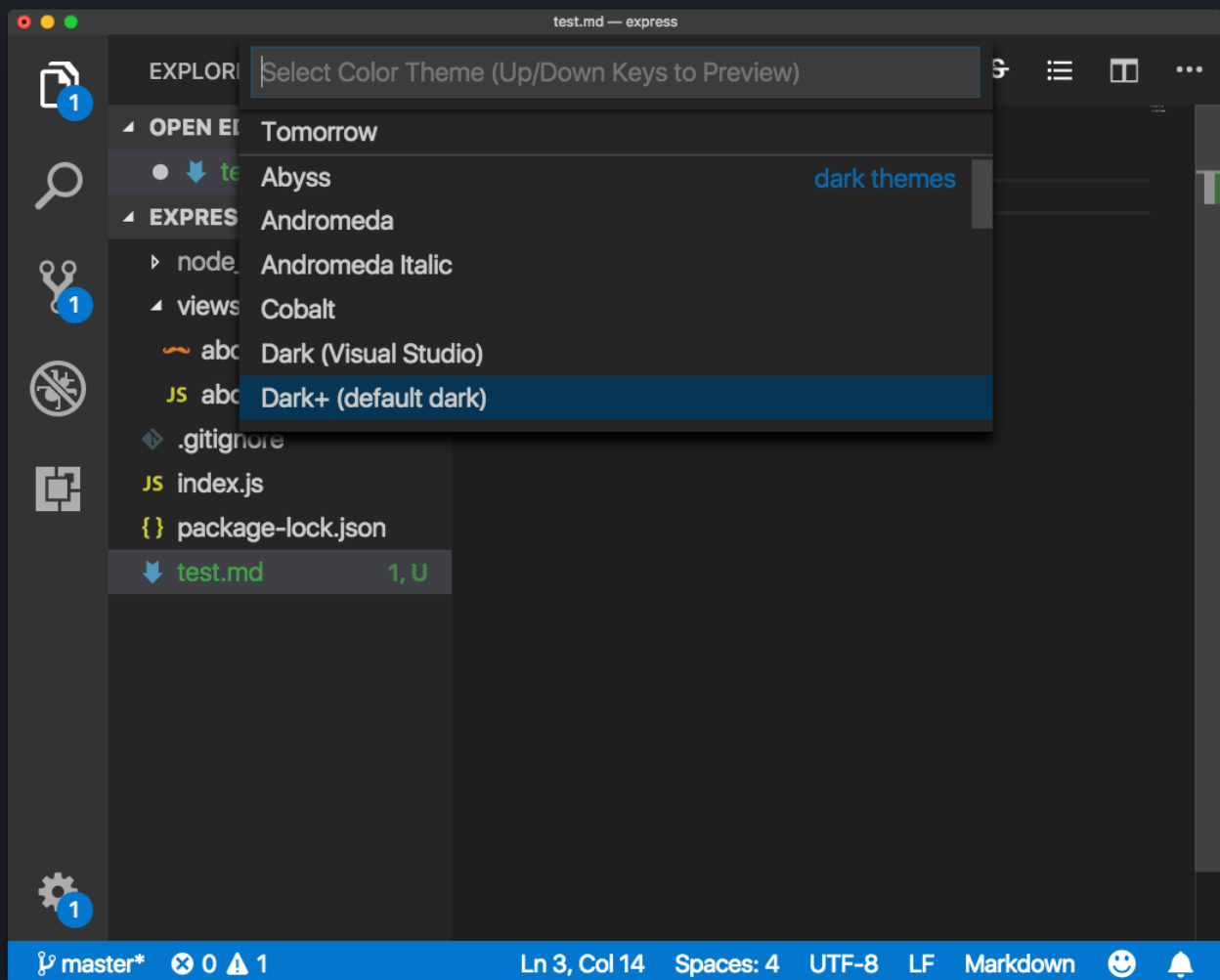
- **Ctrl-Shift-Tab** shows you the active files
- **Ctrl-G** opens the command palette to let you enter a line number to go to
- **CMD+SHIFT+O** shows the list of symbols found in the current file

What symbols *are* depends on the file type. In JavaScript, those might be classes or functions. In Markdown, section titles.

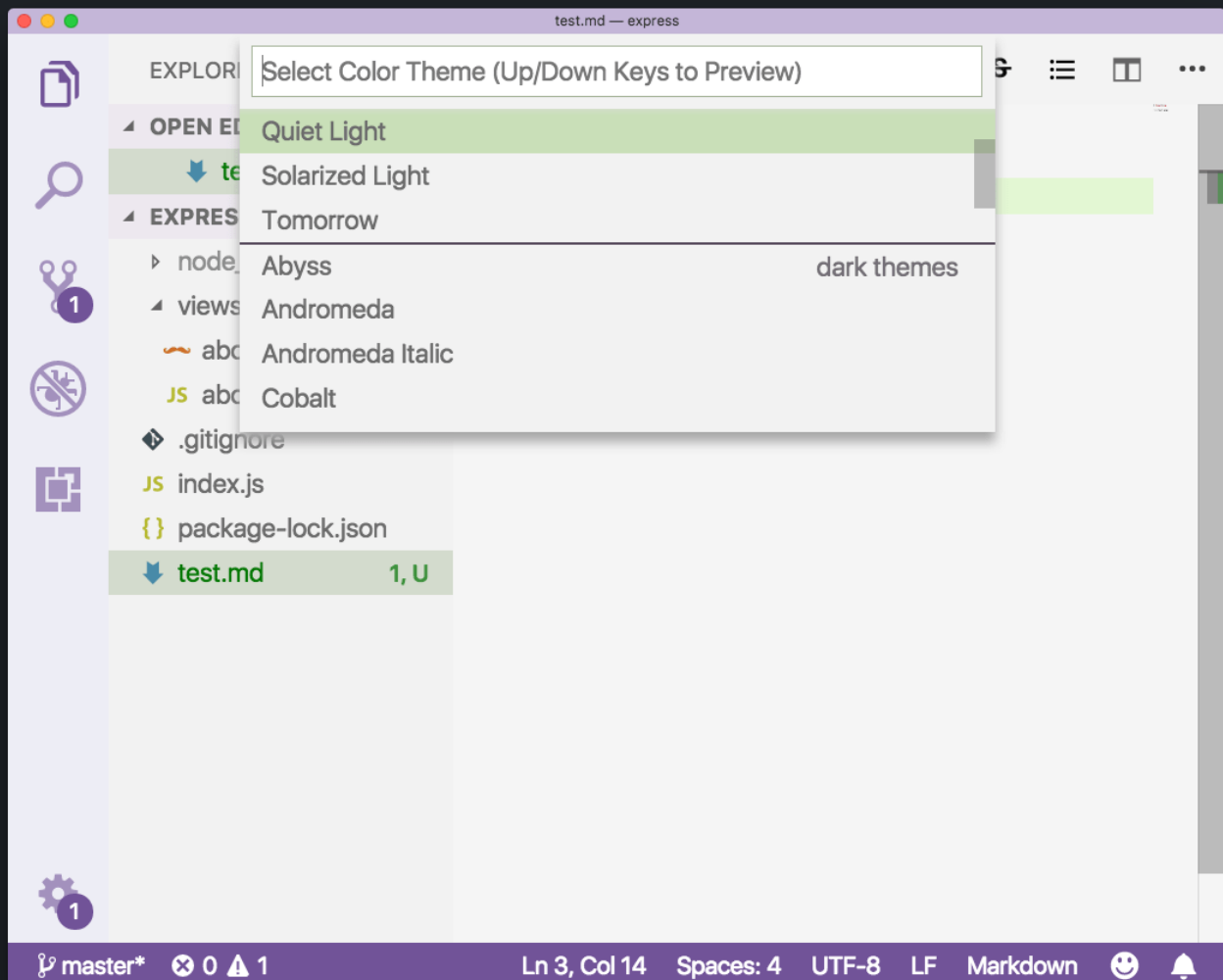
THEMES

You can switch the color theme used by clicking `CMD-k` + `CMD-t`, or by invoking the *Preferences: Color Theme* command.

This will show you the list of themes installed:



you can click one, or move with the keyboard, and VS Code will show you a preview. Click enter to apply the theme:



Themes are just extensions. You can install new themes by going to the extensions manager.

Probably the best thing for discoverability is to use [the marketplace website](#).

My favorite theme is [Ayu](#), which provides a great style for any time of the day, night, morning/evenings and afternoon.

CUSTOMIZATION

Theme is just one customization you can make.

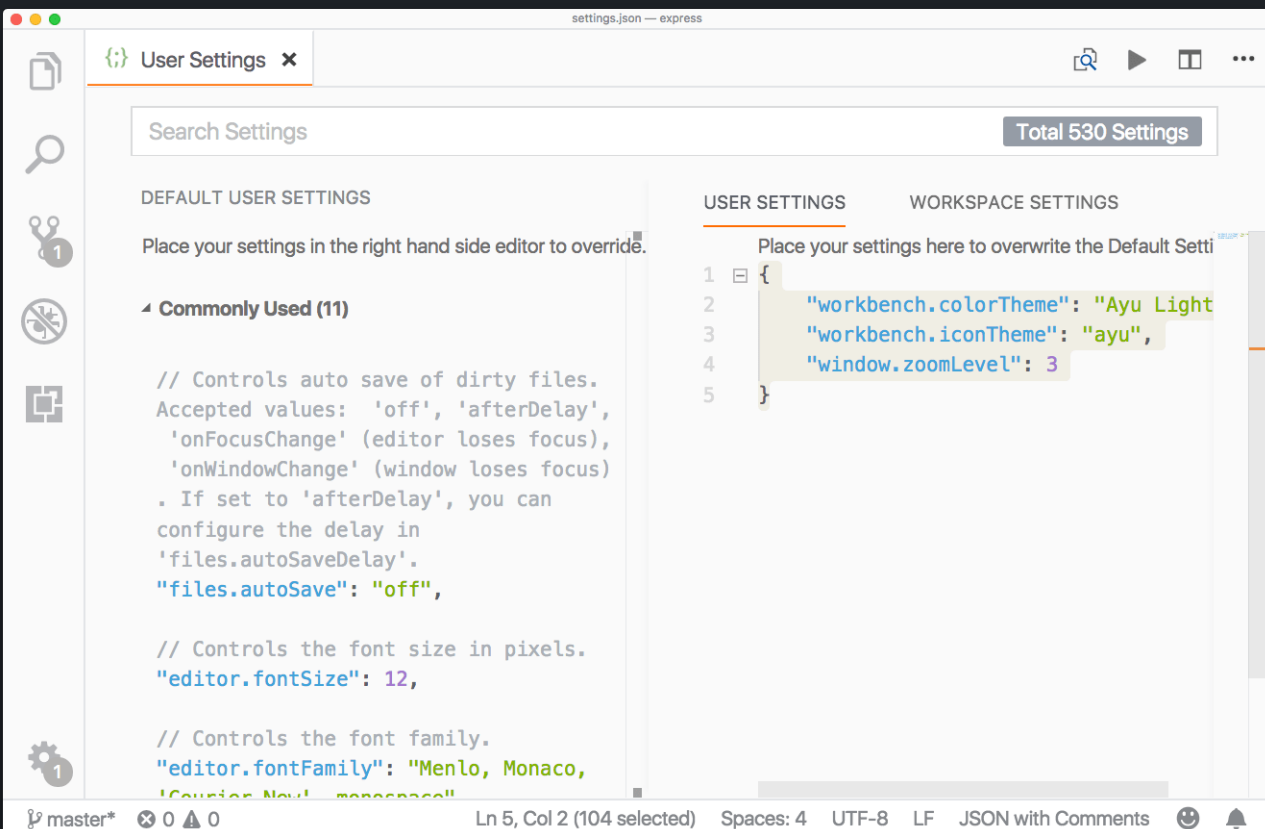
The sidebar icons that are assigned to a file are also a big part of a nice user experience.

You can change those by going to `Preferences > File Icon Theme`. Ayu comes with its own icons theme, which perfectly matches the theme colors:



All those customizations we made so far, the theme and the icon theme, are saved to the user preferences.

Go to **Preferences > Settings** (also reachable via **CMD-,**) to see them:



The view shows the default settings on the left, for an easy reference, and the

overridden settings on the right. You can see the name of the theme and the icon theme we set up, in `workbench.colorTheme` and `workbench.iconTheme`.

I zoomed in using `CMD++`, and this setting was saved as well to `window.zoomLevel`, so the next time VS Code starts up, it remembers my choice for zooming.

You can decide to apply some setting globally, in **User Settings**, or relative to a workspace, in **Workspace settings**.

Most of the times those settings are automatically added by extensions or by the VS Code itself, but in some cases you'll directly edit them in this place.

NICE CONFIGURATION OPTIONS

Some nice configuration options I set in my code:

Option	Description
<code>"editor.minimap.enabled": false</code>	Remove the minimap, which is shown at the right of the editor
<code>"explorer.confirmDelete": false</code>	Stop asking me for confirmation when I want to remove a file (I have source control!)
<code>"explorer.confirmDragAndDrop": false</code>	Disable the confirmation for drag and drop
<code>"editor.formatOnSave": true</code>	Format the code automatically when I save it
<code>"editor.formatOnPaste": true</code>	Format the code automatically when I paste it in my code
<code>"javascript.format.enable": true</code>	Enable formatting for JavaScript code
<code>"files.trimTrailingWhitespace": true</code>	Trim whitespace in files
<code>"editor.multiCursorModifier": "alt"</code>	When clicking the Alt key and clicking with the mouse, I can select multiple lines
<code>"editor.detectIndentation": true</code>	Adapt to the file indentation, useful when editing other people code
<code>"editor.quickSuggestionsDelay": 0</code>	Show the code suggestion immediately, not after some seconds

THE BEST FONT FOR CODING

I like [Fira Code](#). It's free, and has some very nice programming ligatures, which transform common constructs like `!==` and `=>` to nicer symbols:



Enable it by installing the font and adding this to your configuration:

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

WORKSPACES

All User settings can be overridden in Workspace settings. They take precedence. They are useful for example when you use a project that has linting rules different from all the other projects you use, and you don't want to edit your favorite settings just for it.

You create a workspace from an existing project by clicking the **File > Save Workspace as...** menu.

The currently opened folder will be enabled as the workspace main folder.

The next time you open VS code, or you switch project, instead of opening a folder, you open a workspace, and that will automatically open the folder containing your code, and it will remember all the settings you set specific to that workspace.

In addition to having workspace-level settings, you can disable extensions for a specific workspace.

You can just work with folders until you have a specific reason for wanting a workspace.

One good reason is the ability to have multiple, separate root folders. You can use the **File > Add Folder to Workspace** to add a new root folder, which can be located anywhere in the filesystem, but will be shown along with the other existing folder you had.

EDITING

INTELLISENSE

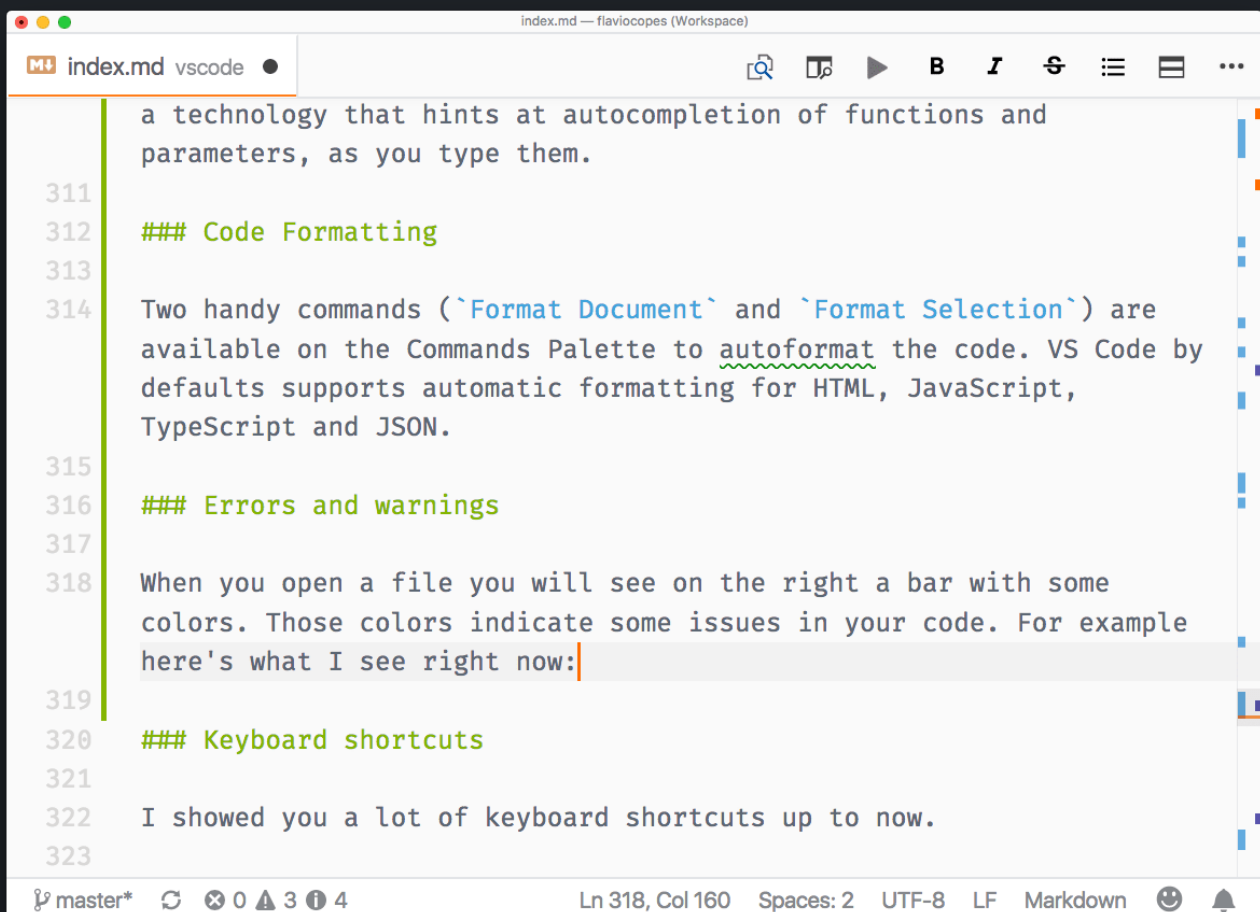
When you edit in one of the supported languages (JavaScript, JSON, HTML, CSS, Less, Sass, C# and [TypeScript](#)) VS Code has IntelliSense, a technology that hints at autocompletion of functions and parameters, as you type them.

CODE FORMATTING

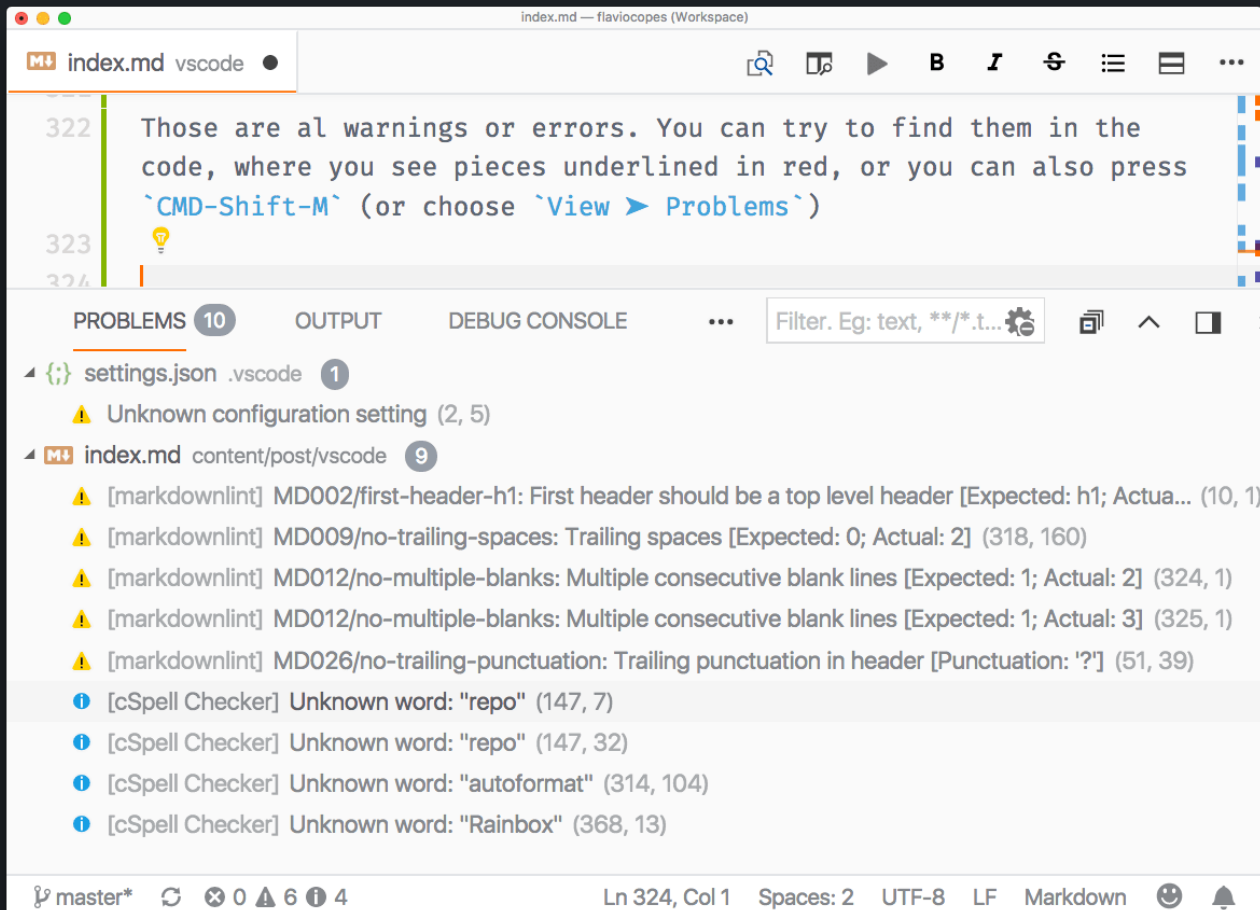
Two handy commands (`Format Document` and `Format Selection`) are available on the Commands Palette to autoformat the code. VS Code by default supports automatic formatting for HTML, JavaScript, TypeScript and JSON.

ERRORS AND WARNINGS

When you open a file you will see on the right a bar with some colors. Those colors indicate some issues in your code. For example here's what I see right now:



Those are all warnings or errors. You can try to find them in the code, where you see pieces underlined in red, or you can also press `CMD-Shift-M` (or choose `View > Problems`)



KEYBOARD SHORTCUTS

I showed you a lot of keyboard shortcuts up to now.

It's starting to get complicated to remember them all, but they are a nice productivity aid. I suggest to print the official shortcuts cheat sheet, for [Mac](#), [Linux](#) and [Windows](#).

KEYMAPS

If you're used to keyboard shortcuts from other editors, maybe because you worked with one editor for a long time, you can use a keymap.

The VS Code team provides keymaps for the most popular editors out of the box: vim, Sublime Text, Atom, IntelliJ, Eclipse and more. They are available as plugins. By opening the **Preferences > Keymaps Extensions** menu.

CODE SNIPPETS

Snippets are very cool.

For every language you might be developing in, there are extensions that provide ready-made snippets for you to use.

For JavaScript/React, one popular one is [VS Code ES7 React/Redux/React-Native/JS snippets](#)

You just type `rfe`, press TAB and this appears in your editor:

```
import React from 'react'

const $1 = props => {
  return <div>$0</div>
}

export default $1
```

there are lots of these shortcuts, and they save a lot of time. Not just from typing, but also from looking up the correct syntax.

You can also define your own snippets. Click `Preferences > User Snippets` and follow the instructions to create your own snippets file.

EXTENSIONS SHOWCASE

- [GitLens](#): visualize who made the last change to a line of your code, and when this happened
- [Git History](#) visualize and search the Git history
- [CSS Peek](#) lets you see and edit CSS definitions by inspecting the class of an HTML element. Very handy.
- [Code Runner](#) lets you run bits of code that you select in the editor, and much more. Supports lots of languages.
- [Debugger for Chrome](#) allows you to debug a JavaScript code running in the browser using the VS code debugger.
- [Bracket Pair Colorizer](#) handy for visualizing brackets endings in your code.
- [Indent-Rainbow](#) colors the indentation levels of your code.
- [Prettier](#) check my [Prettier guide](#)
- [ESLint](#) check my [ESLint guide](#)
- [IntelliSense for CSS](#) improved autocompletion for CSS based on your workspace definitions
- [npm](#) enables [npm](#) utility functions from the command palette
- [Auto Close Tag](#) automatically close HTML/JSEX/* tags
- [Auto Rename Tag](#) automatically renames the closing tag when you change the opening one, and the opposite as well

THE VS CODE CLI COMMAND

When you install VS Code, the `code` command is available globally in your command line.

This is very useful to start the editor and open a new window with the content of the current folder, with `code .`.

`code -n` will create a new window.

A useful thing that's not always known is that VS Code can quickly show the diff between two files, with `code --diff file1.js file2.js`.

SOLVING HIGH USAGE CPU ISSUES

I ran into an issue of high CPU usage, and spinning fans, with a project with lots of files under `node_modules`. I added this configuration and things looked normal again:

```
"files.useExperimentalFileWatcher": true,
"files.exclude": {
  "/.git": true,
  "/.DS_Store": true,
  "/node_modules": true,
  "/node_modules/": true
},
"search.exclude": {
  "/node_modules": true
},
"files.watcherExclude": {
  "/node_modules/": true
},
```

[Home](#)

[Light|dark mode](#)

FREE EBOOKS

[JavaScript](#)

[Vue.js](#)

[Node.js](#)

[Express.js](#)

[React](#) (new!)

Want more? Get my 1250-pages book [Modern Web Development](#)

Express

Node.js

Git

Network

Web Platform

React

Vue.js

Go

JavaScript

CSS

DevTools

Services

Lab

© 2019 [Flavio Copes](#)

