

Intro to VS Code

ALL THIS – JUST TO EDIT SOME TEXT?

Developers have a wide set of tools purpose-made for writing code. Here are a few of them:

- Atom
- Sublime Text
- Brackets
- NetBeans
- Coda
- TextWrangler
- TextMate
- Bluefish
- Dreamweaver

But source code — whether it's HTML, CSS, or JavaScript — is just plain text. Why so many choices? And why not just use a general-purpose text editor?

PRODUCTIVITY – OR NOVELTY?

Ask most developers why they use the code editor they do and you'll probably hear that they chose it to increase their productivity. And code editors *do* help programmers with things like auto-complete, integration of source control like git, automatic formatting, *etc.*

But some programmers prefer bare-bones editors. These programmers embrace tools like vim or emacs and claim that, although the learning curve may seem unbearably long, these tools lets them work far more efficiently than would their fancier cousins.

It's certainly true that programmers seem to have a genetic attraction to anything that's new and shiny, but in the case of code editors, the benefits they offer is so great that we would be hard put to make an argument against them.

One of the newest code editors comes from a surprising source: Microsoft. This is surprising not because Microsoft doesn't make excellent tools for developers — they do — but because their new product, Visual Studio Code (*aka* VS Code) does not require developers to buy into the Microsoft ecosystem, is free, and is cross-platform. It's for these reasons we use VS Code and recommend it to anyone not already committed to a different editor.

INSTALLING AND CUSTOMIZING VS CODE

If you point your browser to code.visualstudio.com, you can download the program for your platform. You can choose from two versions: a "stable build" and an "insider's edition". The insider's edition will keep you up-to-date with the latest additions to the product, but, according to the website, "may lead to the occasional broken build". We recommend you use the stable build.

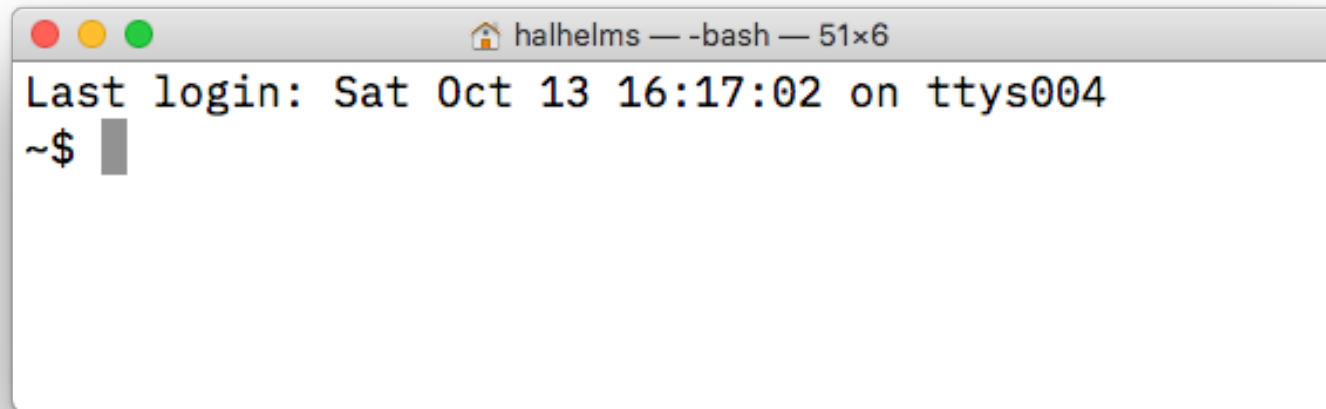
One of the really fantastic features of VS Code is the customizability. Once you've installed VS Code, open it. In this section, we'll introduce which part of the product can be customized — and suggest some ways to make this great tool even better.

1. As you gain familiarity with developing, you'll find yourself spending more time in what truly is a bare-bones tool: Terminal. To open Terminal in your Mac, hold down the command key while pressing the spacebar. This will bring up Mac's Spotlight app — a search app for installed programs.

A screenshot of the Spotlight Search bar on a Mac. It is a light gray rounded rectangle with a magnifying glass icon on the left and the text "Spotlight Search" in a light gray font.

Begin typing **term** — and Spotlight will suggest **terminal.app**. That's the one you want. Select it and you'll be presented with a new window.

Your version may look
somewhat different

A screenshot of a macOS terminal window. The title bar shows 'halhelms — -bash — 51x6'. The terminal content shows 'Last login: Sat Oct 13 16:17:02 on ttys004' followed by a prompt '~\$' and a cursor.

Terminal is a *command line interface* (or CLI). It's quite different from the *graphical user interface* (GUI) that most programs present. Terminal is unapologetic in assuming that, if you're using it, you're okay with CLIs. And you will be — there are only a few commands that you'll commonly use and we predict you'll grow to love how quickly you can accomplish things with it. Let's try one command right now.

We're going to have you create a new directory off of the root — the ultimate parent of all directories. In your terminal window, type...

Changes directory to root

`cd ~` 

Creates a new directory
called "dev"

`mkdir dev` 

Changes current
directory to the newly-
created "dev" directory

`cd dev` 

You could create a new directory, of course, from Finder, but let's do things the geek way. Now that you are in your dev directory, you want to open VS Code — but you want to open it already pointing to the directory you're in (dev, in this case). In order to do that, we're going to go back into VS Code and set things up.

From VS Code's top menu, select **View > Command Palette**.

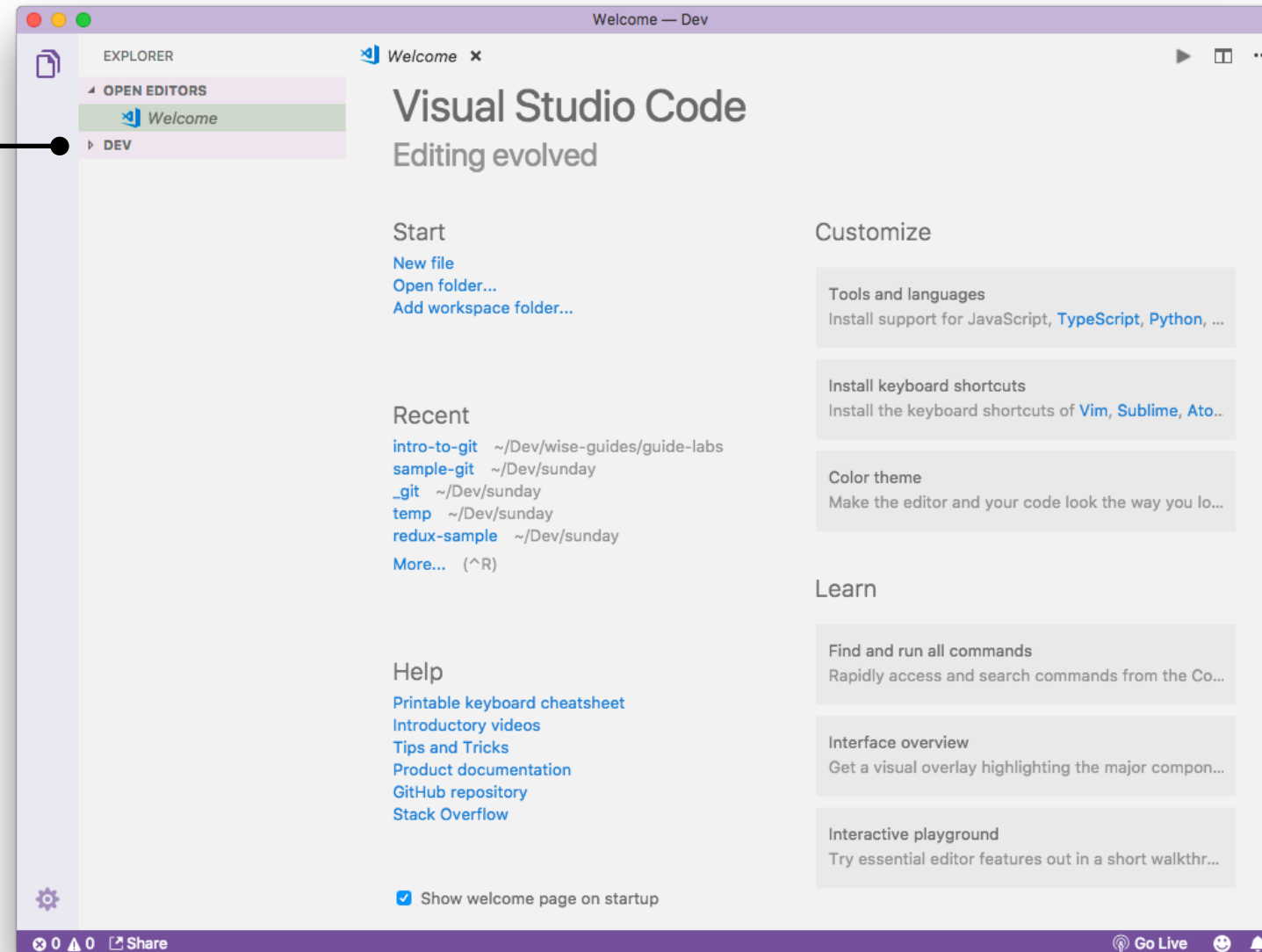
This will bring up the "Command Palette". Begin typing path and you'll see highlighted a selection that reads: **Shell Command: Install 'code' command in PATH**. That's the one you want. Press Enter.

You won't get any feedback, but something cool just happened. To see it, go back to your terminal and type `code .`

You'll see VS Code open and look something like this:

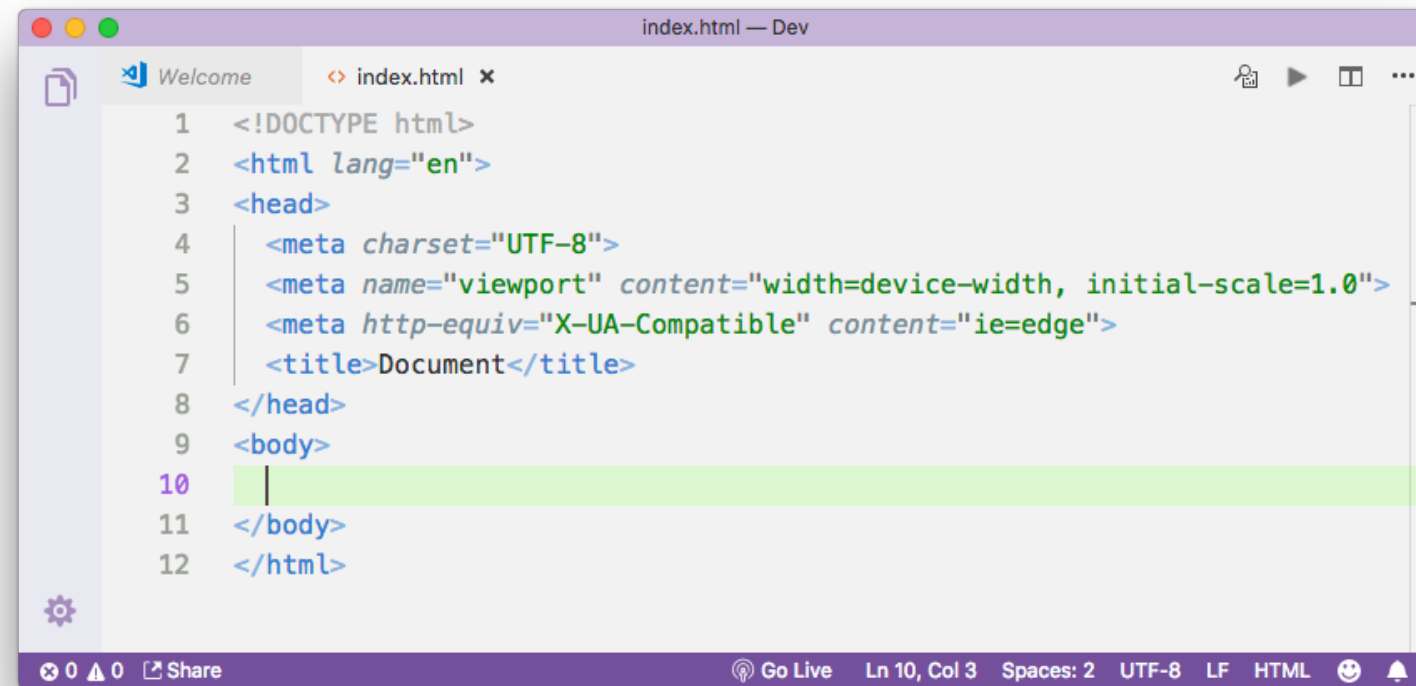
That's the word "code" followed by a period (and pronounced as "code dot")

Notice that VS Code already points its Explorer window to "dev"



You can use `code .` from any directory and VS Code will begin with that directory selected. That will save you a *lot* of time over trying to navigate to your desired directory using VS Code's Explorer function.

2. Speaking of saving time, VS Code comes with a terrific embedded product called Emmet. Let's start by creating a new, empty file called `index.html` with VS Code. All HTML files need some boilerplate code. That's something VS Code/Emmet can help with. Type `!` and press the Tab key.



Standard HTML boilerplate is inserted.

That's just the start of how Emmet can be useful. Within the `<body></body>` tags, type `p` then press Tab. A `<p></p>` element is inserted. (This works with all HTML elements such as `div`, `main`, `container`, etc.)

Let's take the case where we want a `<div>` element with an `id` of `"app"` — something like this...

```
<div id="app"></div>
```

To produce this in VS Code, type `div#id` and press Tab.

How about classes? To produce this...

```
<p class="active product special"></p>
```

...type `p.active.product.special` and press Tab.

Let's take a more complicated example, where we want to produce this...

```
<ol class="current">
  <li id="product-1" class="product"></li>
  <li id="product-2" class="product"></li>
  <li id="product-3" class="product"></li>
</ol>
```

That can be done with this snippet:

```
ol.current>li#product-$.product*3
```

> tells VS Code to make what follows a child element of the one preceding it

\$ tells VS Code to create an incrementing whole number

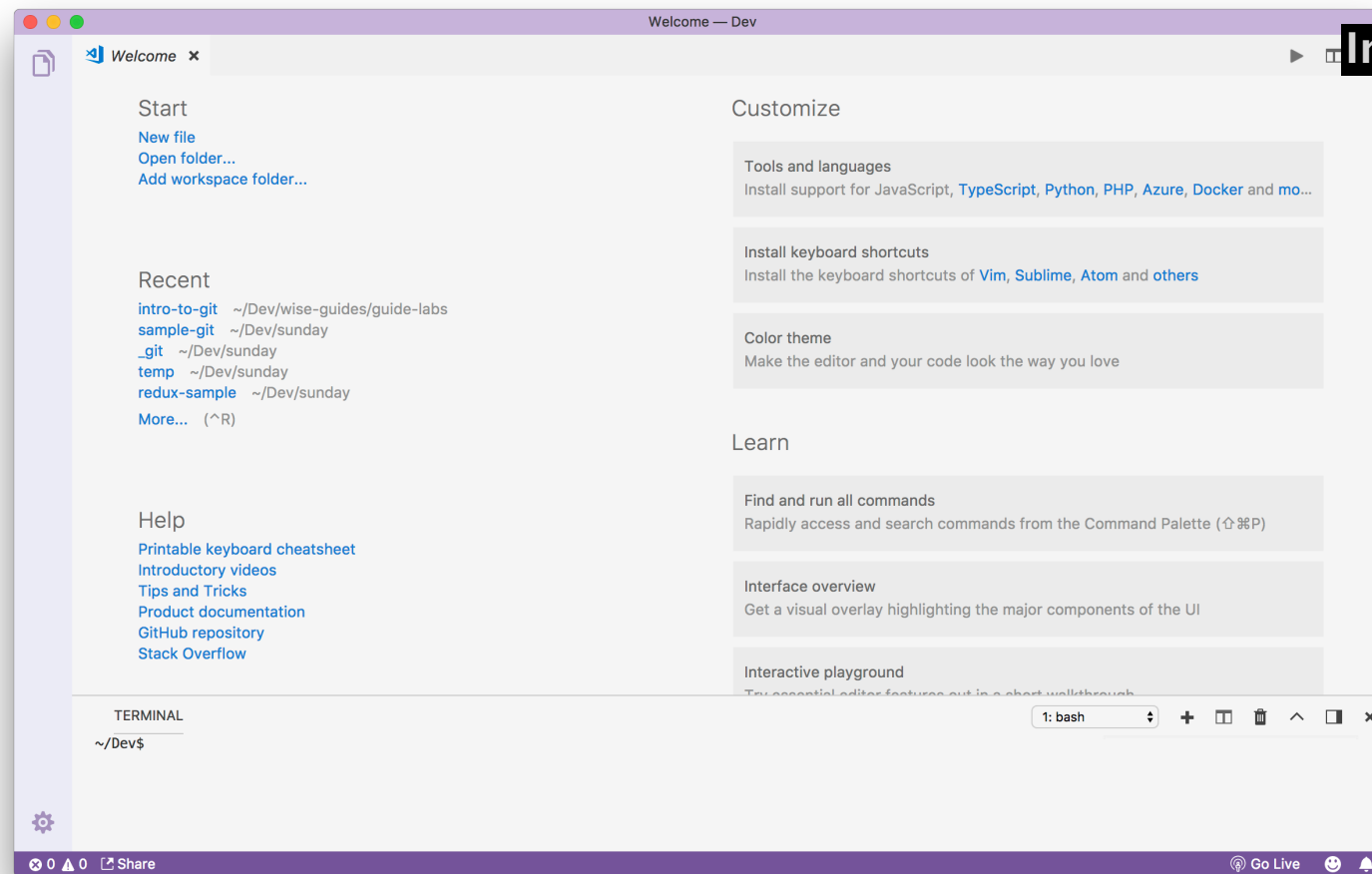
* tells VS Code to create the preceding element a certain number of times

If, instead of wanting to make one element a *child* of another, you want to create *siblings*, you can do so by substituting a + for the >. So this code...

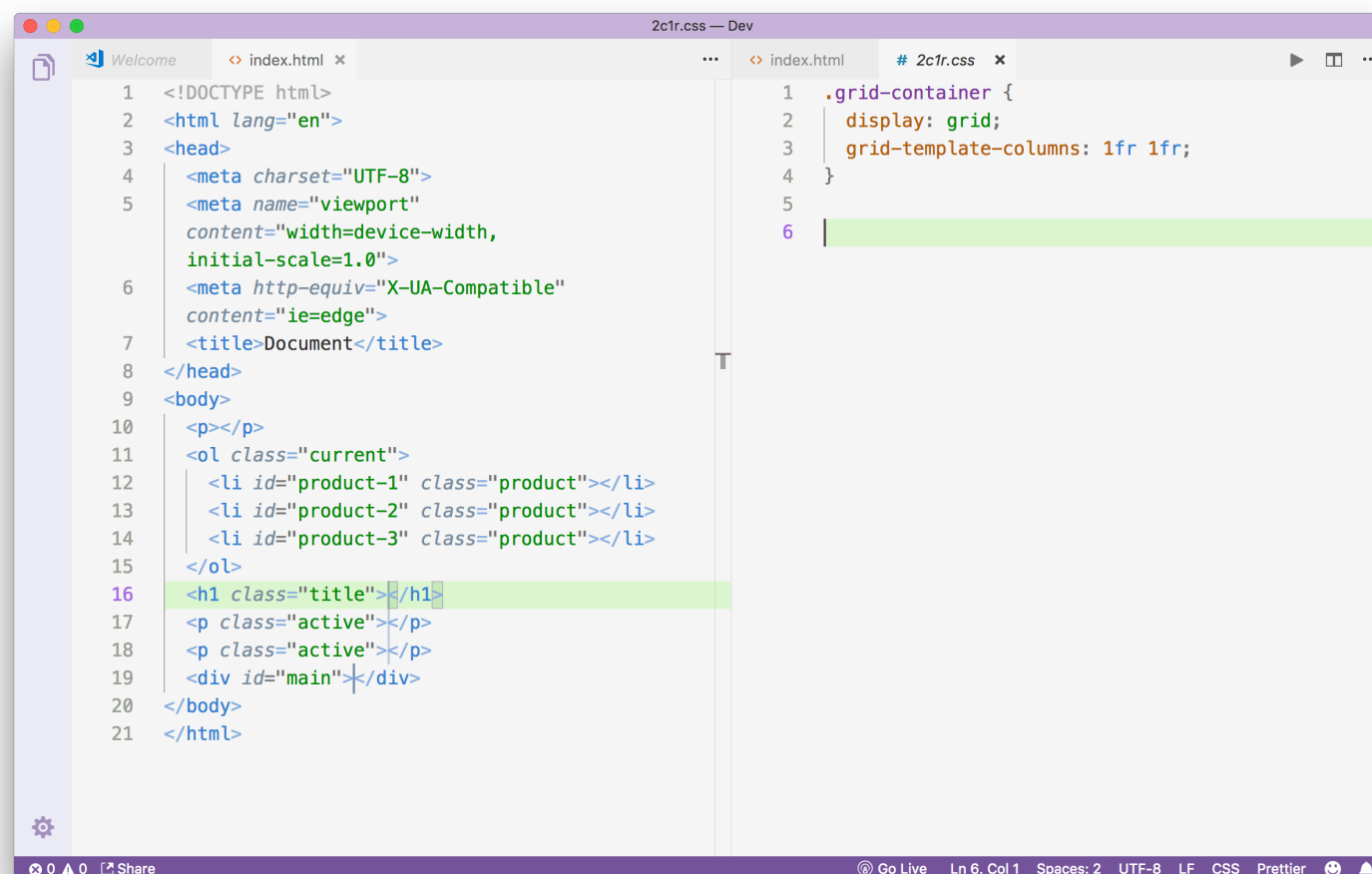
```
<h1 class="title"></h1>
<p class="active"></p>
<p class="active"></p>
<div id="main"></div>
```

...could be accomplished by using `h1.title+p.active*2+div#main`

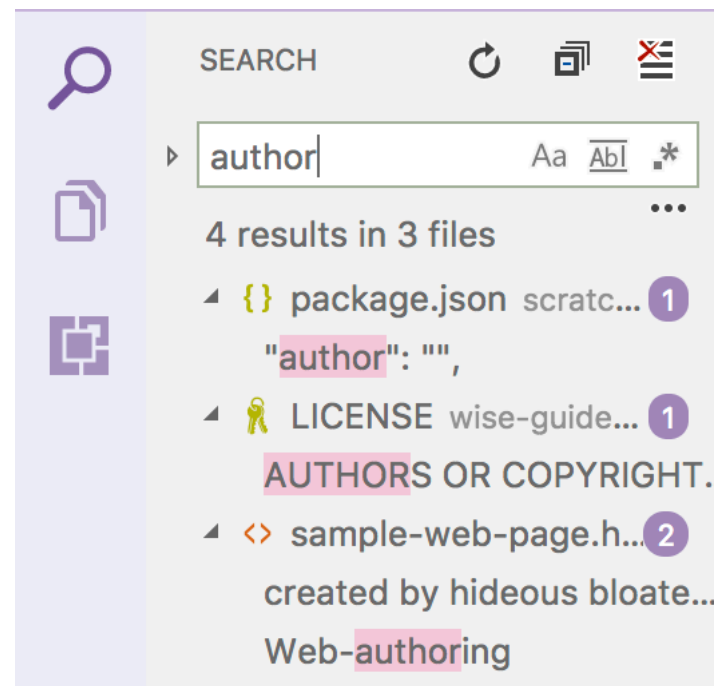
3. VS Code has terminal built into it. From VS Code main menu, select **View > Integrated Terminal**.



4. Sometimes it's useful to be able to see two files open in a split screen. From VS Code main menu, select **View > Split Editor**.

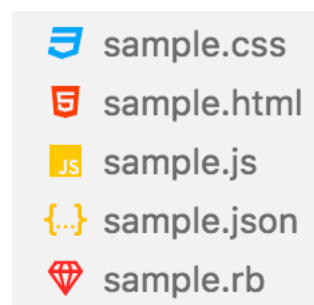


5. Here's something very handy: you know that, somewhere, you wrote some code that had the word, *author*, in it — but you can't remember where. Click on the magnifying glass in the left panel (called the *Activity Bar*) and type **author**, then press Enter. Your results will be all files with the word, *author*, in them — and even including the context that the word is used in.



6. You don't like the default icons used for file types in Explorer.

Boring icons for file types



Cool icons for file types that let people know you're hip

From VS Code main menu, click **View > Command Palette**. Type **icon** and choose **Preferences: File Icon Theme**. Select **Install Additional File Icon Themes**. Select the one named **Material Icon Theme**. Click on the **Install** button. If asked if you want to activate, click **Yes**.

7. Let's add some other extensions to VS Code. From the Activity Bar (on left of screen), click the  icon. You can search and install the following extensions:

- Bracket Pair Colorizer
- Beautify
- React Native Tools
- ESLint
- Live Server
- VS Live Share
- VS Live Share Audio

8. You can select from a somewhat dizzying selection of color schemes for VS Code. Press **Shift-Command-P** and then type `color`. From the accompanying selection choose **Preferences: Color Theme**. If you don't like the limited selection shown, click on the **Install Additional Themes** choice and enjoy!

9. Many of VS Code's defaults can be found (and edited) from VS Code's main menu: **Code > Preferences > Settings**. Here are the settings that we changed. (Feel free to make changes or ignore as you wish.) You can search for these in the search box at the top of the **User Settings** file.

- a. `"files.autoSave": "afterDelay"` ● — Auto-saves files after a short delay specified by this number (where 1000 equals 1 second)
- b. `"files.autoSaveDelay": 1000` ● —
- c. `"files.defaultLanguage": "JS"` ● — Sets default formatting for JavaScript
- d. `"editor.tabSize": 2` ● — Each tab indentation will equal 2 spaces
- e. `"editor.wordWrap": "on"` ● — Editor uses word-wrap
- f. `"explorer.confirmDragAndDrop": false` ● — Removes need to confirm dragging/dropping files in Explorer
- g. `"html.experimental.syntaxFolding": true` ● — See video
- h. `"editor.minimap.enabled": false` ● —
- i. `"emmet.includeLanguages": {"javascript": "javascriptreact"}` ● — Enables Emmet shortcuts for JavaScript and React
- j. `"editor.formatOnSave": true` ● — See video



10. A final few productivity enhancers:

- a. It's not uncommon to need to make exactly the same changes to multiple lines of code (perhaps you forgot to add a class name to an HTML element). VS Code offers a multi-cursor ability so that you can place your cursor at any number of places. Then, any keyboard actions you take will be done in all these places simultaneously. To make that work, use the **option** key while placing your cursor with your mouse.
- b. You can duplicate code easily by selecting one or more lines of code, then pressing **Shift-option-down arrow key**.
- c. If you highlight some snippet of code, you can auto-select the next occurrence of this snippet by using **command-D** key combination. You can do this repeatedly until all occurrences are found.