Creating a VS Code Theme

Sarah Drasner on Jun 8, 2018

Everyone has special and perhaps, particular, tastes when it comes to their code editor. There are literally thousands of themes out there, and for good reason: a thing of beauty and enhancement to productivity for one can be a hindrance to another.

It's been an item on my bucket list to create my own theme. I was coding very late the one night, well into the small hours of the morning. Everyone in my house was sleeping and so, as usual, the only light was the glow of my screen. I know it's not necessarily healthy to code like this, but it's literally the time I'm most productive: there are minimal distractions, I'm not dealing with work stuff, family stuff, friend stuff, or puppy stuff. I can focus.

I had some preferences set for the theme I had been using and, though they all worked well for daytime or plane rides, I always felt like something was missing for late night coding sessions. I decided it was time to craft my own theme.

We'll talk first about the general process for creating a theme in case you'd like to create one of your own, and then we'll dive into some of the research and testing that went into mine in particular to peek into the process.

(#fire-it-up) Fire It Up

Before you do anything, you're going to install vsce (short for Visual Studio Code Extensions) and establish yourself as a publisher. All of the instructions to do so are here (https://aka.ms/U8bd2v) . I know it looks like a lot, but it takes anywhere from 5-10 minutes, and then you'll never have to do it again, for any extension you create.

Now that you've got that under your belt, here are the steps you need to start work.

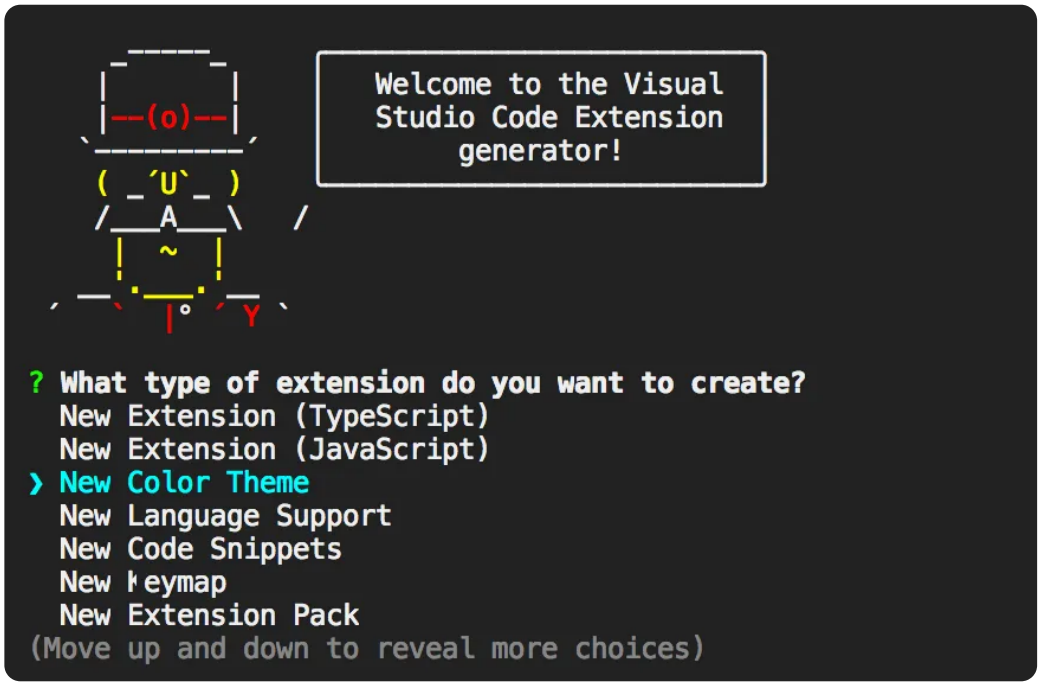
First, you need to run:

```Bash
npm install -g yo generator-code
```

This makes the generator globally available on your machine (meaning you can now create a theme in any directory). You can then execute this command to kick off your theme:

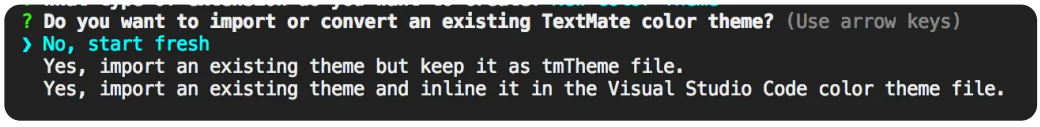
```bash
yo code
```

You will be prompted by a screen that looks like this:



> ***Hey!*** Note that I've used the arrows here to navigate to the "New Color Theme" option. Note also that this is how you'd want to make any other extension.

When selecting this, it asks if this is a new theme or if we want to import from an existing one. We want to create a new one.



Next, you'll have to answer a few other questions, including:

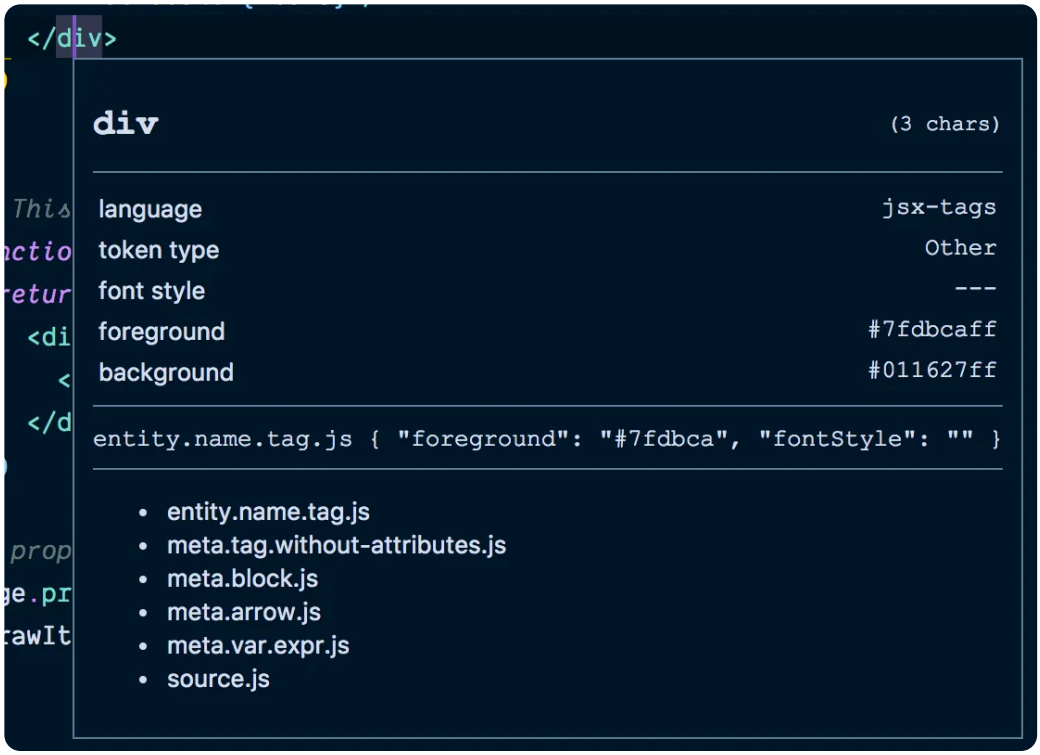
- What's the extension's name?
- What is the the identifier? (I just went with the name, that's probably typical.)
- What is the the description? (I just put something silly in initially. Don't worry, you can update this in your `package.json` in the future.)
- What's the publisher's name? (See earlier instructions.)
- What name should be shown to the user? (I used the same as the extension name.)
- Is this theme dark, light, or high contrast?

It will set you up with a base theme to start skinning your color preferences. The full scoop and all the details are here (https://aka.ms/Wa8ujj) . More details about themes in general are here (https://aka.ms/Cg43ed) .

(#test-drive) Test Drive

We have our base theme and we have some concepts for the palette. So, how do we test it out? When you open the directory with your theme, you can press `fn + f5` on Mac (or just `f5` on Windows) and a new window immediately pops open where you can test your theme! You'll see in the original theme window that you now have a little control panel where you can reload, pause, and stop. Don't forget to save before you do!

OK, now that you have the other window open, hit `Command + Shift + P` to get the command explorer. In there type, "Developer: Inspect TM Scopes" and you'll see a prompt come up that allows you to look through all the tags and attributes: it will tell you their color, their font styles, and how you need to target it.



There is one problem, though. There are a lot of things in the editor you can't target because VS Code will interpret that as you trying to drive the rest of the editor (i.e. the file

viewer, the terminal, and the search boxes). Here are the two ways I found to figure out the rest of the scopes:

- This page is extremely helpful (https://aka.ms/P4x2ct) in understanding some of the base things you need to configure. In fact, you might want to start with some of these.
- There are DevTools! You can open them the same way you do with Chrome: `Command + Option + I`. What I did was look for the color in the computed styles and look them up in the text editor to target them. You'll notice that the default in the DevTools is RGBA, so you will have to `Shift + click on the color` to change it's format until you get to the equivalent hex values. I could then scan through the matching colors in my theme json until I found the matching value and change it.

(#another-small-tip) Another Small Tip!

When I first started to develop the theme, I thought I would try forking someone else's theme as a starting point. I tried out Wes Bos' Cobalt Two (https://aka.ms/Cquk4n). Though I didn't end up using it, one thing he had that I found valuable was a *demos* directory with examples of all of a whole slew of different languages. I started by moving his over, but realized quickly that the files weren't long enough for my testing needs. So I created my own. In the course of correcting issues people filed, I also created a React stateless functional component example, a Ruby example, and of course I created a `.vue` single file component 😀 This is also helpful in maintenance because if people are seeing an issue on a file type I previously didn't test on, they can PR the file into the demos directory, and I can target what they're seeing. It makes duplication and testing really simple.

(#research) Research

Research for a code theme? Isn't that over the top? Probably! But I was genuinely curious: what would work best for legibility for the vast majority of people, while still being something I liked?

(#color-and-contrast) Color and contrast

The first step was considering accessibility. I always liked how solarized themes (http://ethanschoonover.com/solarized#features) made legibility a central theme to their palettes. I read about color retention and accessibility, and it turns out that men have a really high incidence of colorblindness (https://www.allaboutvision.com/conditions/colordeficiency.htm) (around 8% for men,