## **LINTING**

### **Style Guides**

Code style is important! Having a consistent set of style rules for things such as indentation or preferred quote style makes your code more maintainable and easier to read. There are several popular JavaScript style guides on the net that set standards for these types of things, and a little time spent reading them *will* make you a better developer.

1. The [Airbnb Style Guide](https://github.com/airbnb/javascript) is one of the most popular. It is also very well formatted and easy to read.
2. Google also has their own [style guide](https://google.github.io/styleguide/jsguide.html) for JavaScript.
3. The [JavaScript Standard Style](https://standardjs.com/rules.html). Used by companies like NPM and GitHub, among [others](https://standardjs.com/index.html#who-uses-javascript-standard-style).

### **Linting**

The style guides we mentioned above are full of really helpful advice for formatting, organizing and composing your code. But there are a *lot* of rules - it can be difficult to internalize them all. Linters are tools that will scan your code with a set of style rules and will report any errors to you that they find. In some cases, they can even auto-fix the errors! The following articles explain in more detail the benefits of using a linter while you code.

1. [This article](http://mikecavaliere.com/javascript-linting-what-developers-need-to-know/) gets right to the point… start here!
2. [This article](https://hackernoon.com/how-linting-and-eslint-improve-code-quality-fa83d2469efe) goes a little further by discussing exactly *how* linters do what they do.

There are multiple options for linting your JavaScript, but the most popular (and most common in the industry) is [eslint](https://eslint.org/). Getting it installed and the initial set-up is fairly simple.

1. [The official ‘Getting Started’ page](https://eslint.org/docs/user-guide/getting-started) is a good place to start. It covers installation and basic setup. The basic way to use this tool is to simply run the eslint command in your terminal with a specific file.
2. Far more useful are linting plugins for your favorite text editor. Most editor plugins allow you to automatically lint your code as you are writing it, and will show the errors right in the editor, which makes resolving them *much* simpler. We can’t cover *every* editor installation but some of the more popular are:
   1. Visual Studio Code - [The Plugin](https://marketplace.visualstudio.com/items?itemName=dbaeumer.vscode-eslint) and [a tutorial](http://shripalsoni.com/blog/configure-eslint-in-visual-studio-code/).
   2. Sublime Text - [The Plugin](https://github.com/roadhump/SublimeLinter-eslint) and [a tutorial](http://jonathancreamer.com/setup-eslint-with-es6-in-sublime-text/).
   3. Atom - [The Plugin](https://atom.io/packages/linter-eslint) and [a tutorial](https://medium.freecodecamp.org/how-to-set-up-eslint-in-atom-to-contribute-to-freecodecamp-3467dee86e2c).
   4. Vim - [Use the ALE plugin](https://github.com/w0rp/ale). If you use vim you already know what you’re getting into. ALE is a great plugin, but the setup and configuration can be a little tricky.

### **Prettier**

Prettier is *awesome*. It is similar to a linter, but serves a slightly different function. Prettier will take your JS code and then automatically format it according to a set of rules. Unlike a linter, it’s not looking for style errors, but specifically targeting the layout of your code and making intelligent decisions about things like spaces, indentation levels and line-breaks.

1. [This quick talk](https://www.youtube.com/watch?v=hkfBvpEfWdA) from prettier’s creator is a great introduction.
2. Give it a test drive [here](https://prettier.io/playground). Go ahead and copy/paste some of your old JavaScript code into that editor and see what happens.
3. Setup is simple. [The homepage](https://prettier.io/) links to tutorials for most popular editors.

Using prettier makes coding faster and easier! You don’t have to worry about nailing things like indentation, or remembering every semicolon because prettier will take care of those details for you