# Babel

As any language, Javascript also has versions named ECMAScript (short for ES). Currently, most browsers support ES5. ES5 used to be good even though it was painful to code in it. Remember, this not reading from inside callback functions? The new version of Javascript, ES6, also known as ES2015 (specs of the language were finalized in June 2015) makes Javascript great again. If you want to learn about ES6, check out the links at the end of this article. All the great features of ES6 come with one big problem — majority of browsers do not fully support them. That’s when Babel comes to play. Babel is a JS transpiler that converts new JS code into old ones. It is a very flexible tool in terms of transpiling. One can easily add presets such as es2015, es2016, es2017, or env; so that Babel compiles them to ES5.

# Webpack

Now that we know what Babel and ES6+ are, we would like to use that. We would also like to use SASS for our styles, PostCSS for autoprefixing. Plus, we would like to minify and uglify both our CSS and Javascript code. Webpack solves all of these problems using one config file (named webpack.config.js) and one CLI command webpack.

Webpack is a modular build tool that has two sets of functionality — Loaders and Plugins. Loaders transform the source code of a module. For example, style-loader adds CSS to DOM using style tags. sass-loader compiles SASS files to CSS. babel-loader transpiles JS code given the presets. Plugins are the core of Webpack. They can do things that loaders can’t. For example, there is a plugin called UglifyJS that minifies and uglifies the output of webpack.