

Prefixing

In this lesson, we'll learn how to setup autoprefixing.

Let's now move on to adding prefixing into our build.

Recall that in CSS you'll often need to prefix your code in order to ensure cross-browser compatibility. Like in the following example:

```
.myClass {  
  -webkit-transition: all 10s linear;  
  -moz-transition: all 10s linear;  
  -ms-transition: all 10s linear;  
  -o-transition: all 10s linear;  
  transition: all 10s linear;  
}
```

You could do this manually, or use a tool like [Autoprefixer](#), which automates the process entirely. So we don't have to check if a prefix is required.

Let's add autoprefixing into our build process by adding the following line to the end of the script:

```
"prefix-css": "postcss --use autoprefixer -b 'last 5 versions' css/style.c  
oncat.css -o css/style.prefix.css"
```

The script so far should look like so:

```
"scripts": {  
  "watch-sass": "node-sass sass/main.scss css/style.css --watch",  
  "compile-sass": "node-sass sass/main.scss css/style.comp.css",  
  "concat-css": "concat -o css/style.concat.css css/additional.css css/styl  
le.comp.css",  
  "prefix-css": "postcss --use autoprefixer -b 'last 5 versions' css/styl  
e.concat.css -o css/style.prefix.css"  
},
```

So here, `prefix-css` is our script name.

`postcss --use autoprefixer` autoprefixer is selected.

`-b 'last 5 versions'` specifies which browser versions we want our autoprefixes to cover (in this case the last 5).

`css/style.concat.css` is our input file.

`-o css/style.prefix.css` is our specified output file.

We're using the npm `autoprefixer` package, which will need to be installed by running:

```
npm install autoprefixer --save-dev
```

We also need to install `PostCSS` (autoprefixer is part of this plugin). We use the following command:

```
npm install postcss-cli --save-dev
```

And then run the script as follows:

```
npm run prefix-css
```

It will generate our `css/style.prefix.css` file.

Take a look at the code in this file; you'll see the browser prefixes have been automatically added for you.

This is great as we can now forget about prefixing and concentrate on writing clean code!

In the next lesson, we'll add compression to our process.