

I picked node-sass implementer for libsass because it is based on node.js.

Installing node-sass

- (Prerequisite) If you don't have npm, install [Node.js](#) first.
- `$ npm install -g node-sass` installs node-sass globally `-g`.

This will hopefully install all you need, if not read libsass at the bottom.

How to use node-sass from Command line and npm scripts

General format:

```
$ node-sass [options] <input.scss> [output.css]
$ cat <input.scss> | node-sass > output.css
```

Examples:

1. `$ node-sass my-styles.scss my-styles.css` compiles a single file manually.
2. `$ node-sass my-sass-folder/ -o my-css-folder/` compiles all the files in a folder manually.
3. `$ node-sass -w sass/ -o css/` compiles all the files in a folder automatically whenever the source file(s) are modified. `-w` adds a watch for changes to the file(s).

More usefull options like 'compression' [@ here](#). Command line is good for a quick solution, however, you can use task runners like Grunt.js or Gulp.js to automate the build process.

You can also add the above examples to npm scripts. To properly use npm scripts as an **alternative to gulp** read [this comprehensive article @ css-tricks.com](#) especially read about [grouping tasks](#).

- If there is no `package.json` file in your project directory running `$ npm init` will create one. Use it with `-y` to skip the questions.
- Add `"sass": "node-sass -w sass/ -o css/"` to scripts in `package.json` file. It should look something like this:

```
"scripts": {
  "test" : "bla bla bla",
  "sass": "node-sass -w sass/ -o css/"
}
```

- `$ npm run sass` will compile your files.