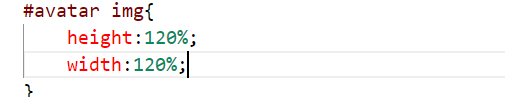
如果想要打包.css文件。





就需css-loader(先)和style-loader（后）。配置如下：

{

test:/\.css$/,

/\*\*

\* 解析.css文件需要style-loader和css-loader的配合。并且要有一定的先后顺序。

\* 不要忘记loader的安装时从下往上，从右向左的。

\* 应该先使用css-loader来解析import的.css文件，并且能分析需要打包的css文件和其他css文件的引用关系，最后将各个css智能的合并在一起，之后在使用style-loader

\* 将css-loader解析得到的cssText（个人认为是类似cssText的东西）,嵌入式的

\* 卸载header标签的style标签内。

\*/

/\*\*

\* 踩坑：注意多个loader的配置写法。

\* 在当使用多个loader的时候，应该让user是一个对象数组，

\* 每一个对象数组中有loader和option属性。

\*

\* （最开始的错误理解和操作）而不能将user写成一个对象，然后loader属性是一个数组，然后option属性

\* 是一个对象数组，每一个对象对应loader中的顺序。就像下面这样

\* use:['style-loader','css-loader']

\* option:[{...},{...}]

\*/

// use:['style-loader','css-loader']

use:[{

loader:'style-loader'

},{

loader:'css-loader'

}]

}

如果我们要打包sass，只需要安装sass-loader以及一定要和他一起安装的node-sass，然后进行配置即可。

npm install sass-loader node-sass webpack --save-dev

use:['style-loader','css-loader','sass-loader']

如果使用css属性，难免会使用浏览器厂商前缀，比如-o- -moz- -webkit- -ms-。

当然可以再编写代码的时候自动添加，但是这样会很乏力。

可以使用postcss-loader并结合autoprefixed插件来对编写的css3属性智能的添加浏览器厂商前缀，所以在写css3属性的时候可以不用考虑浏览器厂商前缀的问题了。

首先安装postcss-loader

npm i postcss-loader -d

然后在安装autoprefixer插件。

npm i autoprefixer -d

Postcss-loader需要一个单独的配置文件。在webpack.config.js同目录下

创建postcss.config.js。

配置postcss.config.js:

module.exports = {

plugins:[

require('autoprefixer')

]

}

其他关于postcss-loader的内容依旧可以查阅文档。

// index.js

body{

#avatar img{

height:120%;

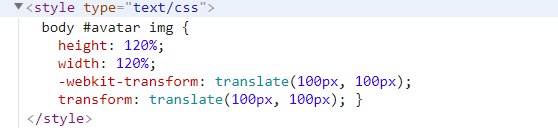
width:120%;

transform:translate(100px,100px);

}

}

浏览器中生成的效果为：



这里是谷歌浏览器，明显自动添加了-webkit-的厂商前缀。

当我们在Sass中使用@import语句引入了其他的Sass文件的时候，可能就会直接跳过最下方的两个loader,也就是postcss-loader和sass-loader，直接去使用css-loader。至于为什么，现在还想不通，老师也没有详细指出。之后需要额外进行探索。

解决方式正是为css-loader配置一个选项importLoaders:2。意思是在执行css-loader之前还需要执行两个loader。这是就能保证执行postcss-loader和sass-loader了。

根据网上同学的问题反馈发现，有些时候可能会遇到了及时按照下面方式也无法自动添加厂商前缀的问题，只是由于postcss属于后加载器，只需要让postcss-loader与sass-loader调换一下位置即可。对于这一点，我目前也只是一知半解，至于具体原因需要日后再深入调查。

{

test:/\.scss$/,

use:[

'style-loader',

{

loader:'css-loader',

options:{

importLoaders:2

}

},

'sass-loader',

'postcss-loader'

]

}