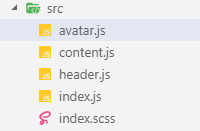
关于css模块化(css-loader)的一些知识，可以查看阮一峰老师的博客。

<http://www.ruanyifeng.com/blog/2016/06/css_modules.html>

一般情况下css都是全局性的。如果像上文index.js一样引用.scss文件，会使所有文档内容都收到css的样式影响。

import "./index.scss";

假如现在在index.js的同级目录下新建一个avatar.js



// avatar.js

import avatar from '../img/类nico.jpg';

function createAvatar(){

var img = new Image();

document.getElementById("avatar").appendChild(img);

img.src = avatar;

}

export default createAvatar;

// index.js

import Header from './header.js';

import Contents from './content.js';

import avatar from '../img/类nico.jpg';

import createAvatar from './avatar.js';

import "./index.scss";

new Header(2);

new Contents(3);

var img = new Image();

img.src = avatar;

console.log(avatar);

document.getElementById("avatar").appendChild(img);

createAvatar();

很明显，由scss引入的css样式就会作用于这两个图片。

但是我并不希望这样，我希望在不为两个图片设置不同类名的情况下，也不修改scss的选择器。

这时候就应该打开css-loader的模块化功能。

为css-loader添加选项：

{

loader:'css-loader',

options:{

importLoaders:2,

modules:true // 开启css模块化

}

}

// avatar.scss

.avatar{

height:120%;

width:120%;

transform:translate(100px,100px);

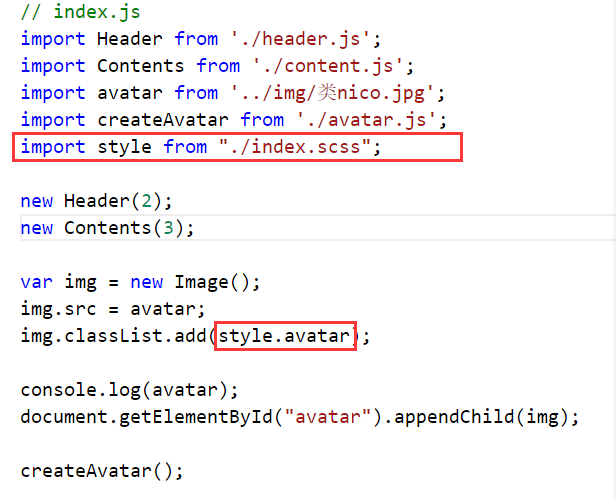
}

修改index.js代码，为只需要应用.avatar的元素进行如下操作：

因为把css作为一个模块，因此需要一个变量来接受css.

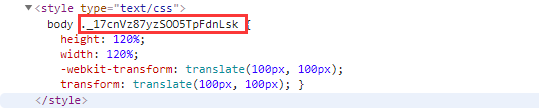
而不能直接import ‘../.scss’;

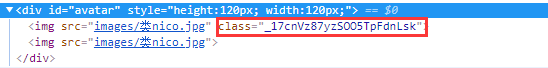
Element.classList.add(style.avatar);



根据我的初步探索，style后面的属性似乎应该是一个类选择器。

可以发现，在index.html的style标签中的代码变成了这个样子。







由此可见，为了实现css局部作用，只不过是为每一个css模块下的类转换成为一个独一无二的hash值。

需要注意的简单问题：

如果scss的文件是这样的：

// avatar.scss

body{

.avatar{

height:120%;

width:120%;

transform:translate(100px,100px);

}

}

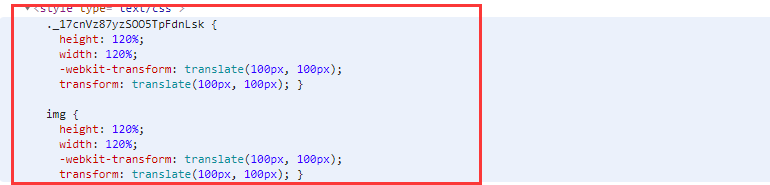
img{

height:120%;

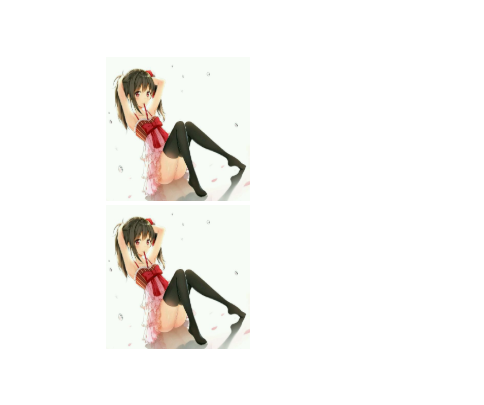
width:120%;

transform:translate(100px,100px);

}

其他不变，最后编译在index.html中的内容却是这样的。

呈现的效果：



事实上需要注意的是，只是把需要转换为局部变量的css属性换成了hash。

其他没有要求的（也就是没有style.xxx）的属性依旧不会变化，还是会被style-loader添加在了style标签内。