

05.Loader 打包 CSS 文件

学习要点：

1. 问题需求
2. 安装部署

本节课我们来开始学习一下在 `module` 下设置打包 `css` 文件的功能。

一. 问题需求

1. 通过之前的学习，我们知道如何打包 `js` 模块和 `html` 页面，并融合；
2. 还有一种东西需要处理，就是 `css` 样式以及 `less`、`sass` 等预处理样式；
3. 思路也是一样的，调试的时候可以试试监听，打包的时候也能融合；
4. 这里需要使用 `module` 这个属性，并需要安装使用以下三种插件：
 - (1) `.css-loader`：读取和编译 `css` 文件，转换为样式字符模块；
 - (2) `.style-loader`：将 `css` 插入到 `JavaScript` 中；
 - (3) `.less-loader`：读取和编译 `less` 预处理样式，转换为 `css` 文件；

二. 安装部署

1. 首先，我们先创建两个基本的 `css` 文件，另一个引入其中一个；

```
/*font.css*/
h1 {
  color: blue;
  font-size: 200px;
}

/*base.css*/
@import "font.css";
body {
  background-color: gray;
}
```

PS：和打包 `html` 文件一样，`css` 文件并不需要你手动去 `link` 引入，让插件处理；

2. 然后在 `.js` 文件中，把 `.css` 文件通过 `require` 引入进来；

```
require('./base.css');
```

3. 安装所需的 `css-loader` 插件和 `style-loader` 插件；

```
npm i css-loader -D
npm i style-loader -D
```

PS：安装完成后，可以在 `package.json` 检查版本号，确认已安装；

4. 在 `webpack.config.js` 中，使用 `module` 进行 `css` 文件部署工作；

```
//模块
module: {
  //规则
  rules: [
    {
      //规则获取需要部署的文件后缀
      test: /\.css$/,
      //style-loader 将 css 字符串插入到 javascript
      //通过浏览器工具样式被动态插入到<style>标签内
      //css-loader 将 css 文件编译成字符串给 style-loader 处理
      //所以，这里数组的执行顺序是从右到左执行，否则失败；
      use : ['style-loader','css-loader']
    },
  ]
},
```

5. 还有一种 `css` 编码方式是：预处理样式，比如 `Less`，`sass` 等；
6. 首先创建一个 `less.less` 预处理样式文件，代码 `css` 一样即可，然后引入；

```
require('./less.less');
```

7. 我们这里简单使用一下 `Less` 先做一个基础了解，先安装 `less-loader`；

```
npm i less-loader -D
```

```
{
  test: /\.less$/,
  //less-loader 会先将.less 文件转换为.css 文件，然后再向左边处理
  use : ['style-loader', 'css-loader', 'less-loader']
}
```