

# 一：css文件

1: 导入css文件

```
9 // 使用 import 语法, 导入 css样式表
10 import './css/index.css'
11 // 注意: webpack, 默认只能打包处理 js 类型的文件, 无法处理 其它的非 js 类型的文件;
```

在webpack格式的js文件中, 只能 import Js类型的文件, 无法处理其他的 非js的文件  
不然会出现如下的报错: 没有合适的加载器

```
ERROR in ./src/css/index.css
Module parse failed: Unexpected token (1:2)
You may need an appropriate loader to handle this file type.
| 11{
|   list-style: none;
| }
|
@ ./src/main.js 10:0-24
@ multi ./node_modules/webpack-dev-server@2.9.3/webpack-dev-server/client?http://localhost:3000 webpack/hot/dev-server ./src/main.js
Child html-webpack-plugin for "index.html":
```

2: 如果想打包处理css文件, 则需要安装

```
cnpm i style-loader css-loader -D
```

```
npm install --save-dev style-loader css-loader
```

```
npm install --sava-dev style-loader css-loader
```

3: 打开webpack.config.js这个配置文件, 在里面, 新增一个配置节点 module,  
它是一个对象, 在这个module对象身上, 有个rules属性, 这个rules属性是一个数组, 在这个数组中, 存放了所有第三方文件的匹配和 处理规则

如下:

```
module: {
  rules: [
    {
      test: /\.css$/,
      use: [
        'style-loader',
        'css-loader'
      ]
    }
  ]
}
```

```
npm install --sava-dev style-loader css-loader
```

4:

注意： webpack 处理第三方文件类型的过程：

1. 发现这个 要处理的文件不是JS文件，然后就去 配置文件中，查找有没有对应的第三方 loader 规则
2. 如果能找到对应的规则， 就会调用 对应的 loader 处理 这种文件类型；
3. 在调用loader 的时候，是从后往前调用的；
4. 当最后的一个 loader 调用完毕，会把 处理的结果，直接交给 webpack 进行 打包合并，最终输出到 bundle.js 中去

## 二： less文件

- 1:安装less的loader :cnpm i less-loader -D
- 2:安装less-loader内部的依赖包 cnpm i less -D
- 3:导入js文件
- 4: 配置webpack.config.js

三： scss文件：

- 1: 安装scss的loader: cnpm i sass-loader -D
- 2:安装node-sass 使用命令 cnpm i node-sass -D

```
18 // ],
19 module:{ //配置第三方模块的匹配规则
20     rules:[
21         { test: /\.css$/, use: ['style-loader', 'css-loader'] }, // 配置处理 .css 文件的第三方loader 规则
22         { test: /\.less$/, use: ['style-loader', 'css-loader', 'less-loader'] }, //配置处理 .less 文件的第三方 loader
23         { test: /\.scss$/, use: ['style-loader', 'css-loader', 'sass-loader'] }, // 配置处理 .scss 文件的 第三方 load
24     ]
25 }
```

```
<> index.html  webpack.config.js
src ▸ JS main.js ▸ ...
1  import $ from 'jquery'
2  import './css/index.css'
3  import './css/index.less'
4  import './css/index.scss'
```

