

17.HMR 热替换

学习要点：

1.HMR 热替换

本节课我们来开始学习一下了解下 HMR 热替换的功能。

一. HMR 热替换

1. 首先，页面调试有一个问题，就是当我们编辑了 `css` 或 `js`，它会自动刷新；
2. 但如果有大量的 `css` 或 `js`，一个小修改，就要全部刷新，影响开发性能；
3. 而 HMR 热替换就是解决这个问题，当 `css` 或 `js` 修改时，只刷新修改的部分；
4. 对于 `html` 文件，不需要热替换，就一个文件；
5. 开启 HMR 热替换，可以在启动服务时加上 `--hot`，或者在 `devServer` 配置：

```
devServer: {  
  hot : true  
},
```

6. 对于 `css` 的热替换，需要使用 `style-loader`，而现在被抽离 `css` 插件替换了；
7. 所以，我们先要注释掉这个插件，改为 `style-loader`；

```
test: /\.scss$/,  
//从右向左执行  
use : [  
  //MiniCssExtractPlugin.loader,  
  'style-loader',  
  'css-loader',  
  'sass-loader'  
]
```

8. 当开启了 HMR 热替换之后，发现 `html` 文件修改后无法自动更新了；
9. 此时，我们需要在入口文件加入这个 `html` 文件，即可；

```
entry: [  
  './src/js/index.js',  
  './src/index.html'  
],
```

10. 对于 `js` 的热替换，只要在入口文件，对加载的其它 `js` 文件进行设置即可；

```
import name from './module'  
console.log(name)  
if (module.hot) {  
  module.hot.accept('./module', () => {  
    console.log(name)  
  })  
}
```