

04.Plugin 部署 Html 插件

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 Plugin 中部署 Html 插件，让调试打包更方便。

一. 问题需求

1. 目前为止，我们只能打包 js 文件模块，事实上：Html 文件也需要打包到一起；
2. 也就是说，我们需要把 .js 和 .html 文件都要打包到 dist 目录中去；
3. 那么 src 目录也应该存放相应的 .html 文件，之前放在外面是方便测试；
4. 如果我们设定 .html 名称为 index.html，放入 src，那如何连接 .js 文件呢？
5. 预想，如果会打包到 dist，那么直接路劲：src="bundle.js"；
6. 可以这样，调试的时候，路径又会产生问题？那这些问题，都直接交给插件处理吧！

二. 安装部署

1. 首先，安装 html-webpack-plugin 插件，本地安装即可；

```
npm i html-webpack-plugin -D
```

PS：安装好后，可以在 package.json 中看到版本号；

2. 我们删除掉 src 外面的 .html 文件，在 src 建立 index.html 文件，用于打包；
3. 我们并不需要在 .html 文件中使用 <script> 标签引入 .js 文件，让插件处理；
4. 在 webpack.config.js 中，引入插件，具体如下：

```
//html 插件
const HtmlWebpackPlugin = require('html-webpack-plugin');

//插件
plugins: [
  new HtmlWebpackPlugin({
    template: './src/index.html',      //值为默认值，源文件
    filename: 'index.html'            //值为默认值，打包生成的文件名
  })
],

//devServer, publicPath 可以忽略
devServer: {
  //publicPath : '/dist',
  port : 3000,
  stats : "minimal",
},
```

5. 经过上述的一系列操作，解决了两个问题，打包后引入.js 文件，会自动引入；
6. 在使用 DevServer 时，localhost:3000 直接访问的就是 index.html；
7. 如果想要.html 和.js 打包后存在不同目录，可设置，并会自动链接；

```
filename : 'js/bundle.js',  
filename: "html/index.html"
```

```
<script src="../../js/bundle.js"></script>
```

PS: 打包后，上节课的 source map 生成了一个.map，这个在火狐刷新会引起弹窗；

PS: 只要使用 inline-source-map，不生成文件，在最后生成一行文本即可；