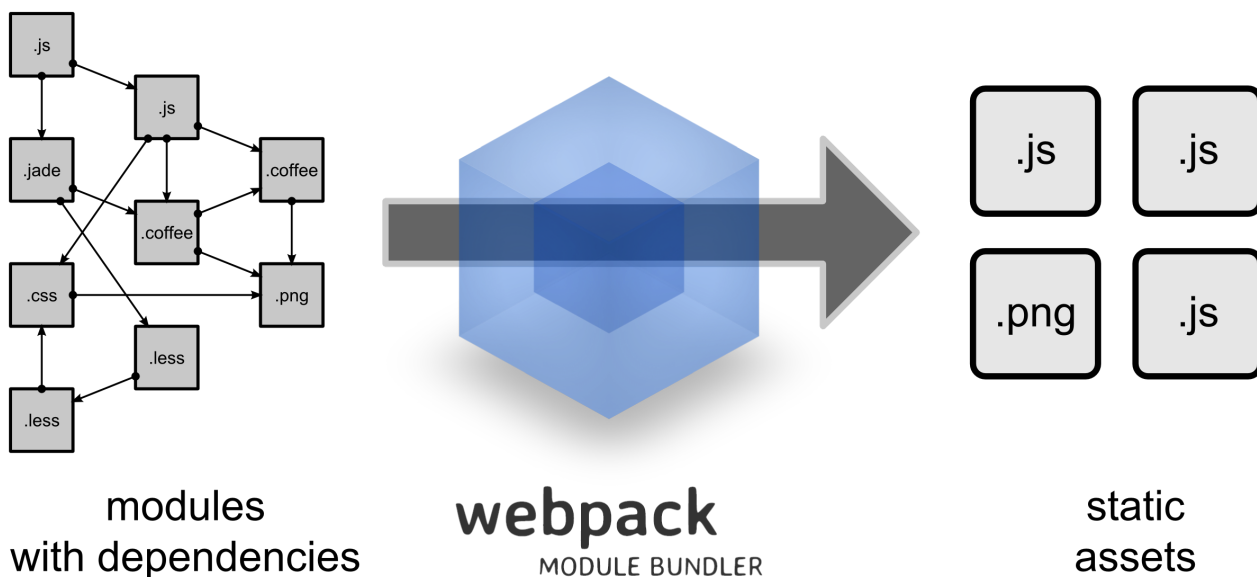


一、什么是Webpack

Webpack 是一个前端资源加载/打包工具。它将根据模块的依赖关系进行静态分析，然后将这些模块按照指定的规则生成对应的静态资源。

从图中我们可以看出，Webpack 可以将多种静态资源 js、css、less 转换成一个静态文件，减少了页面的请求。



二、Webpack安装

1、全局安装

```
1 npm install -g webpack webpack-cli
```

2、安装后查看版本号

```
1 webpack -v
```

三、初始化项目

1、创建webpack文件夹

进入webpack目录，执行命令

```
1 npm init -y
```

2、创建src文件夹

3、src下创建common.js

```
1 exports.info = function (str) {  
2     document.write(str);  
3 }
```

4、src下创建utils.js

```
1 exports.add = function (a, b) {  
2     return a + b;  
3 }
```

5、src下创建main.js

```
1 const common = require('./common');  
2 const utils = require('./utils');  
3  
4 common.info('Hello world!' + utils.add(100, 200));
```

四、JS打包

1、webpack目录下创建配置文件webpack.config.js

以下配置的意思是：读取当前项目目录下src文件夹中的main.js（入口文件）内容，分析资源依赖，把相关的js文件打包，打包后的文件放入当前目录的dist文件夹下，打包后的js文件名为bundle.js

```
1 const path = require("path"); //Node.js内置模块
2 module.exports = {
3   entry: './src/main.js', //配置入口文件
4   output: {
5     path: path.resolve(__dirname, './dist'), //输出路径, __dirname: 当前文件所在路径
6     filename: 'bundle.js' //输出文件
7   }
8 }
```

2、命令行执行编译命令

```
1 webpack #有黄色警告
2 webpack --mode=development #没有警告
3 #执行后查看bundle.js 里面包含了上面两个js文件的内容并惊醒了代码压缩
```

也可以配置项目的npm运行命令，修改package.json文件

```
1 "scripts": {
2   //...,
3   "dev": "webpack --mode=development"
4 }
```

运行npm命令执行打包

```
1 npm run dev
```

3、webpack目录下创建index.html

引用bundle.js

```
1 <body>
2   <script src="dist/bundle.js"></script>
3 </body>
```

4、浏览器中查看index.html

五、CSS打包

1、安装style-loader和 css-loader

Webpack 本身只能处理 JavaScript 模块，如果要处理其他类型的文件，就需要使用 loader 进行转换。

Loader 可以理解为是模块和资源的转换器。

首先我们需要安装相关Loader插件，css-loader 是将 css 装载到 javascript；style-loader 是让 javascript 认识css

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

2、修改webpack.config.js

```
1 const path = require("path"); //Node.js内置模块
2 module.exports = {
3   //...,
4   output:{},
5   module: {
6     rules: [
7       {
8         test: /\.css$/, //打包规则应用到以css结尾的文件上
```

```
9         use: ['style-loader', 'css-loader']
10     }
11 ]
12 }
13 }
```

3、在src文件夹创建style.css

```
1 body{
2     background: pink;
3 }
```

4、修改main.js

在第一行引入style.css

```
1 require('./style.css');
```

5、浏览器中查看index.html

看看背景是不是变成粉色啦？