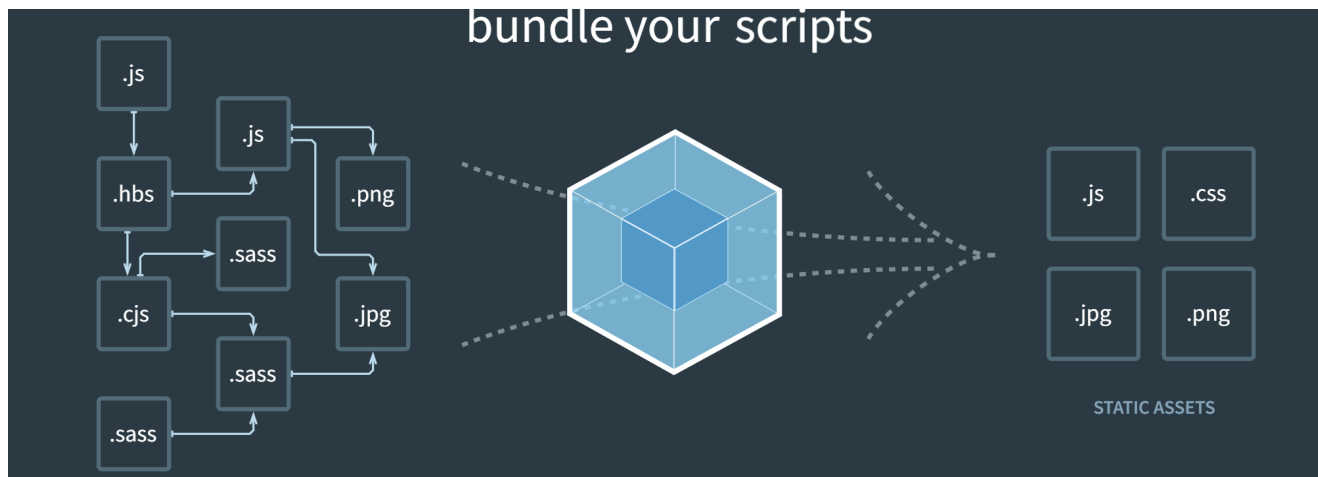


# 1.安装webpack

有很多模块互相依赖的时候，会用到webpack打包工具  
将项目打包成浏览器能直接解析的node包



```
1 npm init -y //初始化项目
2 npm i webpack webpack-cli -D //安装 webpack 和webpack-cli脚手架工具到开发依赖
```

```
1 npx webpack//打包webpack
```

## 入口文件

### 创建入口的js文件

```
1 const path = require('path');
2
3 module.exports = {
4   entry: './src/index.js',
5   output: { //出口的配置
6     path: path.resolve(__dirname, 'dist'),//出口文件地址
7     filename: 'bundle.js',//文件名字
8   },
9   mode: 'production' ///模式名字
10 };
11 };
```

## 出口 地址

```
1 output: { //出口的配置
2   path: path.resolve(__dirname, 'dist'),//出口文件地址
3   filename: 'bundle.js',//文件名字
```

```
4
5  },
```

## 模式 打包的模式

```
1  mode: 'production' //开发模式->development
```

## 插件

插件用来扩展webpack的功能，比如服务器插件，可以自己打开浏览器自己运行，不用去手动运行

```
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1",
  "build": "webpack",
  "dev": "webpack serve --open --port 8888"
```

```
1  npm i webpack-dev-server -D //安装 webpack-dev-server
2  //使用
3  --open 打开浏览器 --port 指定端口
```

```
1  //入口文件
2  const path = require('path');
3
4  module.exports = {
5    //...
6    devServer: {
7      static: { //静态资源的路径
8        directory: path.join(__dirname, 'public'), //public默认路径
9      },
10     compress: true,
11     port: 9000, //端口
12   },
13   };
```

## style-loader css-loader css文件预处理器

webpack是用JS写的，运行在node环境，所以默认webpack打包的时候只会处理JS之间的依赖关系!!!

```
1 npm i style-loader css-loader //css预处理器
2 npm i less less-loader //less 预处理器
3 npm i sass sass-loader //sass 预处理器
```

## less scss 文件预处理器 安装 处理

```
1  module: {
2      rules: [
3          //css 配置文件
4          {
5              test: /\.css$/i,
6              use: ["style-loader", "css-loader"],
7          },
8          //less 配置文件
9          {
10             test: /\.less$/i,
11             use: [
12                 // compiles Less to CSS
13                 "style-loader",
14                 "css-loader",
15                 "less-loader",
16             ],
17         },
18         //sass配置文件 依赖style的loader
19         {
20             test: /\.s[ac]ss$/i,
21             use: [
22                 // Creates `style` nodes from JS strings
23                 "style-loader",
24                 // Translates CSS into CommonJS
25                 "css-loader",
26                 // Compiles Sass to CSS
27                 "sass-loader",
28             ],
29         },
30     ],
31 }
```

## 处理图片

### 安装处理图片的loader

```
1 npm i file-loader -d //处理图片的loader
```

## 引入

```
1 import img from './file.png';
```

## 在配置文件中加载

```
1 {
2     test: /\. (png|jpe?g|gif)$/i,
3     use: [
4         {
5             loader: 'file-loader',
6         },
7     ],
8 }
```

## 更高级的处理转 base64格式，和阈值

### 1, 安装 url-loader

```
1 npm i url-loader -d
```

## 配置

```
1 {
2     test: /\. (png|jpg|gif)$/i,
3     use: [
4         {
5             loader: 'url-loader',
6             options: {
7                 limit: 8192, // 阈值
8             },
9         },
10    ],
11 }
```

```
10     ],  
11     },
```

## 处理html

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

### 配置

```
1  const HtmlWebpackPlugin = require('html-webpack-plugin')  
2  plugins: [  
3    new HtmlWebpackPlugin({  
4      template: './public/index.html',  
5      filename: 'index.html'  
6    })  
7  ],
```

## 处理单文件组件

```
1 npm install -D vue-loader@15 vue-template-compiler  
2 npm i vue -S
```