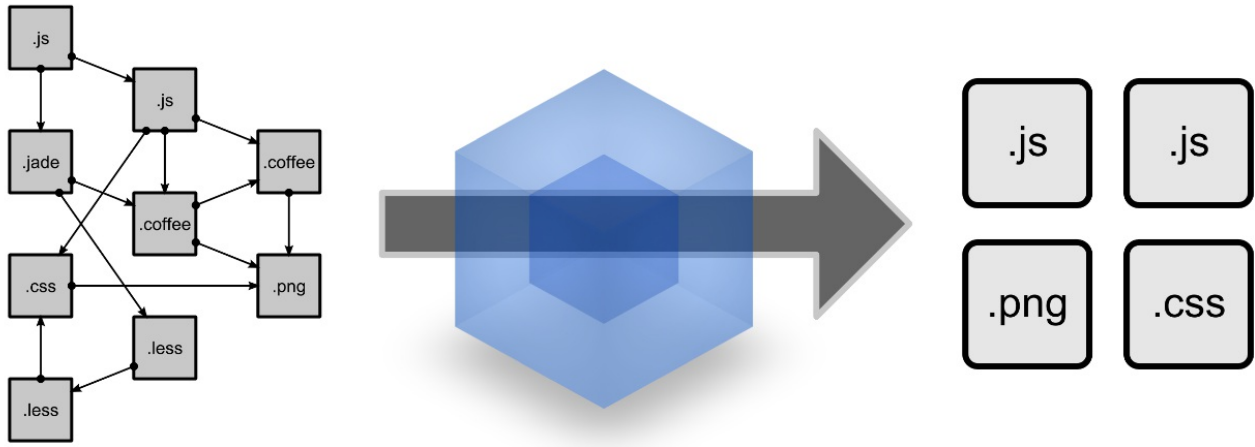


1-基础介绍

<https://github.com/webpack/webpack> 打包工具源代码

<https://webpack.github.io/>

https://webpack.github.io 官网文档



常用设计技术

1-代码分割

2-加载时间更少 loader

3-静态的资源都可以打包

4-适合大型项目

5-模块热更新

安装webpack

初始化项目

```
~/ cd imooc
~/imooc/ mkdir webpack-test
~/imooc/ cd webpack-test
~/imooc/webpack-test/ npm init
```

安装webpack

```
~/imooc/webpack-test/ npm install webpack --save-dev
```

```
IS this OK? (yes)
~/imooc/webpack-test/ npm install webpack --save-dev
```

编写文件hello.js, 开始打包

```
~/imooc/webpack-test/ webpack hello.js hello.bundle.js
```

为了支持CSS

```
~/imooc/webpack-test/ npm install css-loader style-loader --save-dev
```

使用

```
require('css-loader!./style.css')
```

如果要让背景色重新

```
require('style-loader!css-loader!./style.css')
```

也可以 在命令行加loader

```
~/imooc/webpack-test/ webpack hello.js hello.bundle.js --module-bind 'css=style-loader!css-loader'
```

显示进度条

```
~/imooc/webpack-test/ webpack hello.js hello.bundle.js --module-bind 'css=style-loader!css-loader' --progress --display-modules
```

```
require("style-loader!css-loader!./style.css");  
function hello(){  
  document.write("hello word");  
}  
hello();
```

使用webpack.config.js 配制方式运行脚本。

```
index.html  webpack.config.js  Telemetry Consent  
1  module.exports = {  
2    entry: './src/script/main.js',  
3    output: {  
4      path: './dist/js',  
5      filename: 'bundle.js'  
6    }  
7  }
```

面是webpack1写的方式，改写如下

```
var fileDir = __dirname;  
var path = require('path');  
  
module.exports = {  
  entry: './src/script/main.js',  
  output: {  
    path: path.resolve(fileDir, './dist/js'),  
    filename: 'bundle.js'  
  }  
}
```

定义脚本

```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "webpack": "webpack --config webpack.config.js --progress  
--display-modules --colors --display-reason"  
},
```

运行脚本

```
[0] ./src/script/main.js 24 bytes {0} [0]  
~/imooc/webpack-demo/ npm run webpack
```