

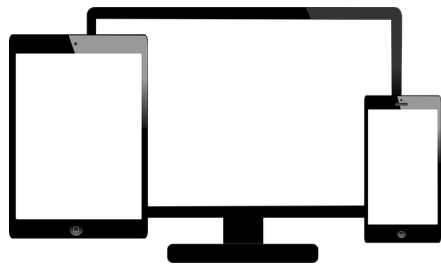
玩转 webpack

腾讯IVWEB 程柳锋



扫码试看/订阅
《玩转 webpack》

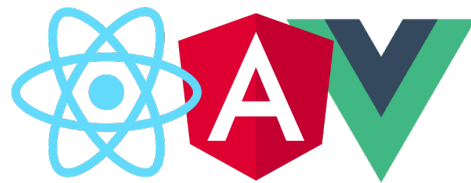
课程介绍



Web 应用多元化

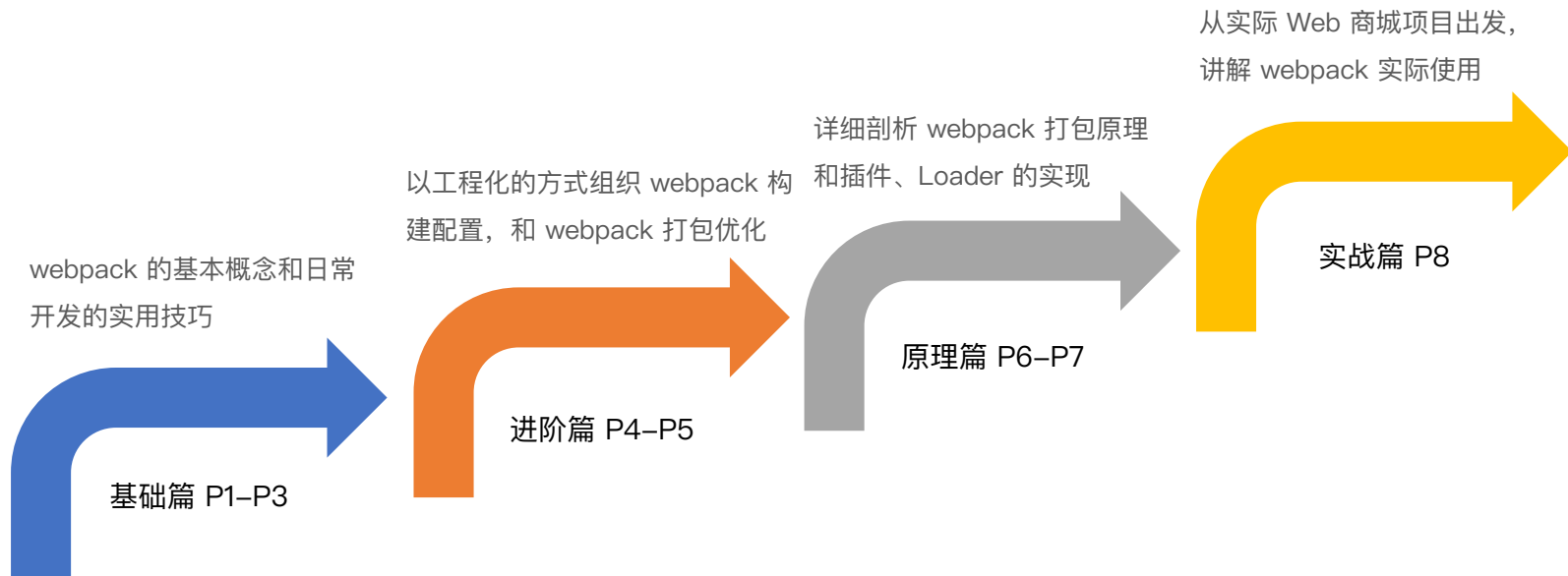


组件化



框架 DSL 解析

内容综述



目录

CONTENTS

- 01 | 基础篇：webpack 与构建发展简史
- 02 | 基础篇：webpack 基础用法
- 03 | 基础篇：webpack 进阶用法
- 04 | 进阶篇：编写可维护的 webpack 构建配置
- 05 | 进阶篇：webpack 构建速度和体积优化策略
- 06 | 原理篇：通过源码掌握 webpack 打包原理
- 07 | 原理篇：编写 Loader 和插件
- 08 | 实战篇：React 全家桶 和 webpack 开发商城项目

为什么需要构建工具?

转换 ES6 语法

转换 JSX

CSS 前缀补全/预处理器

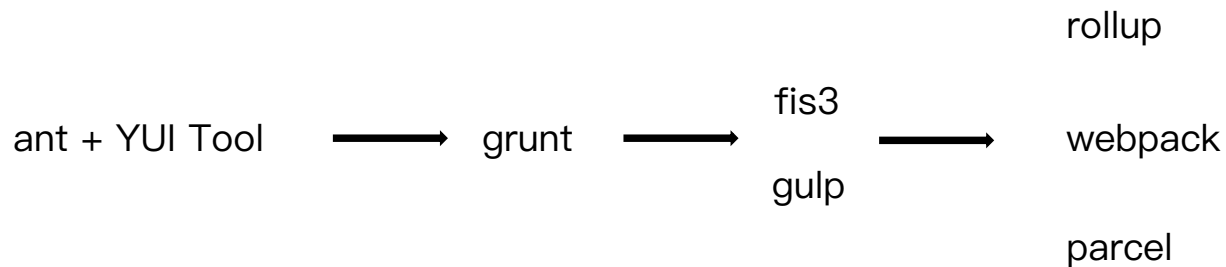
压缩混淆

图片压缩

Current aligned		Usage relative	Date relative	Apply filters	Show all	?	
IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini
	12-14	2-53	4-59	3.1-10	10-46	3.2-10.2	
	1 6 15	2 54-59	1 60	4 5 10.1	1 47	4 5 10.3	
6-10	6 16-17	60-63	61-70	11-11.1	48-56	11-11.4	
11	6 18	64	71	12	57	12.1	all
		65-66	72-74	TP			

ES6 module 主流浏览器支持情况

前端构建演变之路



为什么选择 webpack?

	webpack	grunt	gulp
定义	Module bundler	Task runner	Task runner
语言	JavaScript	Node.js	Node.js
发布时间	2012.3	2012.6	2013.7
GitHub stars	40766	11796	29427
周下载量	3,385,392	478,876	816,228

为什么选择 webpack?

社区生态丰富

配置灵活和插件化扩展

官方更新迭代速度快

初识 webpack：配置文件名称

webpack 默认配置文件：webpack.config.js

可以通过 webpack --config 指定配置文件

初识 webpack: webpack 配置组成

```
module.exports = {  
  entry: './src/index.js', ..... ❶ 打包的入口文件  
  output: './dist/main.js', ..... ❷ 打包的输出  
  mode: 'production', ..... ❸ 环境  
  module: {  
    rules: [ ..... ❹ Loader 配置  
      { test: /\.txt$/, use: 'raw-loader' }  
    ]  
  },  
  plugins: [ ..... ❺ 插件配置  
    new HtmlWebpackPlugin({  
      template: './src/index.html'  
    })  
  ]  
};
```


零配置 webpack 包含哪些内容?

```
module.exports = {  
  entry: './src/index.js',  
  output: './dist/main.js',  
  mode: 'production',  
  module: {  
    rules: [  
      { test: /\.txt$/, use: 'raw-loader' }  
    ]  
  },  
  plugins: [  
    new HtmlWebpackPlugin({  
      template: './src/index.html'  
    })  
  ]  
};
```

① 指定默认的 entry 为: `./src/index.js`

② 指定默认的 output 为: `./dist/main.js`

③

④ 

⑤

环境搭建：安装 Node.js 和 NPM

安装 nvm (<https://github.com/nvm-sh/nvm>)

- 通过 curl 安装: `curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash`
- 通过 wget 安装: `wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash`

安装 Node.js 和 NPM

- `nvm install v10.15.3`
- 检查是否安装成功: `node -v`, `npm -v`

环境搭建：安装 webpack 和 webpack-cli

创建空目录和 package.json

- `mkdir my-project`
- `cd my-project`
- `npm init -y`



安装 webpack 和 webpack-cli

- `npm install webpack webpack-cli --save-dev`
- 检查是否安装成功：`./node_modules/.bin/webpack -v`

Webpack初体验：一个最简单的例子

```
const path = require('path');

module.exports = {
  mode: 'production',
  entry: './src/index.js',
  output: {
    path: path.resolve(__dirname, 'dist'),
    filename: 'bundle.js'
  }
};
```

webpack 脚本



```
<!doctype html>
<html>
  <head>
    <title>webpack示意例子</title>
  </head>
  <body>
    <script src="dist/bundle.js"></script>
  </body>
</html>
```

构建结果

通过 npm script 运行 webpack

```
{  
  "name": "hello-webpack",  
  "version": "1.0.0",  
  "description": "Hello webpack",  
  "main": "index.js",  
  "scripts": {  
    "build": "webpack"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC"  
}
```



通过 npm run build 运行构建

原理：模块局部安装会在 node_modules/.bin
目录创建软链接



扫码试看/订阅
《玩转 webpack》