MPA：多页面应用 ⟶ 每个页面解耦、SEO友好

每个页面对应一个 entry ⟶ 优化：使用 glob.sync。 entry：glob.sync(__dirname, './src/*/index.js') ↪ 同步方式查询文件

glob 类似正则匹配文件路径

Source map ⟶ 将打包编译之后的代码映射回源代码

△ 关于 source map

eval：使用 eval 包裹模块代码

Source map：产生 .map 文件

cheap：不包含列信息 ⟶ 只定位行

inline：将 .map 文件作为 DataURL 嵌入，不单独生成 .map 文件

module：包含 loader 的 source map

webpack 必须选择一种模式，mode 为 'production'、'development' 或者 'none'

可以在 devtool 中指定方式 ⟹ 'eval' 'source-map' 'inline-source-map'

'source-map' 会在生成 .js 文件的同时，生成对应的 .map 文件

不开启 source map 则断点会生成在打包后的代码上

开启 cheap 模式会只显示错误的行信息而不显示列信息 cheap-source-map

△ webpack 提取公共资源

① 基础库分离 思路：将 react、react-dom 基础包通过 cdn 引入，不打入 bundle 中 ⟶ ① CDN 分配资源 ② splitChunks 分离代码

方法：使用 html-webpack-externals-plugin

② Split Chunks Plugin 进行公共脚本分离：webpack 4 内置（替代 CommonsChunkPlugin 插件）

chunks 参数：（↪ 会生成一个文件，就指定的文件）

async：异步引入进行分离 initial：同步引入进行分离 all：将所有引入库进行分离

参数说明：minSize：公共包最小大小 maxSize minChunks：重复次数 maxAsyncRequests：⟹ 同时加载异步数目

test：匹配出需要分离的包 ⟶ 使用 HtmlWebpackPlugin 的 chunks 参数来指定打包的 chunks

↪ 此步不可以省略，否则无法引入打包完成的包

Tree Shaking 的使用和原理分析

概念：1个模块可能会有多个方法，只要其中的某个方法使用到了，则整个文件都会被打包到 bundle，tree shaking 就是只把用到的方法打入 bundle，没用到的会在 uglify 阶段被擦除掉

使用：webpack 默认支持，在 .babelrc 里设置 modules：false 即可

要求：必须是 ES6 的语法，CJS 的方式不支持

DCE（Elimination）

1、代码不会被执行，不可到达 2、代码执行结果不会被用到 3、代码只影响死变量

原理：利用 ES6 模块的特点

- 只能作为模块的顶层出现
- import 模块名只能是字符串常量
- import binding 是 immutable 的

代码擦除：uglify 阶段删除无用代码

↪ 只能进行静态分析

要求模块代码不可以有副作用，有副作用会失效

Scope Hoisting 使用和原理分析 (webpack 4 production mode默认特性)

① 现象：构建后的代码存在大量闭包代码

导致的问题：1、大量函数闭包包裹，导致体积增大(模块越多，越明显)

2、运行代码时创建的作用域变多，内存开销变大

② 模块转换分析

- 被webpack转换后的模块会带上一层包裹
- import 转换为 __webpack_require__ export也会转换

③ webpack 模块机制

- 打包出来是一个IIFE(匿名闭包)
- modules 是一个数组，每一项为模块初始化函数
- __webpack_require__ 用来加载模块，返回 module.exports
- 通过 WEBPACK_REQUIRE_METHOD(0) 启动程序

④ Scope Hoisting 原理

将所有模块代码按引用顺序放在一个函数作用域中，然后适当重命名一些变量以防止命名冲突

可以减少函数代码声明和减少内存开销

必须是ES6 module，CJS不支持