webpack是一个根据入口文件，将所有项目文件与依赖文件，进行统一打包管理处理的打包器。

webpack基本为编辑配置文件

# 核心配置

webpack.config.js

module.exports = {

entry: ‘’, // webpack文件的入口文件配置，根据入口文件去寻找打包项目文件

output: ‘’, // webpack 打包完成的输出配置

loader: ‘’, // 通过相关的加载器来加载打包非js文件的配置

plugins: ‘’, // 用来处理、优化、提升，打包、环境、项目整体配置

mode: ‘’ // webpack的打包模式

}

# entry

三种写法

entry: {

main: path // 入口的一般写法

}

entry: [path, path, ptah] // 多个主入口，但是缺乏灵活性

entry: {

a: path

b: path

c: path

}

// 对象写法可扩展性高，能区分多页面，与三方库，能够根据特定webpack插件进行优化

# output

可配置多个入口，但只能配置一次出口

output: {

filename: ‘filename’ , // 文件名， 可以使用[name].xx来接收entry设定的入口文件名

path: ‘’ // 打包位置 \_\_dirname + ‘’

}

# loader

loader需要预先下载

npm i --save-dev xxx-loader

需要预先在module中告诉webpack使用规则

module: {

rules: [

{test: /\.css$/, use: ‘css-loader’} // 告诉webpack已.css结尾的文件用css-loader

]

}

# plugins

完成loader完成不了的其他事情

需要通过npm安装

npm i somePlugins

也可以访问内置插件

需要再配置文件中引入这个插件

cost somePlugins = require(‘somePlugins’)

插件都为一个类需要通过new去实现

plugins: [

new somePlugins(),

new twoPlugins()

]

# mode

告诉webpack用怎么样的模式优化打包

mode: ‘development’/’production’

development

会将 process.env.NODE\_ENV 的值设为 development。启用 NamedChunksPlugin 和 NamedModulesPlugin。

production

会将 process.env.NODE\_ENV 的值设为 production。启用 FlagDependencyUsagePlugin, FlagIncludedChunksPlugin, ModuleConcatenationPlugin, NoEmitOnErrorsPlugin, OccurrenceOrderPlugin, SideEffectsFlagPlugin 和 UglifyJsPlugin.

# target

target: ‘node’ // 告诉webpack用那种模式进行数据处理， node只会进行require的处理

其他的看官网的文档了解吧~

[https://www.webpackjs.com](https://www.webpackjs.com/)