技术胖视频：纯金网<https://www.chungold.com/>

中文官网：<https://doc.webpack-china.org/guides/>

英文官网：<https://webpack.js.org/concepts/>

相关配置参考文档：

<http://www.cnblogs.com/afrog/p/7130561.html>

<http://www.jianshu.com/p/42e11515c10f>

<http://www.jianshu.com/p/cd123afa196a>

<http://www.cnblogs.com/QxQstar/p/5961387.html>

<https://github.com/liuxcbuxiangzhangda/webpack>

**webpack-dev-server打包速度加快和实现热替换**：

<http://www.cnblogs.com/caideyipi/articles/7080010.html>

**webpack与gulp，grunt,require区别**：

<http://www.cnblogs.com/-walker/p/6056529.html>

**为什么使用webpack，**

由于流行SPA的流行，js变得复杂，还得依赖很多包，多了很多技术比如less sass所以webpack变得很重要，要成为前端工程师而不仅仅是切图小白就必须会webpack。

Webpack就是学习配置文件，配置文件相当于一个说明书。webpack相当于一个机器，入口文件相当于原料，按说明书（配置文件）操作，然后输出产品（出口文件）。配置文件里面配置要做的事情，webpack根据需要调用第三方软件去解决要做的事情，webpack实现自动化，让编程更快速，修改很容易（相比自己写的代码很好具有可读性），把很多事情整合起来做。它可以将许多松散的模块按照依赖和规则打包成符合生产环境部署的前端资源。还可以将按需加载的模块进行代码分隔，等到实际需要的时候再异步加载。通过 loader 的转换，任何形式的资源都可以视作模块，比如 CommonJs 模块、 AMD 模块、 ES6 模块、CSS、图片、 JSON、Coffeescript、 LESS 等

**安装webpack官方提示**：不要全局安装webpack，会影响后期项目的维护和版本升级，所以最开始初始化项目npm init,然后npm install –save-dev webpack（非全局安装的时候必须最开始初始化项目，生成packge.json，因为非全局安装是安装在node-modules里面的，保存在packge.json的。如果带dev就代表安装在开发环境，不带dev就是安装在生产环境的，webpack只是在开发时候程序员使用的）

**如果安装不成功**：

1.node版本问题，必须6以上

2.网络问题，有些是屏蔽了npm，使用cnpm

3.安装权限，不要window7开发

**更新到最新版本**

首先修改package.json文件中webpack的版本，然后删除node-modules,最后npm install（安装之前项目所有依赖包）

**webpack可以做的事情：**> 压缩js文件——让我们可以编写模块化的javascript代码  
> 使用npm包管理前端代码（模块优化）——NPM最大的开源代码生态系统  
> 转译ES6和ES7——es6和es7为js增加了很多新特性  
> 缩小和优化代码  
> 编译LESS和SASS  
> 使用HMR（Hot Module Replacement/实时的模块监听改变）——每次保存代码的时候，只要他是被引入到该页面那么不需要完整的页面刷新。这在我们编辑代码时候是非常方便的。  
> 把任何类型的文件包含到javascript——减少其他构建工具的需要，并允许我们用代码的方式修改和使用这些文件。  
**开始简单项目（实际工作中是配置文件进行使用，因为涉及多入口多出口）：**

建src文件夹：放开发环境的代码（源代码）

建dist文件夹：放生产环境的代码（开发好了，测试好了，提交给组长放在服务器上的代码）

Entery.js：入口文件（开发的原生代码），在src下面

（终端执行webpack src/entery dist/bundle.js）

Bundle.js：打包的文件，在dist下面

Live-server：npm install –g live-server 使用live-server开启本地服务器启动项目

**配置文件 webpack.config.js：**

基本机构：暴漏接口（module.exports）---entry—output---module---plugins---devServer（plugins是数组，其他都是对象）



**Entry:**可以多入口，直接写多个地址

**Output**：配置两个path和filename

**Path(绝对路径)：**node语法,先要引用path接口（require(“path”)），然后配置output的path选项，path.resolve(\_dirname,”dist”)

**Filename：**打包后的文件名，如果是多出口就用[name].js,就会生成与源文件相同名字的打包后文件

devServer:四个contentBase,host,compress

webpack的组成部分：入口，出口，加载器（规定webpack要做的任务），插件  
  
webpack使用3.0版本  
  
全局安装存放在系统，非全局安装存放在项目的node\_modules,yarn的全局安装有问题，要用npm  
  
安装的库：  
> yarn add require  
  
> npm install webpack-dev-server -g 必须全局安装，构建本地服务器，自动刷新更改的代码  
  
  
webpack配置文件：  
> require（webpack）  
  
> require（path）  
  
> 进出口文件：一个入口js（把其他的模块js依赖到这个js来），一个出口js（打包好的js），index.html引用出口文件就好了  
  
> 配置devtool，找到出错了的地方对应的源代码的位置，使调试更容易  
  
> 配置webpack-dev-server构建本地服务器，自动刷新更改  
  
> 运行webpack：在packjson文件里面设置start  
  
加载器：  
loader从右至左，从下至上执行  
> style-loader ：将解析的css嵌入js  
  
> css-loader：解析css代码  
>   
> less-loader：编译less成css  
  
代理服务器：配置devServer  
> 把本地的代码放在服务器上访问