1. 资料  
   http://webpackdoc.com/usage.html  
     
   确定已经有 package.json，没有就通过 npm init 创建  
     
   2.原理  
   文件（模块）打包到 bundle.js 。Webpack 给每个模块配一个 id 并通过这个 id 索引和访问模块  
   在页面启动时，会先执行 entry.js 中的代码，其它模块会在运行 require 的时候再执行。  
     
   3.loader

（1)项目目录安装loader  
 npm install css-loader style-loader --save-dev

（2）安装后引入CSS方式  
 require("!style!css!./style.css")

(3)指定类型后的引入方式  
 webpack entry.js bundle.js --module-bind "css=style!css"  
  
（4）引入方式最终简化版，最后直接执行webpack即可  
 webpack.config.js

4.监听模式  
直接全局安装： npm install webpack-dev-server -g  
不能局部安装。

5.根据不同的命令打不同版本的包

在process上新增个属性 然后打包的时候进行判断

https://blog.csdn.net/a0405221/article/details/79608711  
  
  
  
6. webpack webpack-cli webpack-dev-server 要一起安装 不然会出现版本不兼容

npm i webpack webpack-cli webpack-dev-server –D

7.代码分离

<https://www.webpackjs.com/guides/code-splitting/#%E5%85%A5%E5%8F%A3%E8%B5%B7%E7%82%B9-entry-points->

* [webpack-chart](https://alexkuz.github.io/webpack-chart/): webpack 数据交互饼图。
* [webpack-visualizer](https://chrisbateman.github.io/webpack-visualizer/): 可视化并分析你的 bundle，检查哪些模块占用空间，哪些可能是重复使用的。
* [webpack-bundle-analyzer](https://github.com/webpack-contrib/webpack-bundle-analyzer): 一款分析 bundle 内容的插件及 CLI 工具，以便捷的、交互式、可缩放的树状图形式展现给用户。