## 一：工具的两种安装方式

1：npm  i webpack -g 全局安装

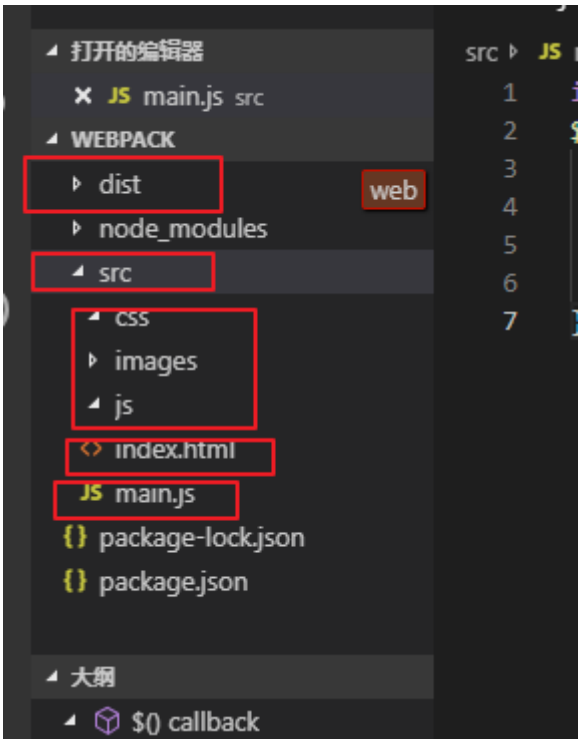
2：在项目根目录中运行 npm i webpack --save-dev 安装到项目依赖中

cnpm install --save-dev webpack

## 二：打包一个项目

在vscode的终端中

项目的结构如下：（几个框中的）




1: npm init -y  初始化项目

2: 引入包，将jquery导入项目中

npm i jquery -S(其中的i代表install)

即：  npm install jquery -S

3：

后我们尝试进行打包,回到命令框

原本输入了webpack .\src\main.js .\src\bundle.js,然后报错了

```
 @ multi ./src/main.js ./dist/bundle.js main[1]
PS C:\Users\GouQian\Desktop\vscode学习\webpack> webpack .\src\main.js .\src\bundle.js
Hash: 0c49ec27d449552d387d
Version: webpack 4.39.1
Time: 4351ms
Built at: 2019-08-07 3:59:33 PM
 1 asset
Entrypoint main = main.js
[1] multi ./src/main.js ./src/bundle.js 40 bytes {0} [built]
[2] ./src/main.js 178 bytes {0} [built]
    + 1 hidden module

WARNING in configuration
The 'mode' option has not been set, webpack will fallback to 'production' for this value. Set 'mode' opt
ion to 'development' or 'production' to enable defaults for each environment.
You can also set it to 'none' to disable any default behavior. Learn more: https://webpack.js.org/config
uration/mode/

ERROR in multi ./src/main.js ./src/bundle.js
Module not found: Error: Can't resolve '.\src\bundle.js' in 'C:\Users\GouQian\Desktop\vscode学习\webpack
'
 @ multi ./src/main.js ./src/bundle.js main[1]
```

这是为什么呢?原因是我的webpack版本过高,原来的命令已经不适用了

如下查询版本号:

```
F:\demo\test01\webpack-demo>webpack -v
4.12.0
```


那应该如何解决?

更换打包命令为: `webpack .\src\main.js -o .\dist\bundle.js`

这个命令是手动去 生成运行内存中 js文件

```
 @ multi ./src/main.js ./src/bundle.js main[1]
PS C:\Users\GouQian\Desktop\vscode学习\webpack> webpack .\src\main.js -o .\dist\bundle.js
Hash: 10c1b3c5d458bcccec6b
Version: webpack 4.39.1
Time: 3304ms
Built at: 2019-08-07 4:02:50 PM
    Asset      Size  Chunks             Chunk Names
bundle.js  87.6 KiB       0  [emitted]  main
Entrypoint main = bundle.js
[1] ./src/main.js 178 bytes {0} [built]
    + 1 hidden module

WARNING in configuration
The 'mode' option has not been set, webpack will fallback to 'production' for this value. Set 'mode' opt
ion to 'development' or 'production' to enable defaults for each environment.
You can also set it to 'none' to disable any default behavior. Learn more: https://webpack.js.org/configuration/mode/
PS C:\Users\GouQian\Desktop\vscode学习\webpack>
```

webpack 能够处理 js文件的相互依赖的关系，

webpack能够处理js的兼容问题，把 高级的，浏览器不支持的别的语法转变为低级的，能真正识别的语法

# ////// cls 清空终端的命令

**想要通过webpack命令对更改完的文件进行打包，需要在**

根目录下新建 webpack.config.js文件，然后将下面的代码写进去就可以了

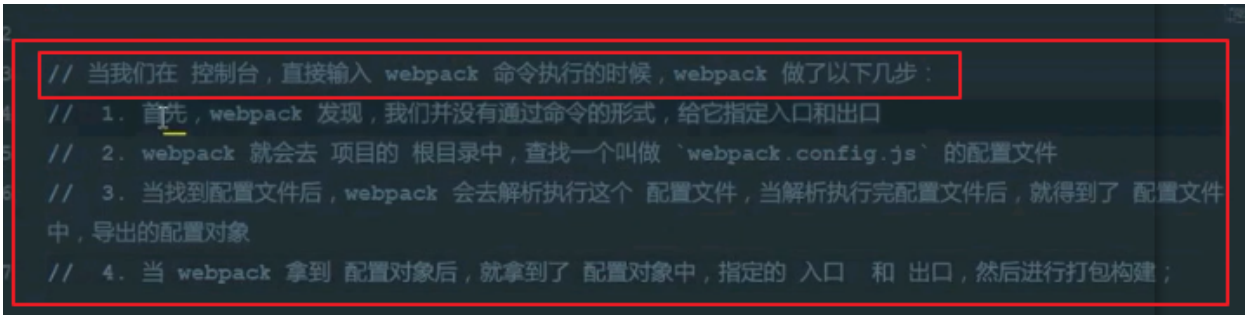
## 3：通过配置文件打包（及时通过webpack官网去查看相关命令和写法因为 ----更新太快了！！！）

```
//这个配置文件，其实就是一个js文件，通过node中的模板操作，向外暴露了一个配置对象

const path = require('path');

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

写完如上的配置文件，就可以通过在命令行直接写 webpack 及进行打包了，

此时需要在html文件中将需要的js文件引进去

## 4：当在控制台输入webpack时，执行了哪些步骤




因为每次写完都还需要通过在命令行直接写 webpack 及进行手动打包了，多以