# Webpack：模块加载器

Node.js（后台模块化） -> require，export

后台能模块化，那么前台也需要模块化

Broserify（最早的前台模块化）：模块加载，但只能加载.js文件

Webpack：一切东西都能模块化（css文件，html文件，jpg文件…）

比如说reuqire（‘style.css’）；

但是要能进入此操作，需要其他的一些辅助模块或工具的支持，

loader就是这样的一个工具

要想使用require（‘style.css’），则需要先引入css-loader， style-loader

Webpack是多个模块分开写，最后打包到一起

简单的目录结构（以vue为例子）：

|-index.html

|-main.js 编写webpack代码的入口文件

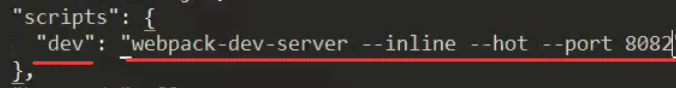
|-App.vue vue文件，官方推荐命名法

|-package.json 工程文件(项目依赖、名称、配置)

npm init --yes 生成

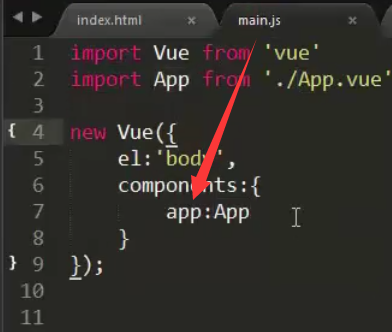
|-webpack.config.js webpack配置文件

Package.json里的script代码块里的一些属性是方便输入代码用的，比如

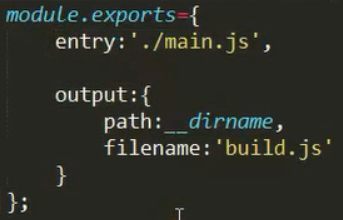
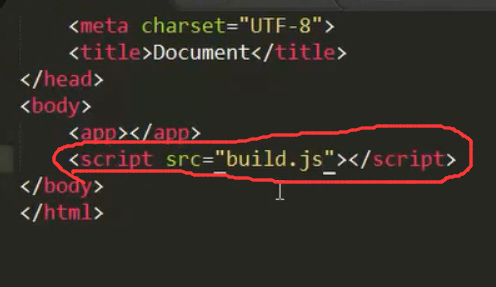


只要运行npm run dev就能实现和webpack-dev-server --inline --hot --port 8082一样的功能

webpack.config.js里的entry（入口文件）属性：意思是，如果html文件里引用了一个名为“app”的标签<app></app>，那么这个app标签是存在于入口文件里的



Output属性：意思是，webpack里的模块一起打包到某个文件（filename），并且需要在html文件里引用该文件，path是指该文件目录

因为webpack推崇的是模块化开发，所以在引用vue.js时不在html文件里通过

<script src=”vue.js”></script>来引用，而是在入口文件里引用



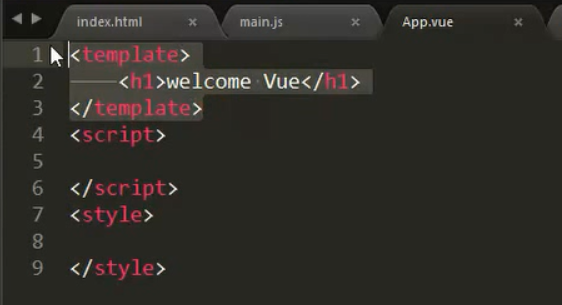
ES6: 模块化开发

导出模块：

export default {}

引入模块:

import 模块名 from 地址



因为浏览器不能识别类似与import，<template></template>之类的语法，所以需要引入loader

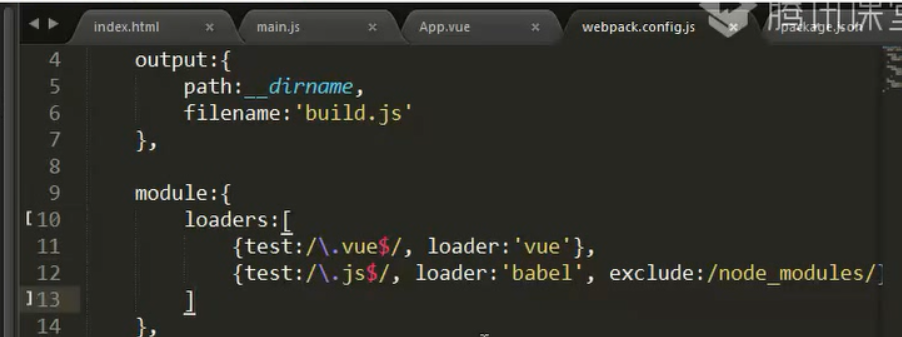
编译<template></template>的loader是vue-html-loader

编译<script></script>的loader是vue-hot-reload-api

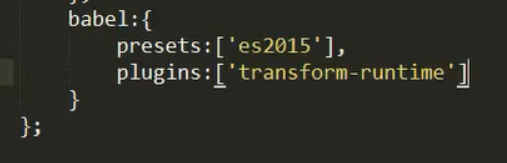
编译<style></style>的loader是css-loader、vue-style-loader

让浏览器识别ES6语法的loader是babel-loader

而loader是存放在webpack.config.js里的module对象里的loaders数组里



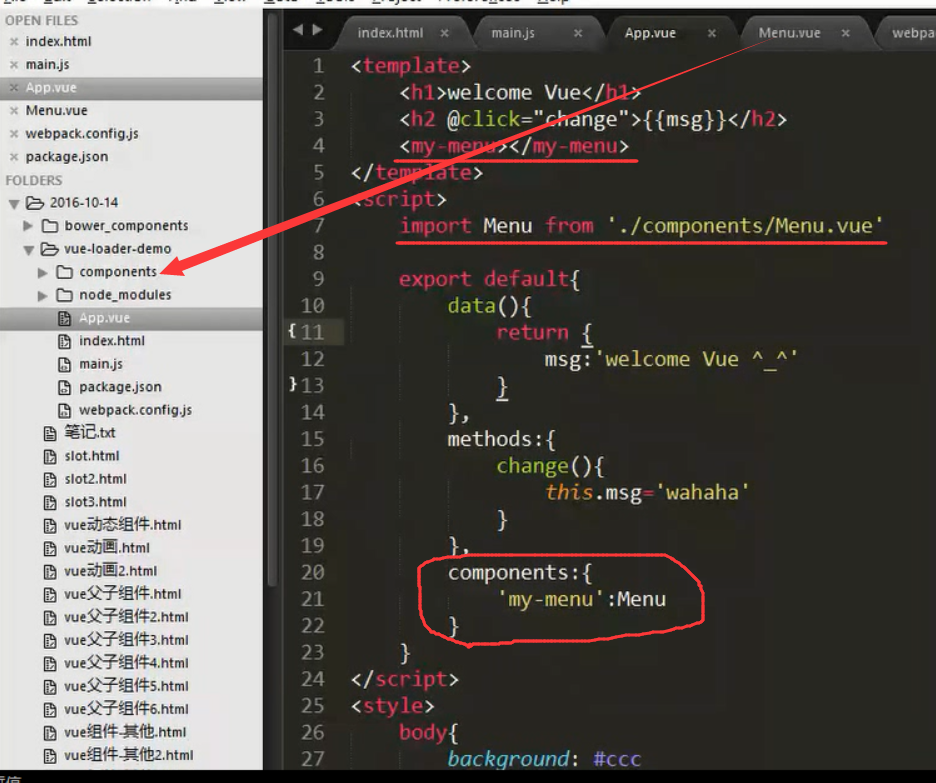
babel存放在babel对象里



完整的APP.vue文件代码



在一个组件模块里引用其他组件模块



当然，你要使用webpack的话需要先下载webpack的一些模块

webpak准备工作:

cnpm install webpack --save-dev //没有服务器的版本

cnpm install webpack-dev-server --save-dev //有服务器的版本

App.vue -> 变成正常代码 vue-loader@8.5.4

cnpm install vue-loader@8.5.4 --save-dev

cnpm install vue-html-loader --save-dev

cnpm install vue --save //--save 和--save-dev的区别是前者是项目必须依赖 的框架（存放在dependencies对象里），后者是开发依赖的框架（存放在devDependencies对象里）

vue-html-loader、css-loader、vue-style-loader、

vue-hot-reload-api@1.3.2

babel-loader

babel-core

babel-plugin-transform-runtime

babel-preset-es2015

babel-runtime