

Axios – 他的AJAX

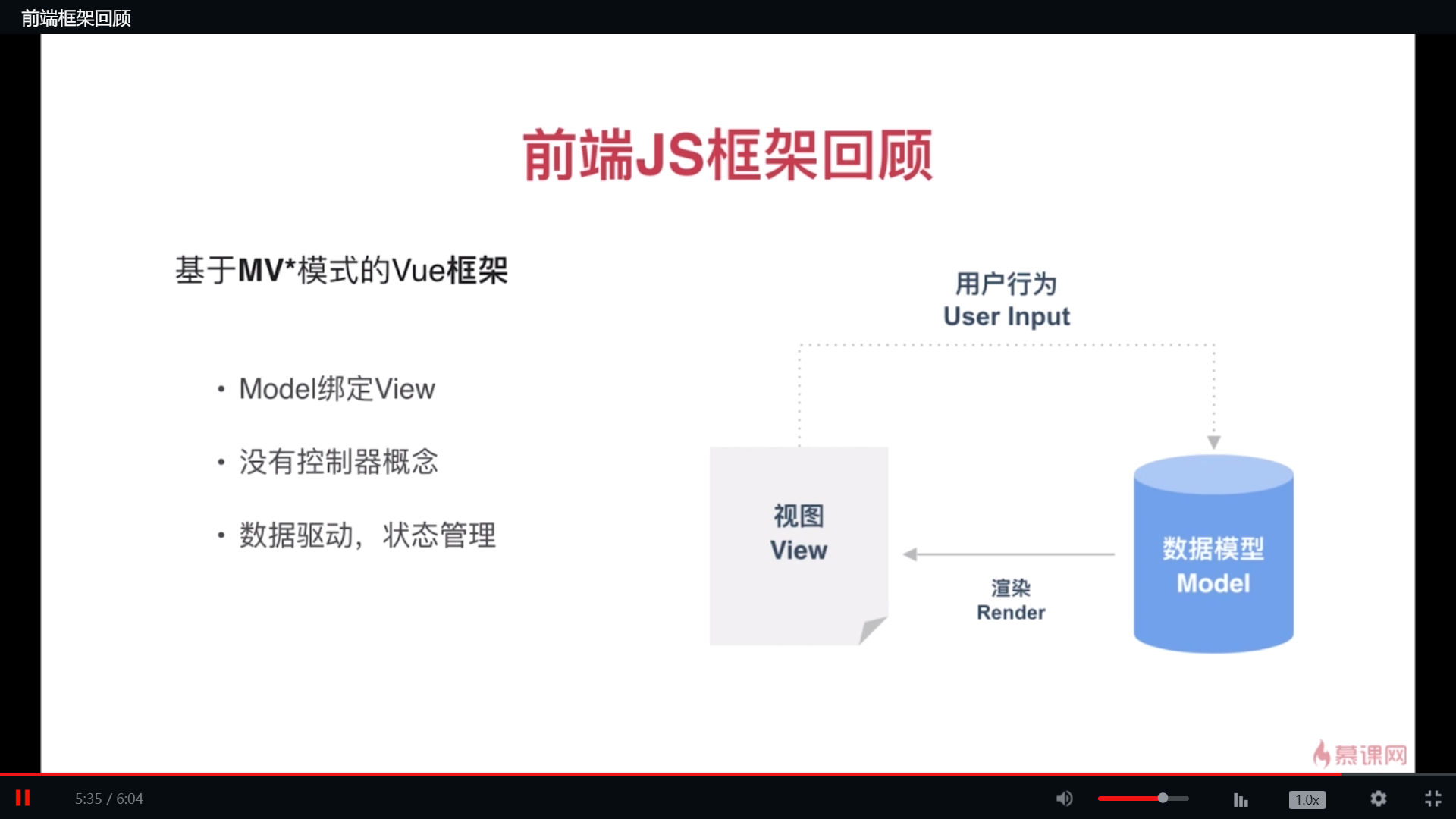
<http://ithelp.ithome.com.tw/articles/10188510>

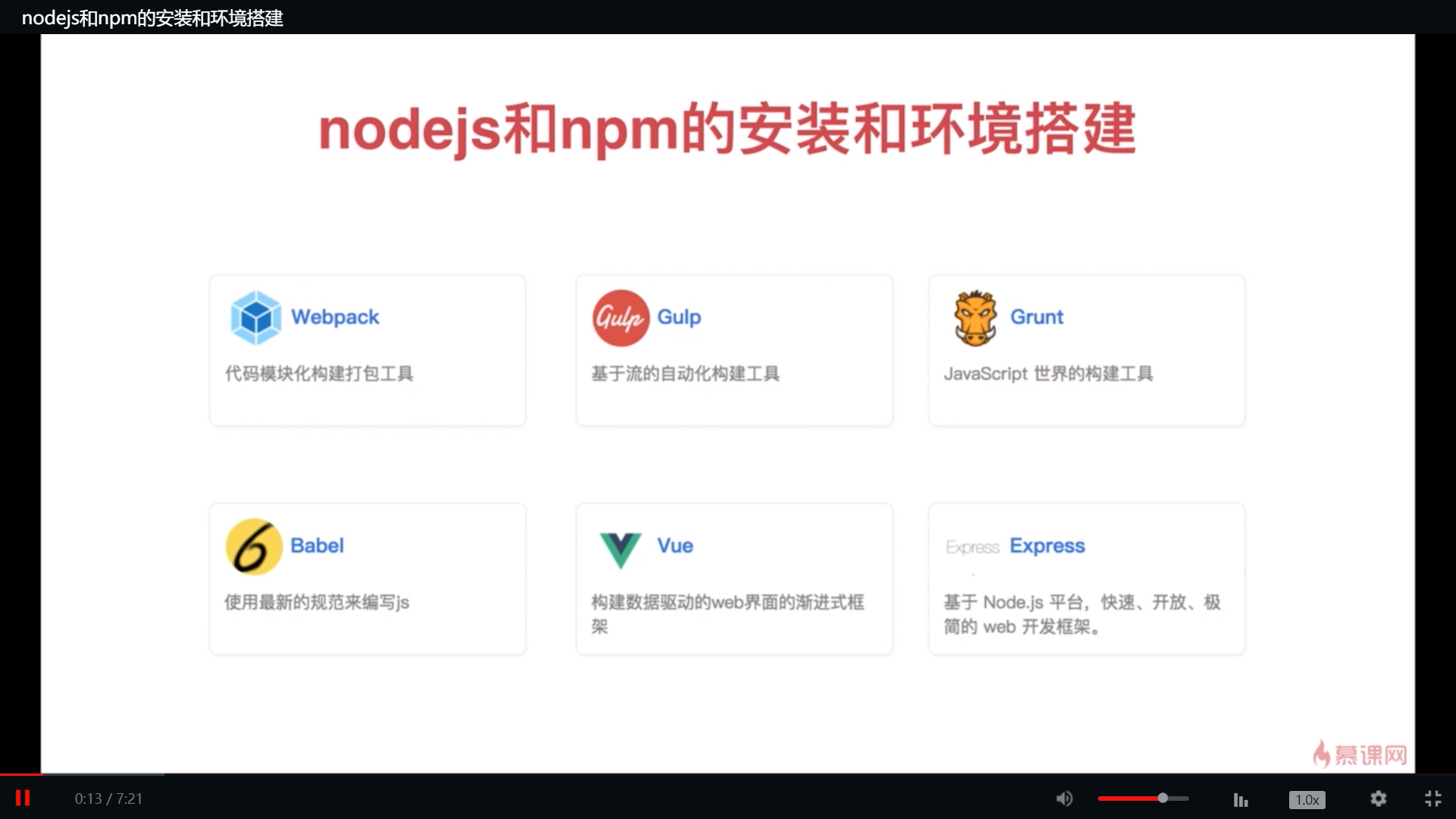
Vuex是一個專為Vue.js應用程序開發的**狀態管理模式**。它採用集中式存儲管理應用的所有組件的狀態，並以相應的規則保證狀態以一種可預測的方式發生變化

<https://vuex.vuejs.org/zh-cn/intro.html>

SPA 單頁面應用: <https://baike.baidu.com/item/SPA/17536313>

2. 





2-1nodejs和npm的安裝和環境搭建

1. Npm init (進入點預設indesx.js)
2. Npm install vue –save
3. 可以新增index.js 然後輸入測試code:

Ex:

function sum(a, b){

return a + b;

}

let s = sum(10, 7);

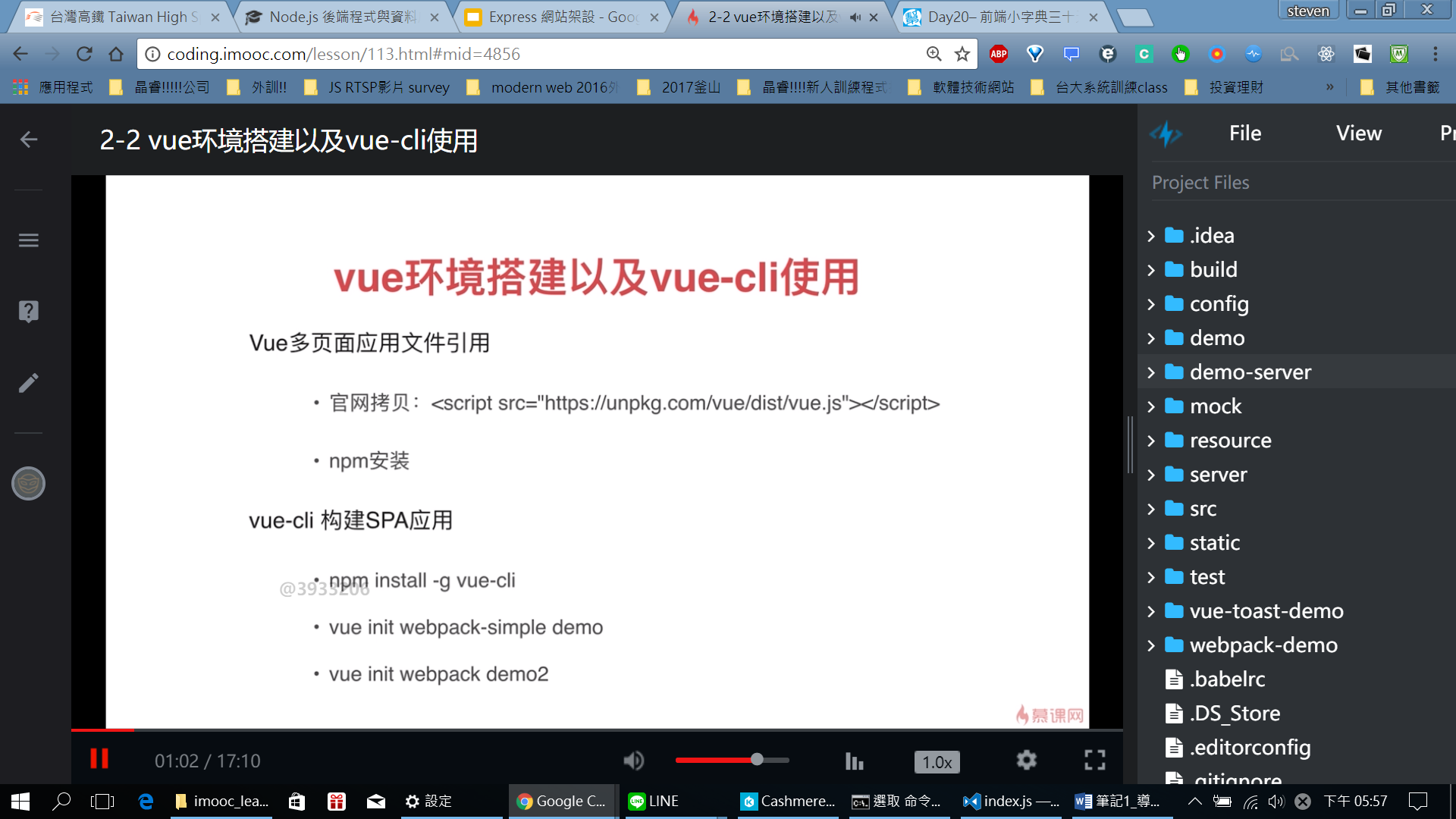
console.log("sum=", s)

在cosole 輸入node index.js就可以看到結果了

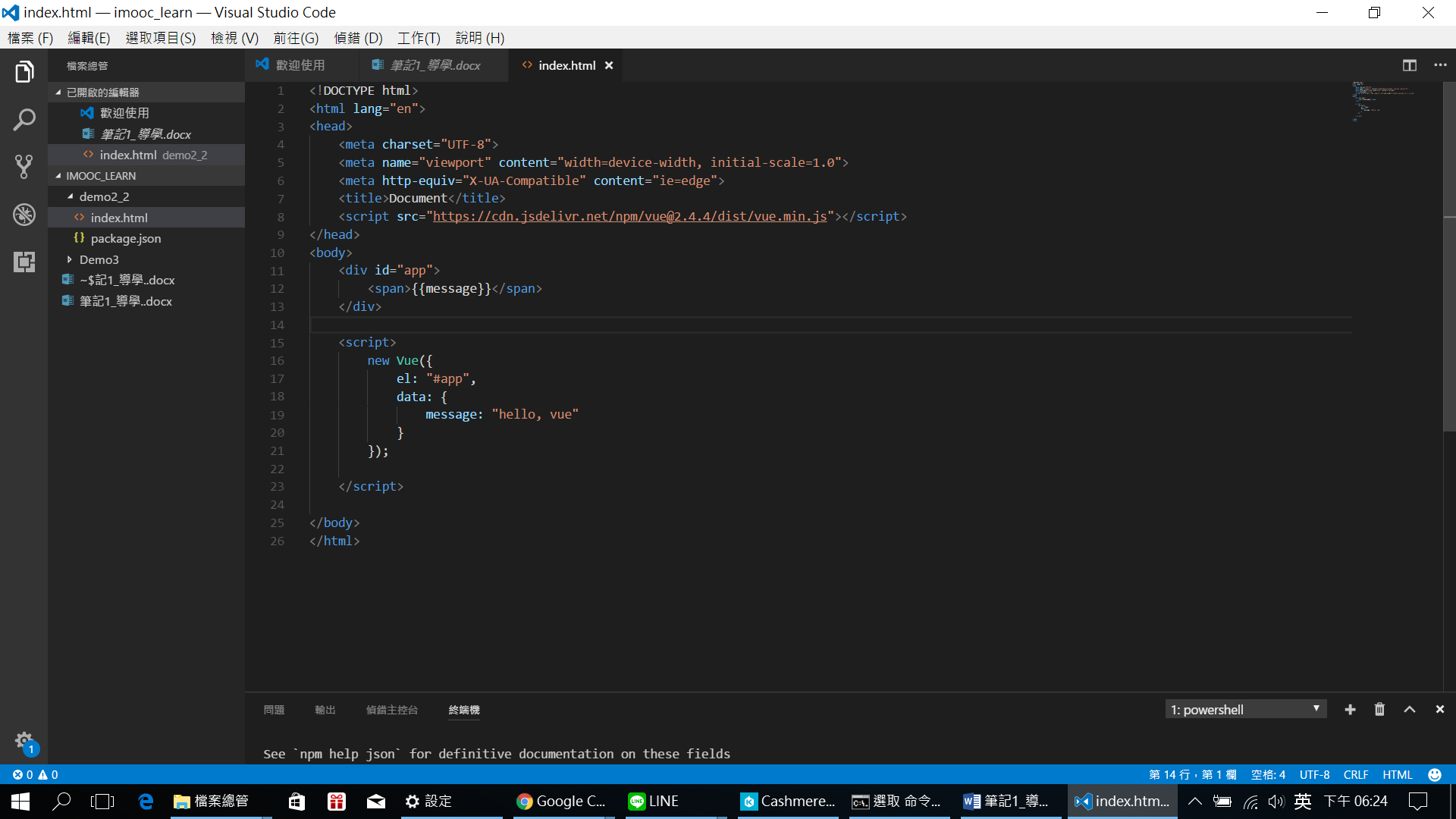
2-2 vue環境搭建以及vue-cli使用

了解SPA應用: my: <https://ithelp.ithome.com.tw/articles/10160709>

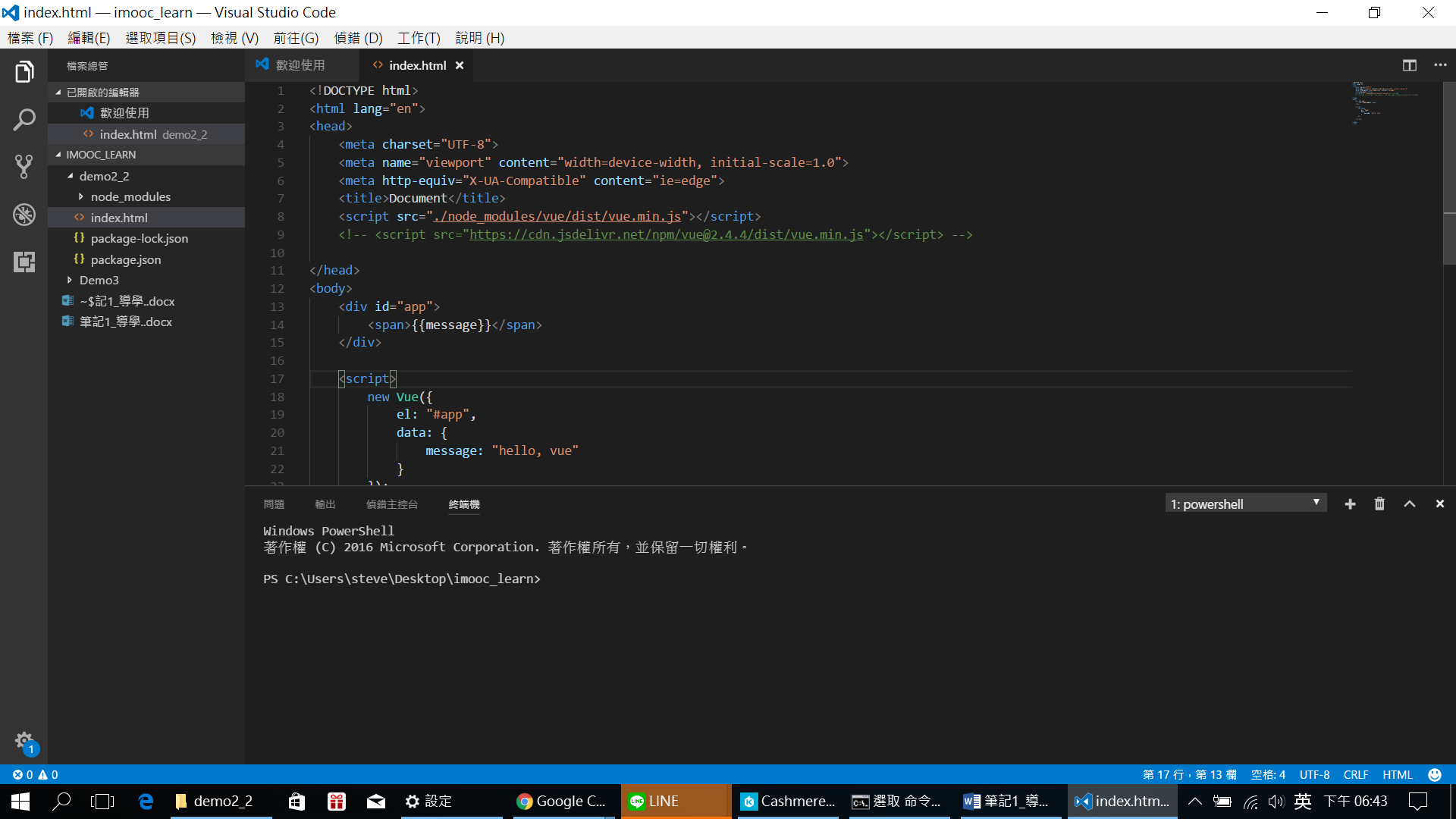
Vue可以使用vue-cli構建SPA應用



1. 透過cdn:



1. 或透過npm : (npm install vue –save 再把src路徑改掉)



1. 安裝vue-cli

npm install --global vue-cli

# 全局安装 vue-cli

$ npm install --global vue-cli

# 创建一个基于 webpack 模板的新项目

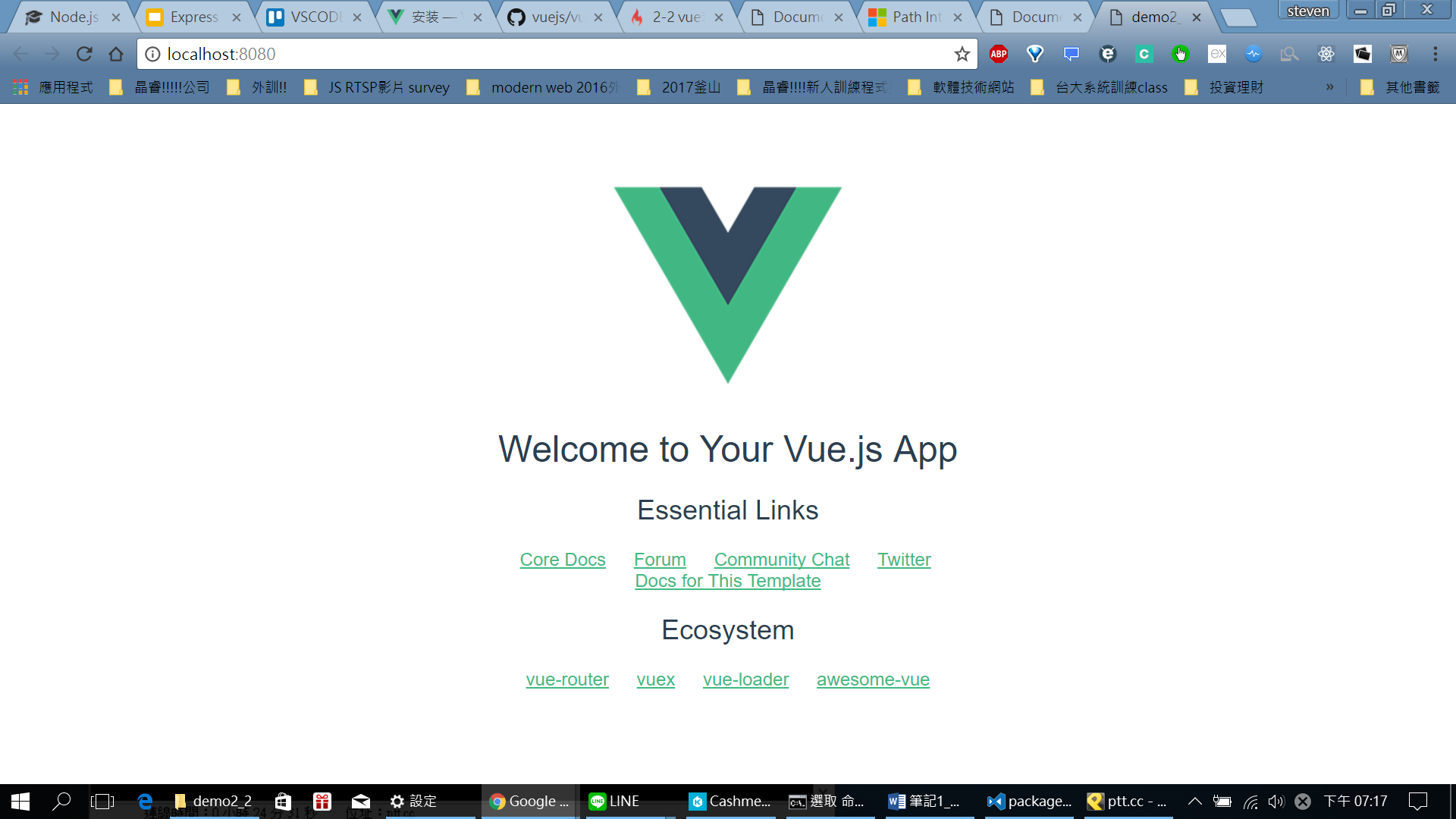
$ vue init webpack my-project

# 安装依赖，走你

$ cd my-project

$ npm install

$ npm run dev

1. 就會開啟server囉! 
2. 從package.json可以看到:

即為當執行npm run dev還是npm run start等所跑的指令

"scripts": {

"dev": "node build/dev-server.js",

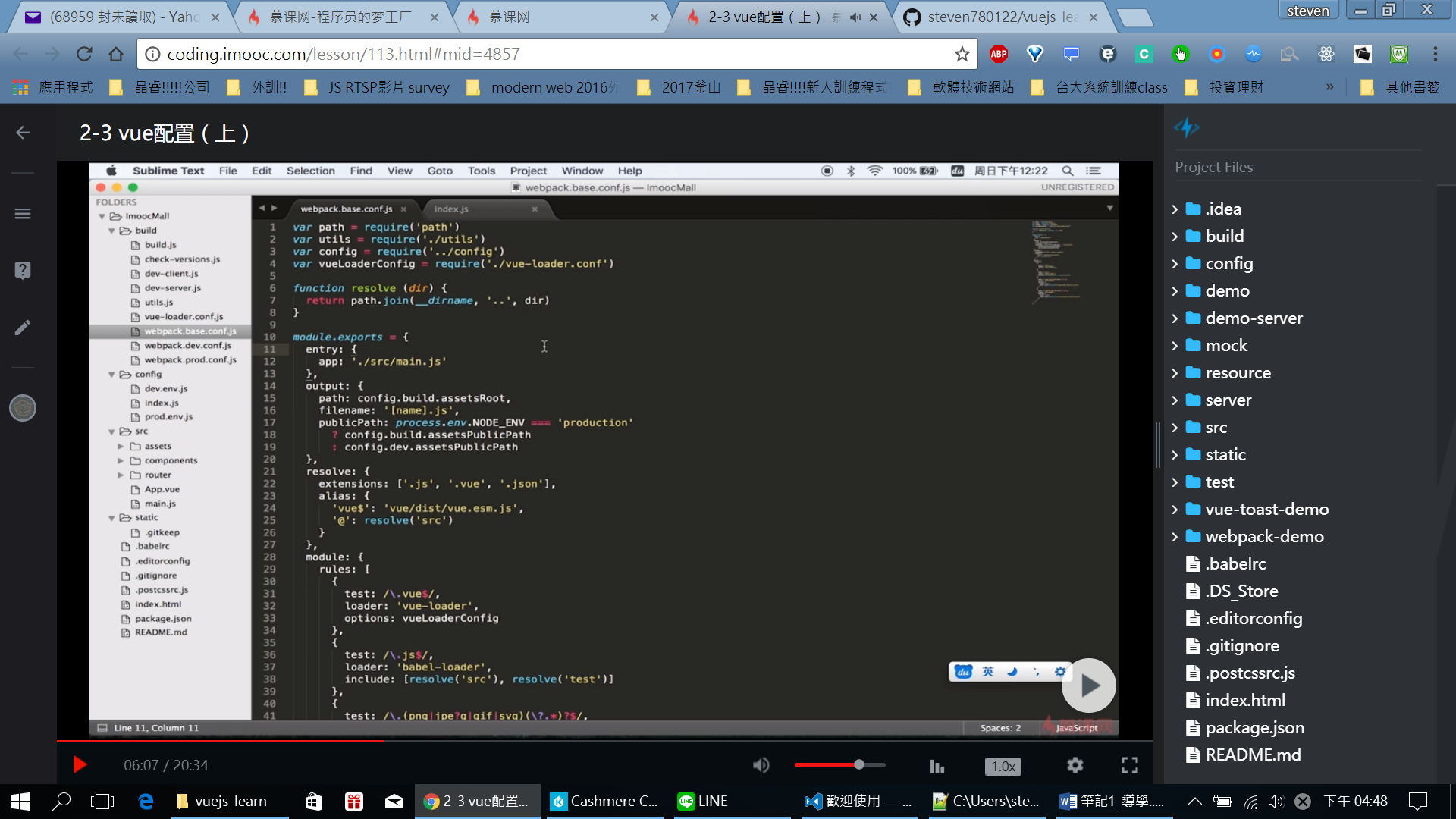
"start": "npm run dev",

"build": "node build/build.js"

},

建立Steven Mall

1. Vue init webpack steven\_mall
2. 需要關注的主要會在build的webpack.base.conf.js & config的index.js

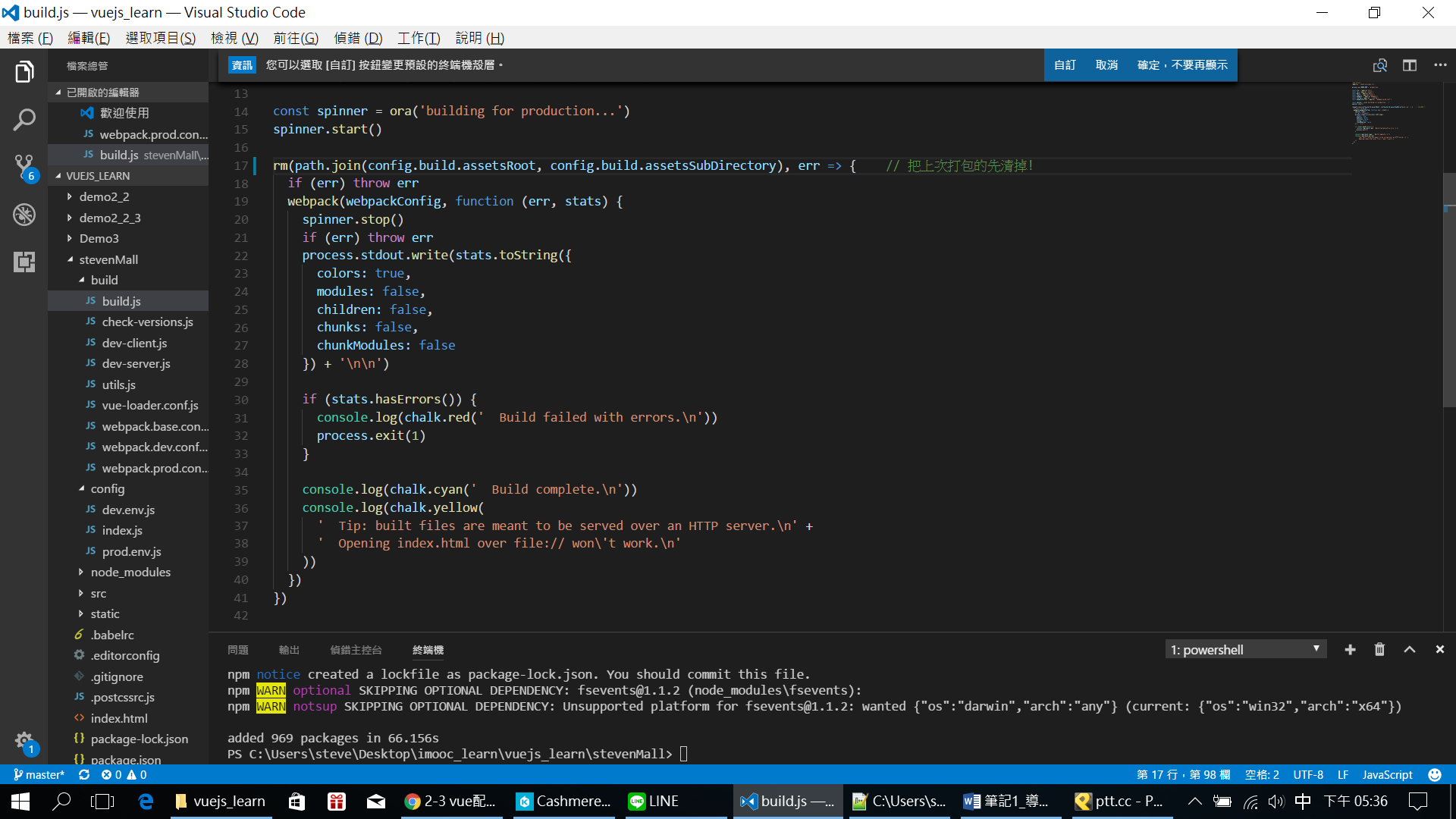


1. Build/build.js通常是不需要我們去改的，通常會改的話是改config/index.js的配置像是port等等，再來的話就是改build/webpack.base.conf.js
2. 根據webpack.dev.conf.js和webpack.prod.conf.js都是繼承自webpack.base.conf.js

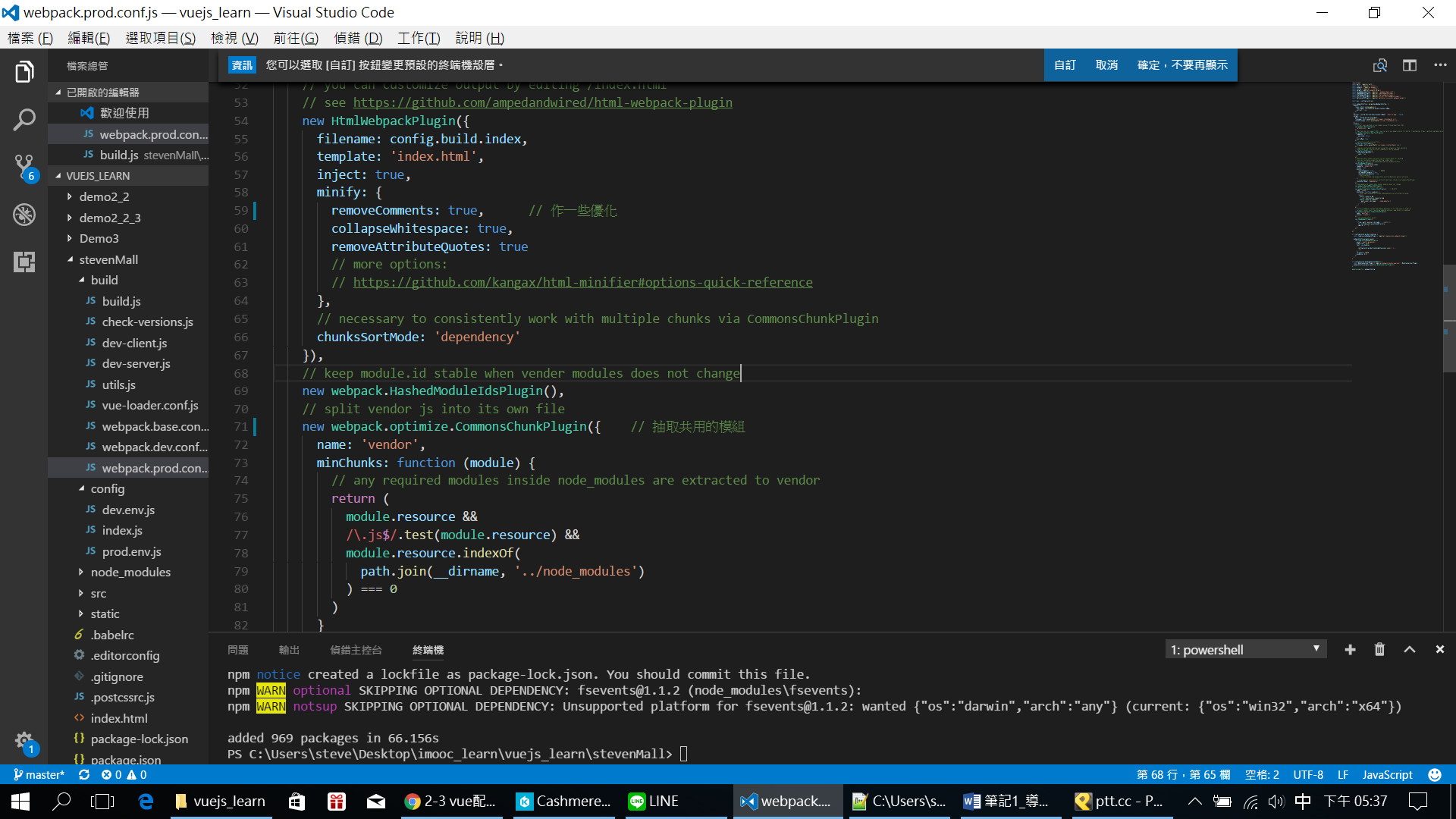
(是因為引用了此module):

const merge = require('webpack-merge')

1. Build/build.js:



的rm是將前一次打包先清掉，然後再用webpack進行打包

1. 

HtmlWebpackPlugin

打包時有作一些優化

new webpack.optimize.CommonsChunkPlugin({

會抽取共用的模組

1. 可以試著npm install後輸入npm run dev伺服器將啟動!!!
2. 所以其實build/build.js做的事很簡單，:就是將base(webpack.base.conf.js)和prod(webpack)根據設定購過webpack進行打包
3. Build/dev-client.js

hotClient.subscribe(function (event) {

if (event.action === 'reload') {

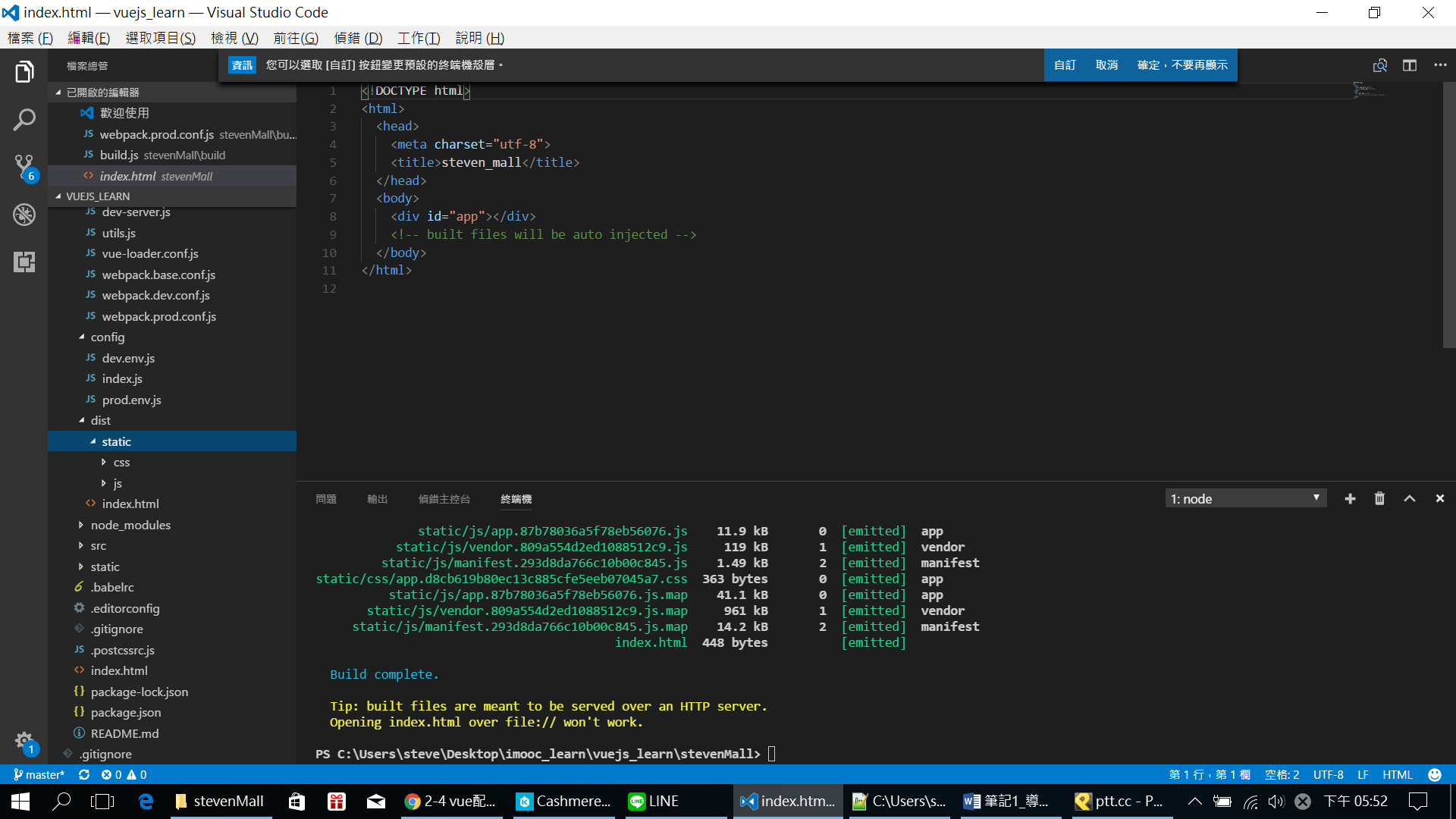
window.location.reload()

}

})

這就是熱更新的插件

1. 輸入npm run build看生產包長什麼樣子! (會生成一個dist資料夾):



上面的Build complete.

Tip: built files are meant to be served over an HTTP server.

Opening index.html over file:// won't work. 是正常的喔!

1. 為何打包完較dist/XXX呢?
   1. 可以從config/index.js看到 (build(走prod所以可以從

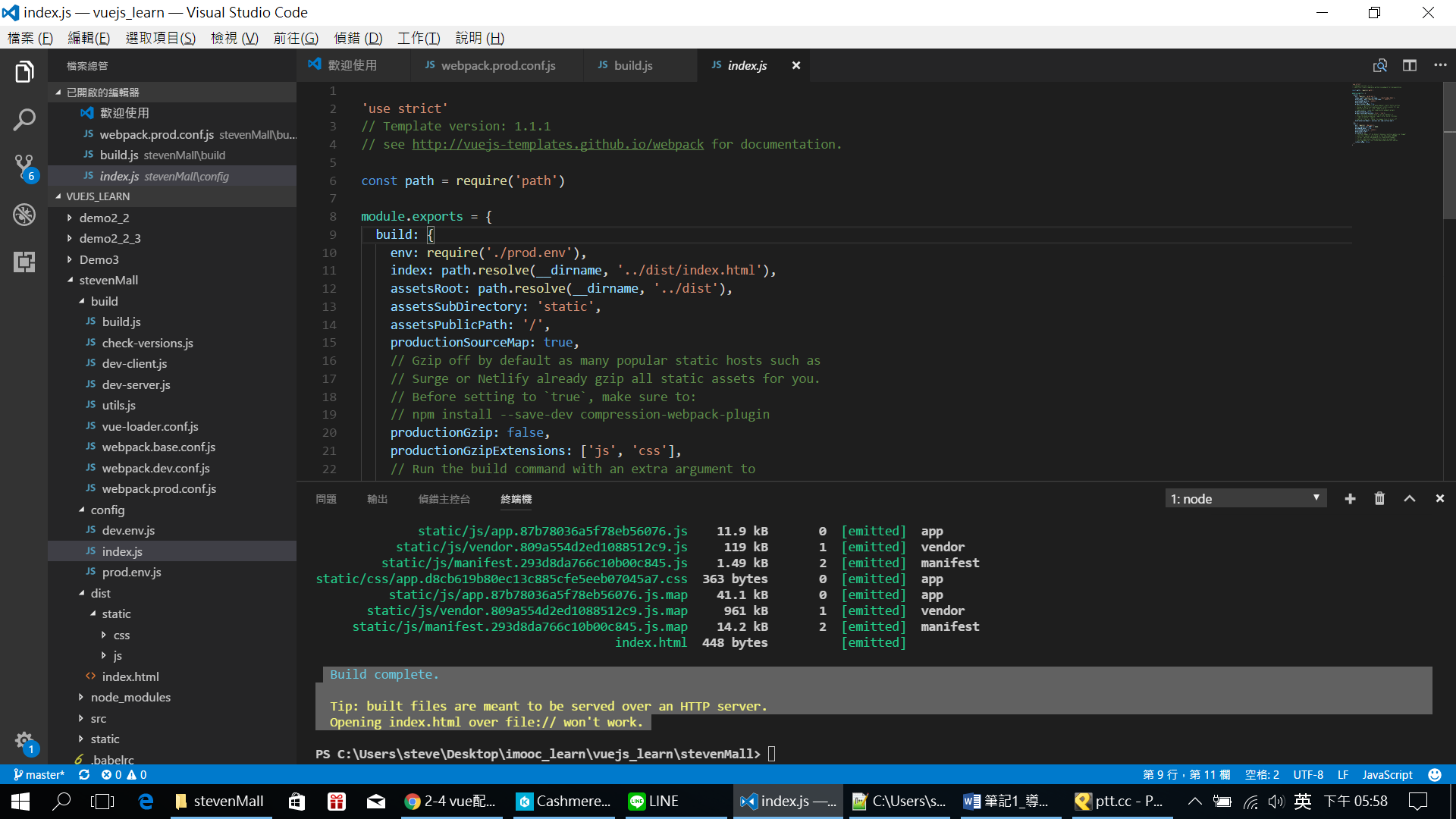
(webpack.prod.conf.js以下的)

new HtmlWebpackPlugin({

filename: config.build.index,

看到

)):



1. 在其中

assetsPublicPath: '/',

若改掉為XXX，則會發現重build出來的dist/index.html的<link href變為XXX/static/…> 因此這是非常有用的，例如有時候要用CDN地址就可以來改這塊! 也就是說例如如果自己打包完的靜態檔案如果已經放到了雲上面，就可以透過改這個地址來導入囉 !!!!!

1. 接下來介紹dev-server.js(利用express建立server端口):

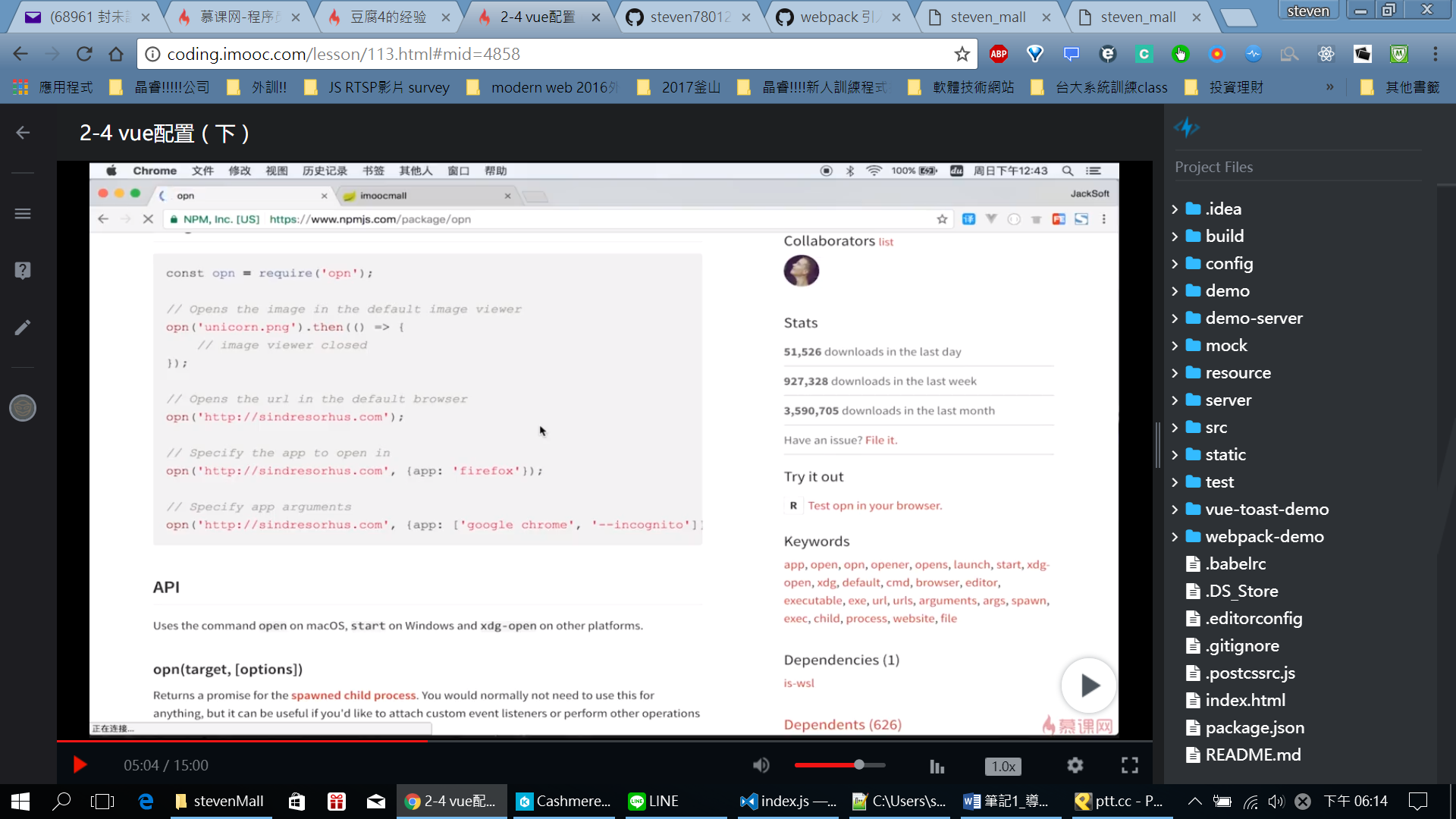
My: 難怪package.json中script如果是走npm run dev會開啟server

因為他走dev-server.js壓!!! (裡面有express做Server的端口)，但走npm run build是走生產包就沒有開啟server囉!

* 1. 其中的

const opn = require('opn')

很厲害，去npm 搜尋就可以發現是一個很強大的打開工具，還可以指定要用什麼瀏覽器打開網址或圖片等



1. 再來utils.js:

可以發現切很細:

Utils.js有:

exports.cssLoaders = function (options)

而

Vue-loader.js中有:

module.exports = {

loaders: utils.cssLoaders({

再..

Webpack.base.conf.js有:

const vueLoaderConfig = require('./vue-loader.conf')

所以可以看到他切非常細!

熟了以後就可以自己加!

<建議2-4 vue配置下再聽一次!!>

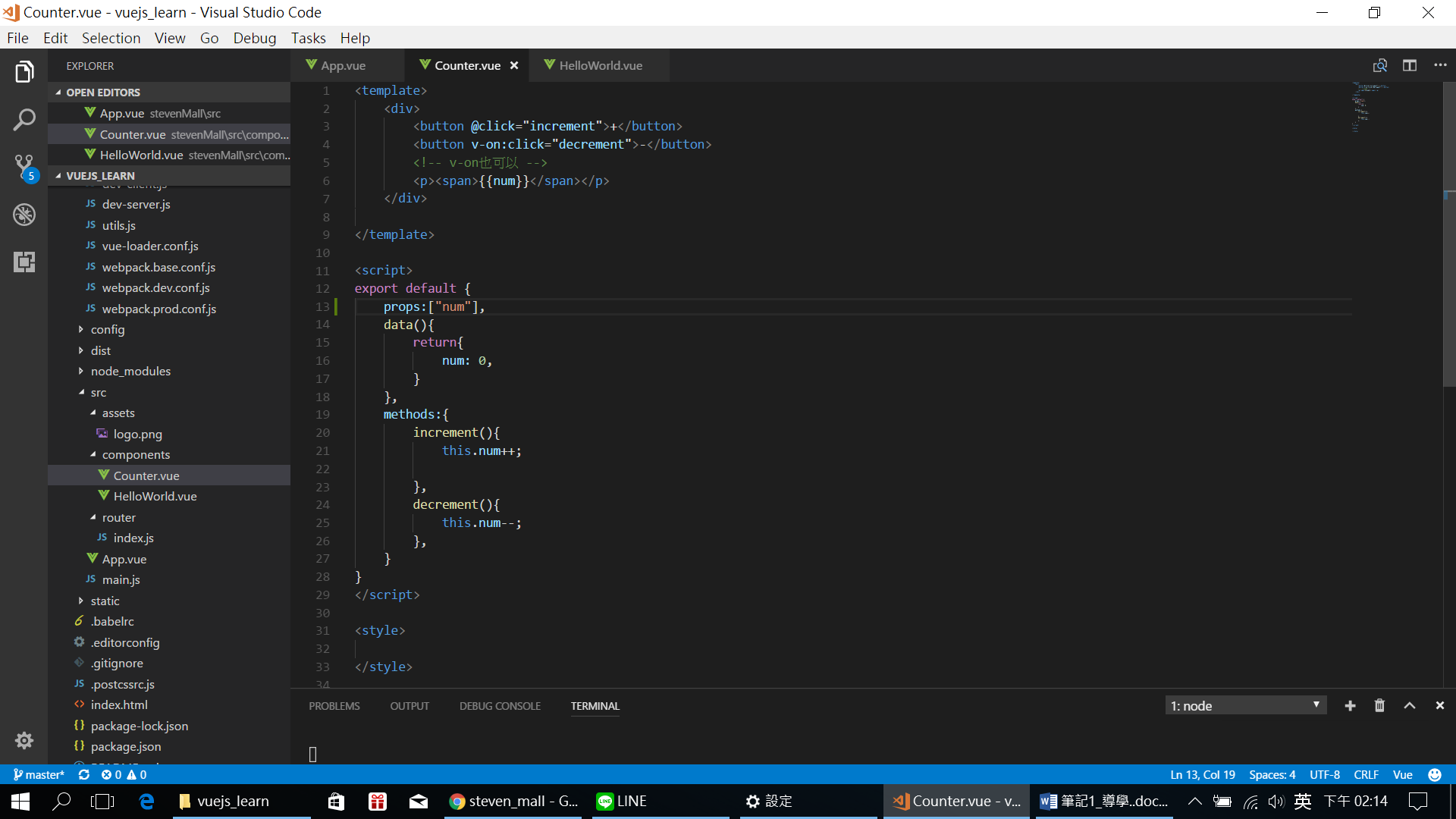
<2-5>

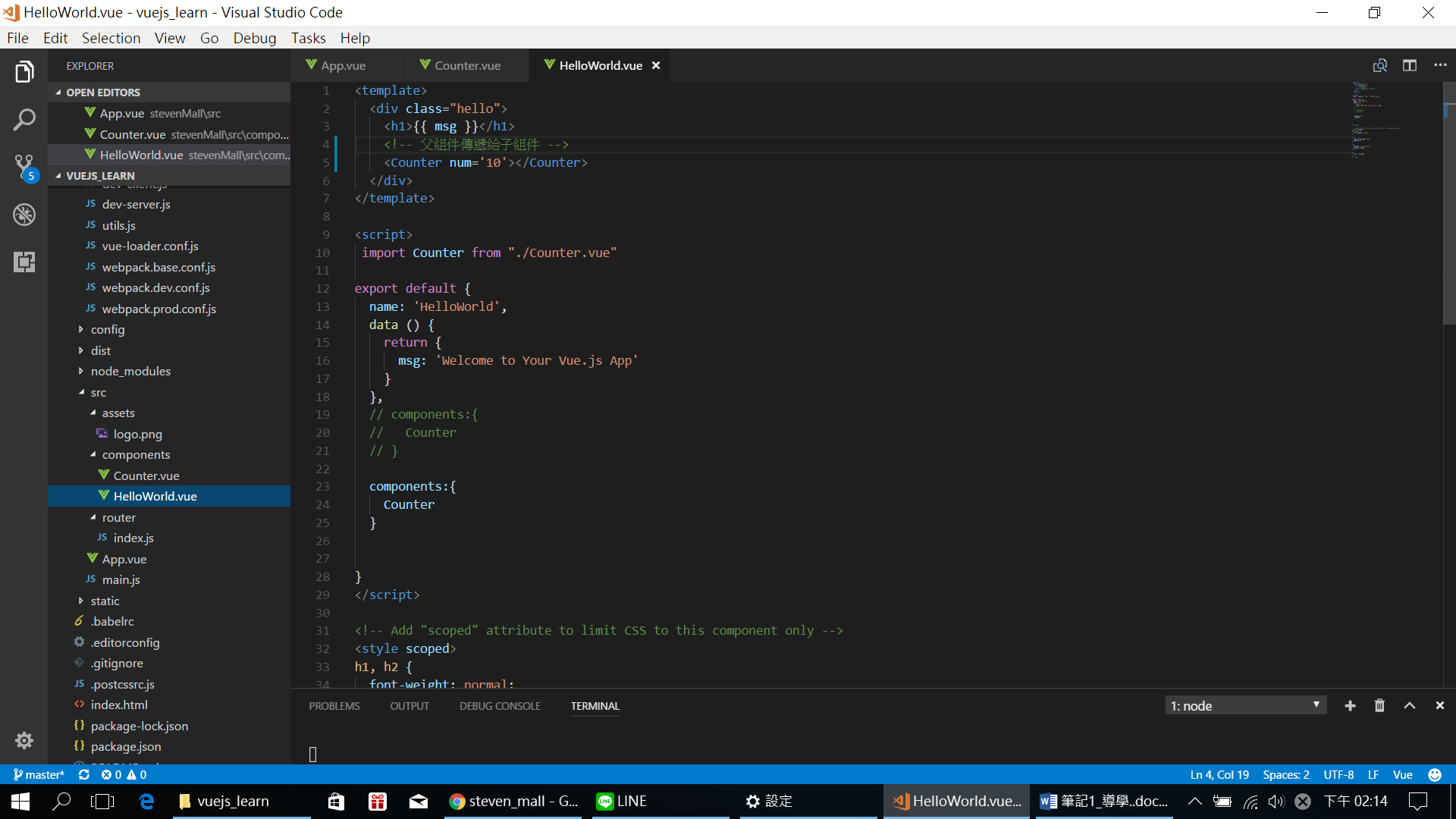
My: VScode建議裝個 Vuter

1.

新增Components :

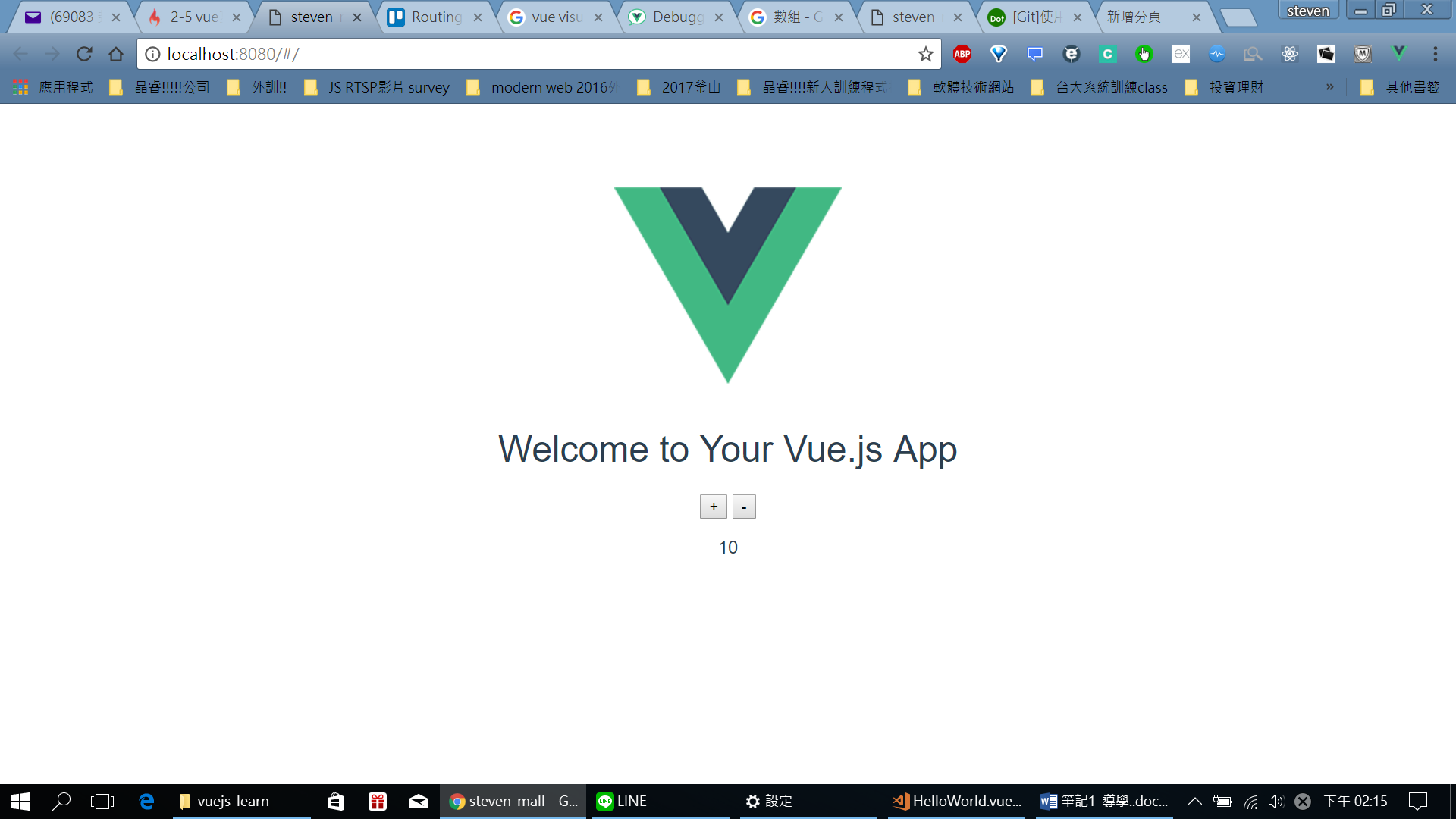
Counter.vue是子 HelloWorld是父



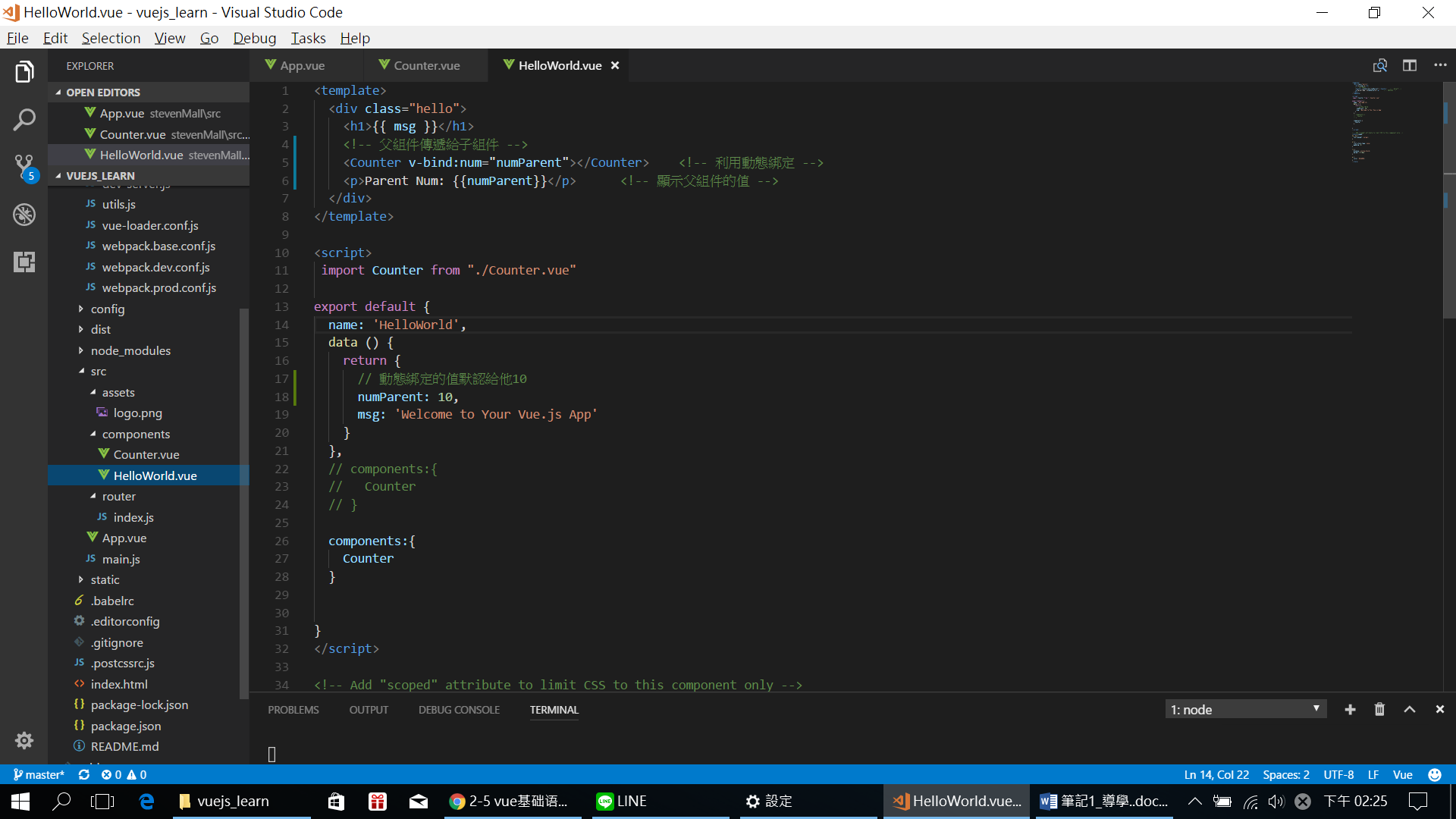


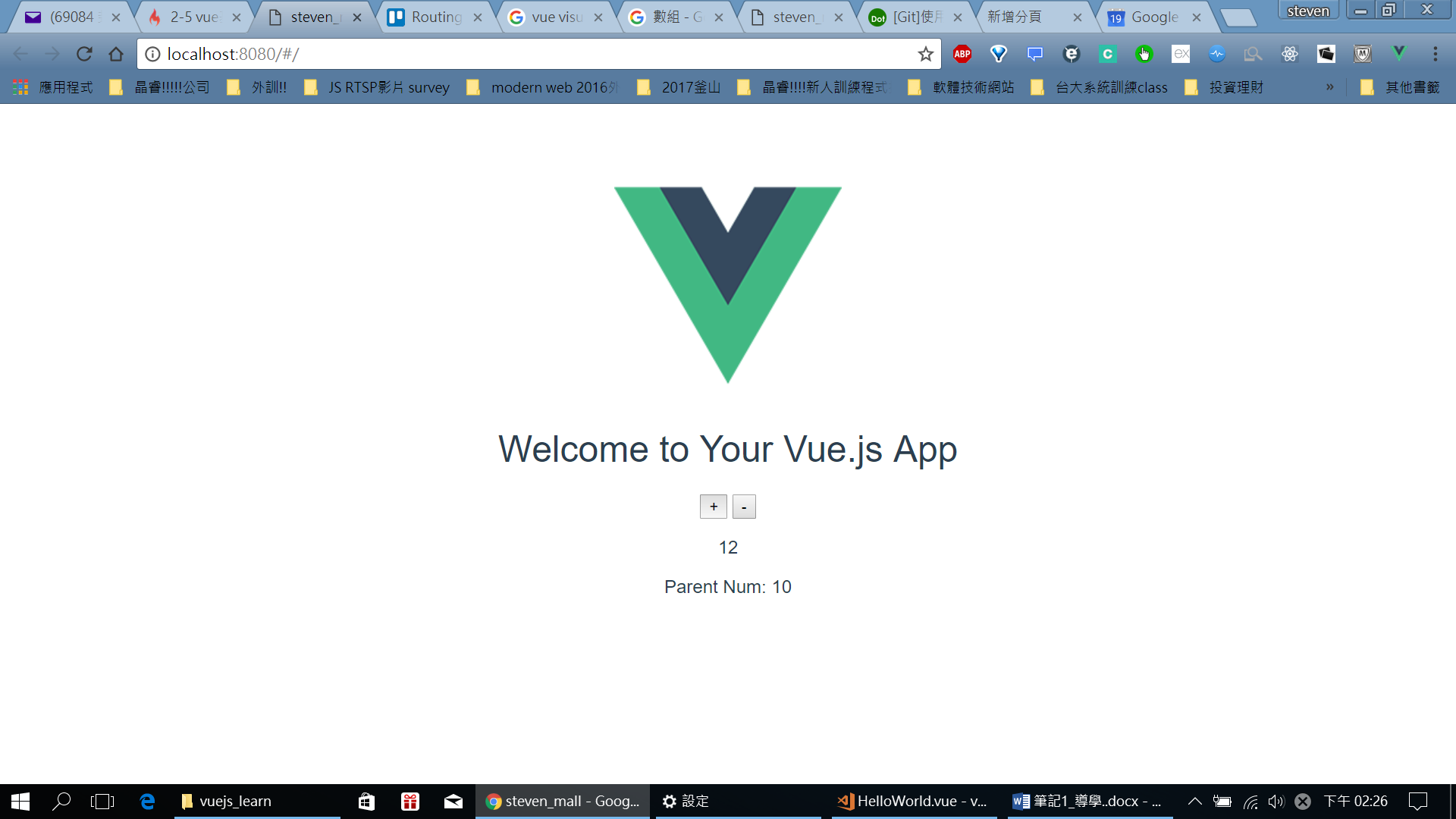
因為有用一個num=10 將值從爸爸(HelloWorld)傳給兒子(Counter)

所以UI一開始會是10



2. 若改成動態綁定





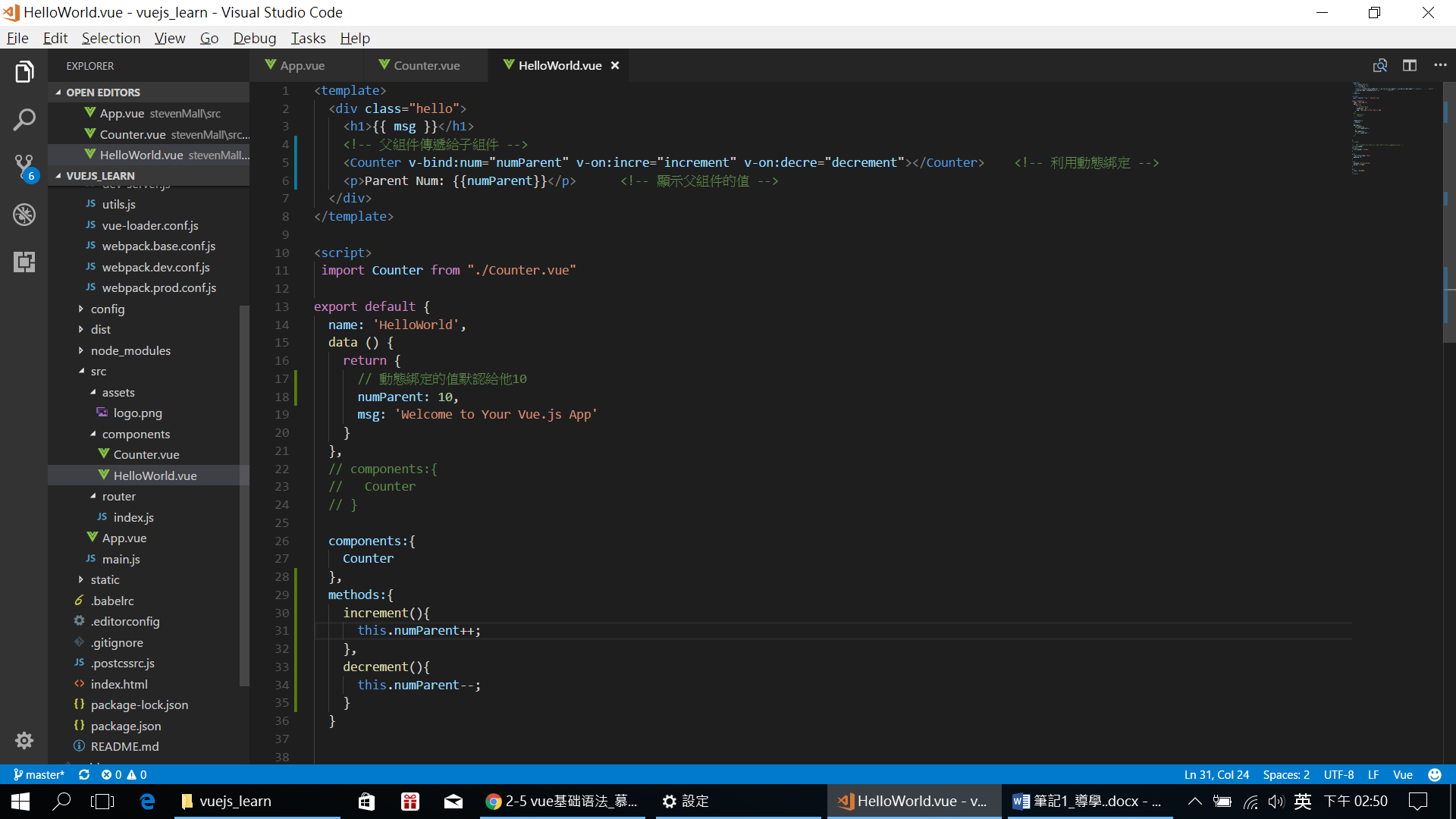
會發現父組件的值不會變!!

(因為是單向，不會從子來到父)

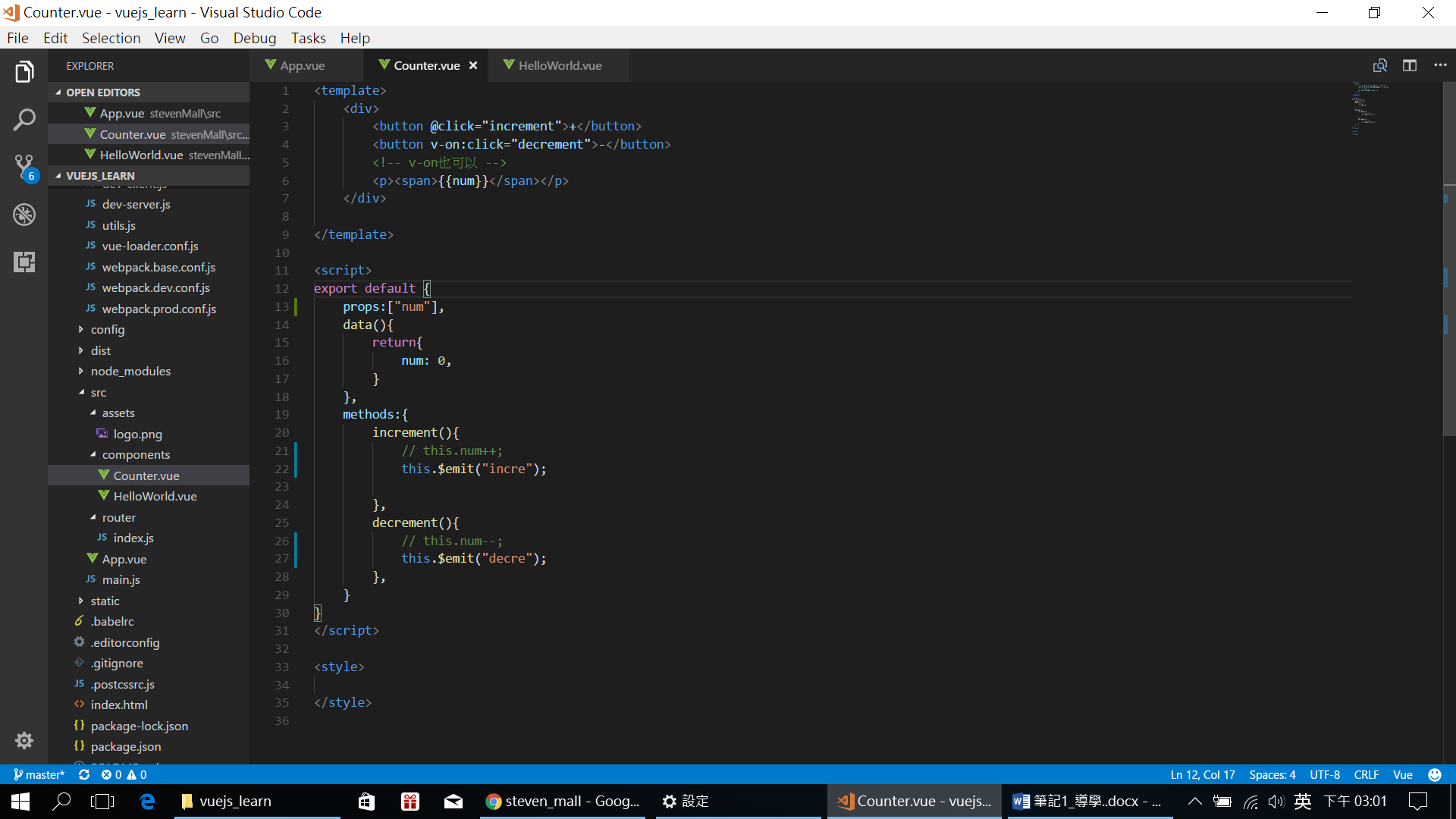
那要如何改變父的值呢?

以下要說明要如何實現雙向綁定呢?!!!!!! (其實是在子組建透過emit實現 父組件v-on綁定!!!!!)

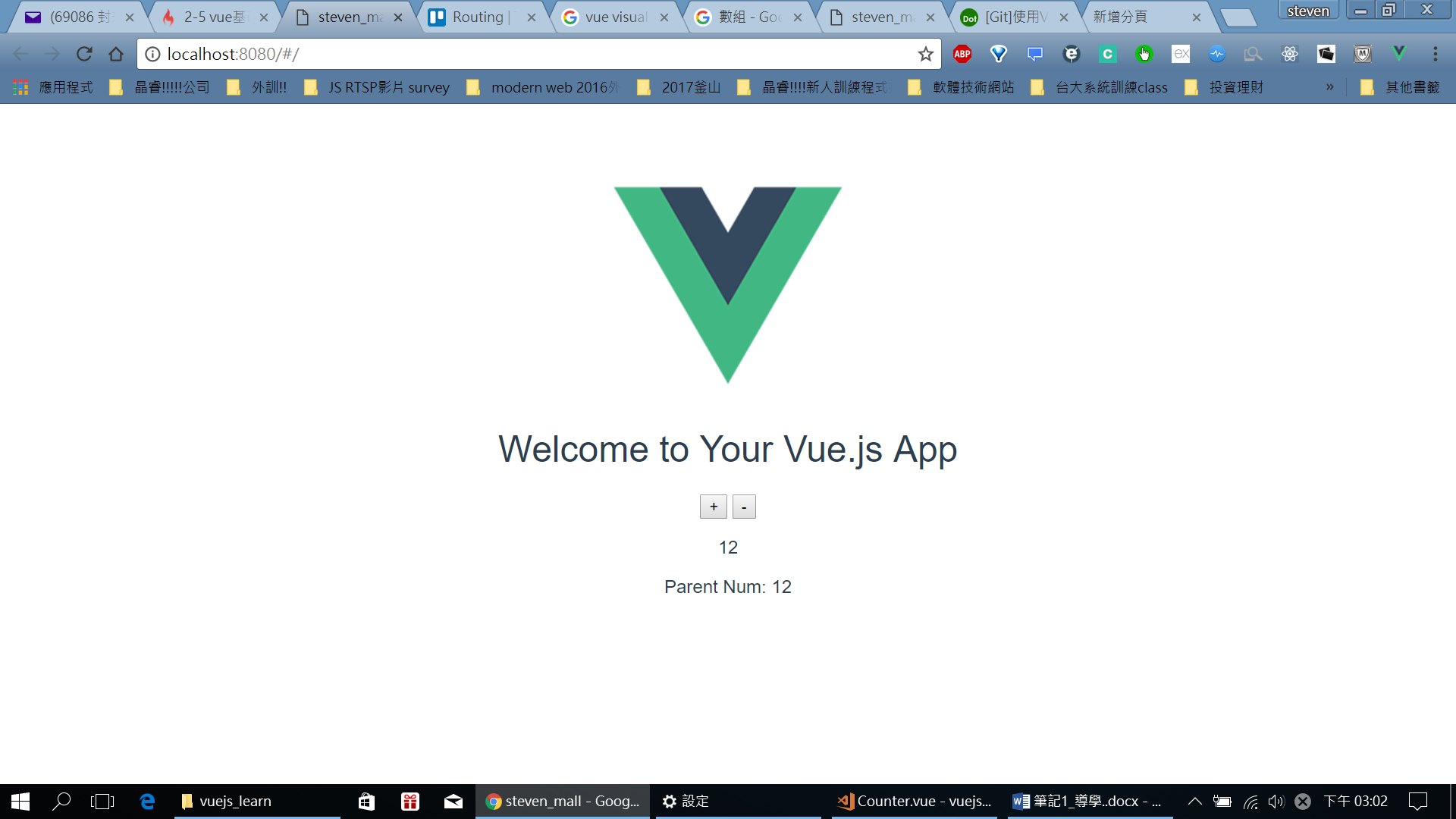
1. 先在父定義v-on:incre(自定義事件)=””



1. 在子組件的自定義increment decrement方法中，使用emit方法，當中代的參數即是父組件中的v-on綁定的文字 v-on:和子組件綁定的文字:”父組件的方法”



就會發現UI的2個值同步了!!!!



這就是因為子組件在按下button時，觸發了父組件的方法，父組件的方法改變了父組件的值

那可能會問: 那子組件的值怎麼也跟著改了呢?

這是因為>>>

看一下父組件的num綁定:

<Counter v-bind:num="numParent" v-on:incre="increment" v-on:decre="decrement"></Counter> <!-- 利用動態綁定 -->

因為我們子組件的num已經和父組件的numParent綁定了!!!!!!!

如此就達到雙向同步的功能了!!!!

<這種方法可用在一些特殊情況例如，菜單 子菜單新增的渲染等等!>

比較複雜的 之後會教學使用vuex!