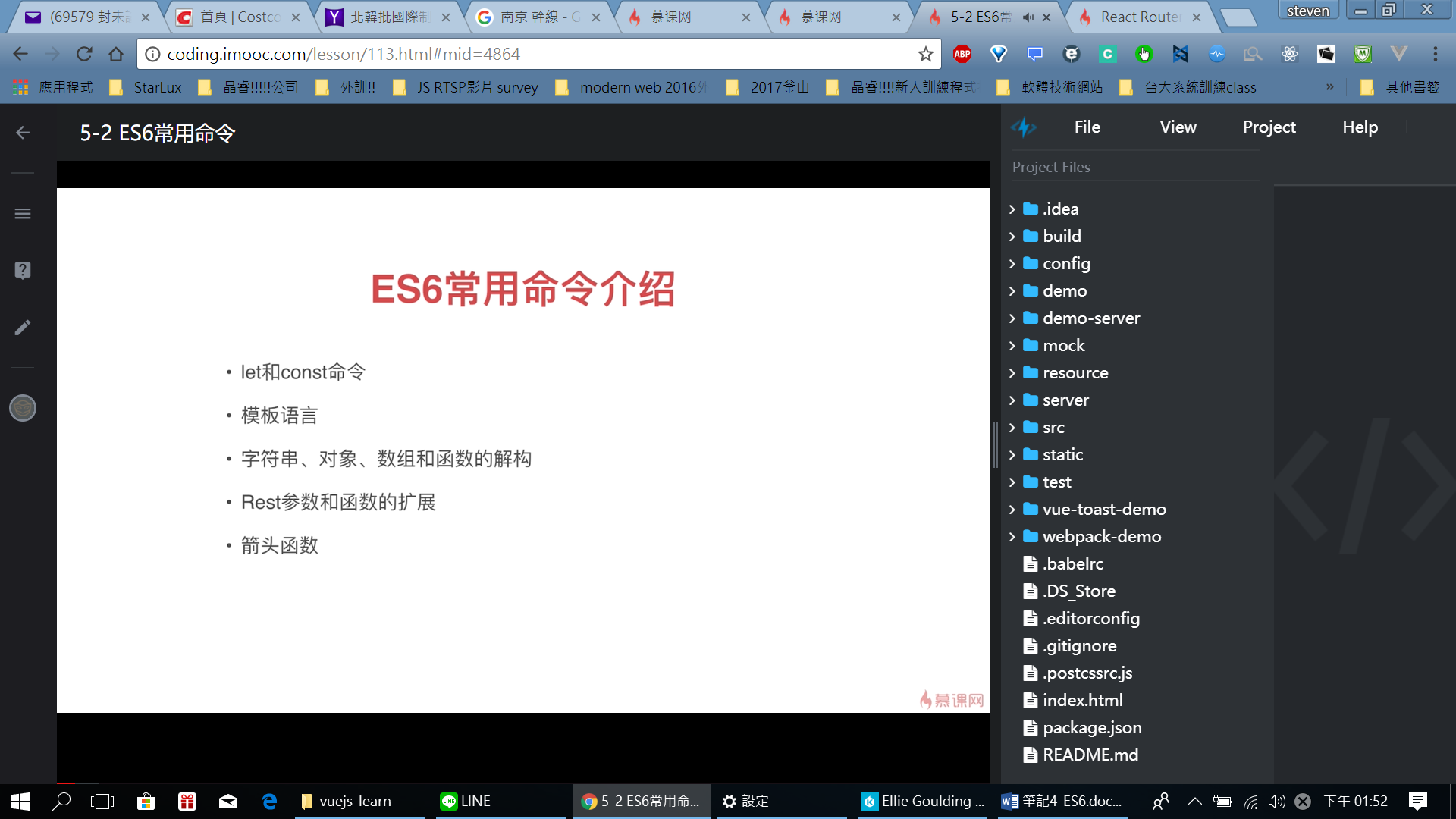
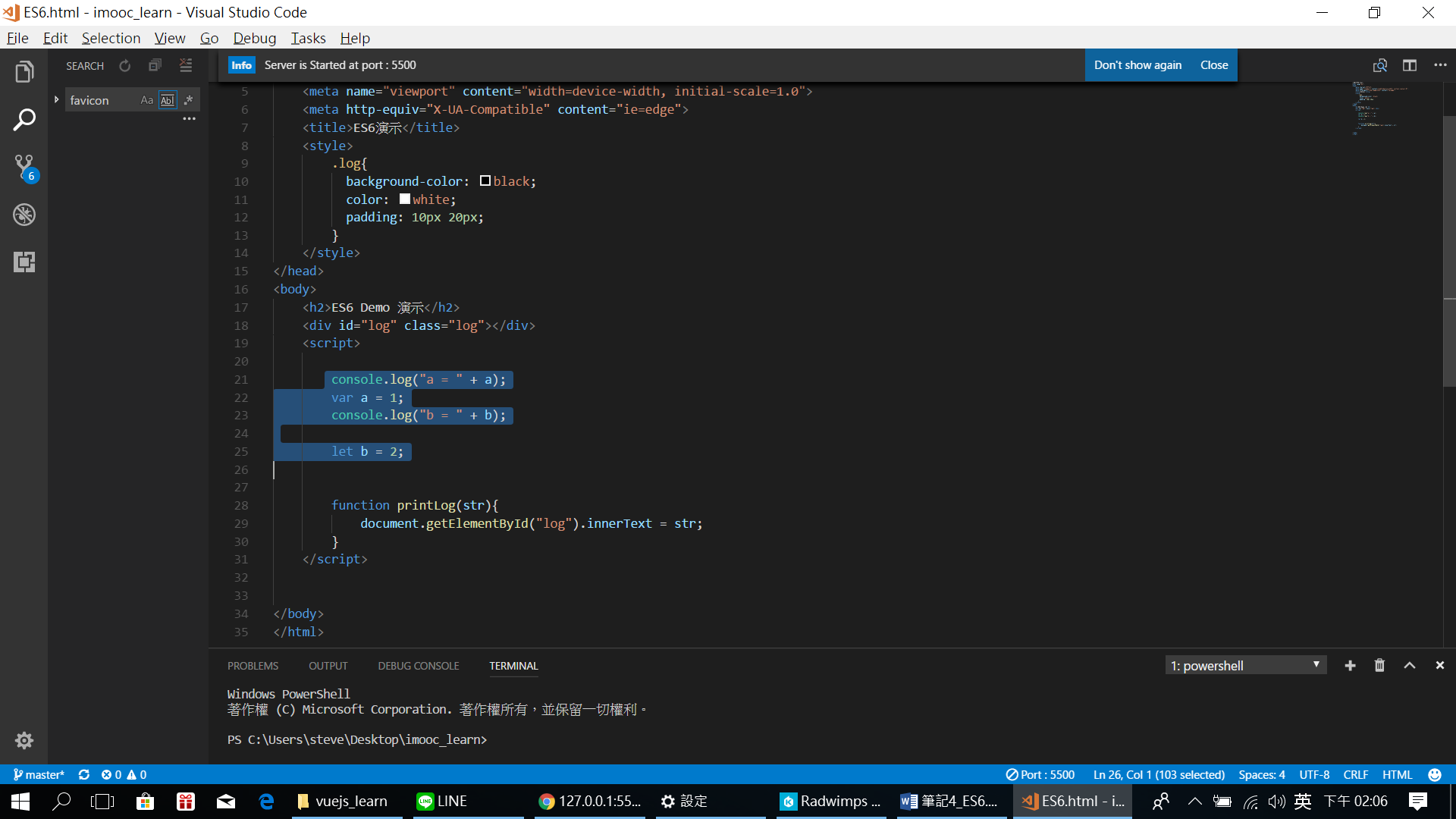
1. 
2. 接下來的解析不用babel解析，因為最新瀏覽器版本幾乎都可以支持

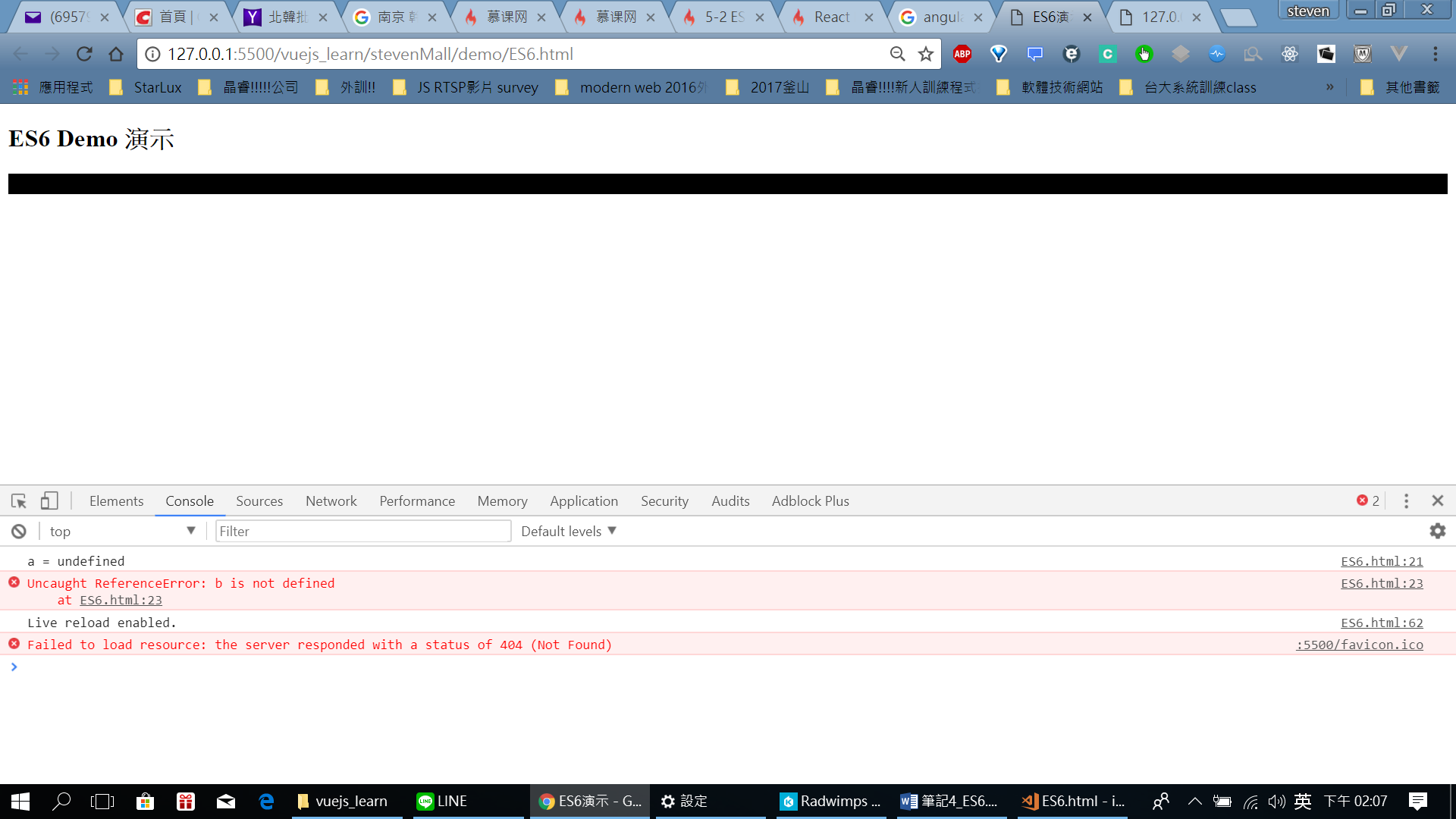
那為啥vue還要babel，是因為開發應用要盡量解決最大多數人(盡量要100%用戶)

<var let>

Var有變量提升> 其實會先定義



會看到:



a顯示undefined b卻是 is not define

>> 這就是因為var 有變量提升的作用， 所以a 其實以定義了只是還沒賦值

B則是沒有

A相當於:

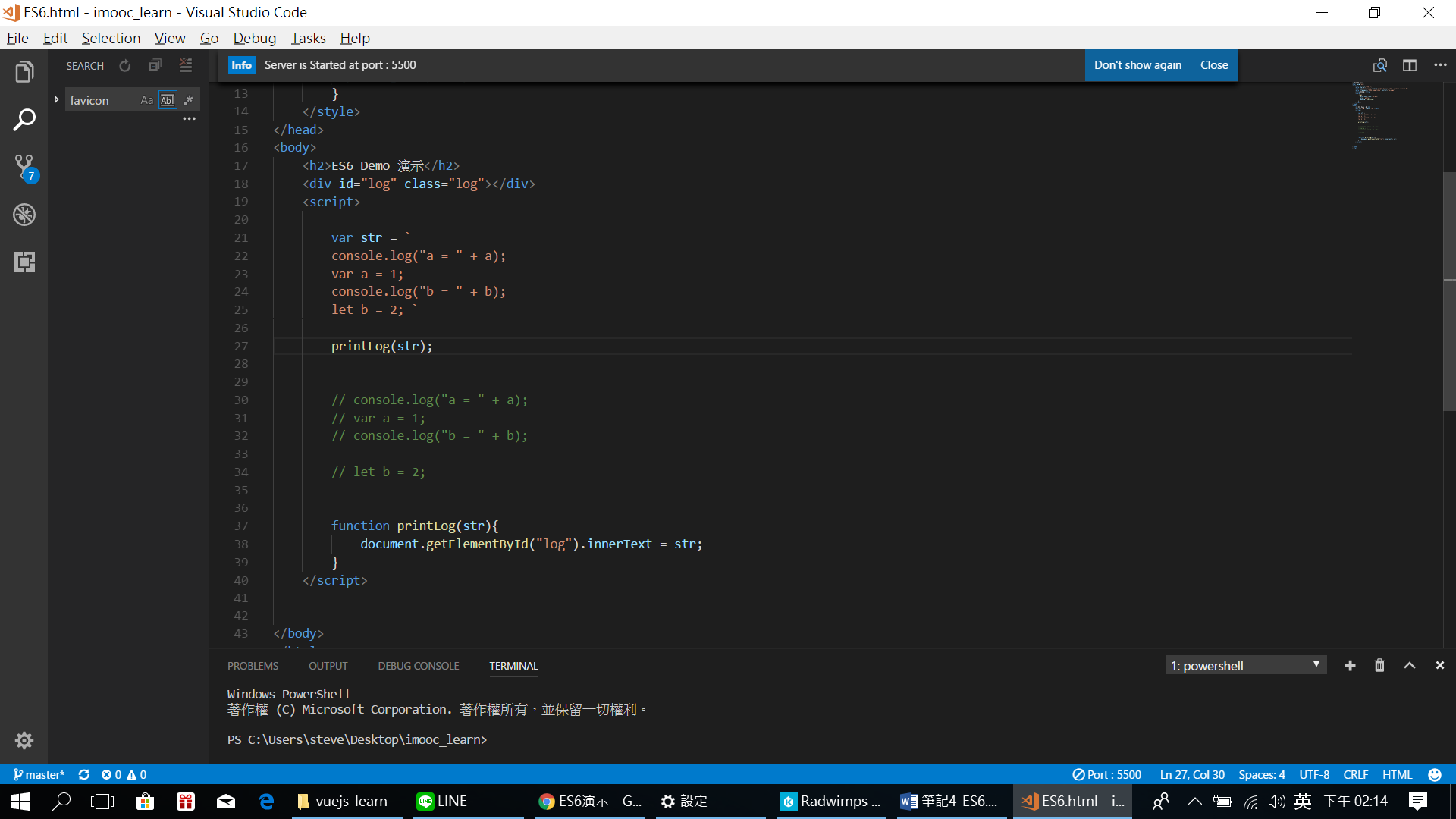
Var a;

Console.log(a);

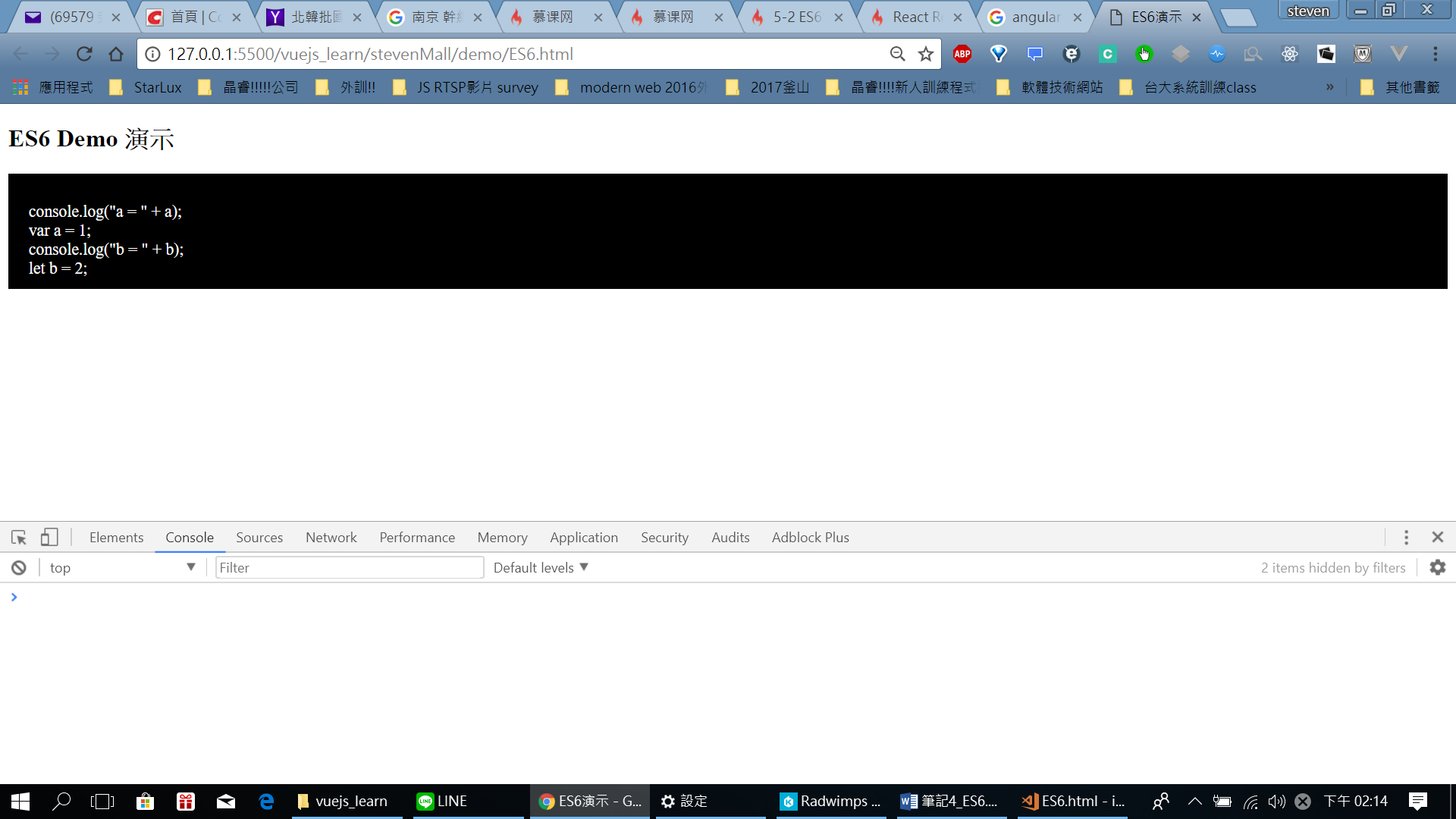
a = 1;

<模板語言!>

使用` 符號:



就可以看到:



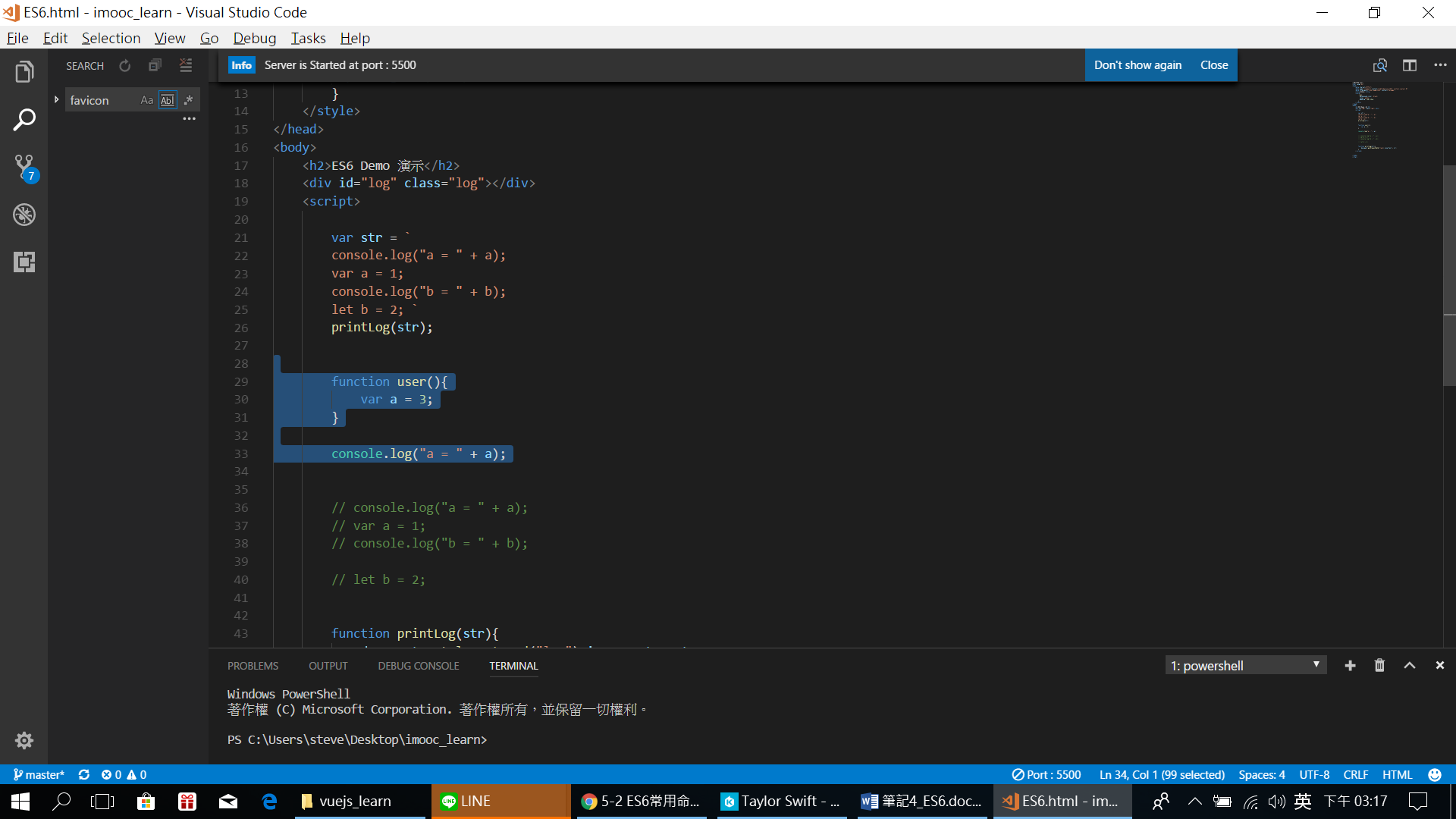
如果是使用傳統string +=方式

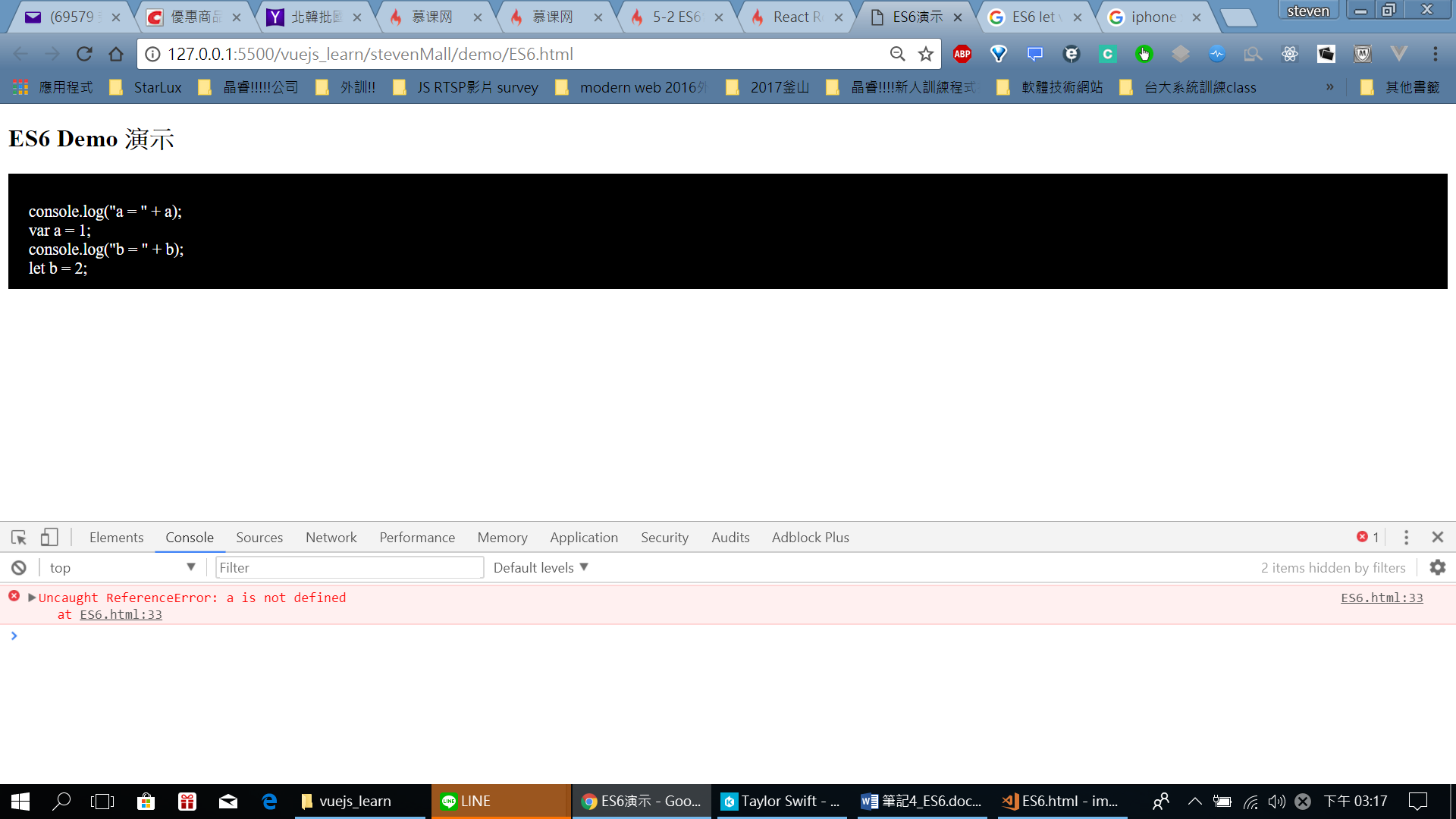
會變成黏在第一列!

<作用域>

Var 是函數型做用域

Let 和const是塊級{} 作用域

因為var是函數級作用域(作用域僅限於函數內部)，會顯示is not defined:



但同剛剛如果在外面下方Var 是可以的

塊級作用域:

{

let a = 1;

var b = 2;

}

console.log("a:" + a);

>> ES6.html:40 Uncaught ReferenceError: a is not defined

at ES6.html:40 (因為a只在{}內)

{

let a = 1;

var b = 2;

}

console.log("b:" + b);

b是正常可以拿到b:2的

>> 因為var 是函數行作用域，在塊級仍然可以被提升到外面!!!

以後盡量用let >> 可即時消毀 var會有一些內存洩漏!!

<Const>

定義的內容不可改變>

但:

const b = {a:1};

b.a = 2;

b.c = 10;

console.log(b)

b會變成{a: 2, c: 10}

因為是指b隊到的記憶體位址(因為是定義為物件類型!!! )不能被改變，但該位址隊到的內容是可以的

但如果寫成:

b = {d:1}

>>>> 就會噴錯囉!

<模板語言>

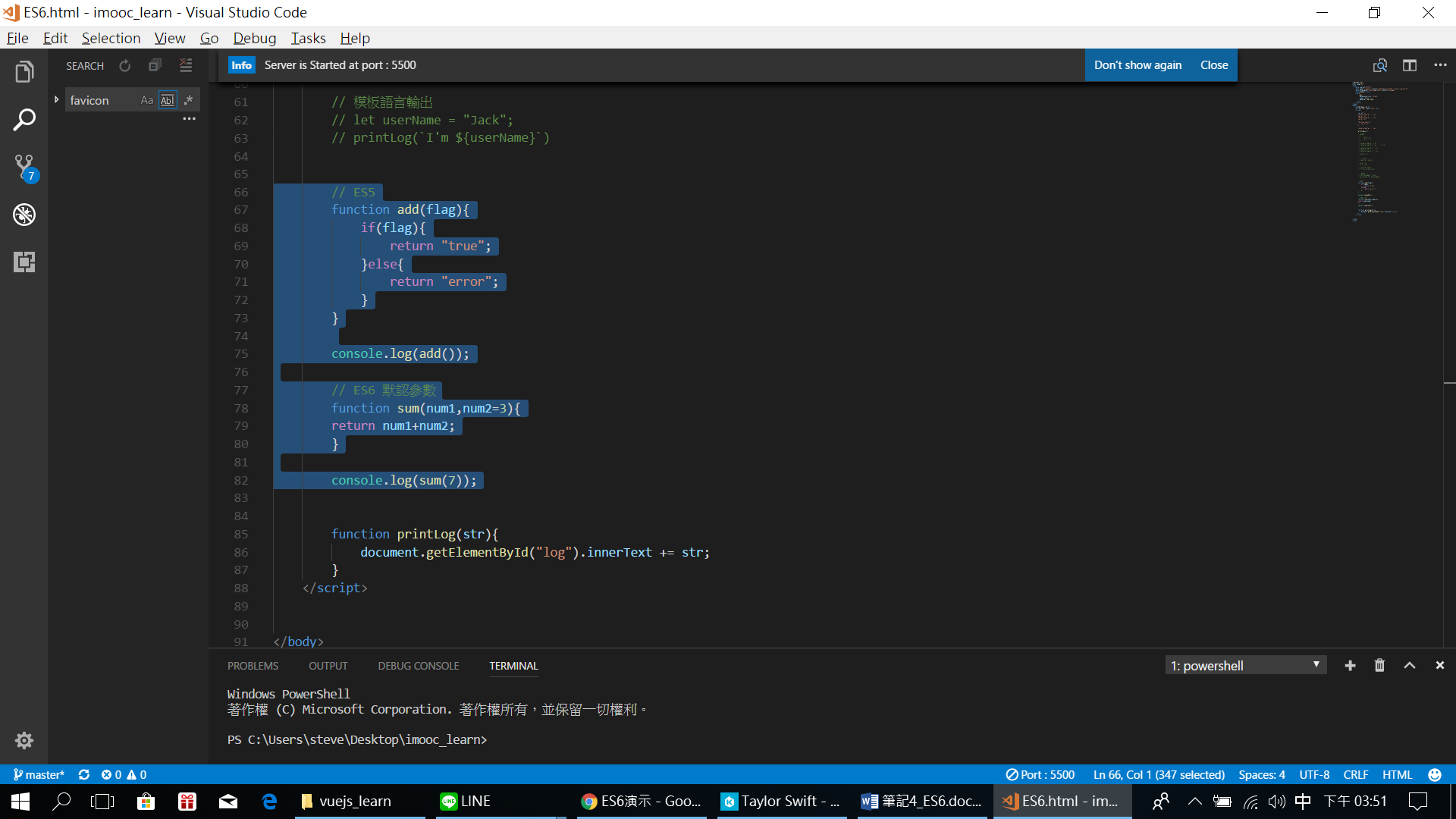
ES6新的魅力:

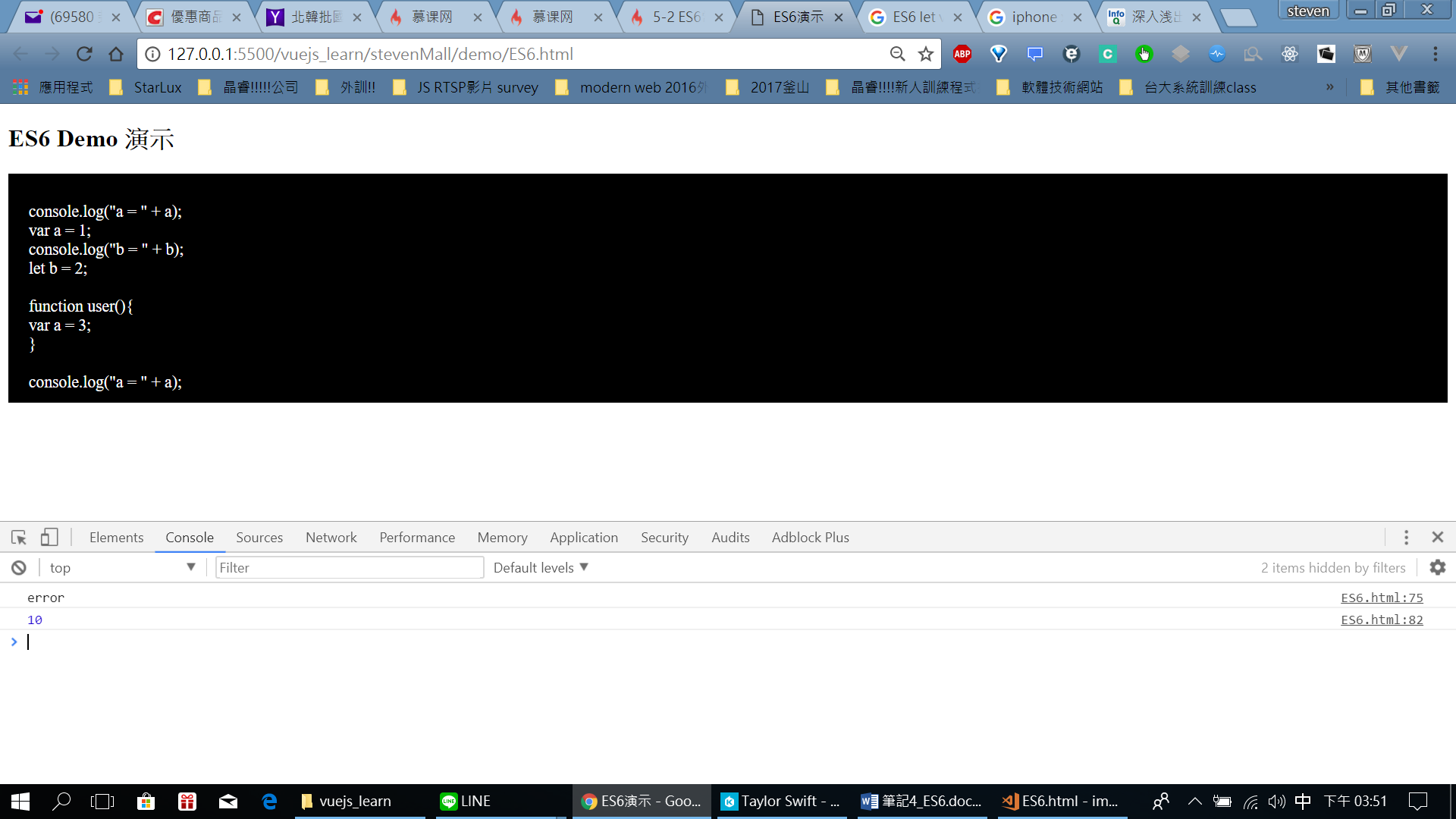
let userName = "Jack";

printLog(`I'm ${userName}`)

用`` 和 ${} 就可以打印出來了! (保存固有格是 不用+)

<默認值> ES6 運許默認參數





<箭頭函數>

// 箭頭函式:

var arr = [3,6,9];

var newArr = arr.map(function (item) {

return item+2;

})

console.log(newArr);

var newArr1 = arr.map((item)=>item+2); // 如果不是只有item+2這麼一行> 需要用大括號! 箭頭函式可以想成就是ES5傳統的function後面的值

// 另外箭頭函式有this的差異>> 在筆記會解說!!!

console.log(newArr1);

(3) [5, 8, 11]

(3) [5, 8, 11]

<箭頭函數 在前面this. 的差異!!!>

在不用箭頭函數 和使用箭頭函數的this的範圍可能不同，請小心!!