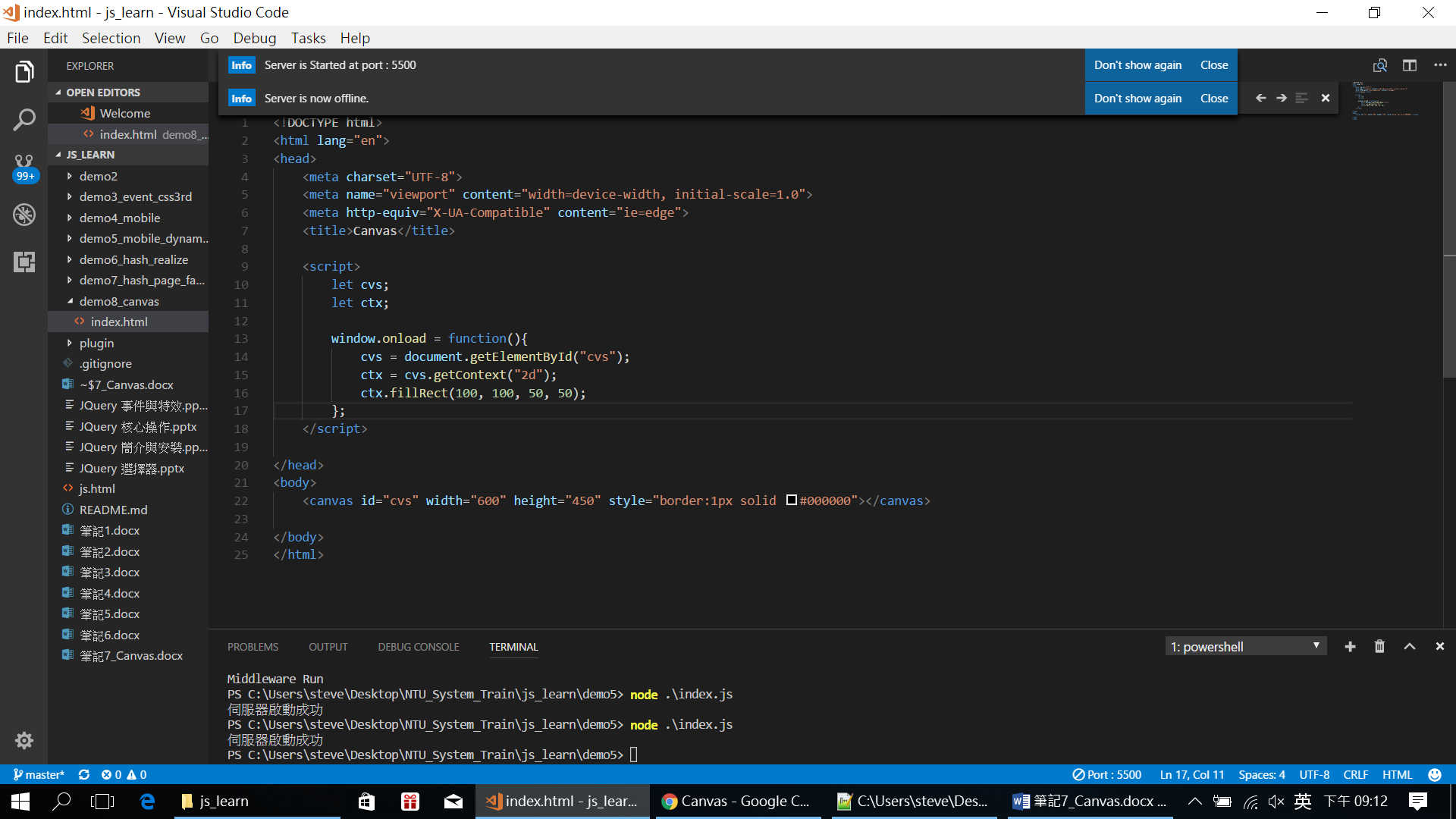
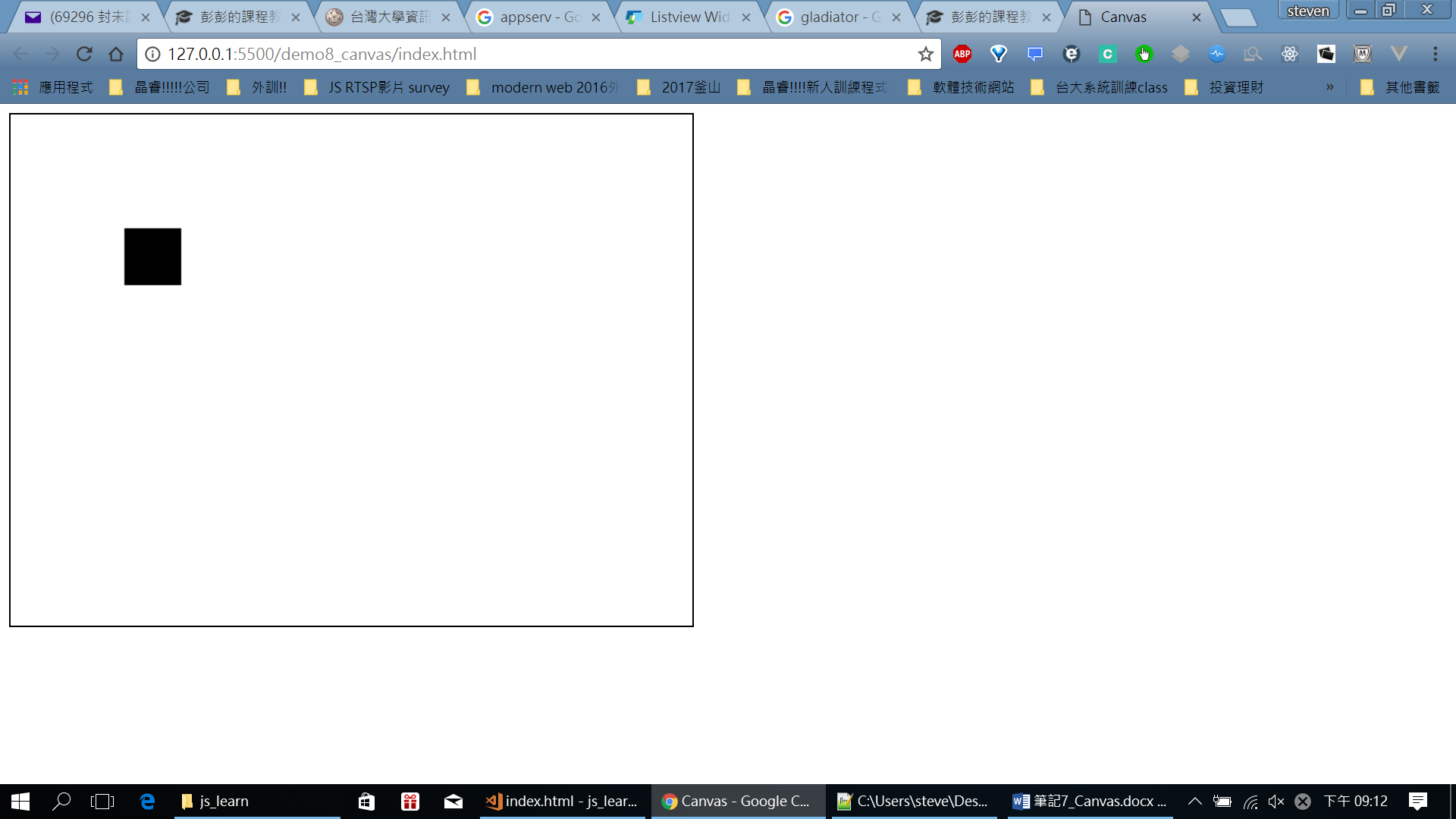
Canvas (2d, webgl(複雜) )

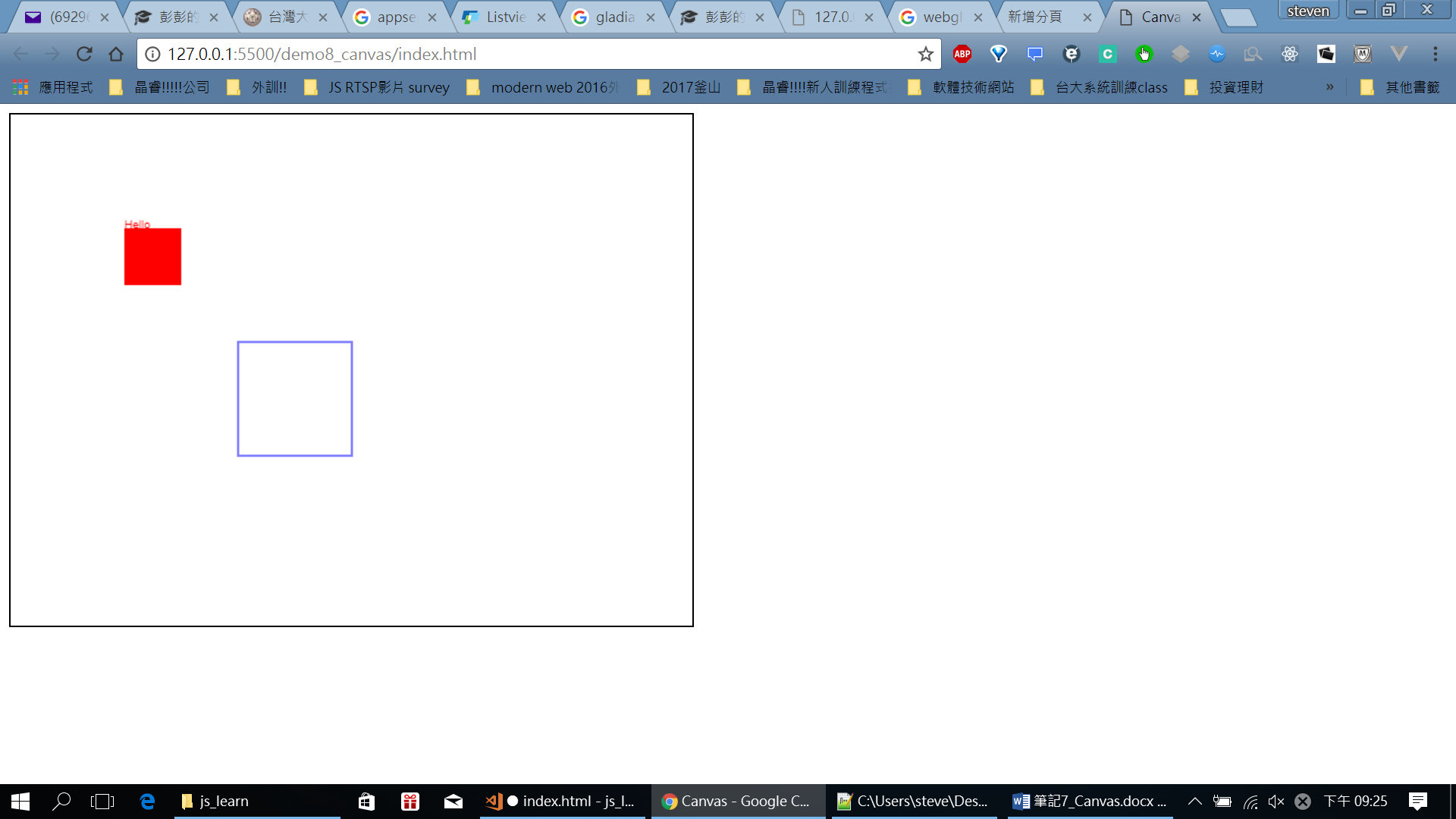
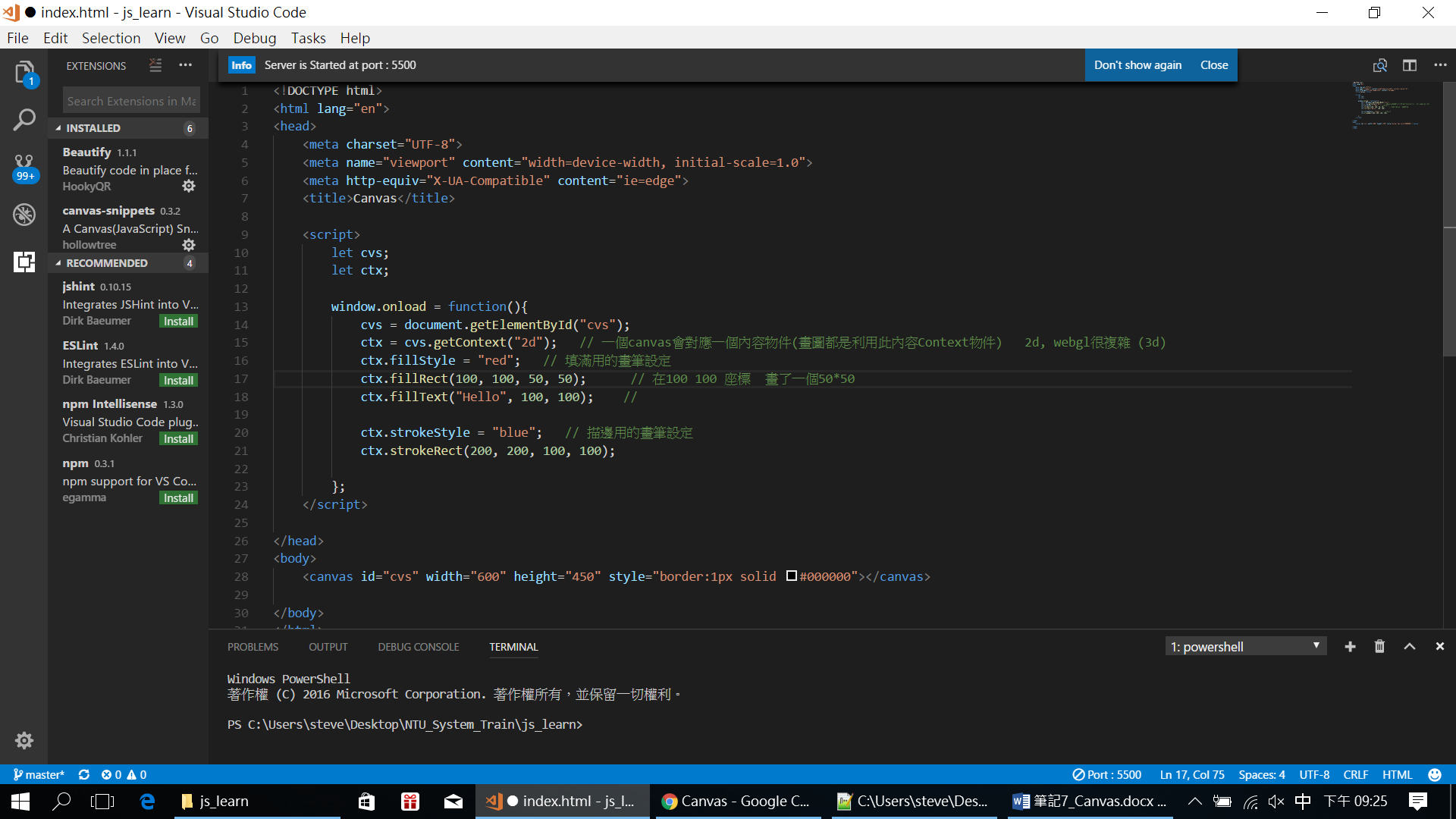
1.



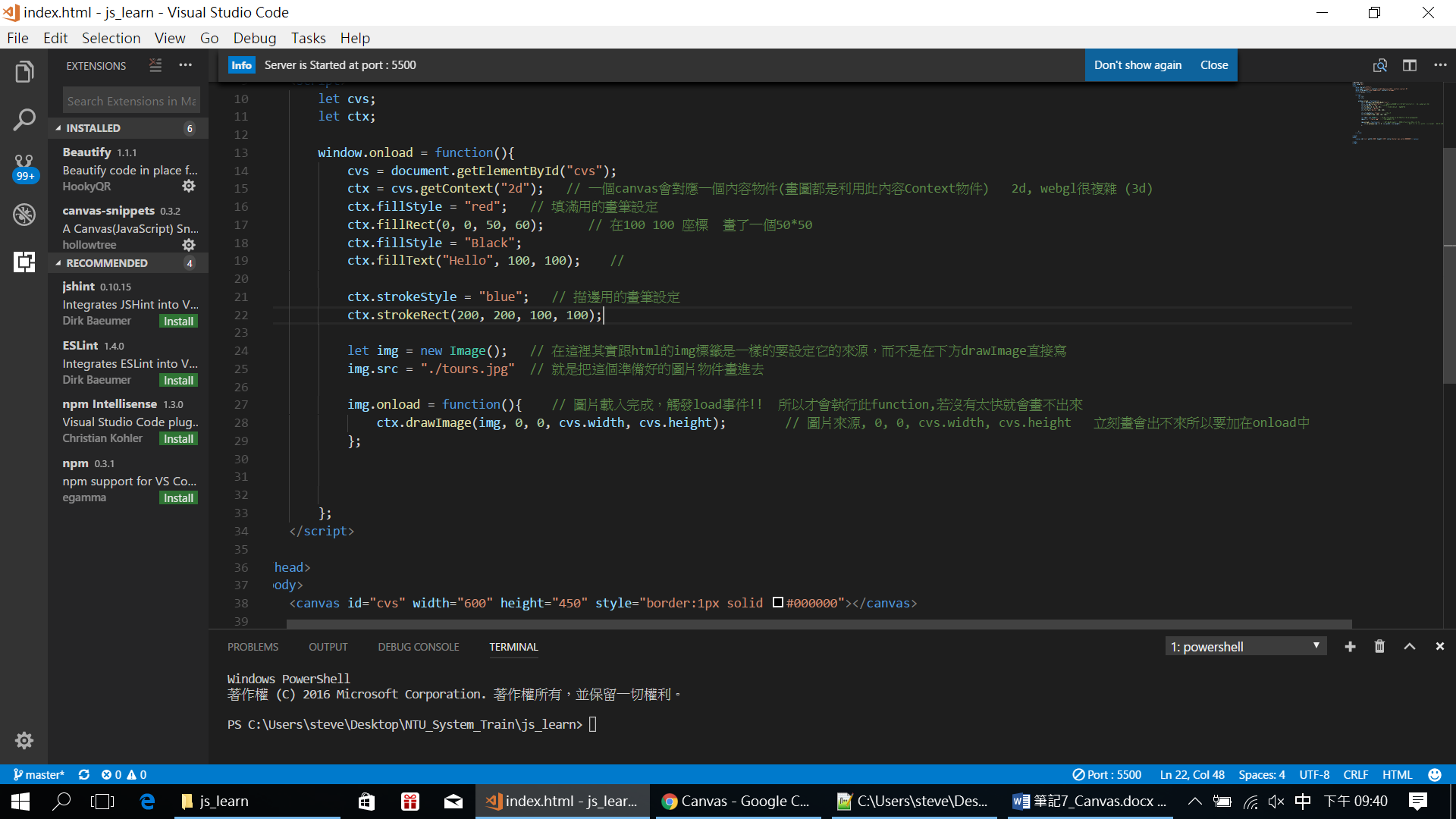


Cancas提供2種畫筆: fill填滿(預設黑色)、stroke描邊

Fillstyle可以設定填滿用畫筆



2. 練習載入圖片

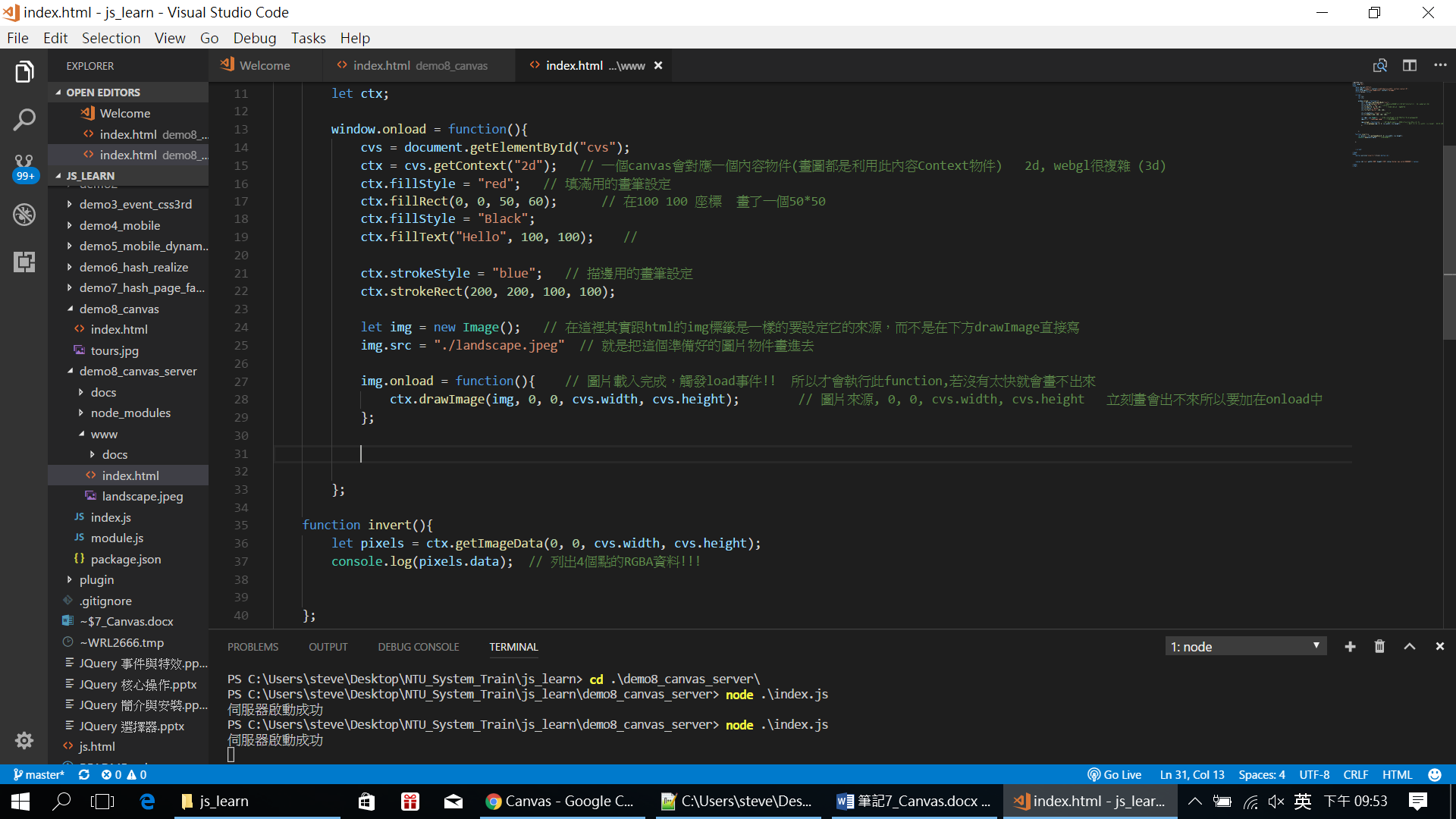


3. 練習做濾鏡吧!!!

EX, 油畫濾鏡就是把某項速周昭出現頻率最高的點來決定附近周遭某些像素的顏色

RGBA 3顏色(紅 綠 藍)+透明度 0~255

看一下



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Canvas</title>

<script>

let cvs;

let ctx;

window.onload = function(){

cvs = document.getElementById("cvs");

ctx = cvs.getContext("2d"); // 一個canvas會對應一個內容物件(畫圖都是利用此內容Context物件) 2d, webgl很複雜 (3d)

ctx.fillStyle = "red"; // 填滿用的畫筆設定

ctx.fillRect(0, 0, 50, 60); // 在100 100 座標 畫了一個50\*50

ctx.fillStyle = "Black";

ctx.fillText("Hello", 100, 100); //

ctx.strokeStyle = "blue"; // 描邊用的畫筆設定

ctx.strokeRect(200, 200, 100, 100);

let img = new Image(); // 在這裡其實跟html的img標籤是一樣的要設定它的來源，而不是在下方drawImage直接寫

img.src = "./landscape.jpeg" // 就是把這個準備好的圖片物件畫進去

img.onload = function(){ // 圖片載入完成，觸發load事件!! 所以才會執行此function,若沒有太快就會畫不出來

ctx.drawImage(img, 0, 0, cvs.width, cvs.height); // 圖片來源, 0, 0, cvs.width, cvs.height 立刻畫會出不來所以要加在onload中

};

};

function invert(){

let pixels = ctx.getImageData(0, 0, cvs.width, cvs.height);

console.log(pixels.data); // 列出4個點的RGBA資料!!!

};

</script>

</head>

<body>

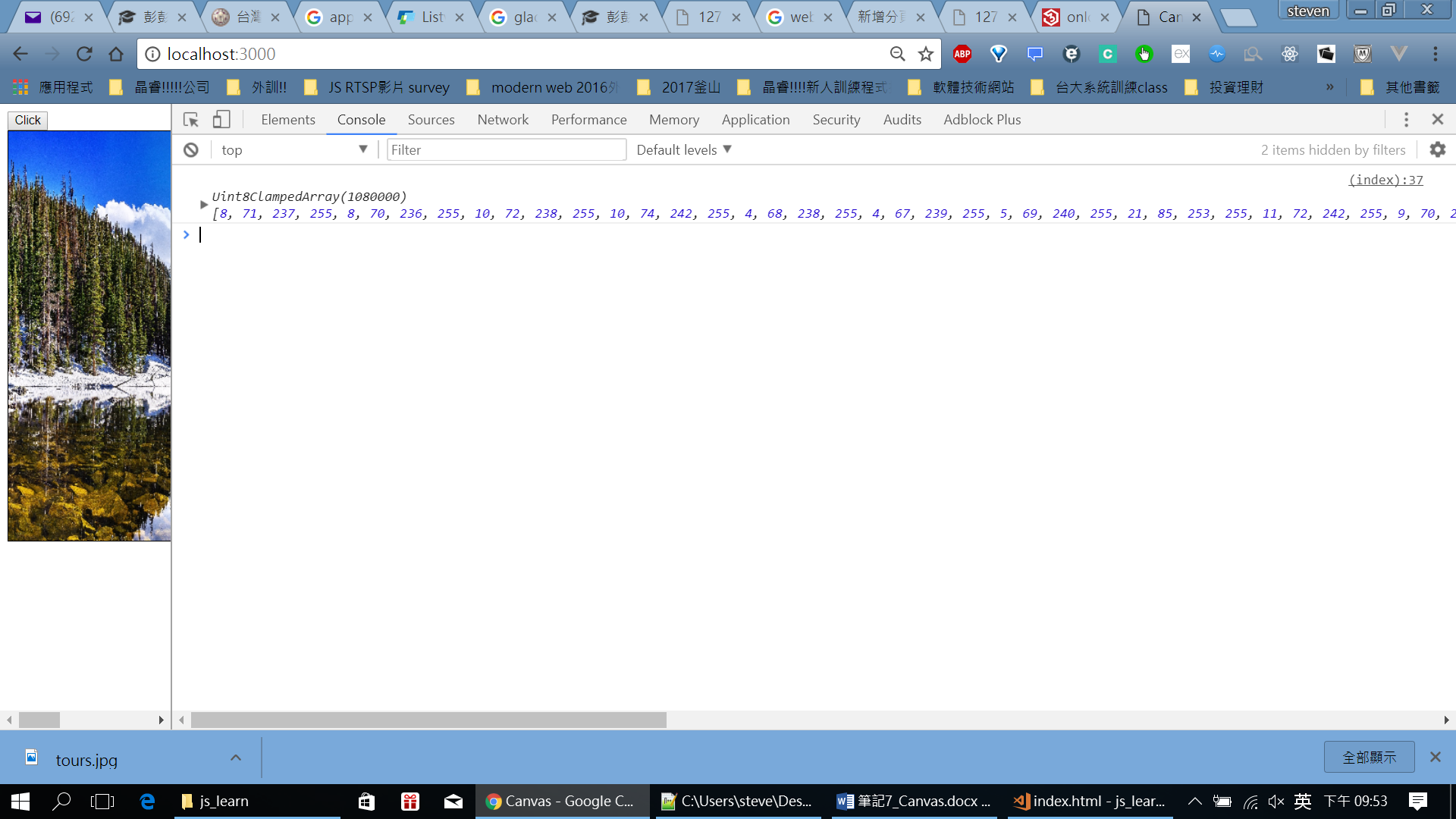
<button onclick="invert()">Click</button><br>

<canvas id="cvs" width="600" height="450" style="border:1px solid #000000"></canvas>

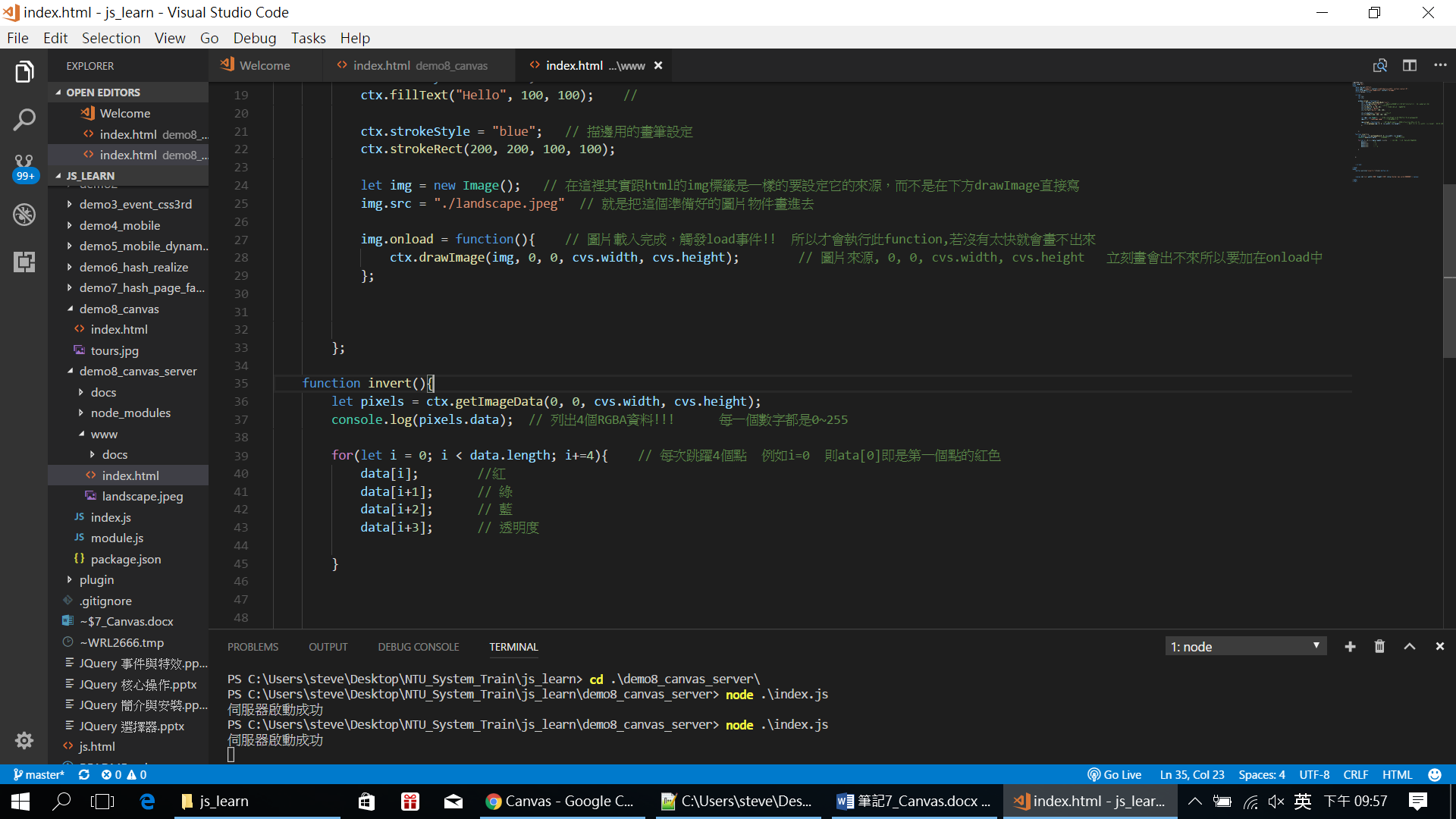
</body>

</html>

Pixels data



來loop:



反轉就是用255去減囉:



成功拉:

