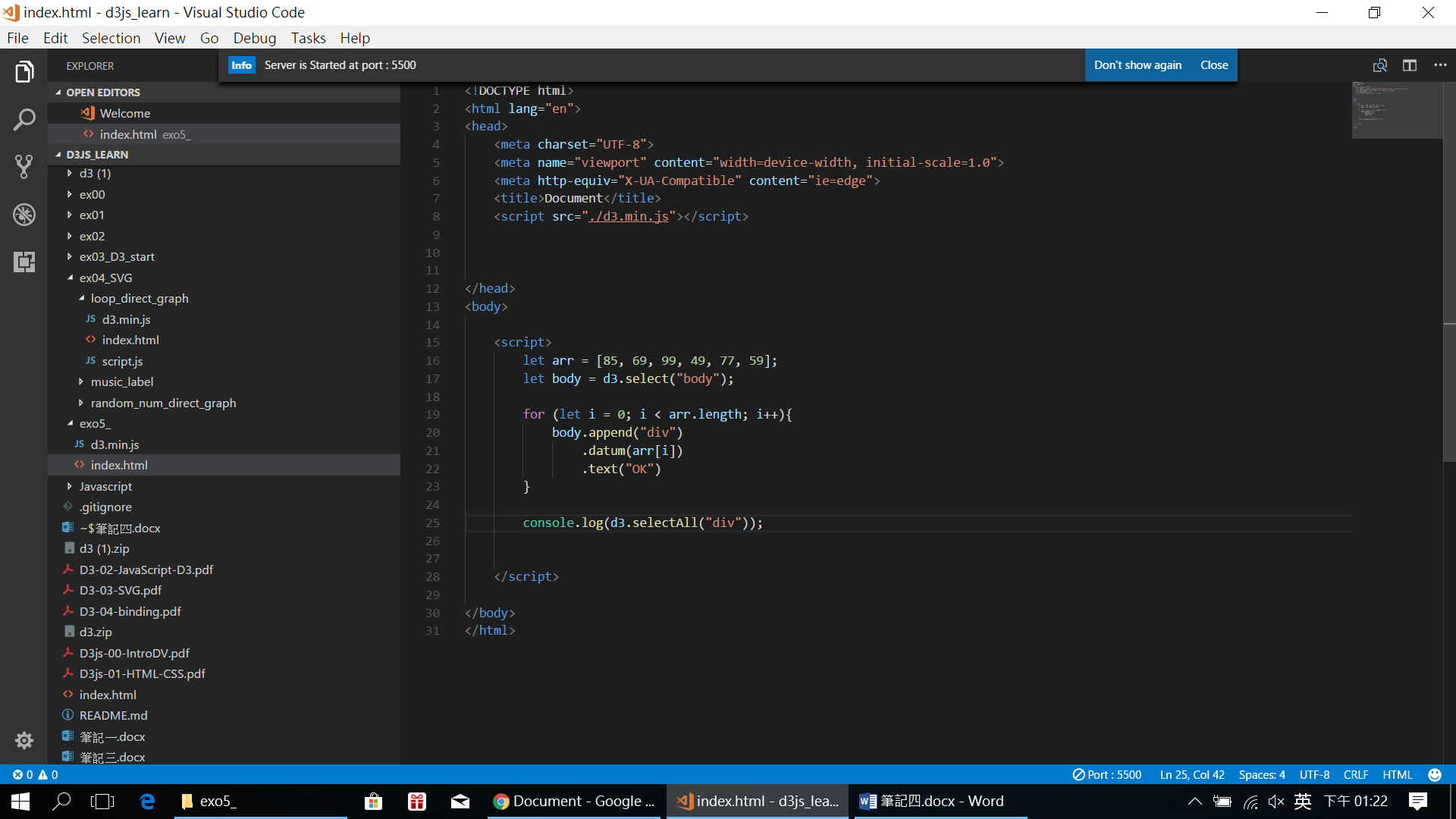
資料綁定

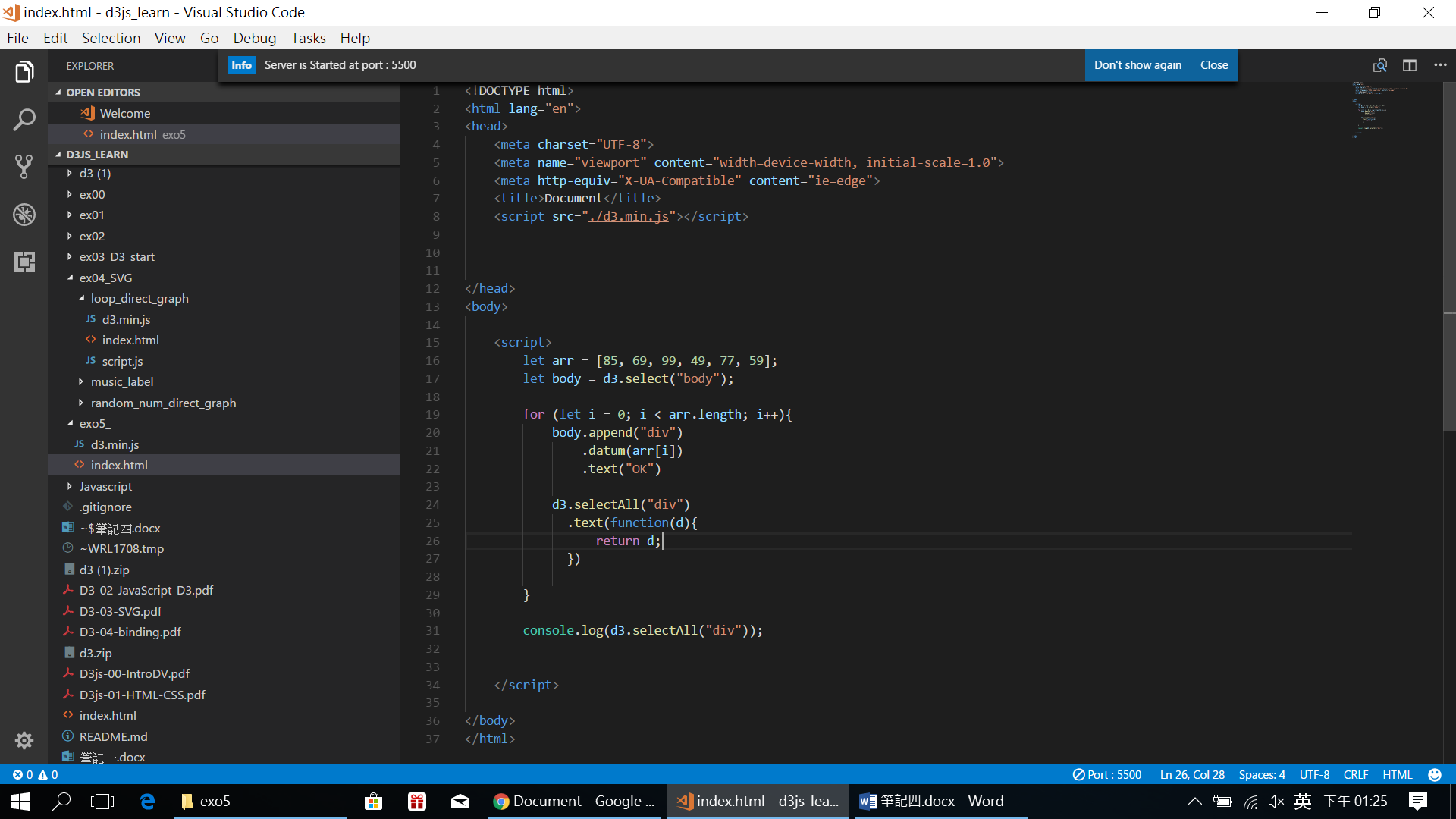
.datum();

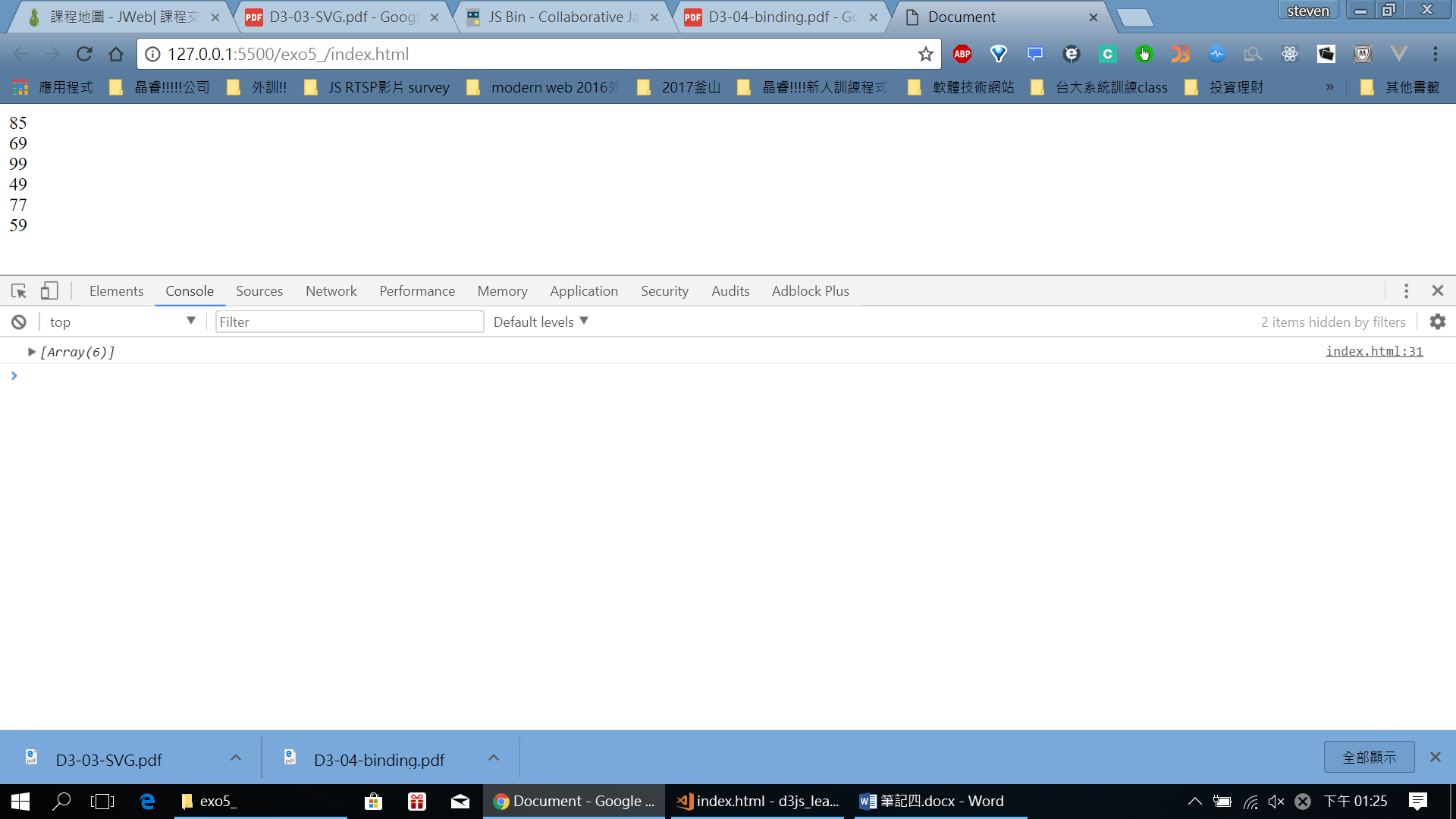


可以看到\_\_data\_\_ 看到值

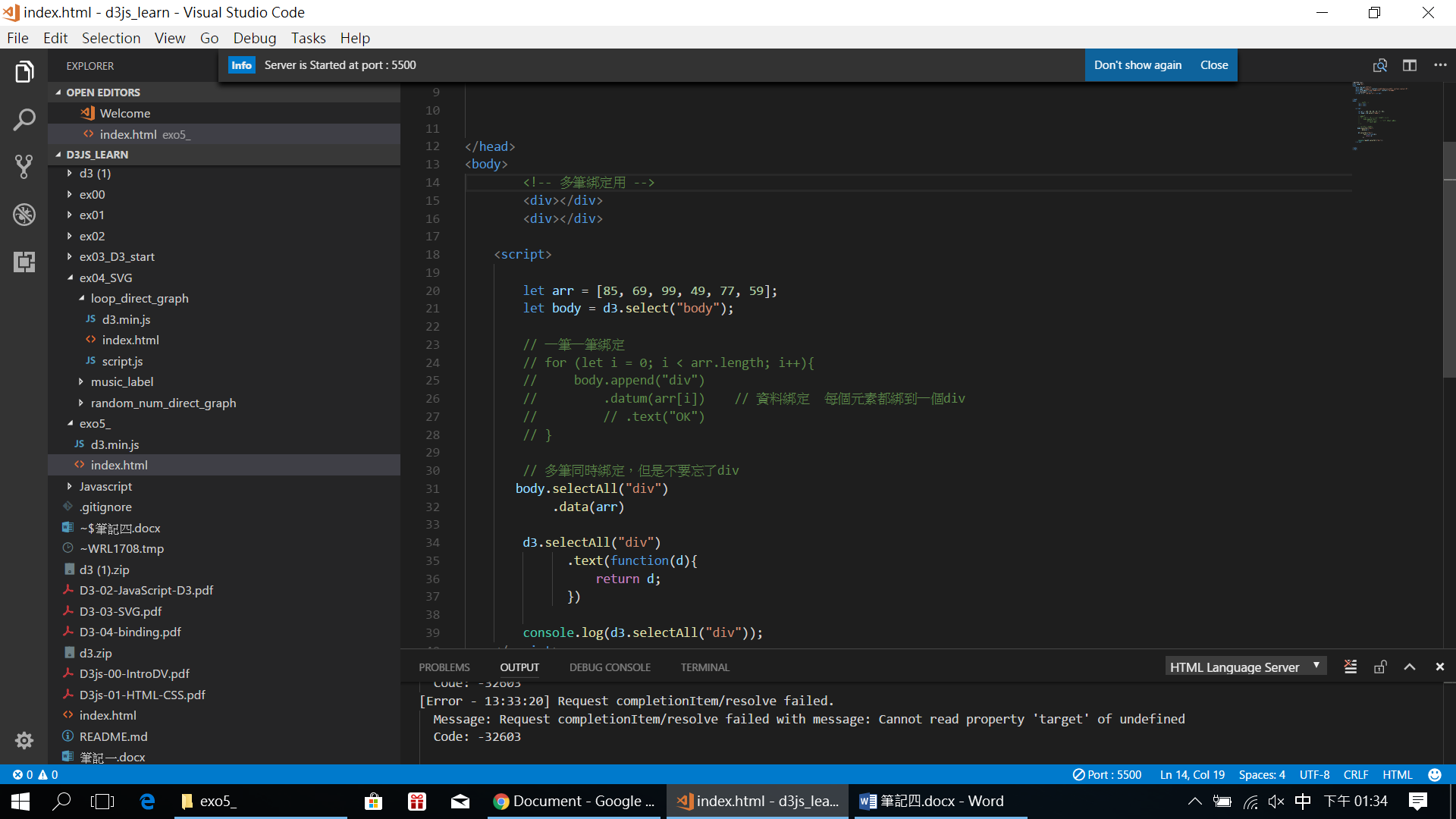


這時候改寫為:

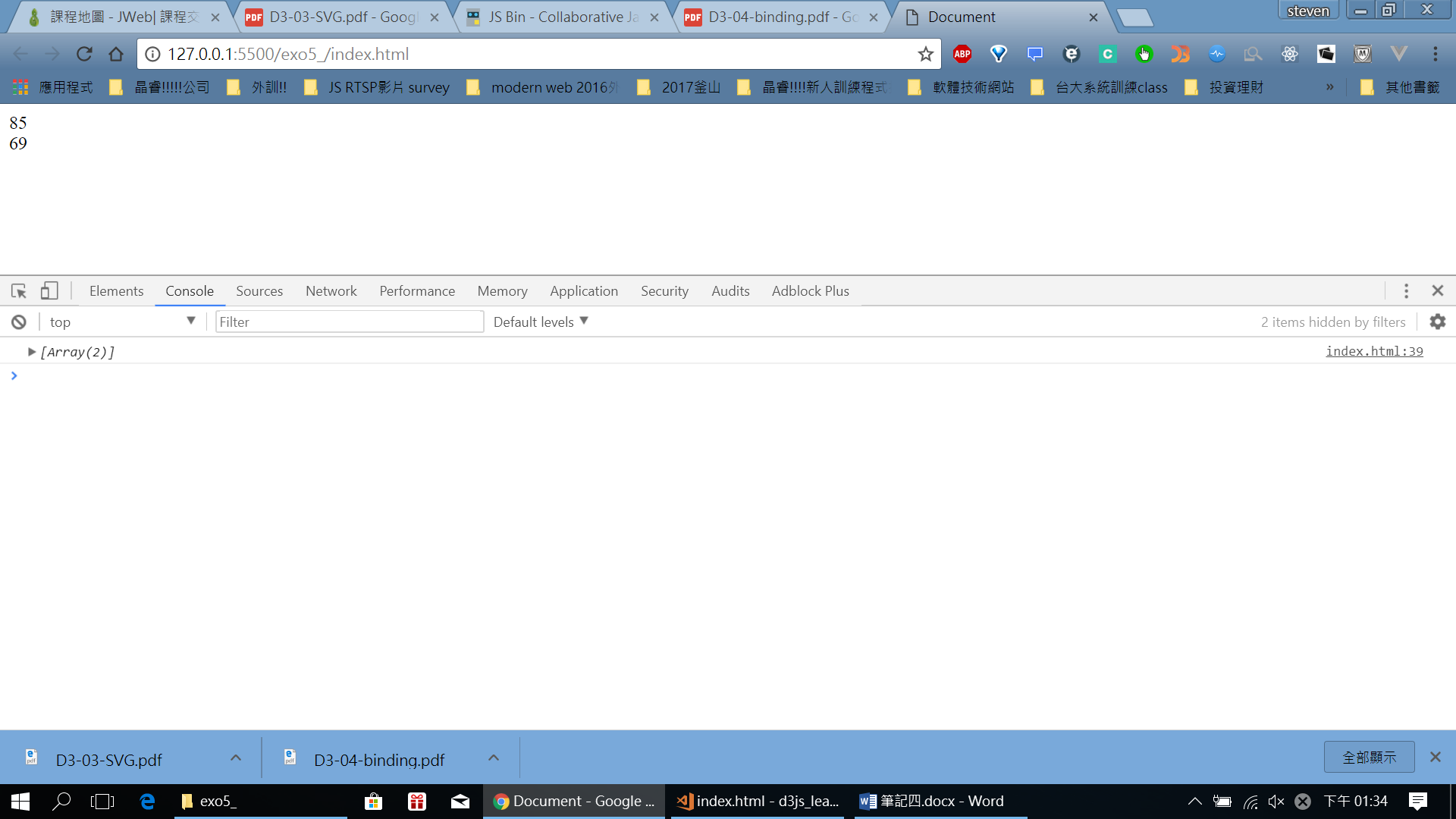




2. 多筆綁定:

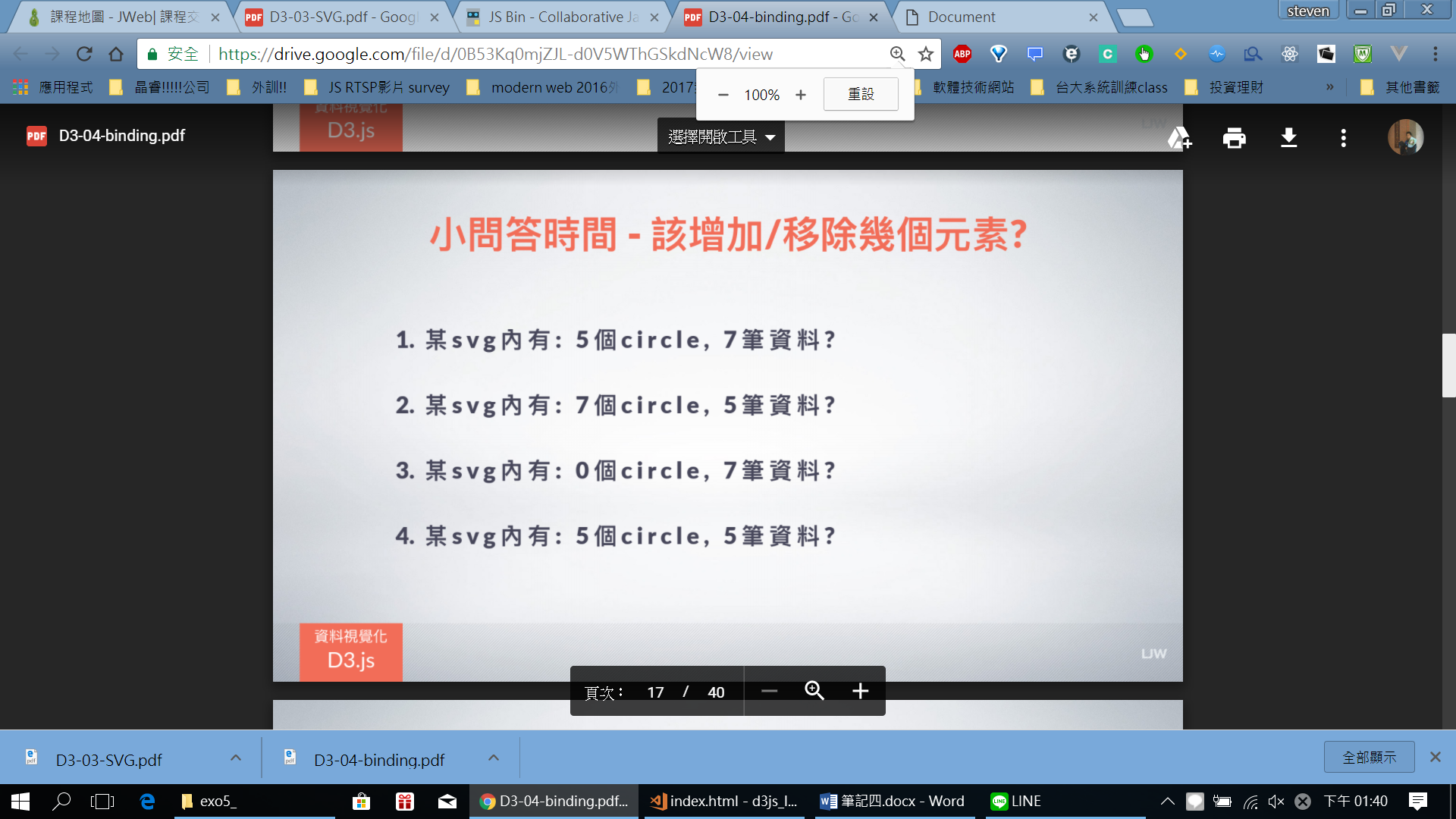


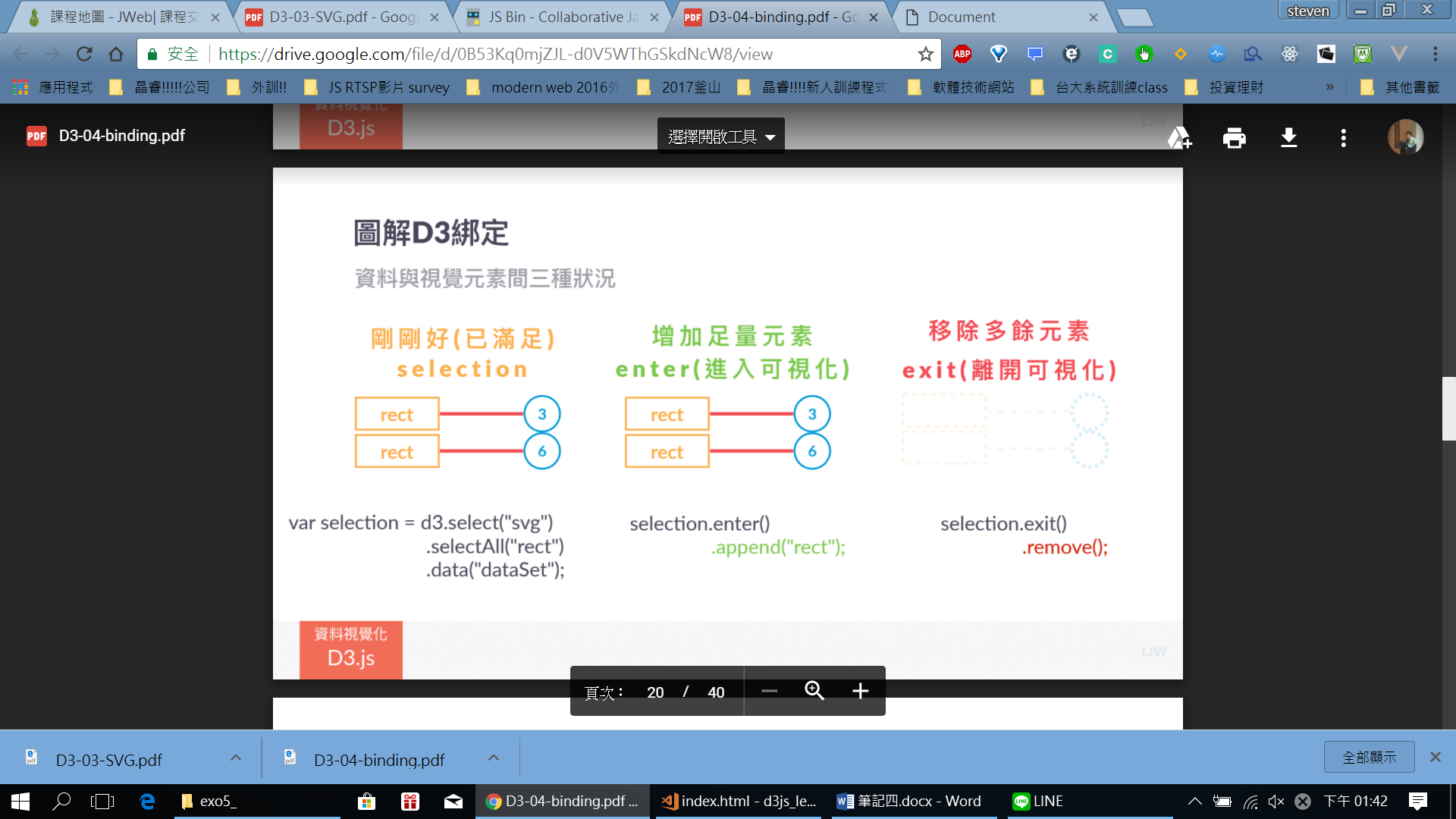
但要自己加<div>:



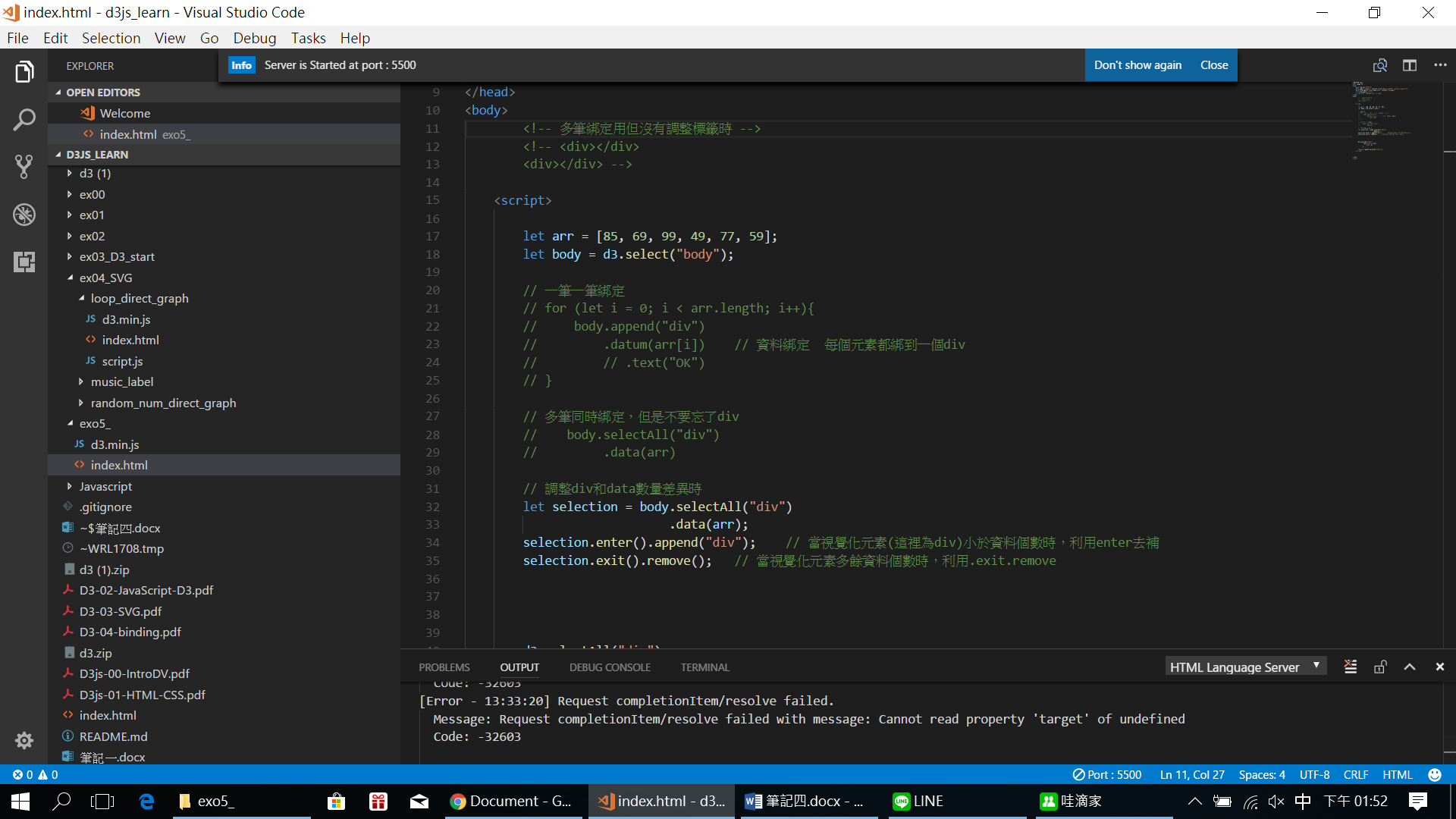
2個div就才顯示2筆

所以:

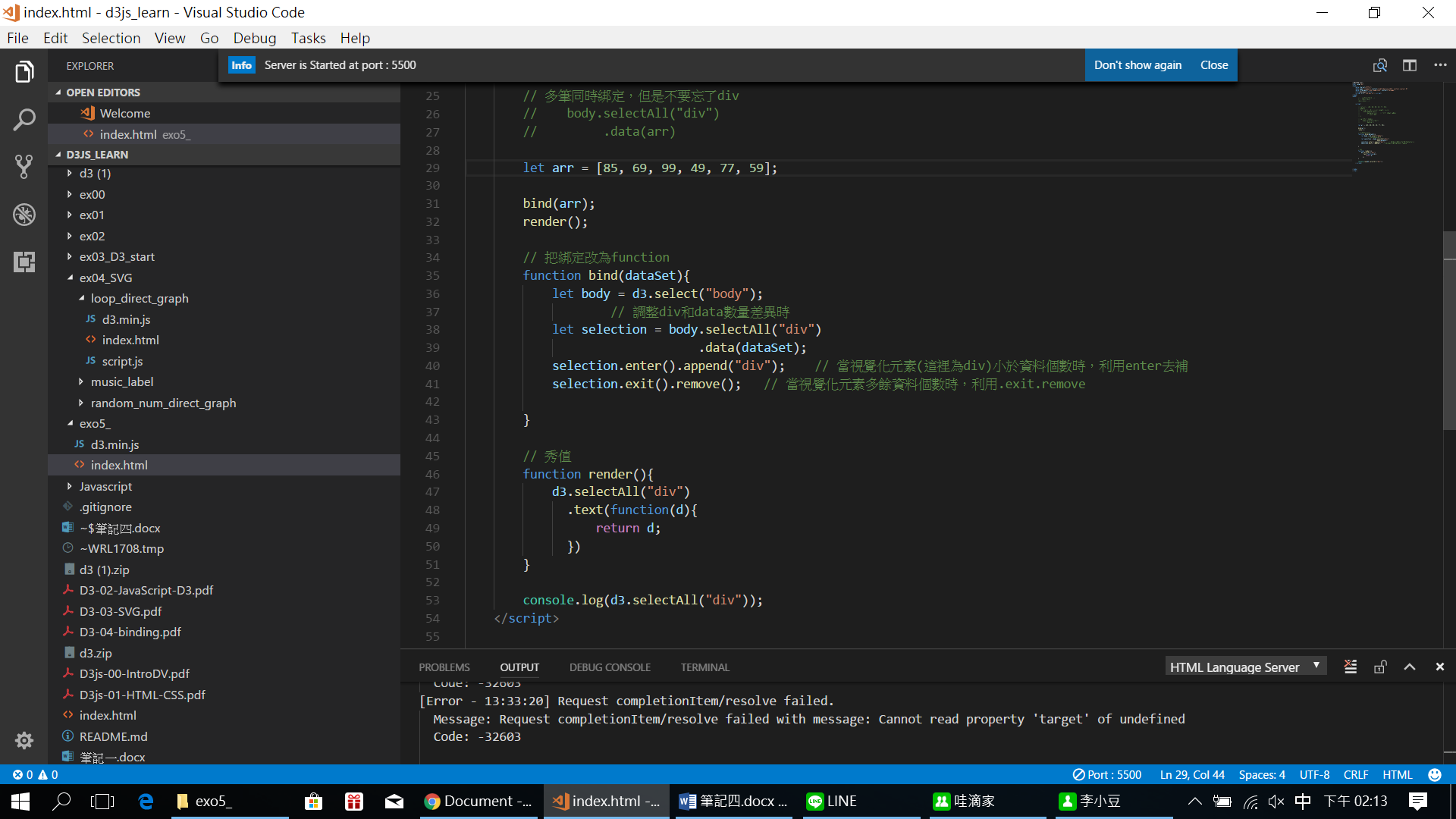




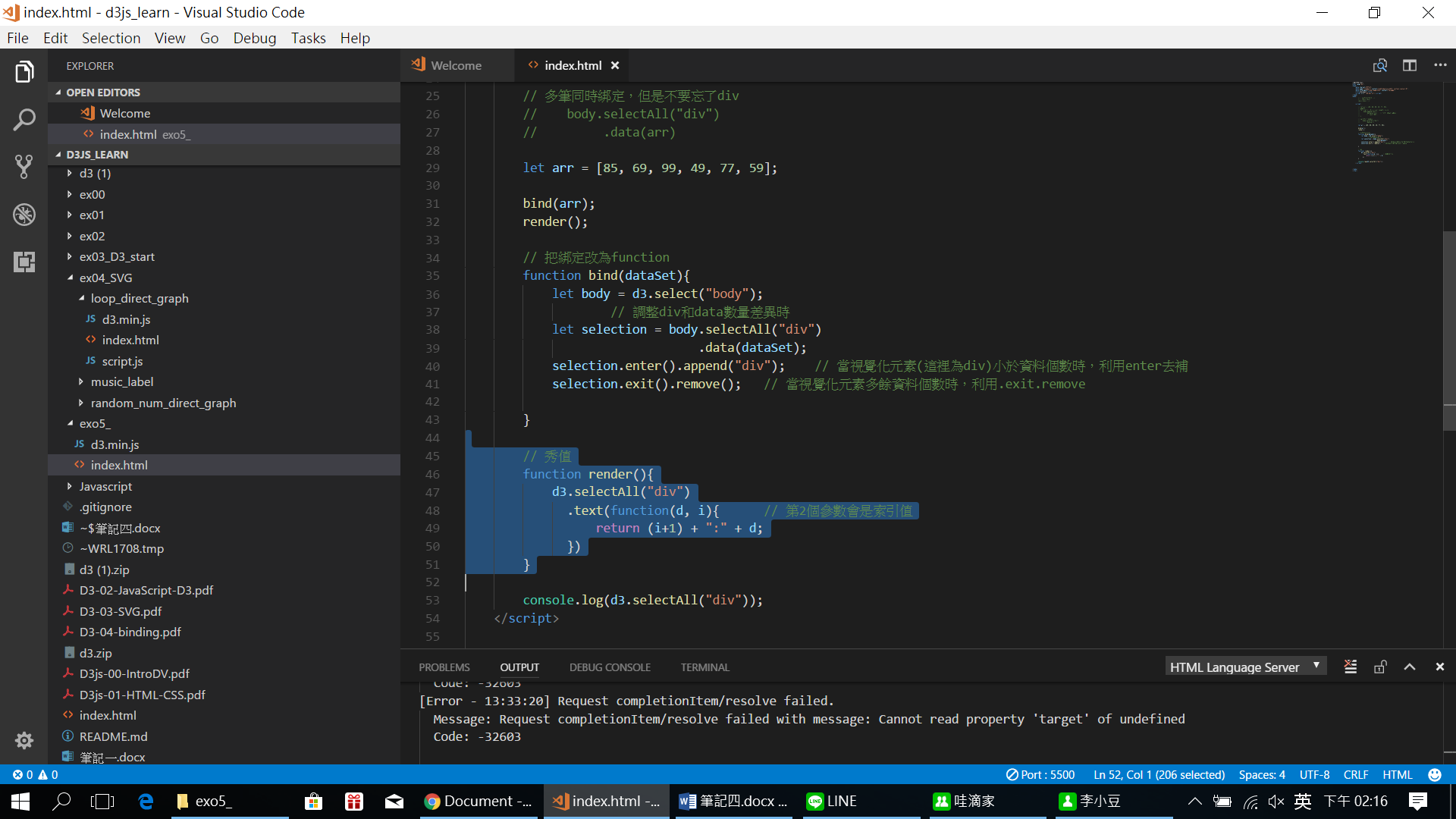
所以改寫為:

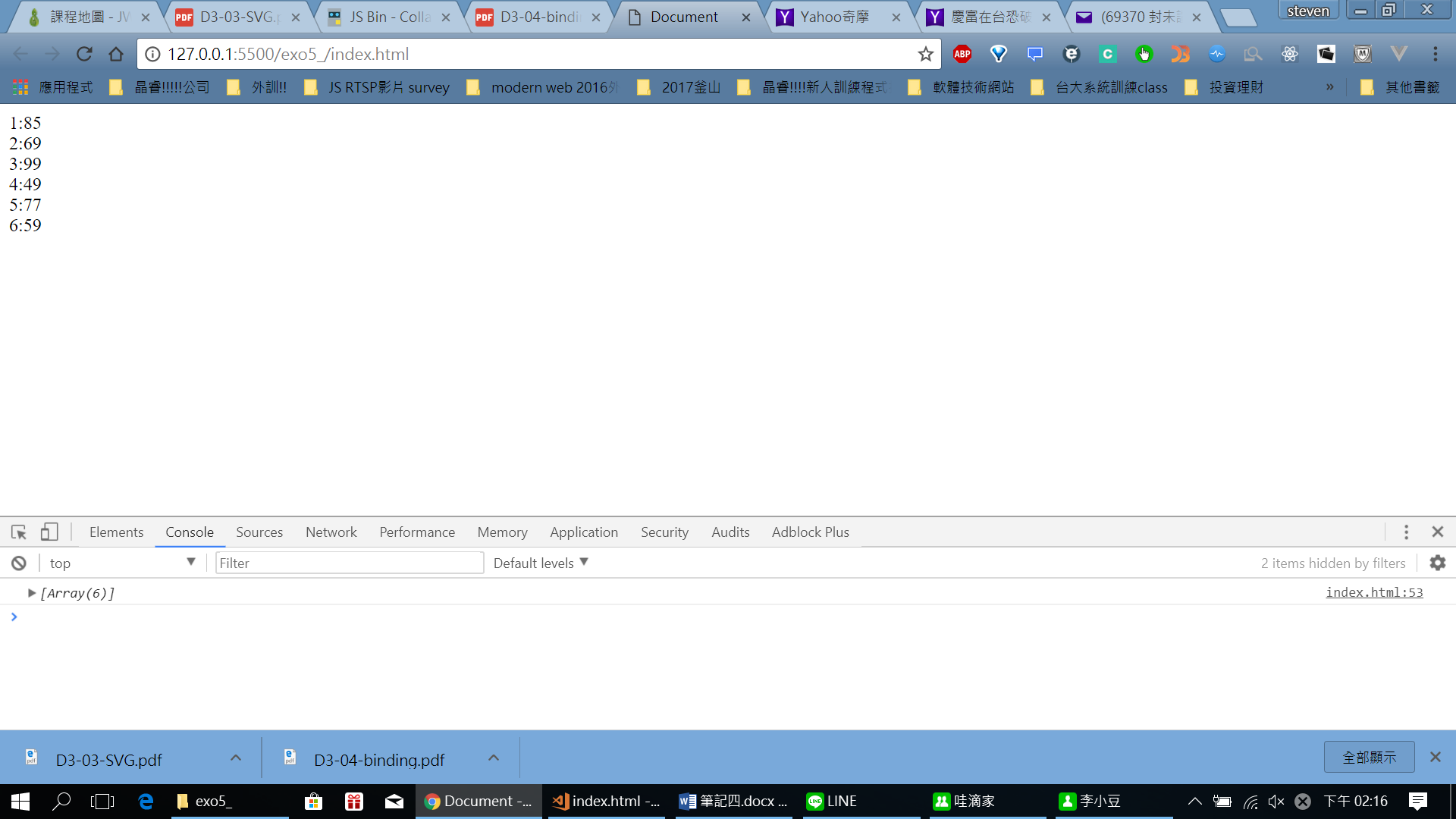


改為function:



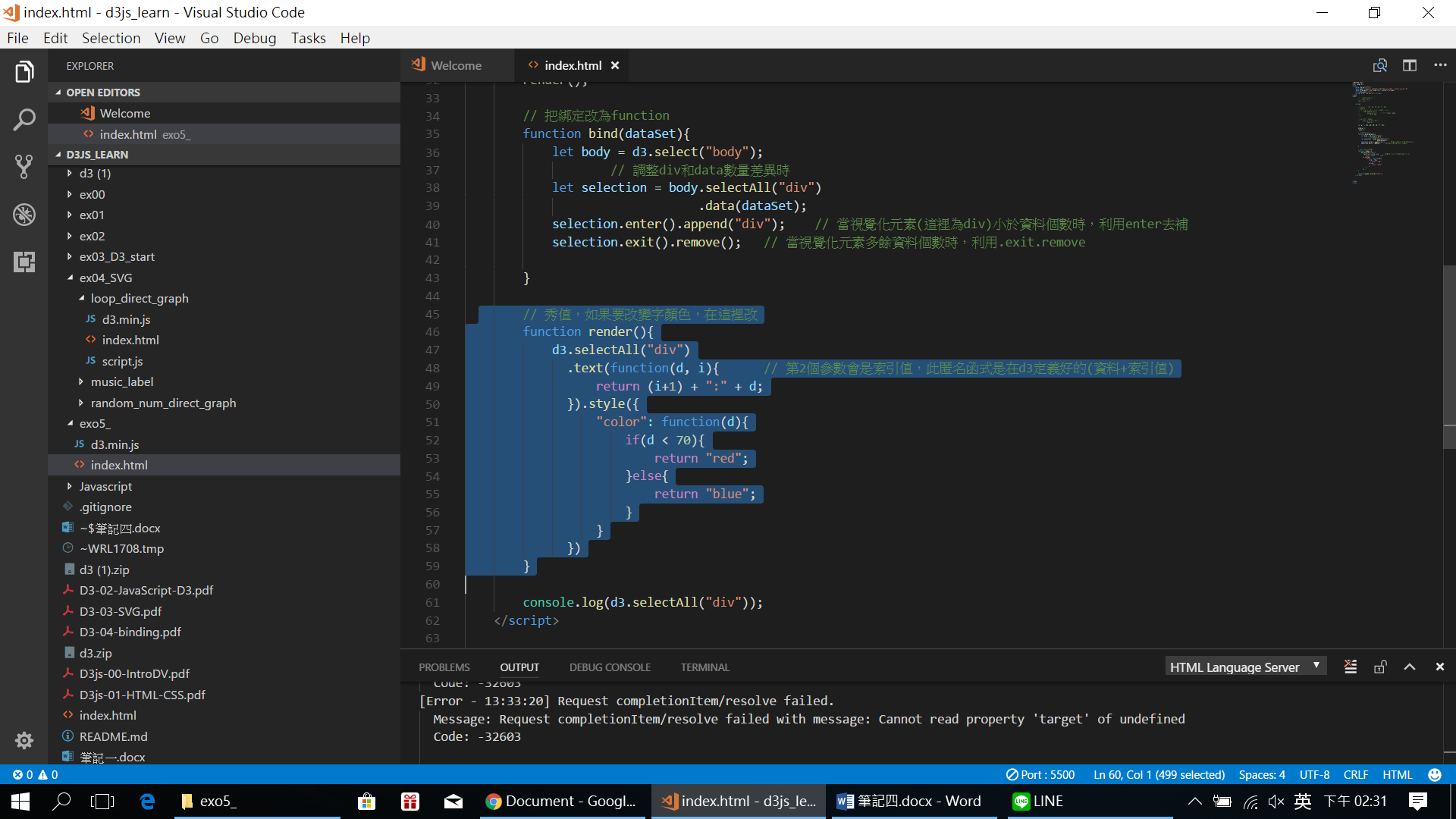
.text 匿名函式加入第二個參數為索引值

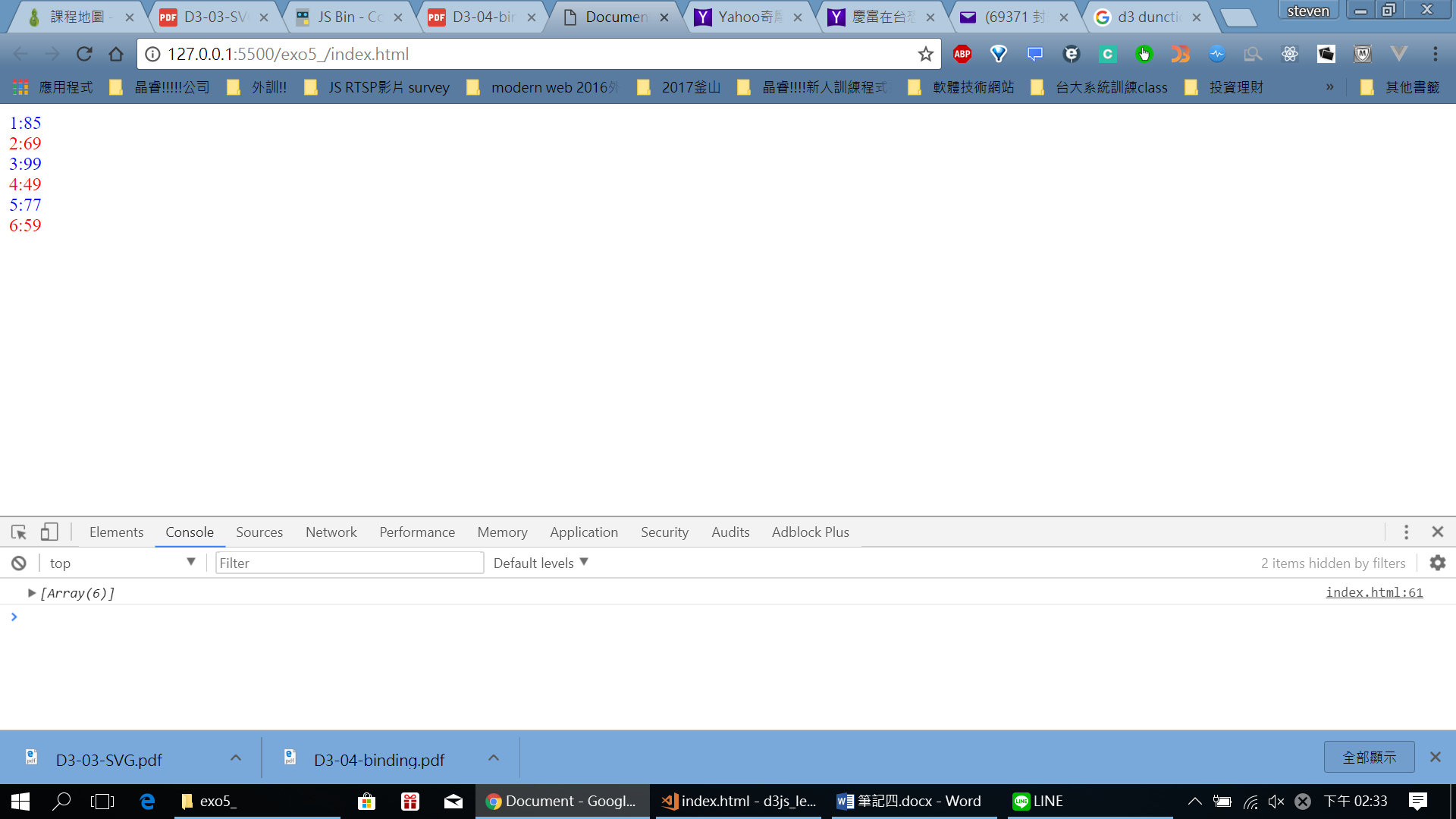




如果要改成顯示在頁面時，小於70的秀紅色:

在.text後再.style({XXXX }) 然後依樣用匿名函式取data做if else





3. 顯示於SVG



每個rect的y比較複雜，因為是從左上定位

所以都是用h(全高)- p(底部間距) -d(資料高)

程式碼:

<!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>Document</title>

<script src="./d3.min.js"></script>

</head>

<body>

<script>

let arr = [85, 69, 99, 49, 77, 59];

let w = 900, h = 600, p = 100;

svg();

bind(arr);

render();

function svg(){

d3.select("body").append("svg")

.attr({

"id": "barChart",

"width": w,

"height": h

})

}

// 把綁定改為function

function bind(dataSet){

let svg = d3.select("#barChart");

// 調整div和data數量差異時

let selection = svg.selectAll("rect")

.data(dataSet);

selection.enter().append("rect"); // 當視覺化元素(這裡為div)小於資料個數時，利用enter去補

selection.exit().remove(); // 當視覺化元素多餘資料個數時，利用.exit.remove

}

// 秀值，如果要改變字顏色，在這裡改

function render(){

d3.selectAll("#barChart rect")

.attr({

"x": function(d, i){

return p + 43 \* i

},

"y": function(d){ // 因為要拿到每個資料的data值，所以要寫個匿名函式

return h - p -d

},

width: 40,

height: function(d){

return d

}

})

}

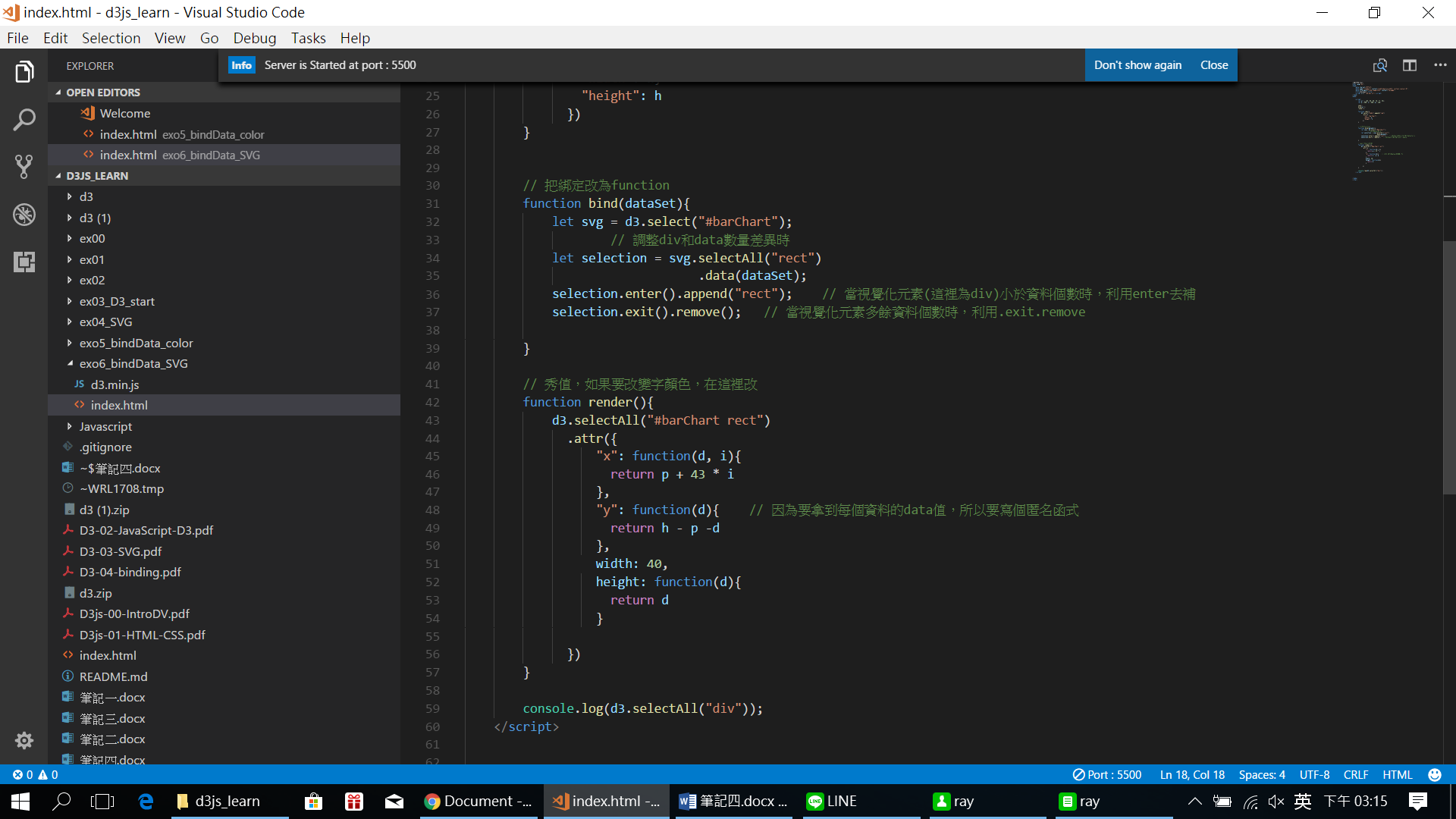
console.log(d3.selectAll("div"));

</script>

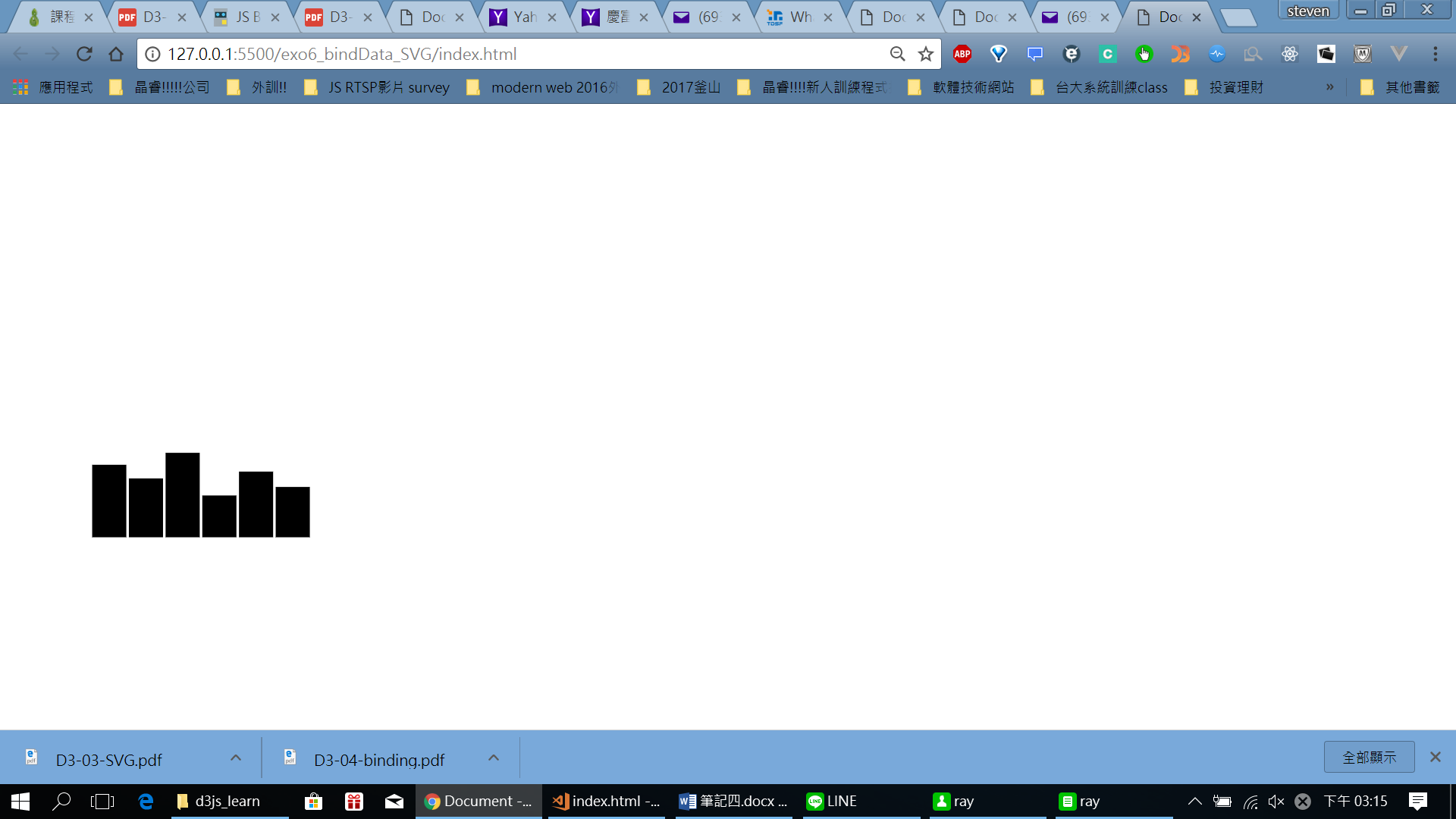
</body>

</html>

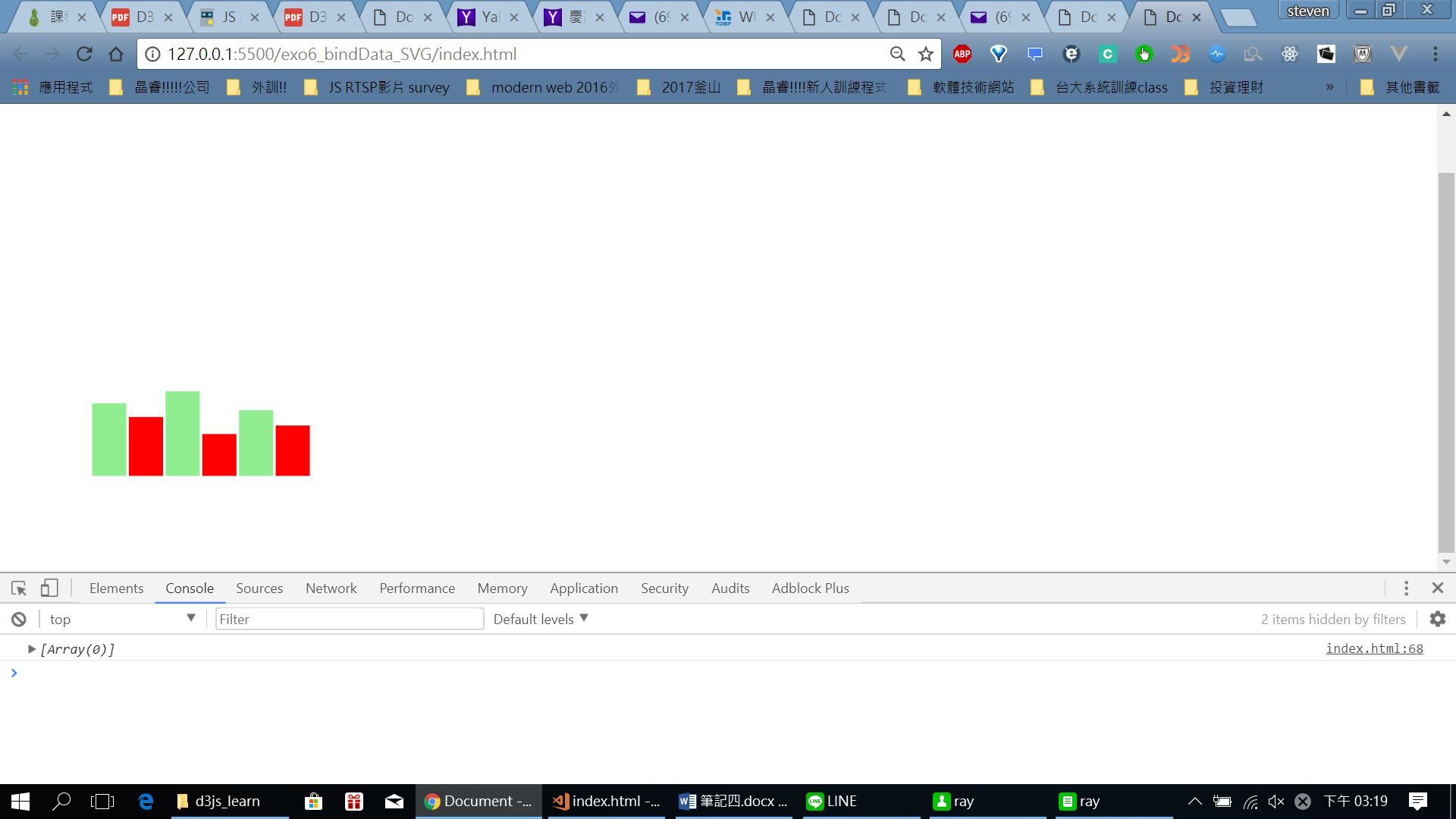
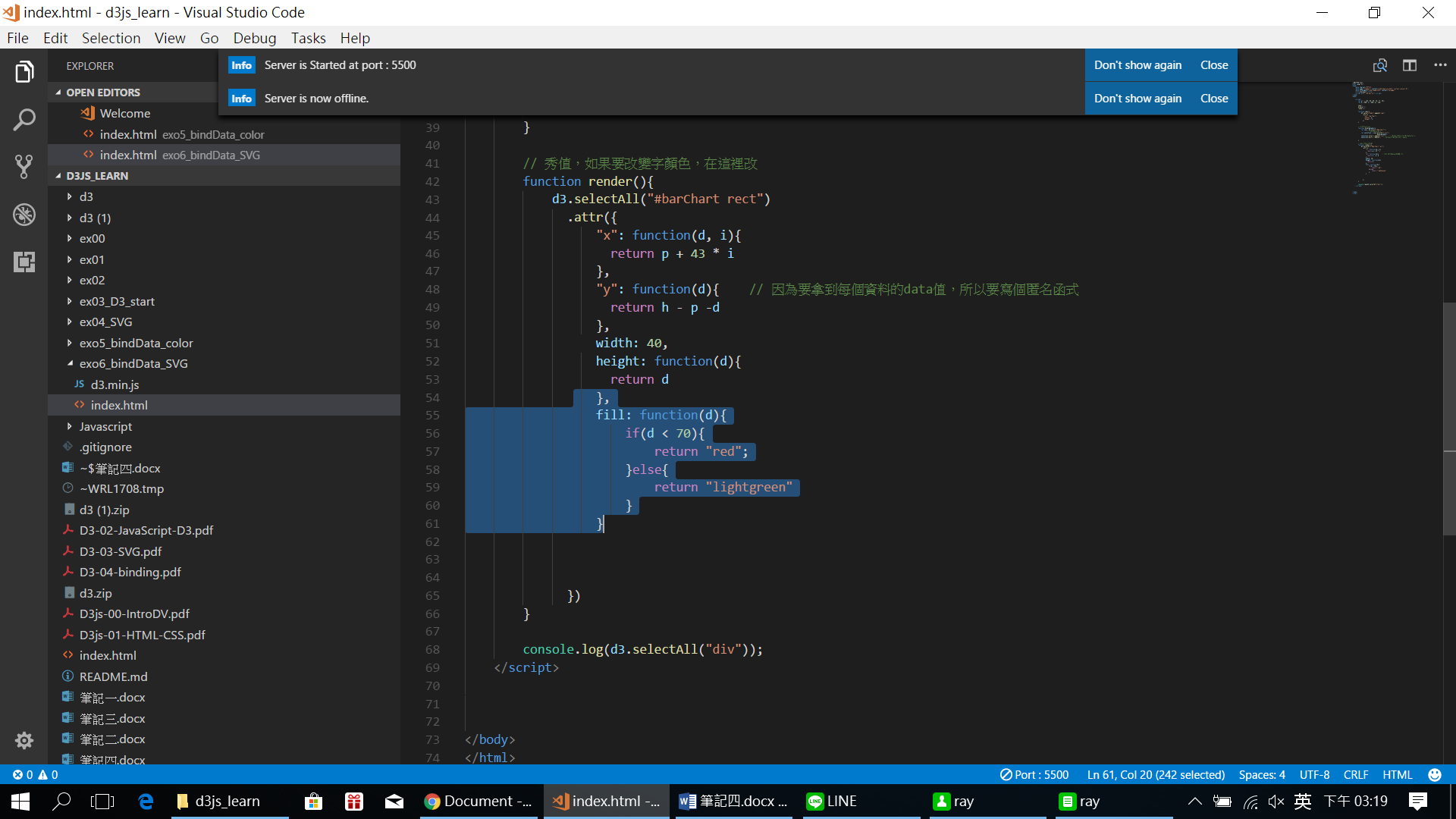
(主要改的就是render:



就可以看到圖形囉:



加上< 70的塗上紅色:



新增文字:

<!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>Document</title>

<script src="./d3.min.js"></script>

</head>

<body>

<script>

let arr = [85, 69, 99, 49, 77, 59];

let w = 900, h = 600, p = 100;

svg();

bind(arr);

render();

function svg(){

d3.select("body").append("svg")

.attr({

"id": "barChart",

"width": w,

"height": h

})

}

// 把綁定改為function

function bind(dataSet){

// 綁定矩形

let svg = d3.select("#barChart");

// 調整div和data數量差異時

let selection = svg.selectAll("rect")

.data(dataSet);

selection.enter().append("rect"); // 當視覺化元素(這裡為div)小於資料個數時，利用enter去補

selection.exit().remove(); // 當視覺化元素多餘資料個數時，利用.exit.remove

// 綁定文字

let selectionText = svg.selectAll("text")

.data(dataSet);

selectionText.enter().append("text");

selectionText.exit().remove();

}

// 秀值，如果要改變字顏色，在這裡改

function render(){

// 劃出矩形

d3.selectAll("#barChart rect")

.attr({

"x": function(d, i){

return p + 43 \* i

},

"y": function(d){ // 因為要拿到每個資料的data值，所以要寫個匿名函式

return h - p -d

},

width: 40,

height: function(d){

return d

},

fill: function(d){

if(d < 70){

return "red";

}else{

return "lightgreen"

}

}

})

// 寫文字

d3.selectAll("#barChart text")

.attr({

"x": function(d, i){

return p + 43 \* i + 20; // 把文字在平移寬的一半

},

"y": function(d){ // 因為要拿到每個資料的data值，所以要寫個匿名函式

return h - p + 20;

},

"text-anchor": "middle"

}).text(function(d){

return d;

})

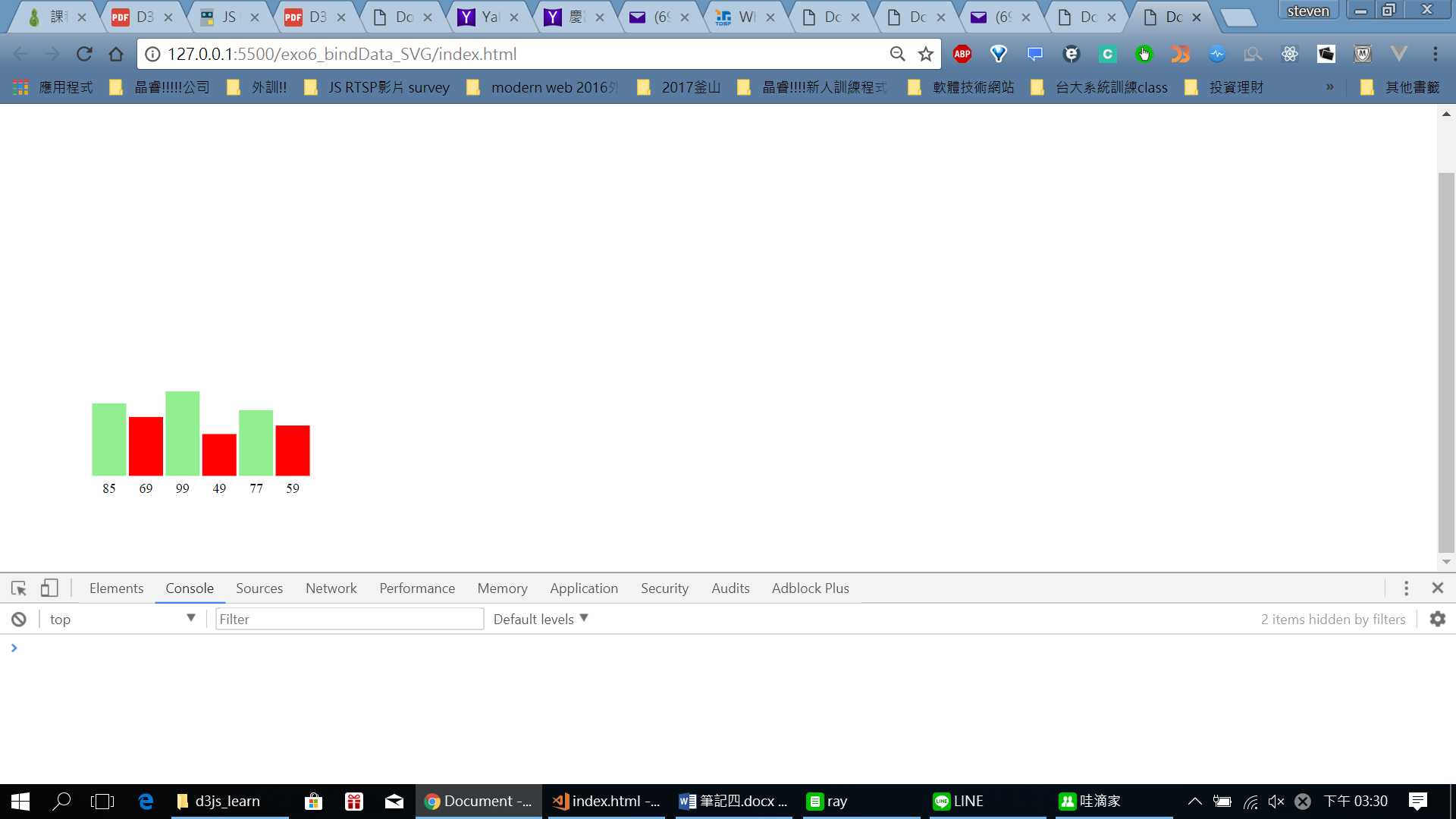
}

// console.log(d3.selectAll("div"));

</script>

</body>

</html>



接下來要寫互動式!

<!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>Document</title>

<script src="./d3.min.js"></script>

</head>

<body>

<input type="button" value="新增" onclick="update()"> <!-- 新增亂數到Arr -->

<input type="button" value="移除" onclick="kickItem()">

<script>

let arr = [85, 69, 99, 49, 77, 59];

let w = 900, h = 600, p = 100;

svg();

bind(arr);

render();

function svg(){

d3.select("body").append("svg")

.attr({

"id": "barChart",

"width": w,

"height": h

})

}

// 把綁定改為function

function bind(dataSet){

// 綁定矩形

let svg = d3.select("#barChart");

// 調整div和data數量差異時

let selection = svg.selectAll("rect")

.data(dataSet);

selection.enter().append("rect"); // 當視覺化元素(這裡為div)小於資料個數時，利用enter去補

selection.exit().remove(); // 當視覺化元素多餘資料個數時，利用.exit.remove

// 綁定文字

let selectionText = svg.selectAll("text")

.data(dataSet);

selectionText.enter().append("text");

selectionText.exit().remove();

}

// 秀值，如果要改變字顏色，在這裡改

function render(){

// 劃出矩形

d3.selectAll("#barChart rect")

.attr({

"x": function(d, i){

return p + 43 \* i

},

"y": function(d){ // 因為要拿到每個資料的data值，所以要寫個匿名函式

return h - p -d

},

width: 40,

height: function(d){

return d

},

fill: function(d){

if(d < 70){

return "red";

}else{

return "lightgreen"

}

}

})

// 寫文字

d3.selectAll("#barChart text")

.attr({

"x": function(d, i){

return p + 43 \* i + 20; // 把文字在平移寬的一半

},

"y": function(d){ // 因為要拿到每個資料的data值，所以要寫個匿名函式

return h - p + 20;

},

"text-anchor": "middle"

}).text(function(d){

return d;

})

}

function update(){

let num = random(10, 100);

arr.push(num); // python是append // 這裡將Array更新

bind(arr); // 變更後需要重新綁定!

render(); // 重新畫

}

function random(min, max){

return Math.floor(Math.random() \* (max - min + 1) + min);

}

// console.log(d3.selectAll("div"));

</script>

</body>

</html>

Javascript移除array前面用shift()

function kickItem(){

arr.shift(); // python是append // 這裡將Array更新

bind(arr); // 變更後需要重新綁定!

render(); // 重新畫

}

就可以動態新增移除

