Var:

<script>

if(true){

var x = 3;

}

alert(x);

</script>

可以show

但

<script>

if(true){

let x = 3;

}

alert(x);

</script>

卻出不來!!!!!!!!

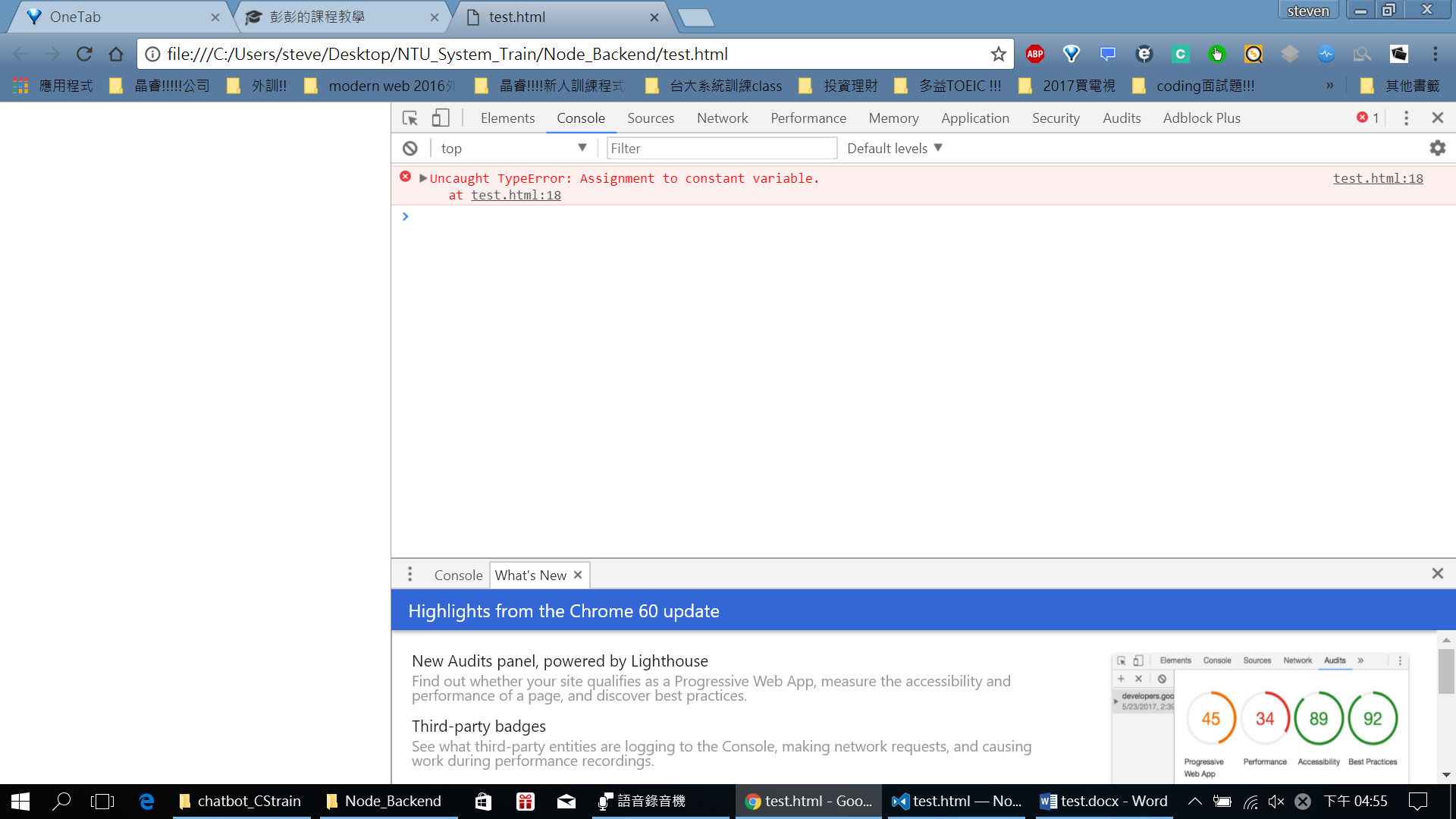
是因為scope: var就算再判斷是裡面，沒有用函示區別都算同一個範圍

若用新的let: 變數範圍除了函式，所有的大括號都在範圍內!!!

所以小心2者的差別!!!!

1. Const
2. <script>
3. const x = 3;
4. x=5;
5. </script>

會噴錯: 因為const後不能更動資料:



須給初始值

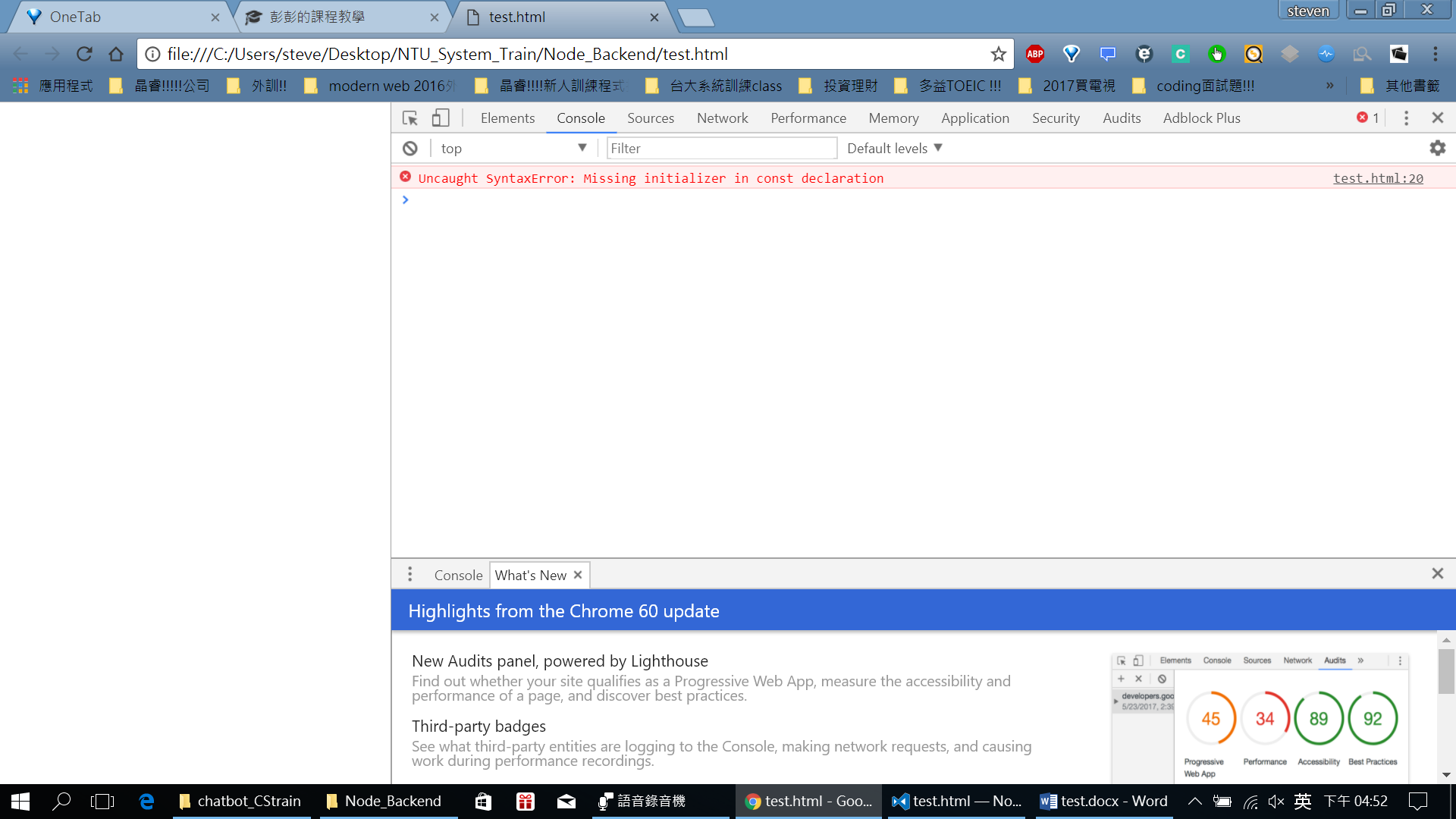
<script>

const x = 3;

alert(x);

const y;

</script>



3. 箭頭函式:

<script>

// 傳統:

function t1(){

alert("Hello")

};

let t2 = function(){ // 右邊是匿名函式

alert("Hello")

};

// 箭頭函式:

let t3=(參數列表)=>{

函式的本體

};

</script>

Ex:

// 箭頭函式:

let t3=()=>{ // 如此即沒有參數

alert("Hello3")

};

t3();

以下就會alert: 12

// 箭頭函式:

let t4=(n1, n2)=>{ // 如此即沒有參數

// alert(n1\*n2)

return n1\*n2

};

let result = t4(3, 4);

alert(result)

另一種寫法:

let t5=(n1, n2)=>(n1\*n2)

let result2 = t5(3,4);

alert(result2)

參數給預設值(不一定要箭頭函式，舊的函式也可以用):

以下噴6

<!-- TEST5 參數的預設值-->

<script>

let t5=(n1, n2=2)=>(n1\*n2)

let result2 = t5(3);

alert(result2) //噴6

</script>

Ex舊函式:

let t2 = function(mes="Hello"){ // 右邊是匿名函式

alert(mes)

};

t2(); //噴"Hello"

t2("XD"); // 噴"XD""