CSS

圖片變形的話: object-fit: cover

background-position:

background-size: cover

**flex:**

**.flex\_container** 父元素{

*/\* flex 排版 \*/*

**display**: flex | inline-flex;

*/\* 水平對齊 \*/*

**justify-content**: flex-start | flex-end | center | space-between | space-around;

*/\* 垂直對齊 （ 單一行 ） \*/*

**align-items**: flex-start | flex-end | center | baseline | stretch;

*/\* 排列方向 \*/*

**flex-direction**: row | row-reverse | column | column-reverse;

*/\* 換行方式 \*/*

**flex-wrap**: nowrap | wrap | wrap-reverse;

*/\* 結合 排列方向 與 換行方式 \*/*

**flex-flow**: <flex-direction> <flex-wrap>;

*/\* 多行的垂直對齊 \*/*

**align-content**: flex-start | flex-end | center | space-between | space-around | stretch;

**.flex\_item** {子元素

*/\* 該元素的排列位置 \*/*

**order**: <number>;

*/\* 該元素佔用容器剩餘的比例 \*/*

**flex-grow**: <number>;

*/\* 該元素壓縮比例 \*/*

**flex-shrink**: <number>;

*/\* 該元素的最小值 \*/*

**flex-basis**: <length> | auto;

*/\* 結合上面 3 種屬性 \*/*

**flex**: <'flex-grow'> <'flex-shrink'> <'flex-basis'>

**margin**: auto 隔開其他東西

}

JavaScript

Number()

isNaN() //Not A Number

parseInt()

parseFloat()

toString(無,2,8,16) //以10進制(默認)或2,8,16進制

1

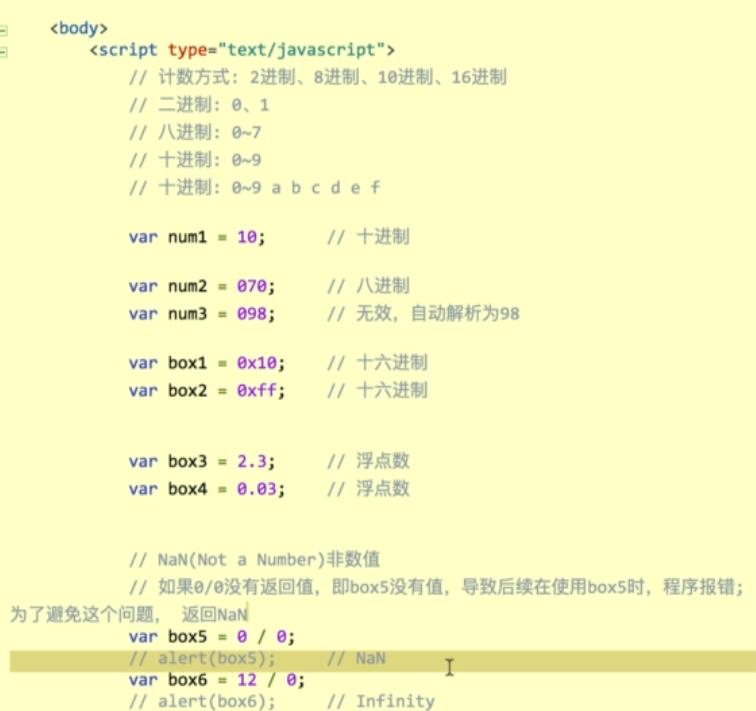
++i與i++的區別為賦值先後的差別

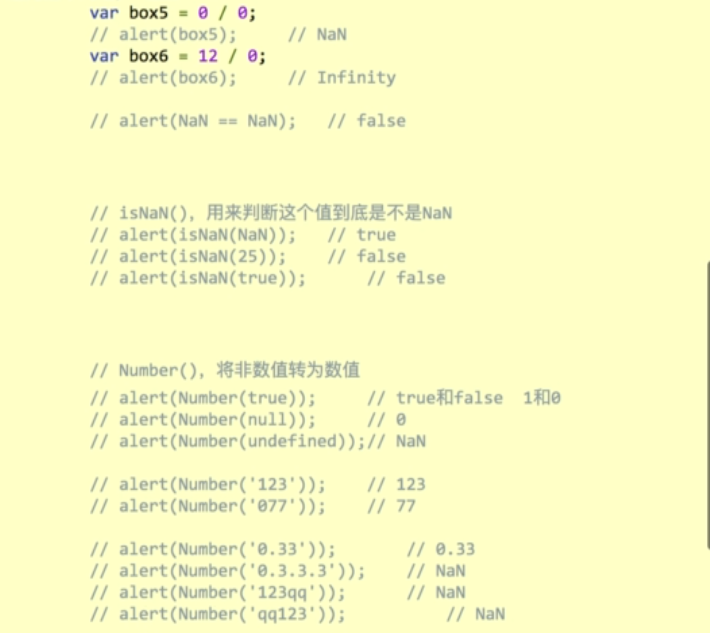
++i為先+1 , 後賦值

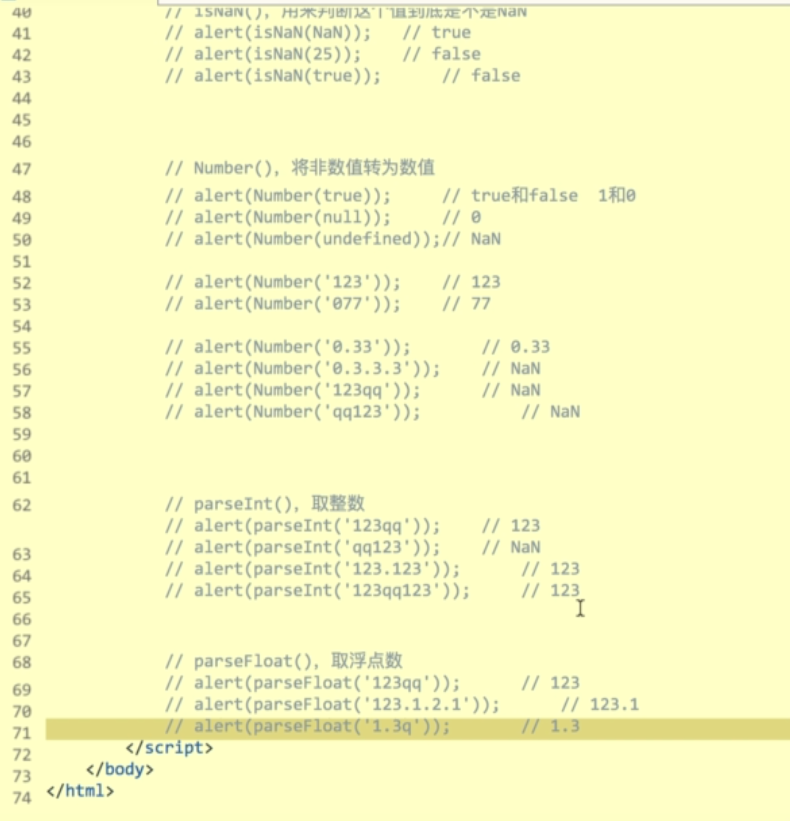
I++為先賦值 , 後+1

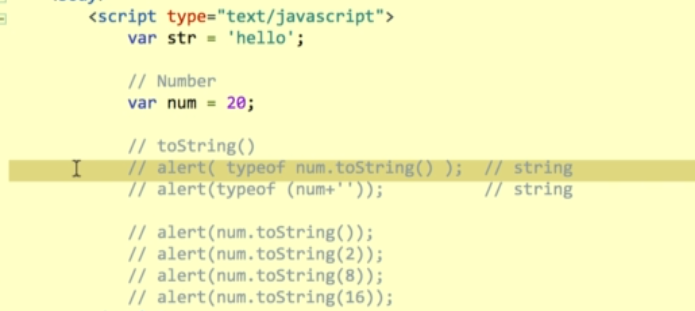
do-while尋歡不管怎樣都會先執行一次, 而while循環在符合條件下才會執行

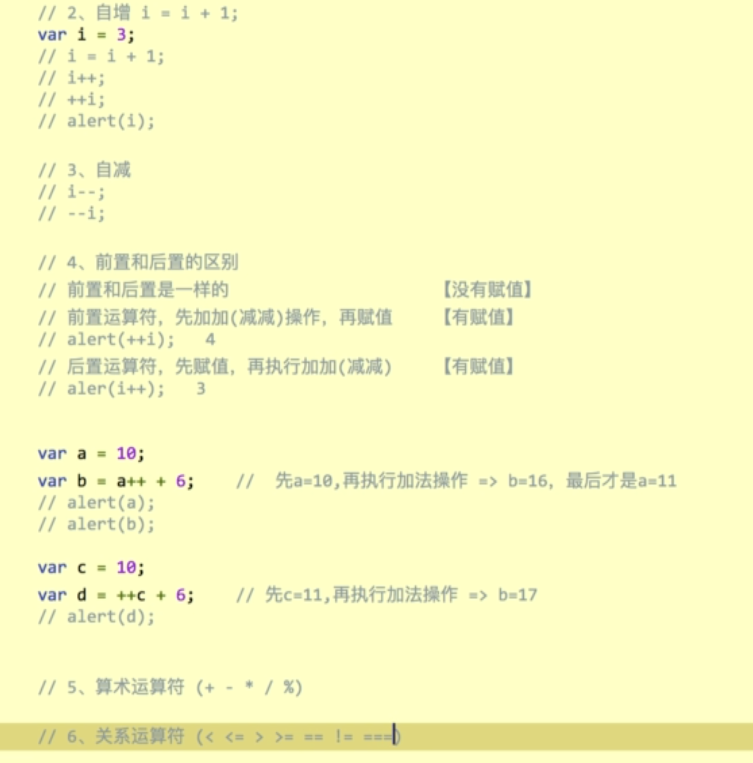
while循環適合次數不固定的操作, for循環適合次數固定的操作

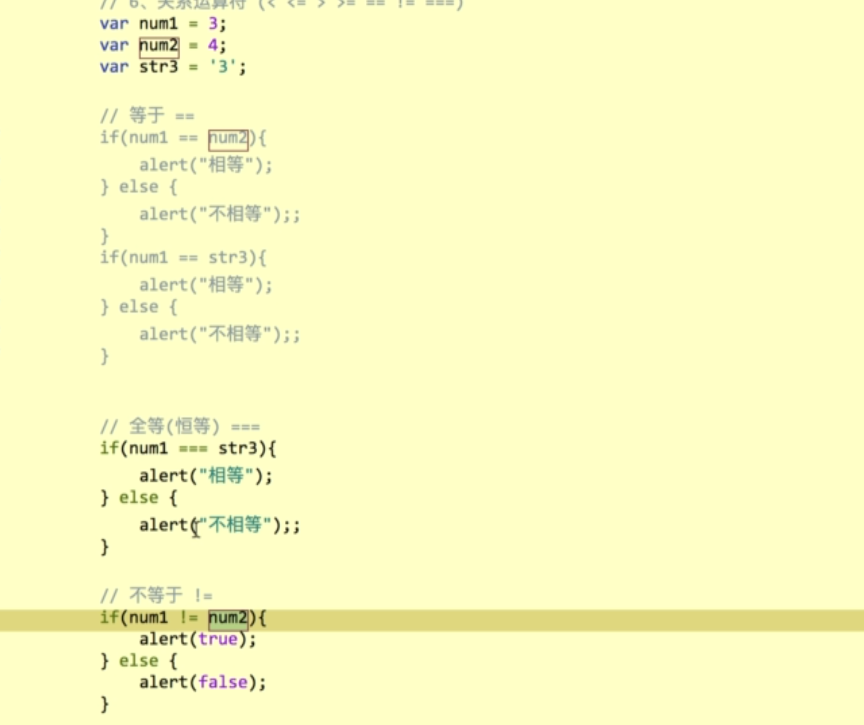


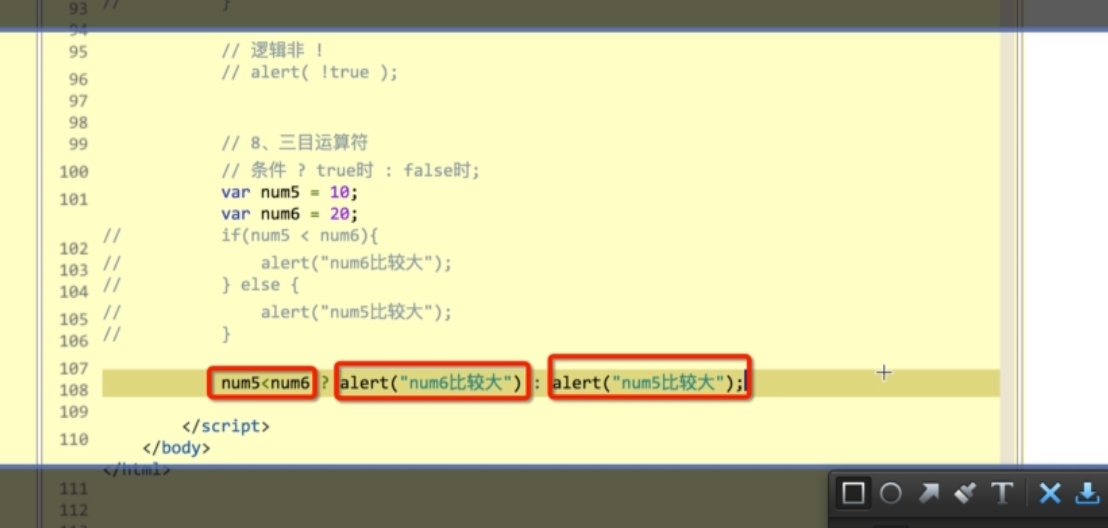


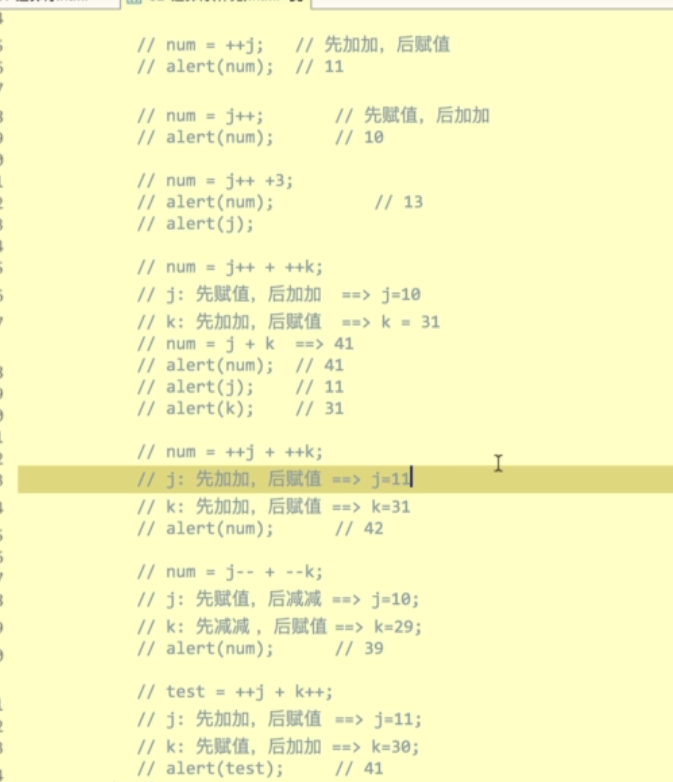






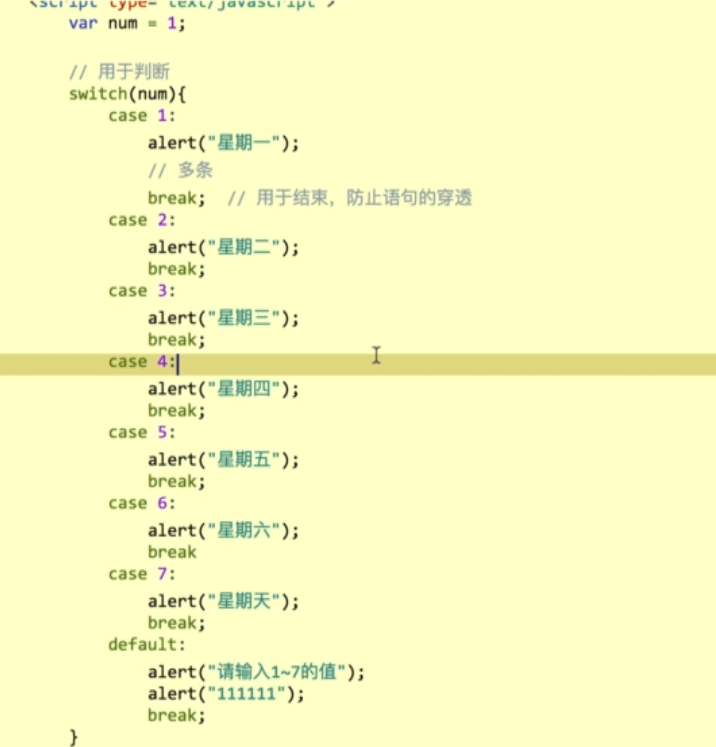


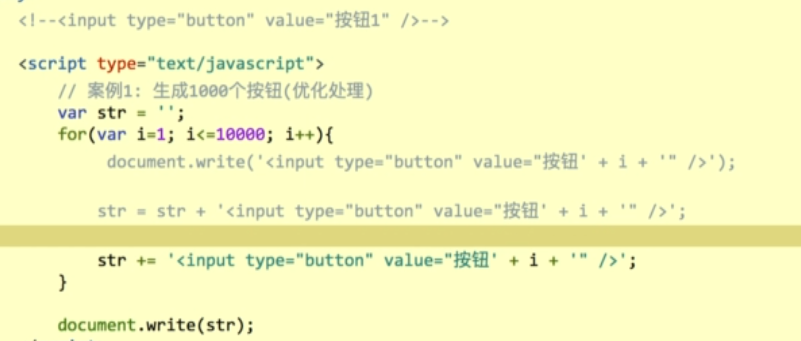


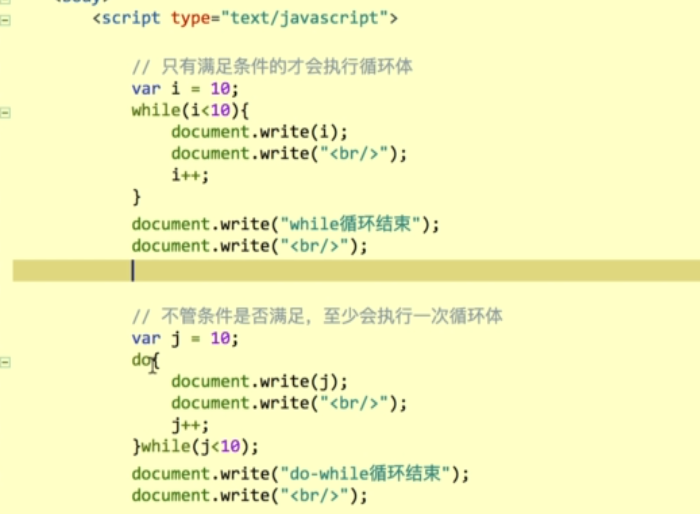


j=10 k=30





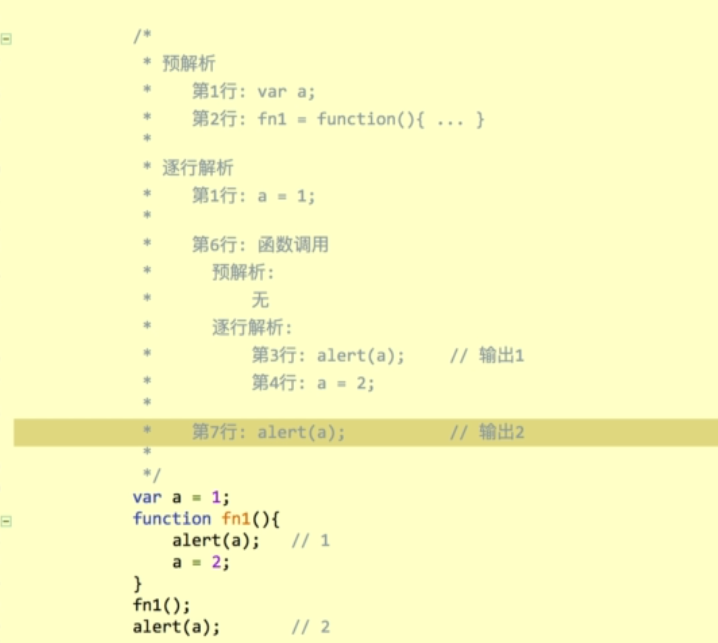
優化for循環

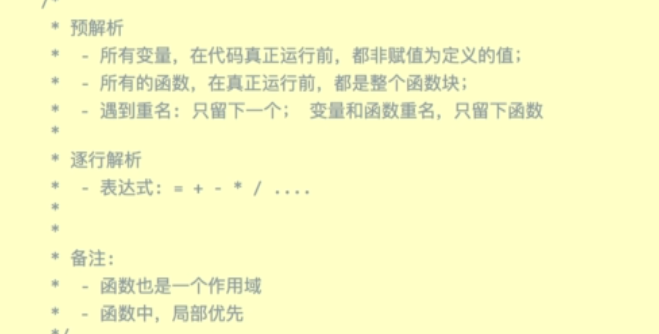


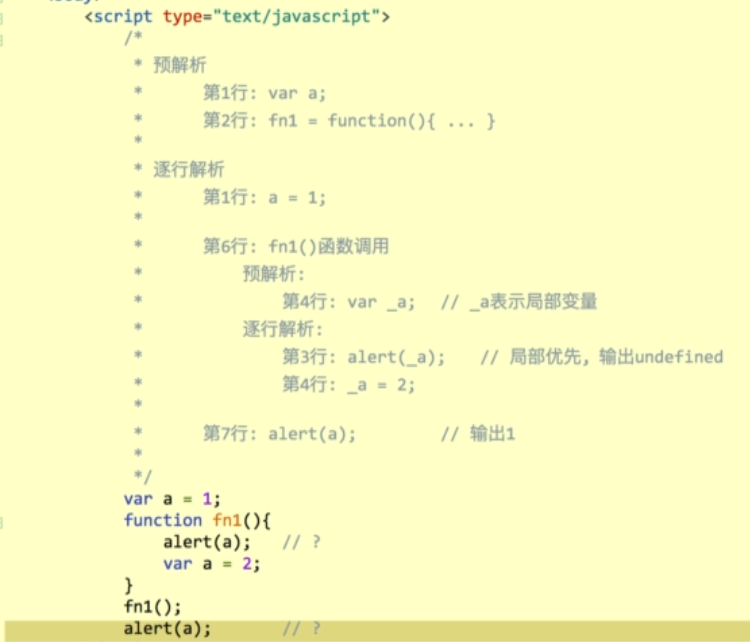




形參 實參(函數)





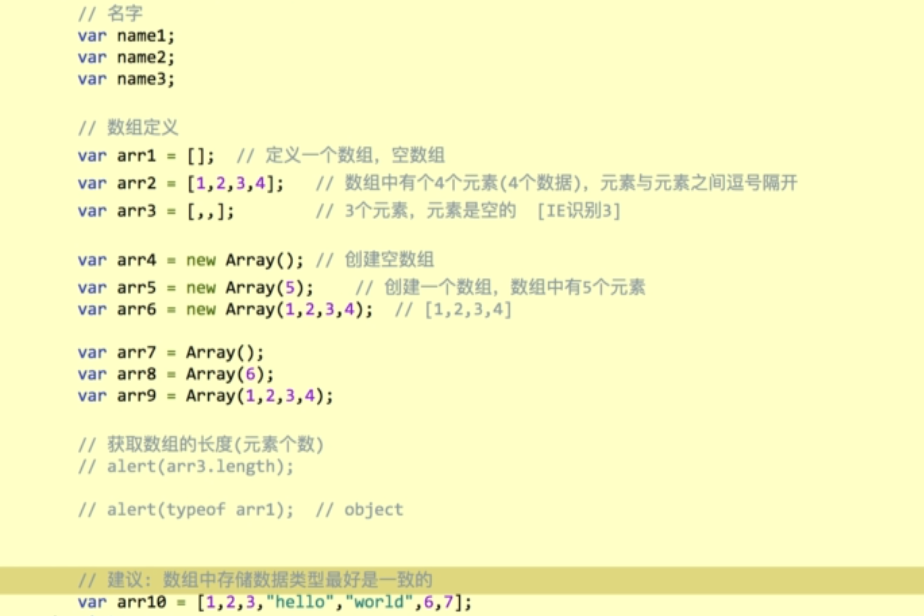
\*

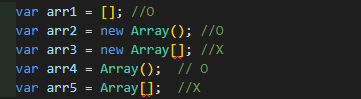


\*

\*\*\*

預解析 逐行解析 (JS解析的運作)





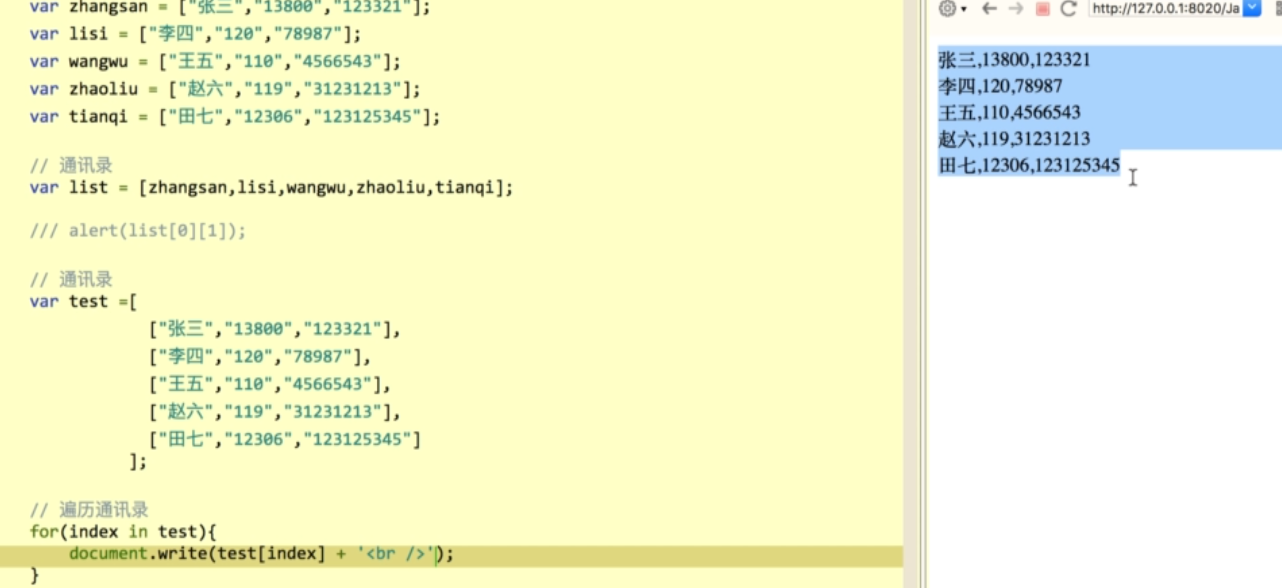




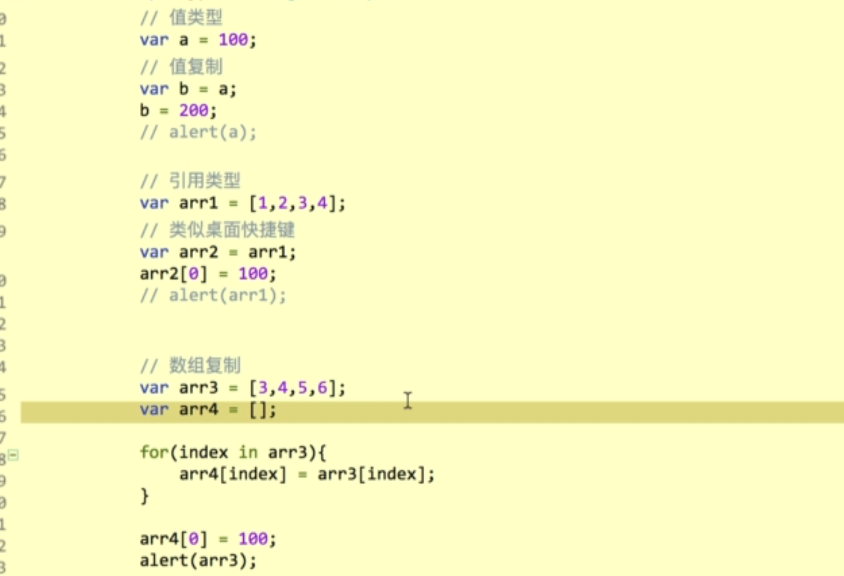
\*\*\*



\*



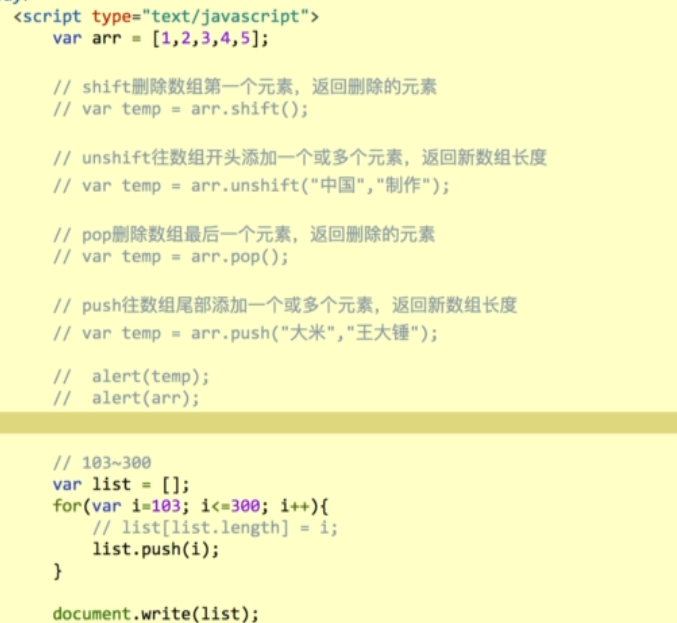
\*



Arr3 = [3,4,5,6], Arr4 = Arr3 => Arr4與Arr3都指向同一位置

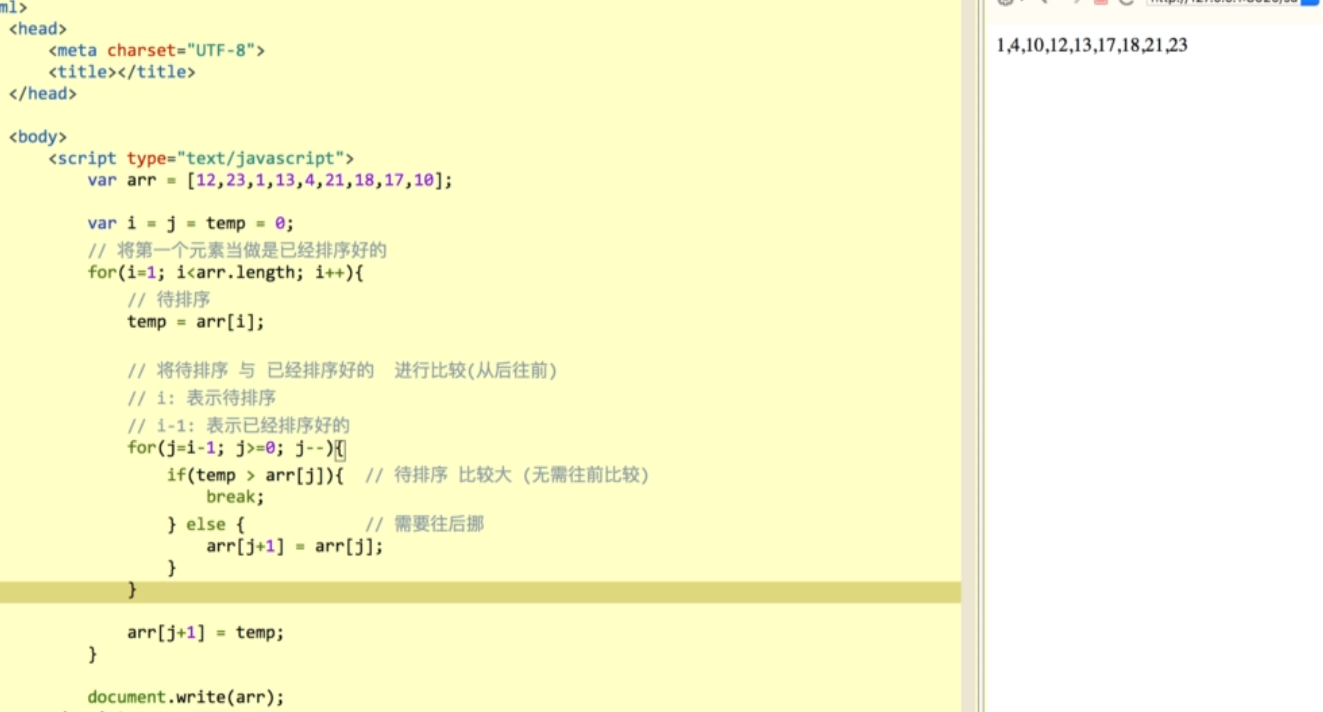
(改變Arr4會改變Arr3)

\*\*

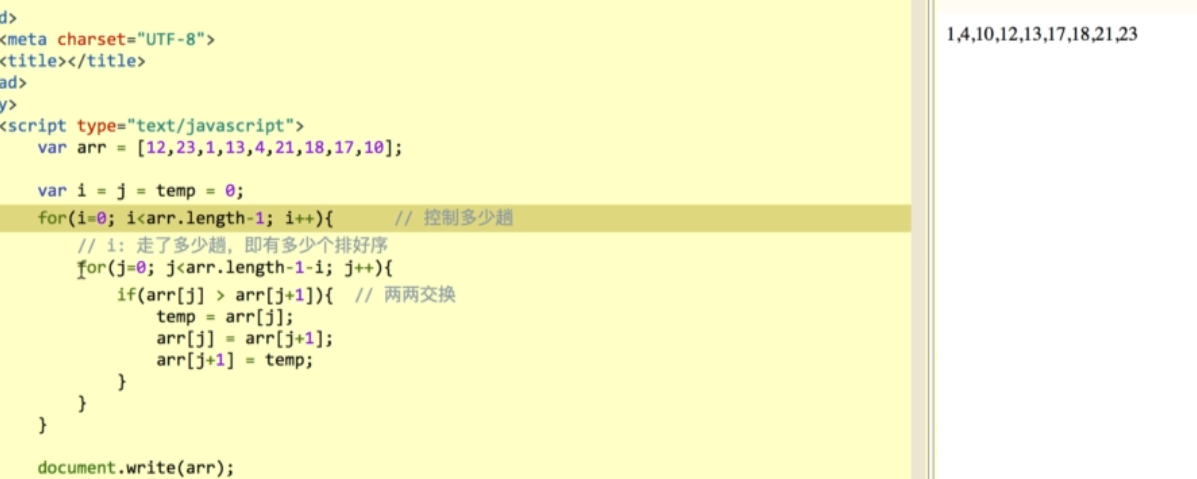


shift,unshift,pop,push

數組



插入排序法



冒泡排序法

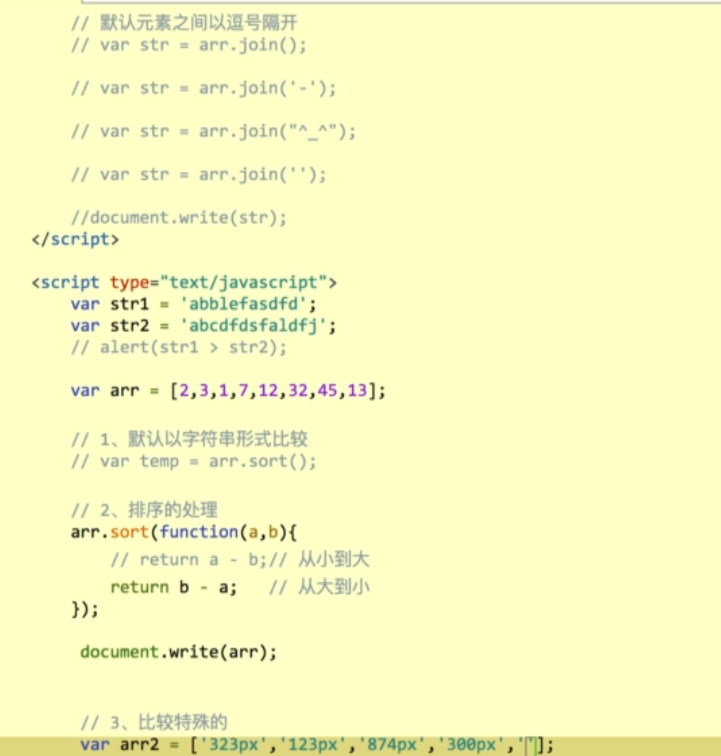
\*\*\*\*\*

排序



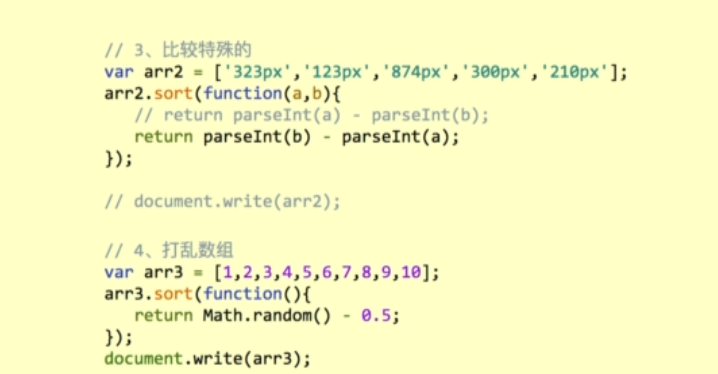
關於this的指向

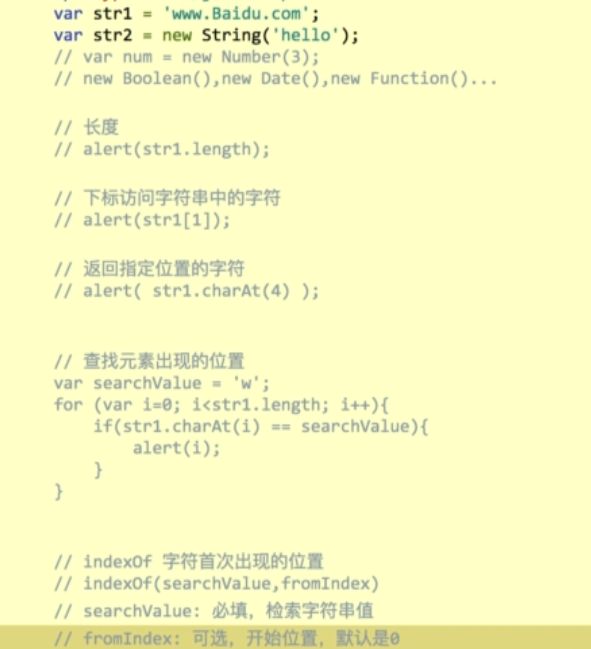
數組

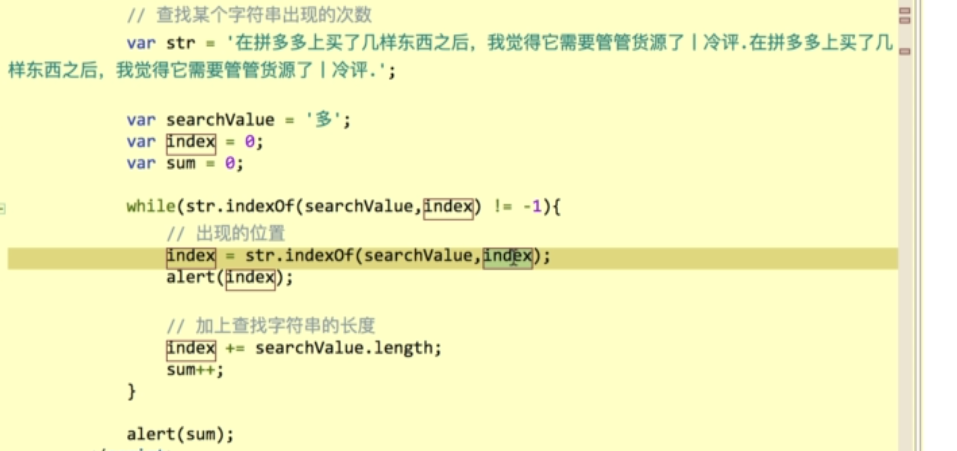
arr.join(), arr.sort() 指定arr各個元素隔開的字串, 排大小

Join:數組轉為字符串

Split:字符串轉為數組

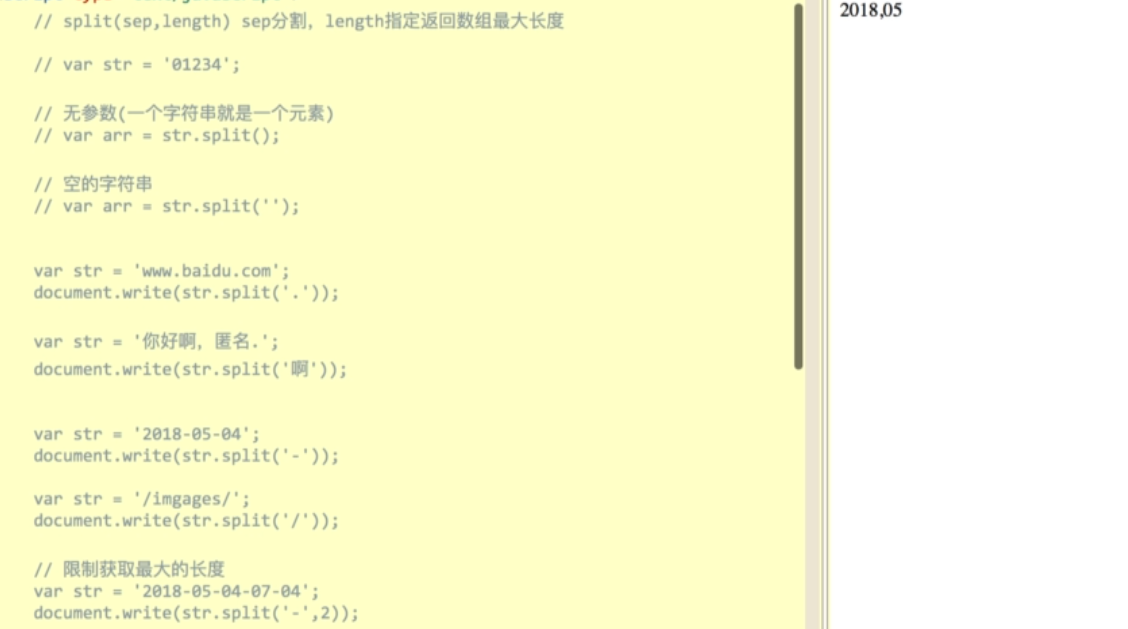
特殊的數組辨別, 打亂數組

查找字符串

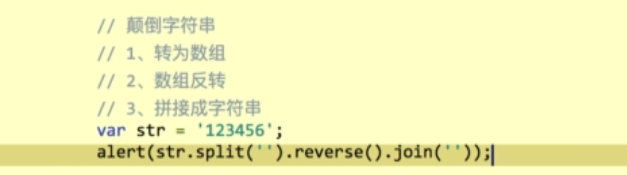


2, 3, 31, 32

4(sum)



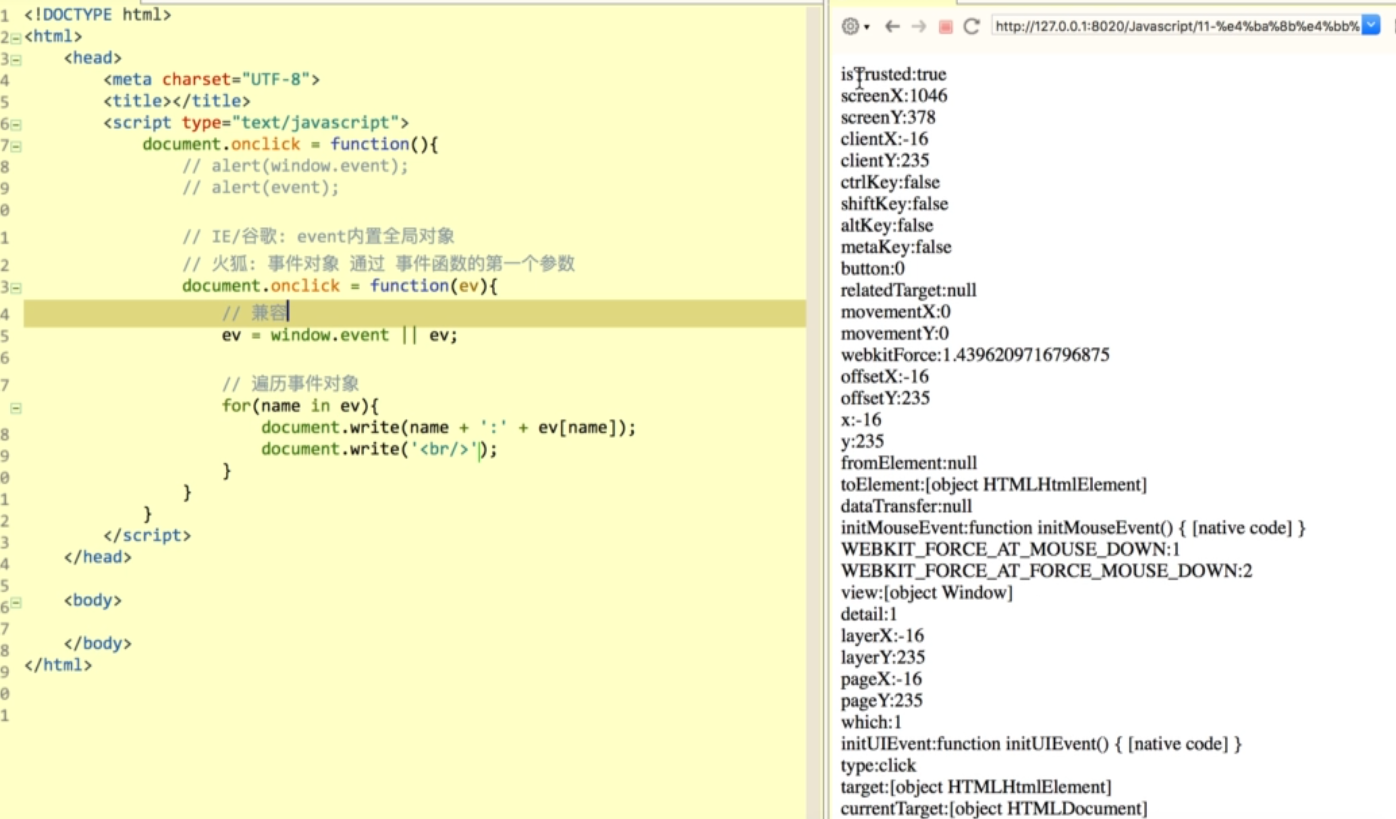
Split 切割arr



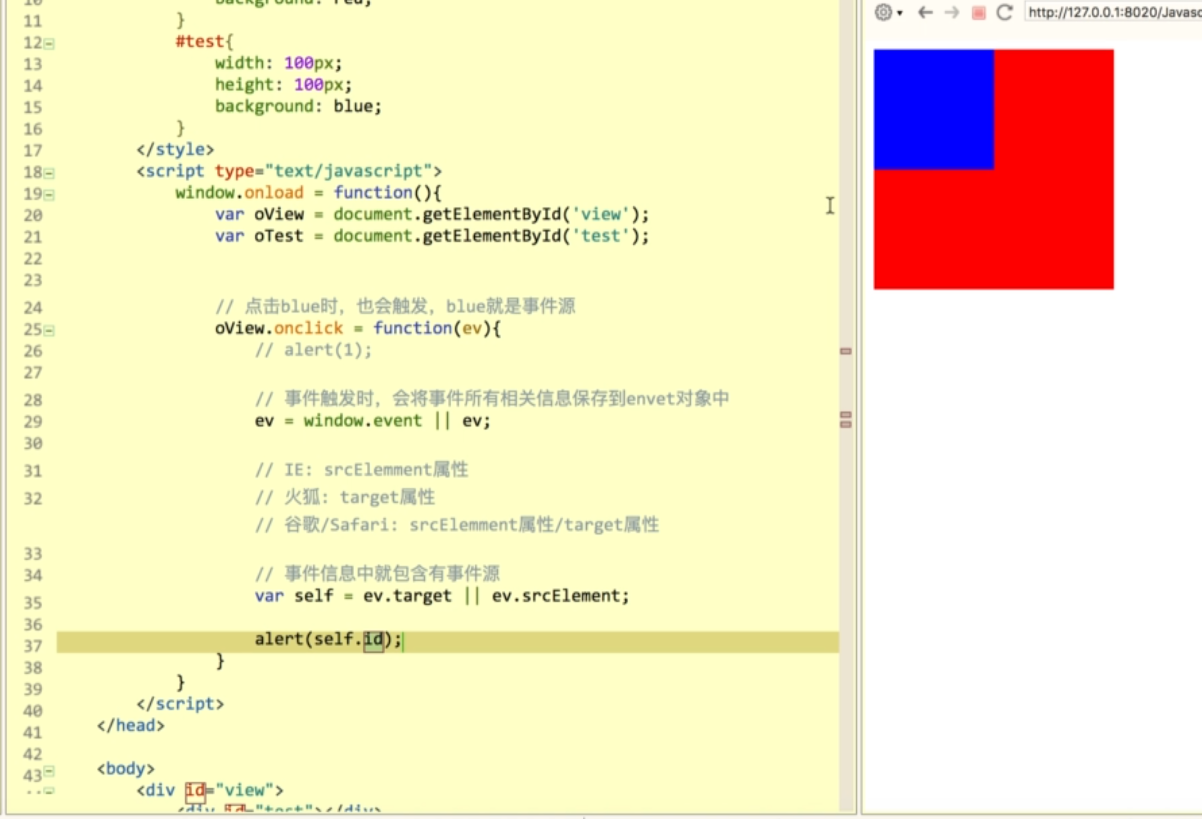
反轉字符串



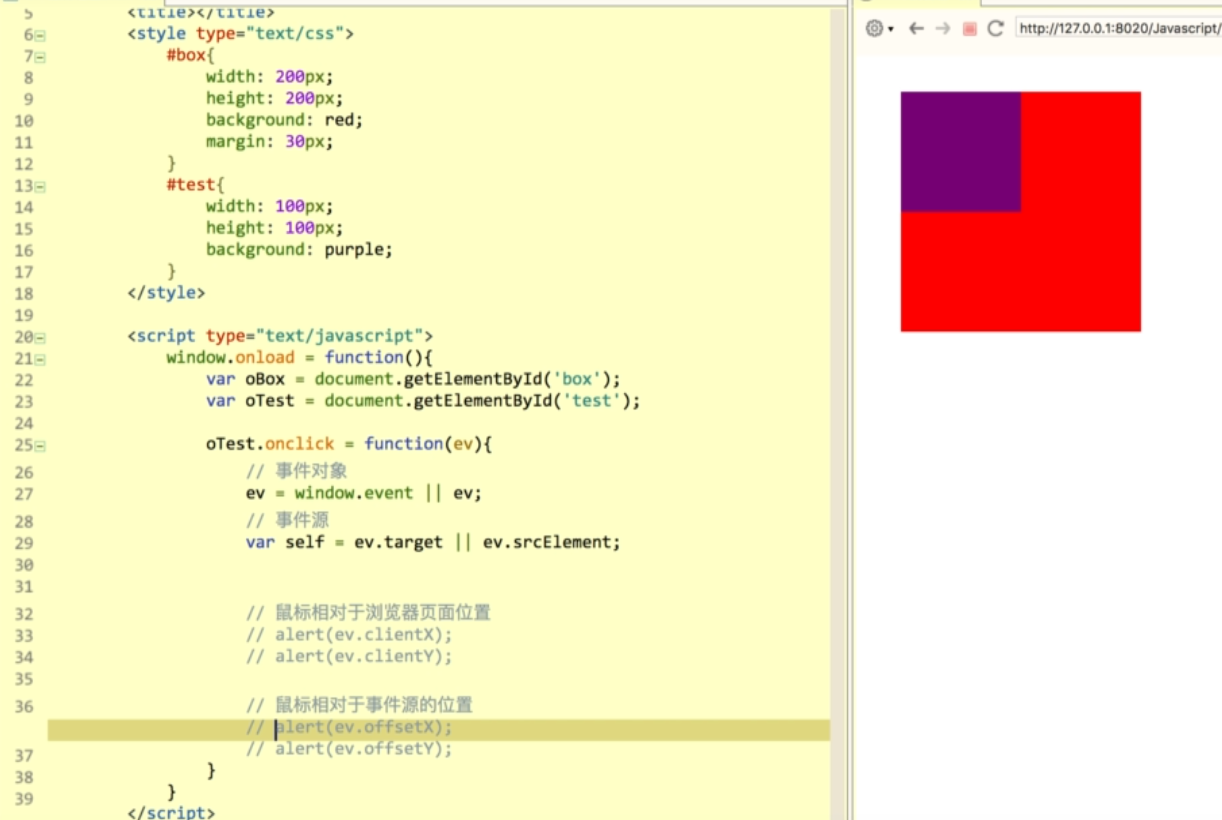
.onfocus, .onblur



事件對象



事件源(兼容瀏覽器)



鼠標位置