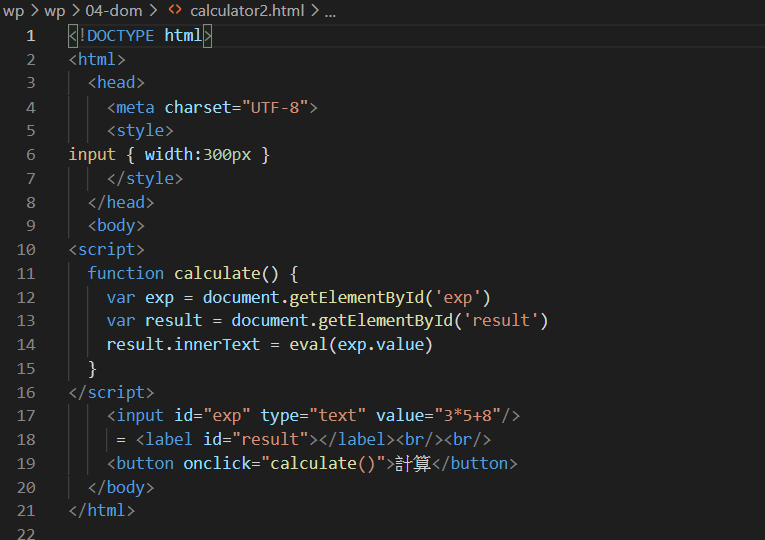
前面先複習5/7號的內容

新增內容 : calculator2, clock2, e2c2, e2c解說

calculator2主要把運算都放到一個格子裡面



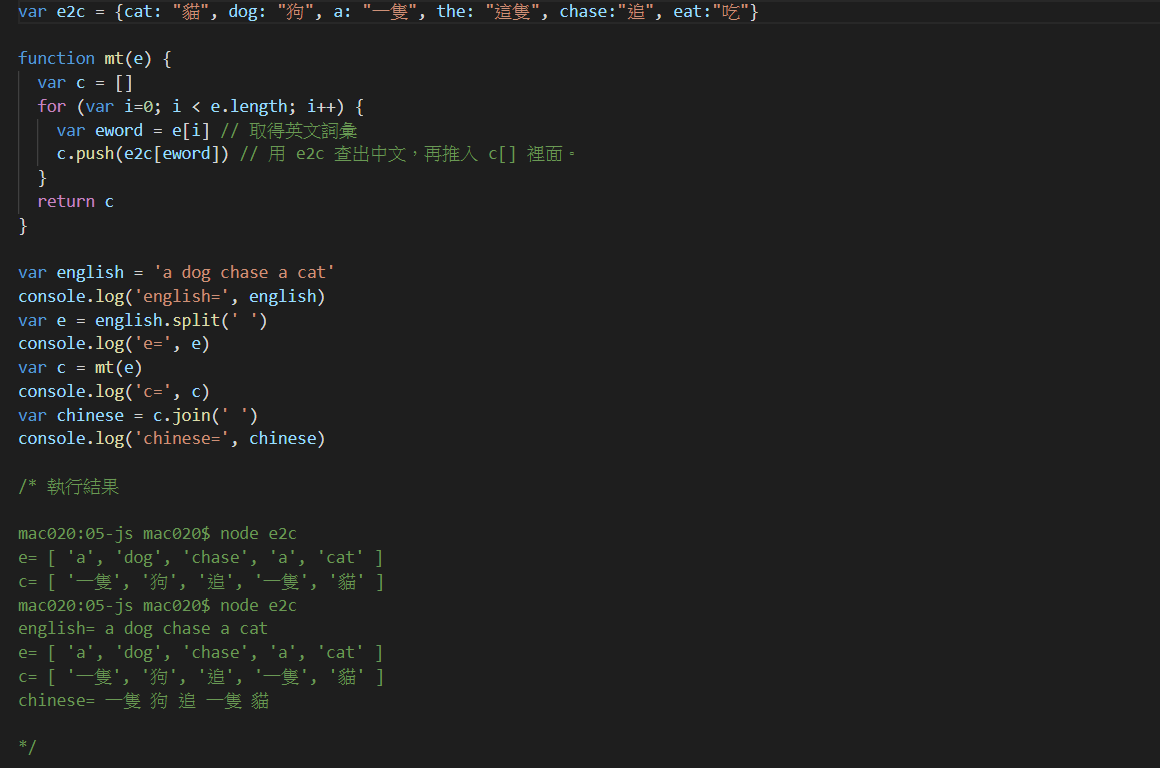
用eval完成運算(再次強調，eval很危險，不要隨便使用)

clock2主要是把之前的setTimeout(showTime,1000); 改成 setInterval



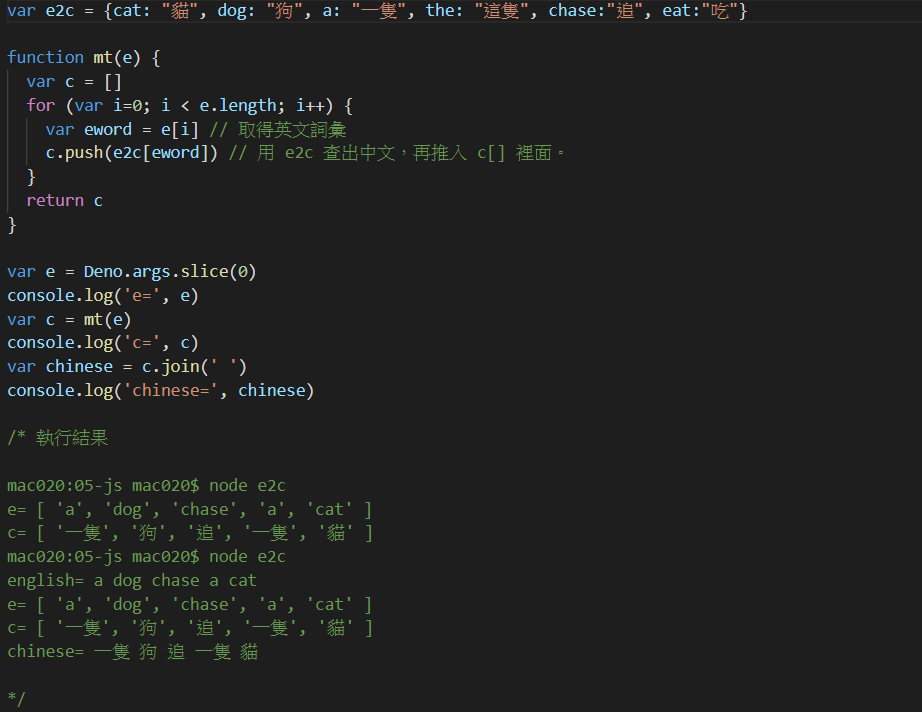
setTimeout只會跑一次要用遞迴，setInterval本身就會自己跑

e2c是英中翻譯機，利用字串a dog chase a cat再把它split(‘ ’)利用空白分割裡面的內容，成為一個一個字，最後再放到e2c矩陣裡翻譯



Join是把矩陣裡面的所有物件都加入後面的”?”，這裡是加空白

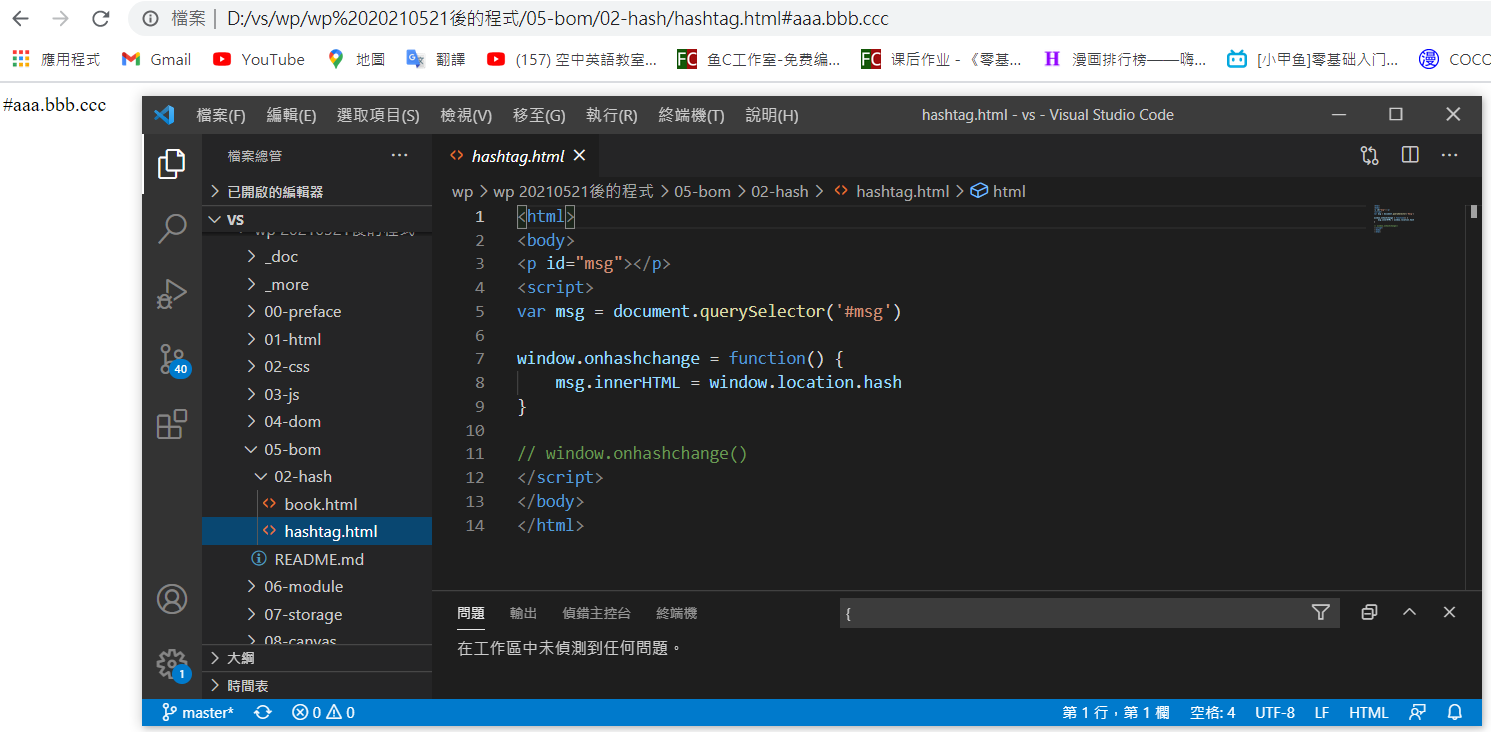
e2c2是上面e2c的強化版，跟5/7號裡面的translator.html類似，利用的是Deno.args裡的功能(這是額外的函式庫)



Deno.args.slice(0)把下面輸入的字逐字翻譯(從第1個)

BOM

## hash

hash是指在網址後面加#...的東西

像上面的這個是把hash丟入網站裡面

#符號是用來當書籤用的，當網頁很長時，可以透過書籤做整理

(#後面輸入id)

其他的BOM在5/14就說得差不多了

MODULE

<script type="module"> import {hello} from './hello.js' hello() </script>

這個是比較新的寫法，如果希望程式更有彈性且模組化，就可以用

但是它需要伺服器才可以用Q^Q

現在就先簡單介紹如何開伺服器

1. 先到需要開伺服器的檔案那裡開終端機
2. 在終端機打deno upgrade
3. 把伺服器安裝進去，在終端機打

deno install --allow-net --allow-readhttps://deno.land/std@0.97.0/http/file\_server.ts

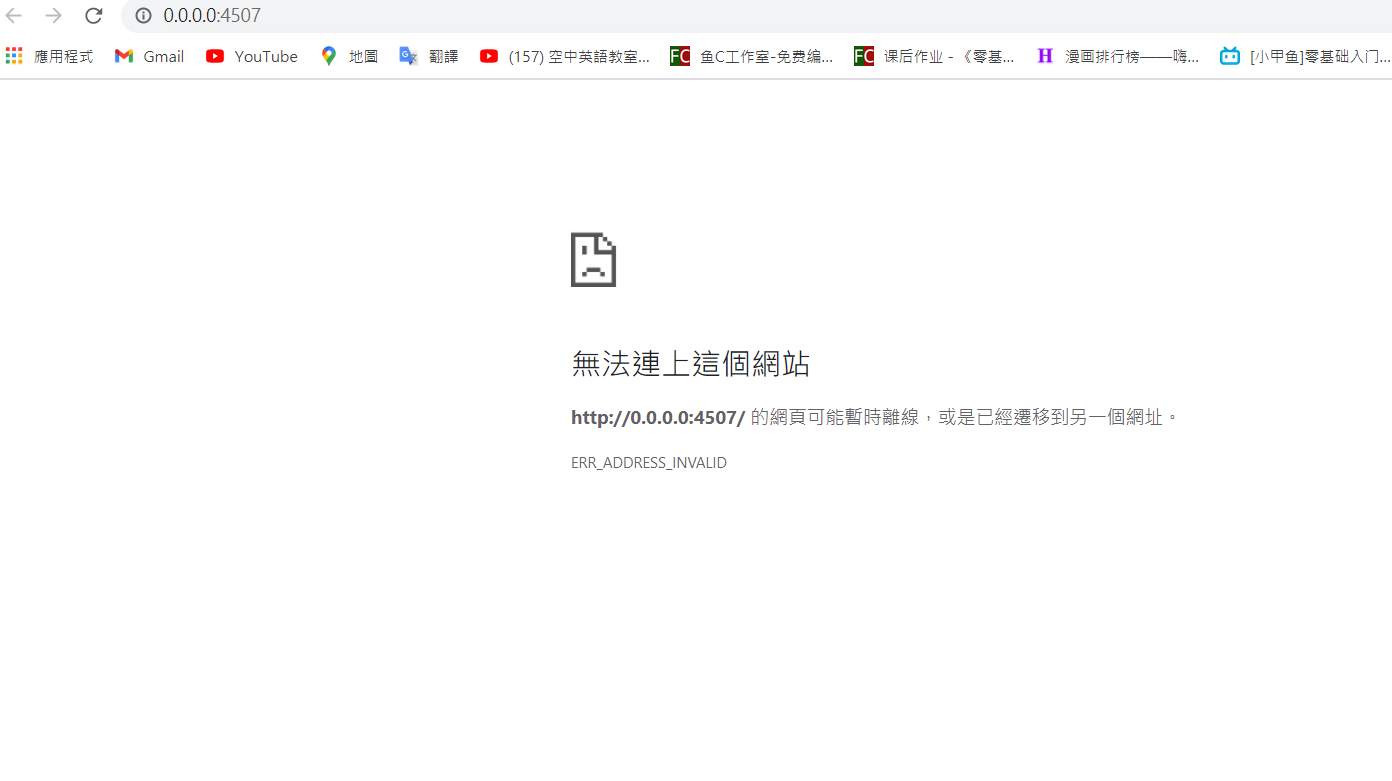
出現✅ Successfully installed file\_server就是成功了!!

如果安裝過要強制安裝要輸入 : deno install --allow-net --allow-read -f https://deno.land/std@0.97.0/http/file\_server.ts

之後指定要上去的文件，在終端機打 file\_server ./(目前資料夾，如果是../就是上個資料夾)

之後他就會顯示文件在伺服器的位置了

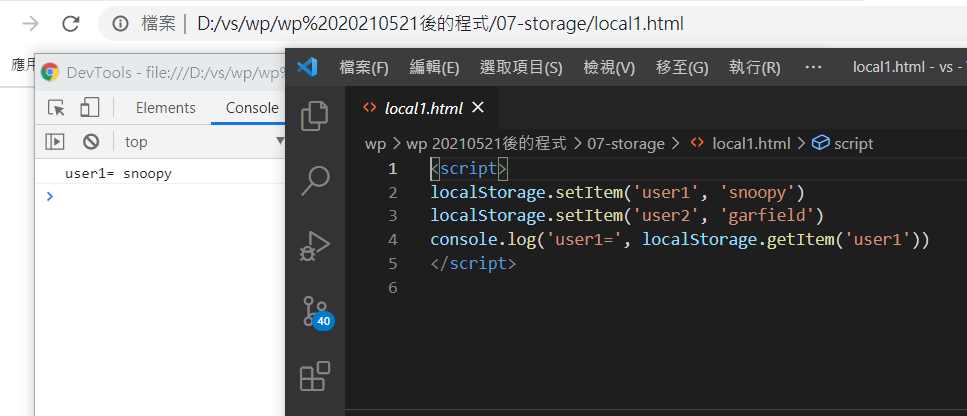
我的找不到位置



STORAGE

這個有分localStorage(存在本地)和sessionStorage(存在本地可以重整，但離開就會不見)

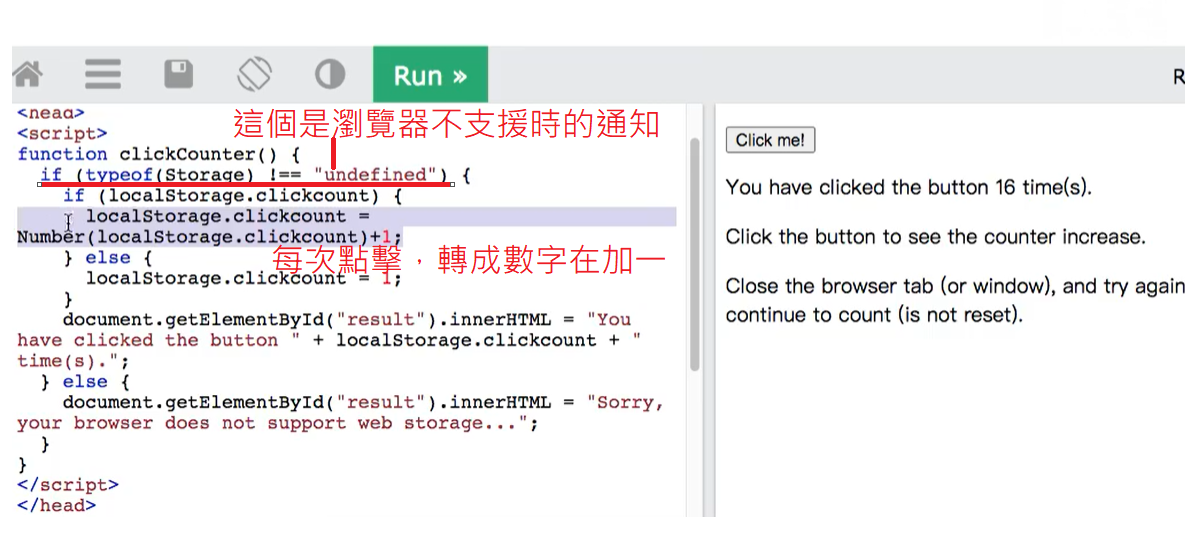
## local





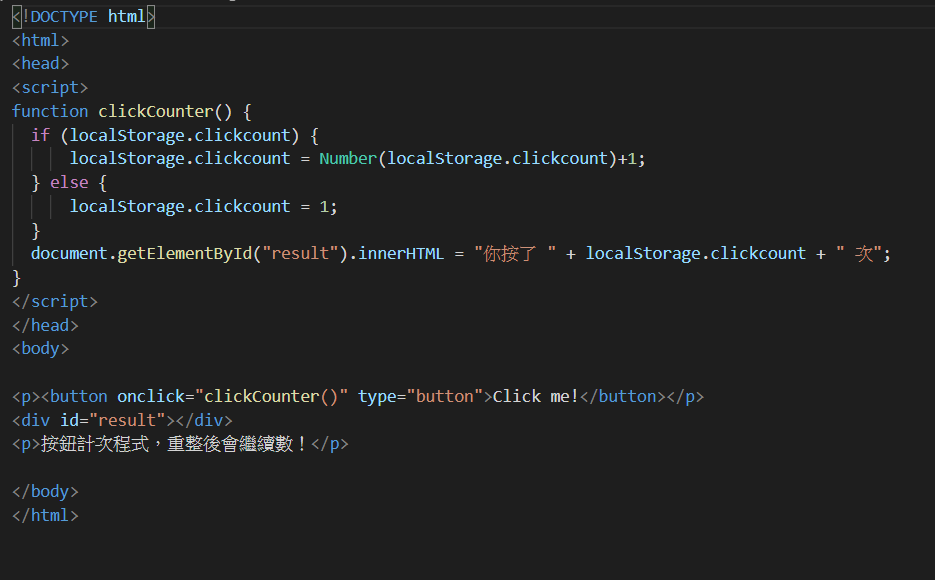
利用localStorage可以跨網頁取得資訊，前面第一張圖是先存(‘user1’)的資訊(‘snoopy’)，之後另一個網頁呼叫localStorage.getItem(‘user1’)就會印出snoopy

這種方法只可本地使用，自己設定的東西只有自己能看到



## clickCount

現在的瀏覽器基本上都支援localStorage，所以不用加前面的那個也可以，就直接寫像下面這樣



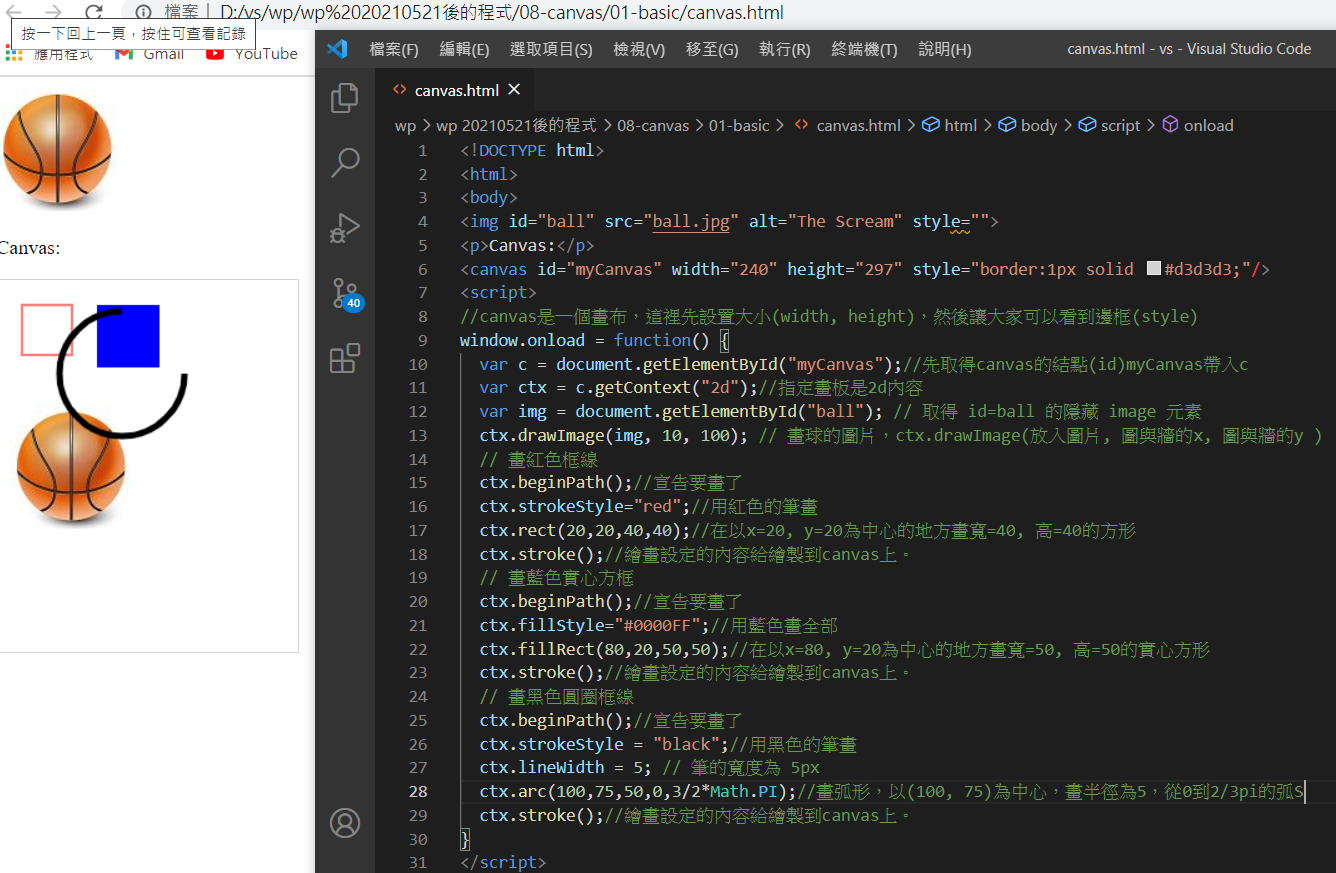
這個是沒伺服器目前網頁唯一儲存的方法

下面這個是sessionStorage，和localStorage寫法差不多，不過sessionStorage重整資料還在，離開就會重來



CANVAS

## basic



上面的程式最後會長的像左邊那樣

//主要是先生成畫板

var c = document.getElementById("myCanvas");

c var ctx = c.getContext("2d");

//放入圖片

document.getElementById("ball");

//再拿畫筆畫

ctx.beginPath();

//回傳話板

ctx.stroke();

## ballmove1



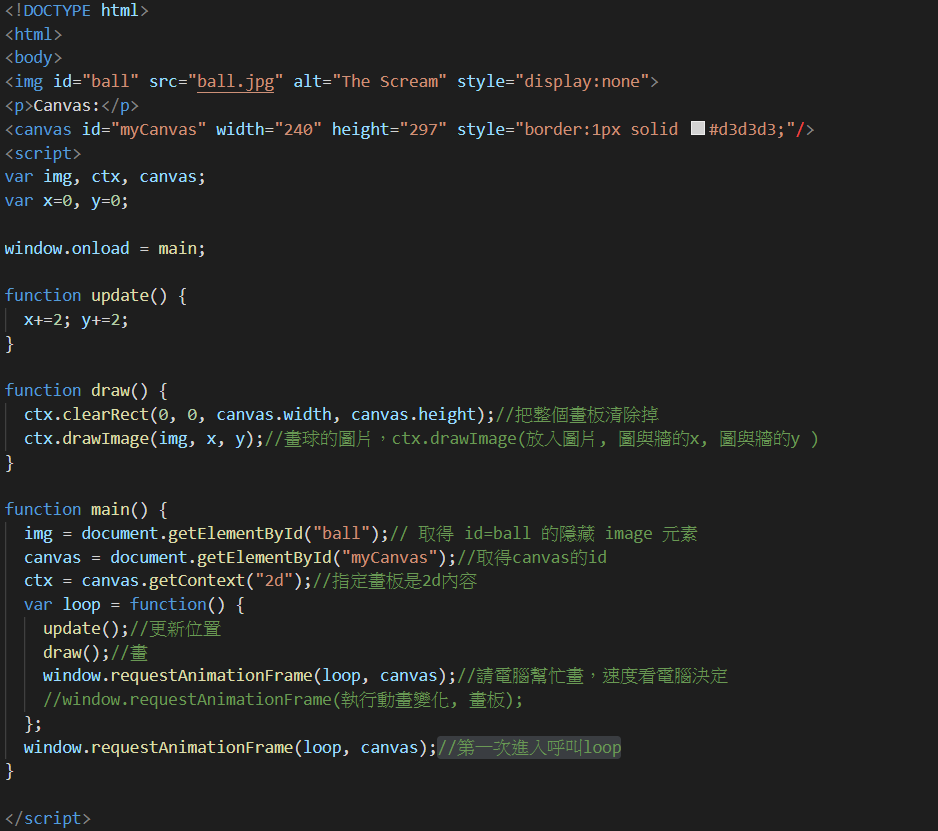
windows.onload = main指的是一開始帶入main函式

ctx.clearRect(矩形的的x, 矩形的的y, 矩形的的width, 矩形的的height);

這裡是設置ctx.clearRect(0, 0, canvas.width, canvas.height);

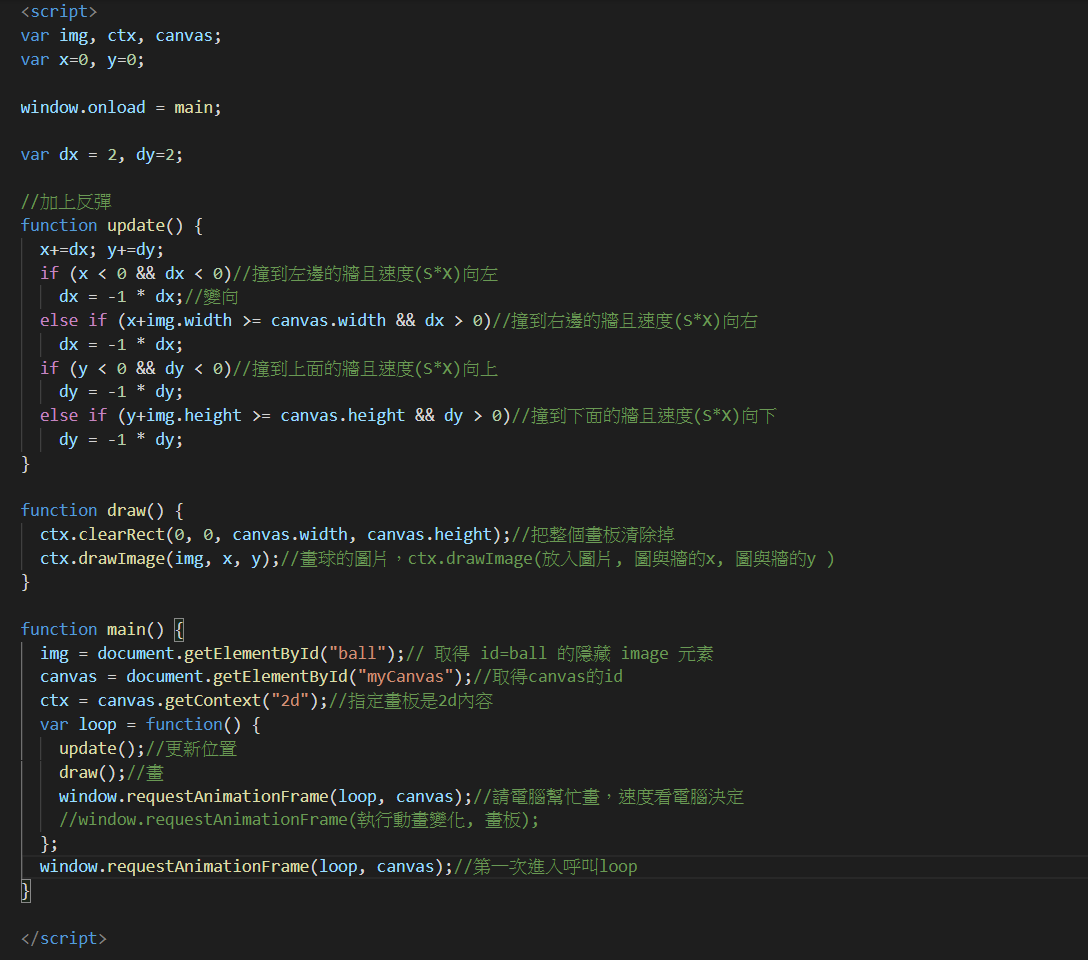
代表從0到整個畫板做清除

## ballmove2



這個是順暢的動畫，用windows.requestAnimationFrame做動畫

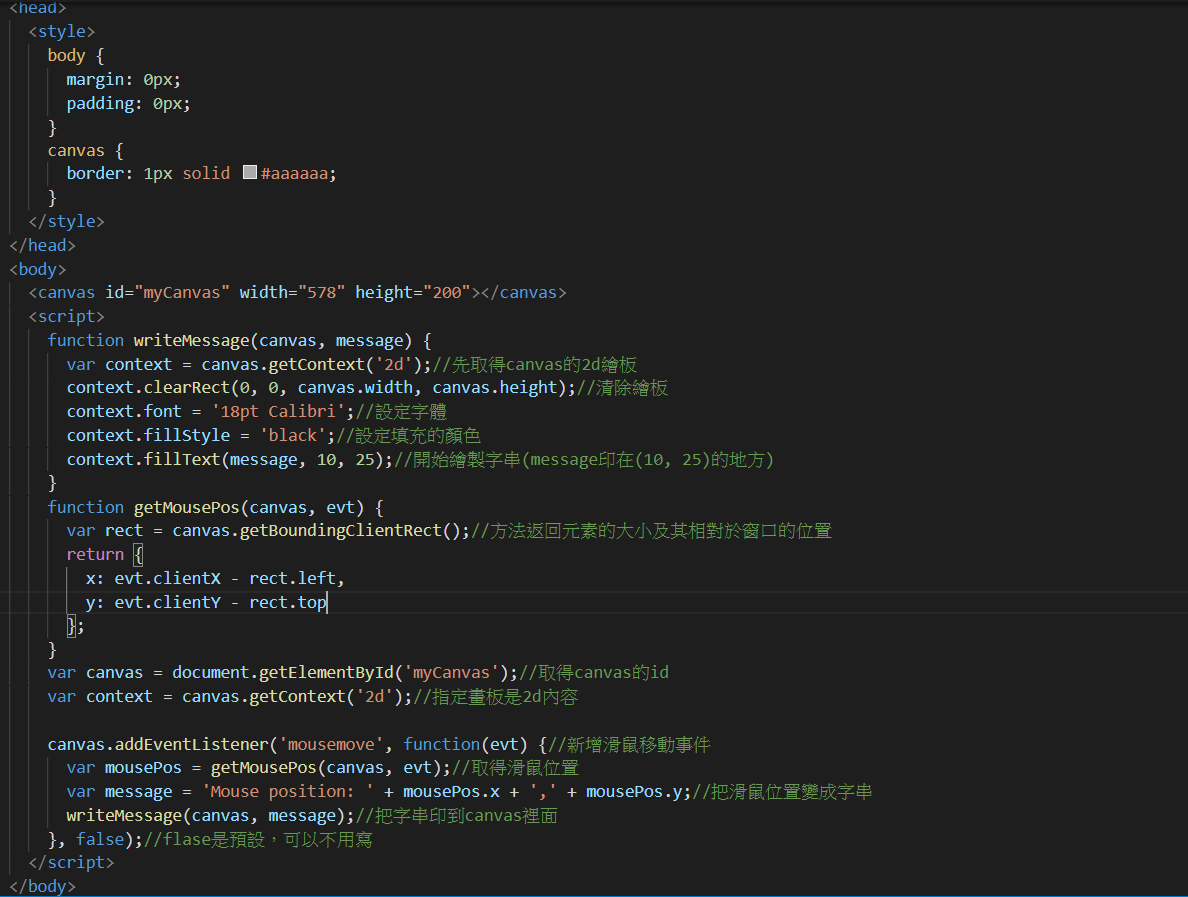
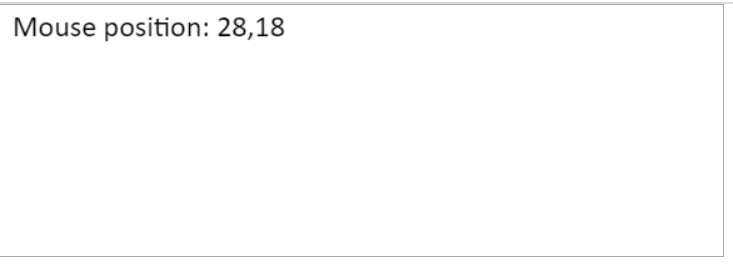
## ballgame



上面這個和剛剛上上面的那個差在update()，這個會反彈

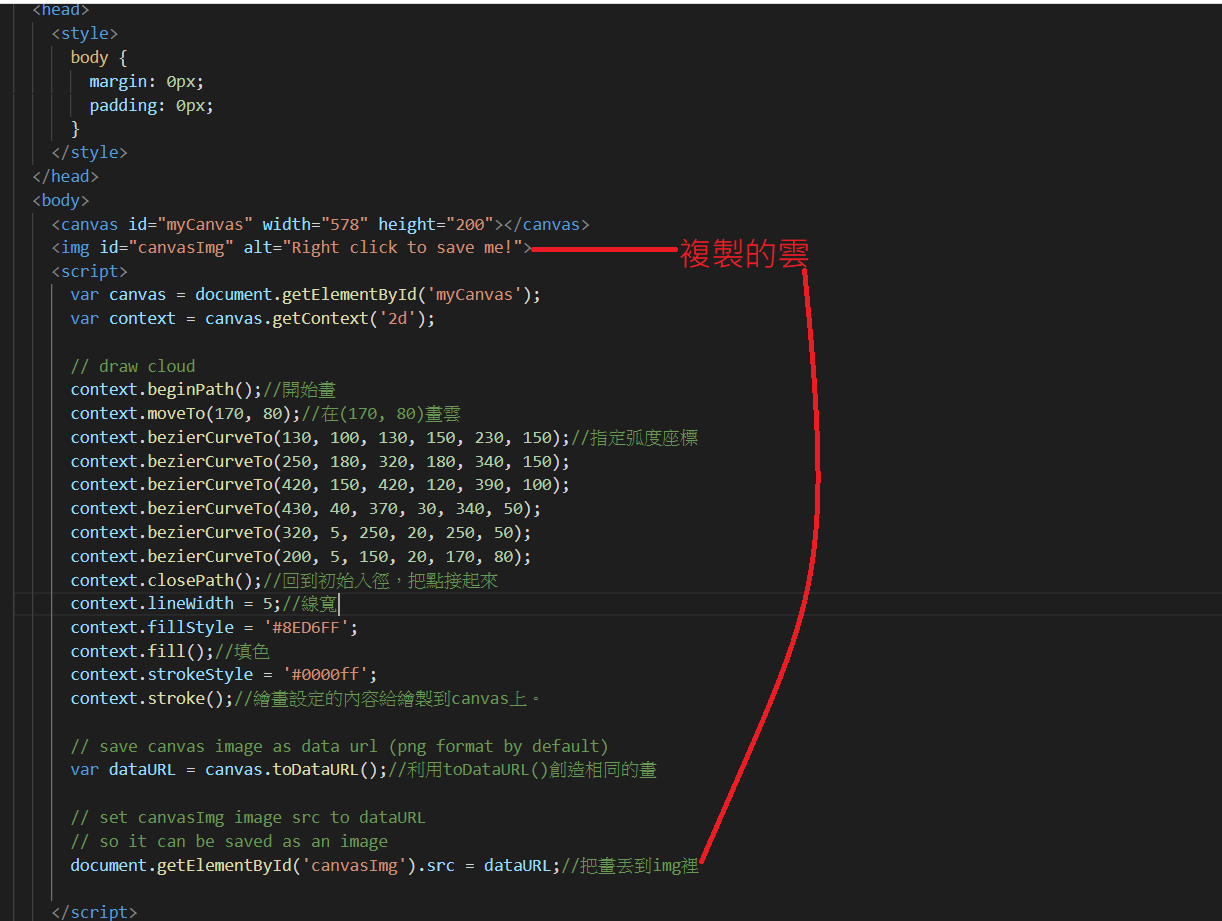
程式碼不一定要從上看到下，有些程式碼就要從下看到上

## mouseMove

這個是可以取得滑鼠位置的程式碼，如下圖  


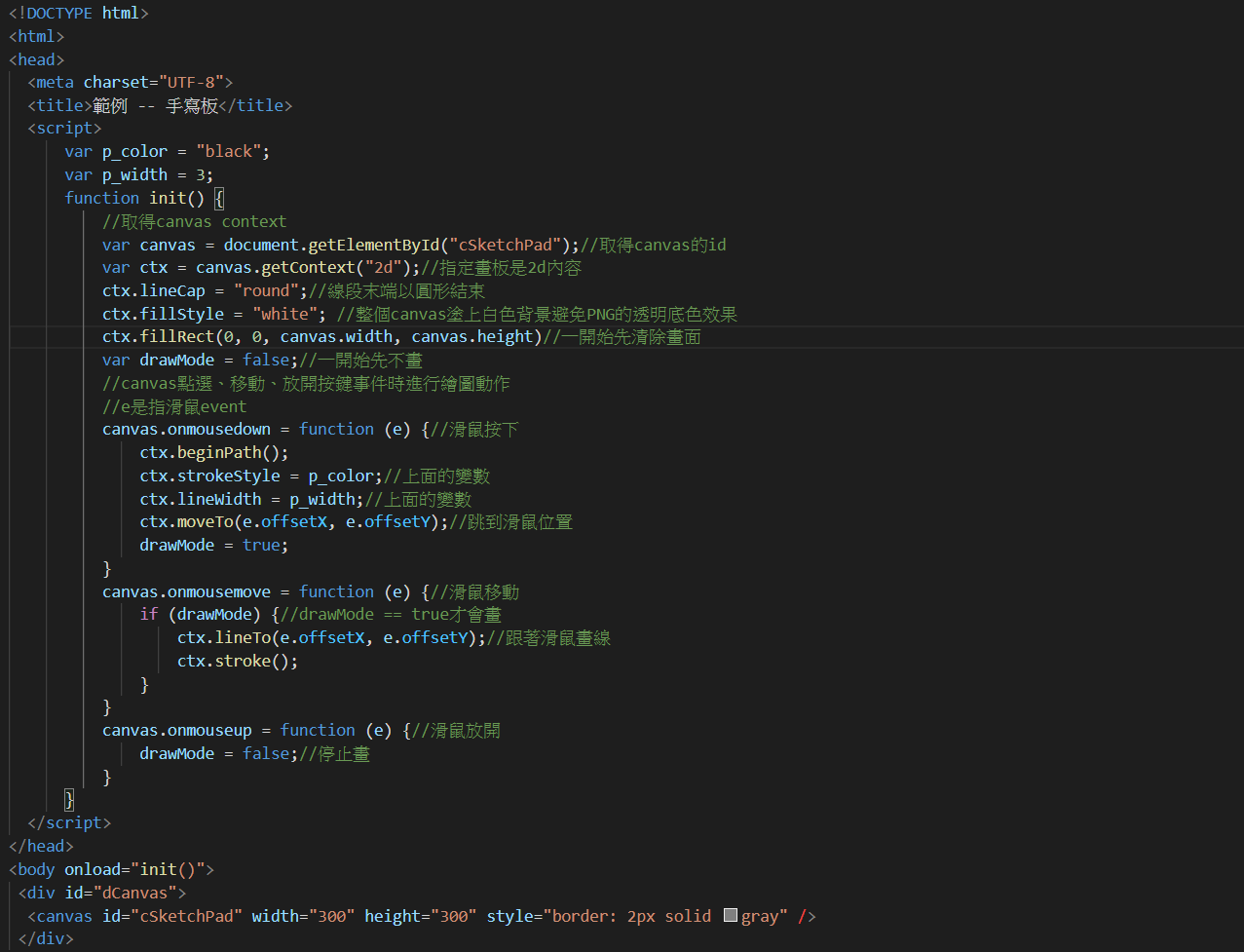
## canvasSave





document.getElementById(‘myCanvas’).toDataURL()可以存canvas裡面的圖

## painter

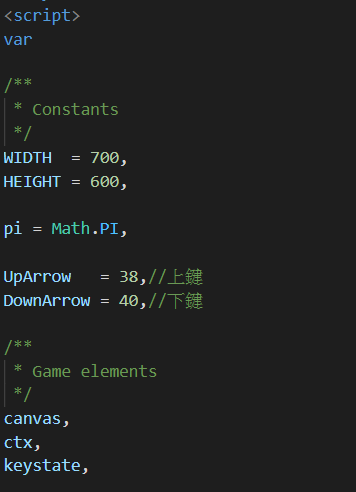


這個是繪畫板，用點選、移動、放開按鍵事件(e)進行繪圖動作

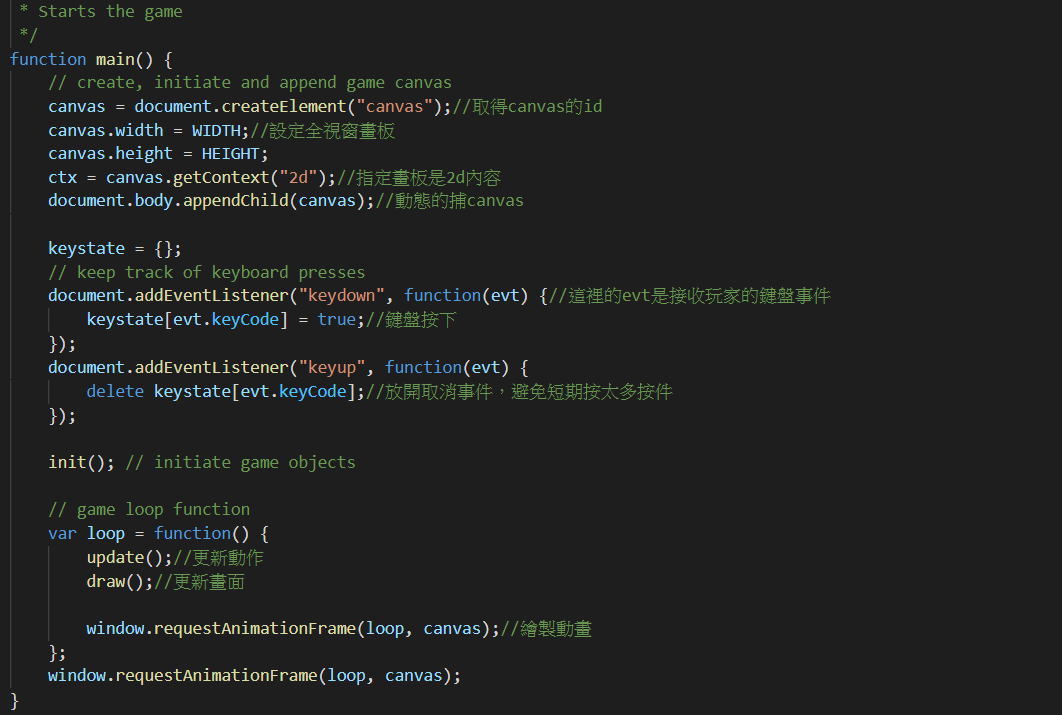
## pingpong

下面是乒乓球遊戲，很複雜

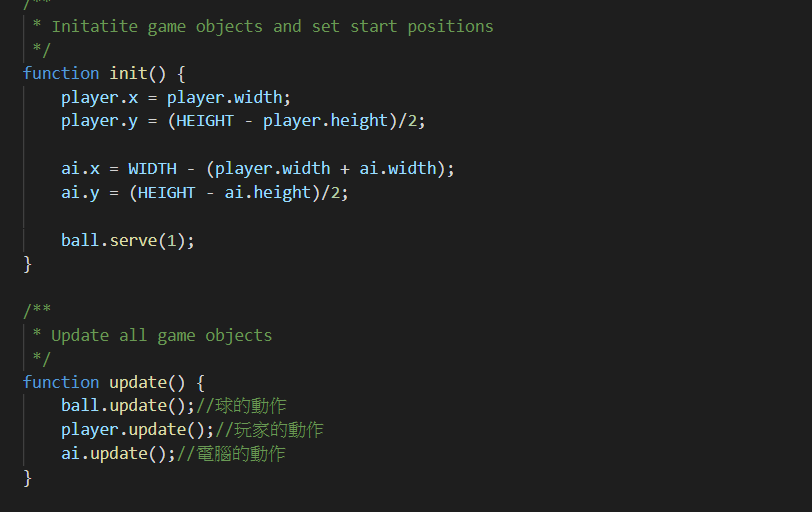
遊戲大變數 :



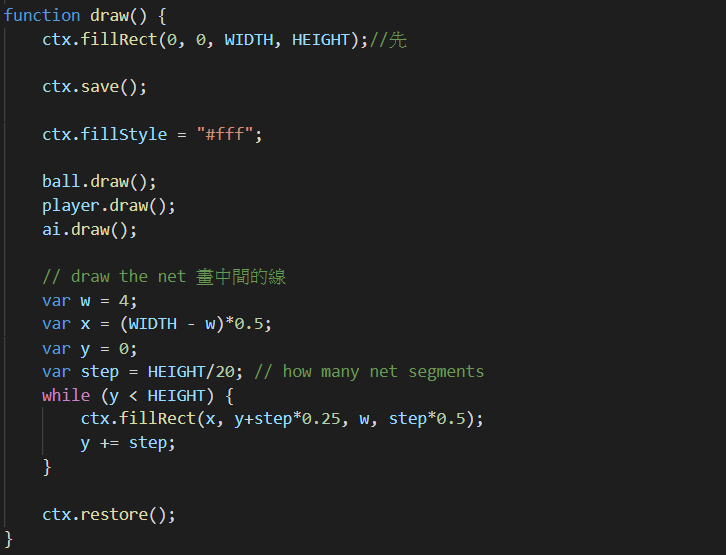
主要框架 :



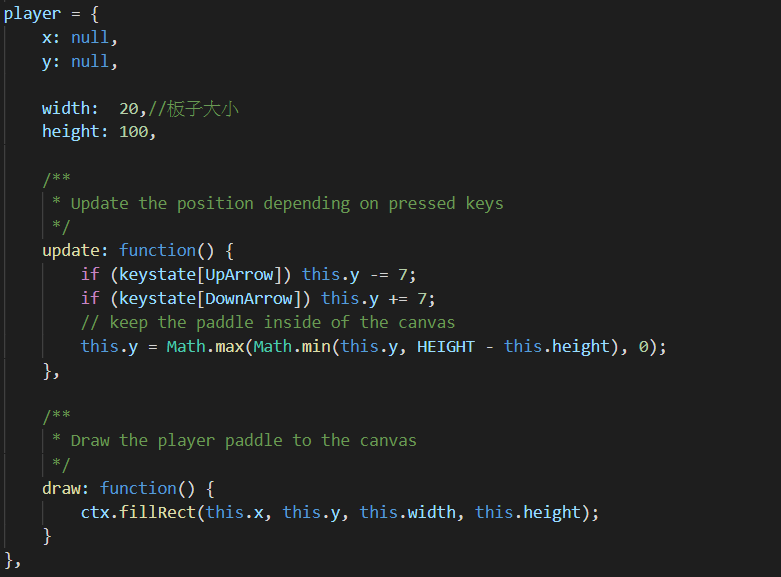
遊戲初始化和遊戲動作 :



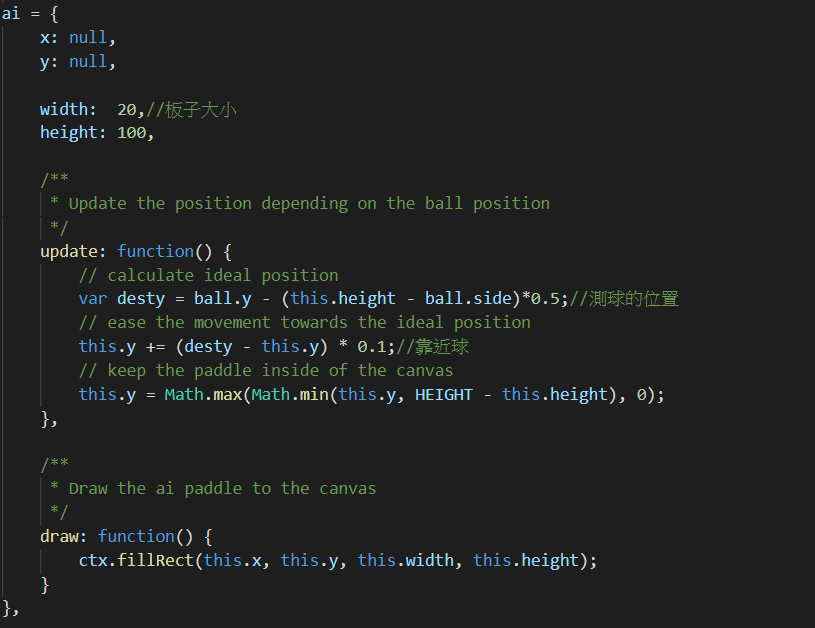
遊戲繪製 :



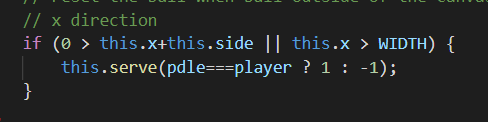
玩家和內部內容 :



電腦和內部內容



球的部分跟上一個反彈的球差不多，不過有奇怪的符號 ? 和 :



而且超長

如果要遊戲模板可以用下面的

遊戲框架 phaser -- <https://phaser.io/>