**JavaScript 物件與類別**

let和var的區別

let的作用範圍只在block，而var是在整區生效。 EX:

function varTest() {

var x = 1;

{

var x = 2; // 這裡的 x 與 function 區塊內部的 x 是一樣的，因此會影響 function 區塊內所有的 x

console.log(x); // 2

}

console.log(x); // 2

}

function letTest() {

let x = 1;

{

let x = 2; // 這裡的 x 與 function 區塊內部的 x 是不同的，只會作用在這層 block 區塊中

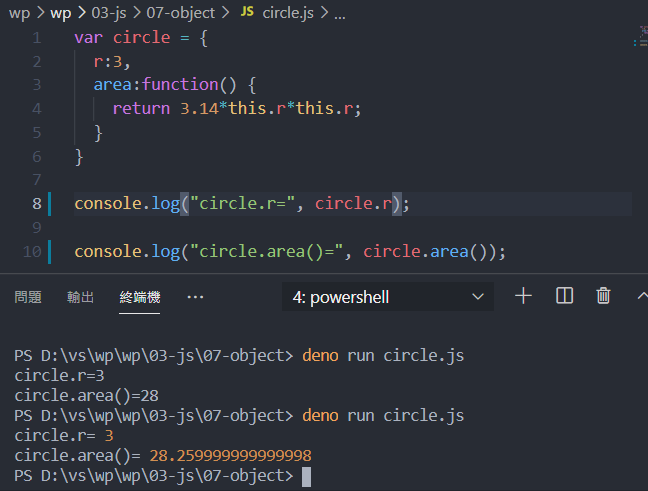
console.log(x); // 2

}

console.log(x); // 1

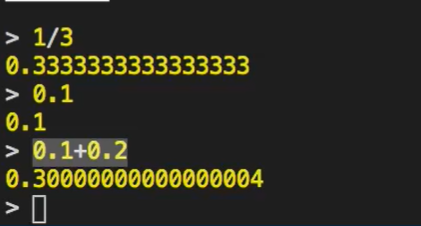


上面這是物件搭配for的用法，可以將陣列裡的物件印出



This 是代表函式本身

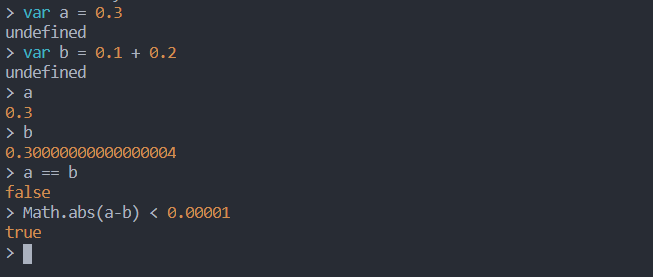
上面的面積答案會有誤差是因為用的是浮點數(雙精度浮點數)運算



如果要做浮點數運算判斷的話，最好不要用等於，這很容易是bug的來源

用兩個差異的絕對值是不是最小判斷比較好

EX :



Math是JavaScript的函示庫Math.abs是代表後面a-b的絕對值

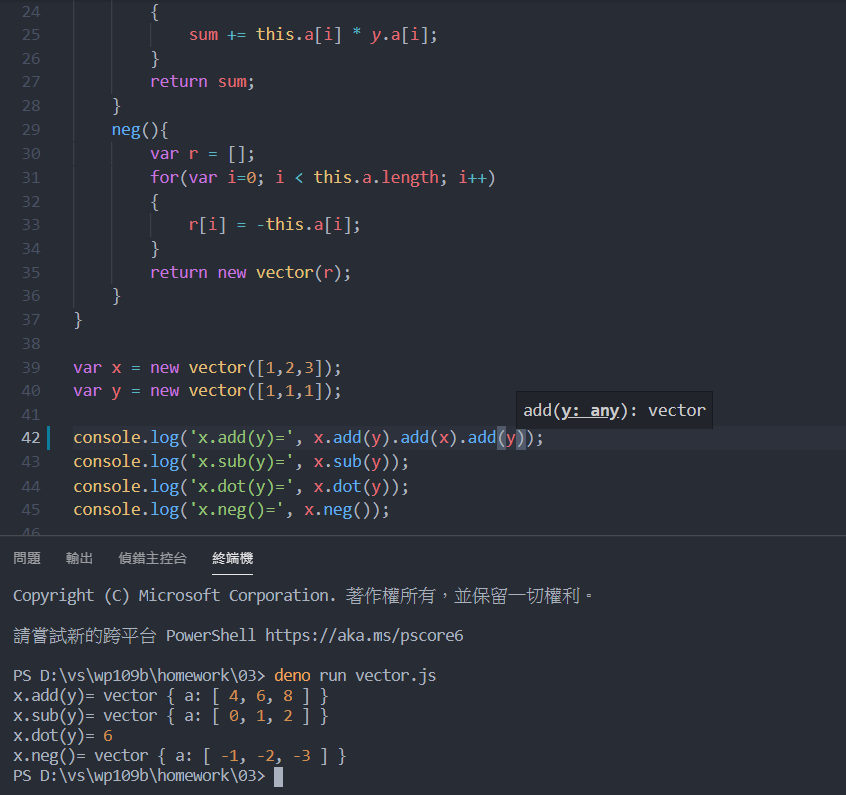
歷史上有很多大災難的bug就是浮點數造成的，千萬不要用等於在浮點數上



上面的setRadius可以把r帶入之後再回傳circle所以後面可以再接他自己

***JavaScript的寫法是 var c1 = new Circle(1);***

***但在C++ 的寫法是 Circle c1 = new Circle(1);***



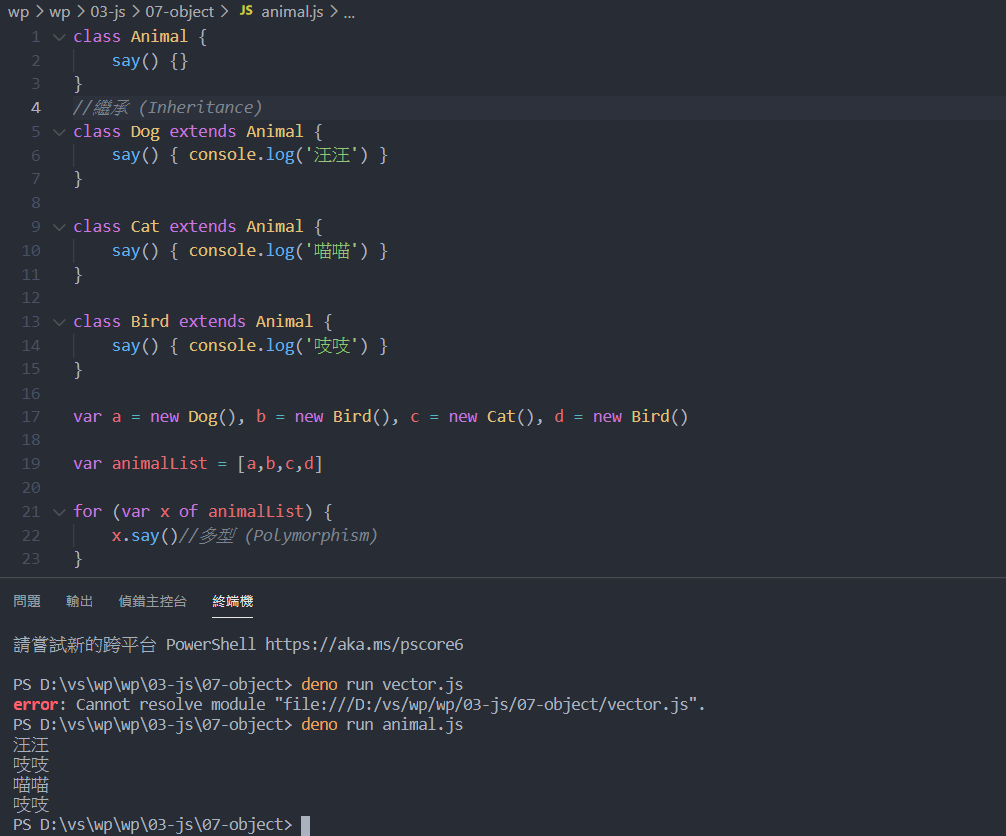
物件導向的好處是可以比較好看，好統整。

這邊的累加不用物件導向，就會很麻煩

物件導向有很多特性

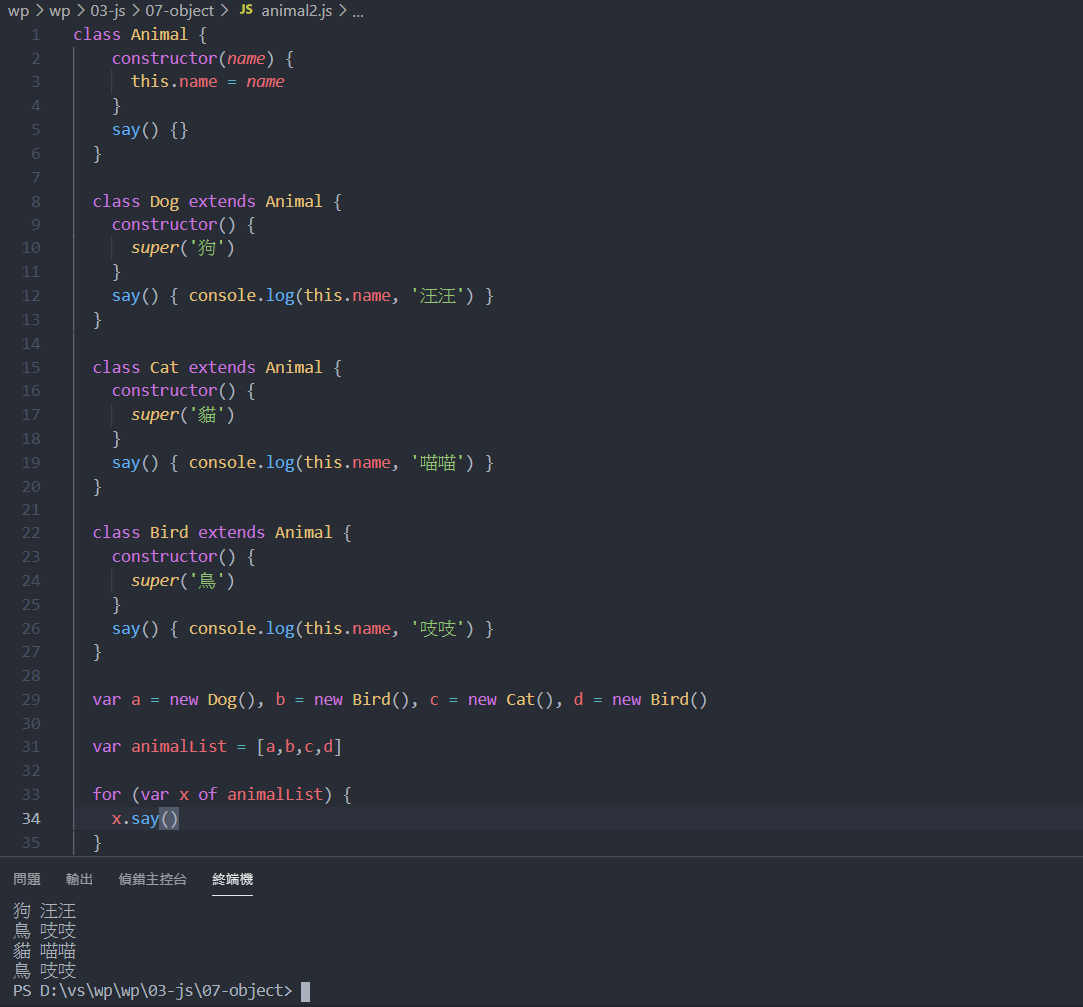
封裝 (Encapsulation) 繼承 (Inheritance) 多型 (Polymorphism)

要寫網頁，或伺服器的話，基本上都要去買雲端，會給你伺服器流量



上面的Dog, Cat, Bird 都繼承在animal裡面，用得好的話，會讓程式變漂亮

在物件裡面的function一般都稱作method



子類別需要在constructor()中呼叫super()來使用父類別的建構函(constructor)

有一點像共用事件

super() 只能在constructor()中執行