JavaScript

js輸出:

document.getElementByID(id).innerHTML //適用於和HTML一同使用

//id為定義HTML元素

//innerHTML則是元素內容,可在此設定HTML的內容,或是稱將程式的執行結果放在此

//用處:可更改具有該id的區塊內容

ex: ex1.html

**comsole函數** comsole.log(內容); //適用於純js或放在js函數內呼叫

//文字需加””,變數和文字一起需用 , 隔開

js輸入:

**prompt()函數**

* prompt 函數會顯示一個對話框，讓使用者輸入文字。
* 使用者輸入的內容將作為字串返回。

ex: var 储存輸入值的變數名 = prompt(“內容”+ 變數 );

//prompt會將()內的內容輸出並在末端等待輸入

定義變數:

* 全局變數 var //宣告後全範圍可使用

var 變數名 = 初始值;

//無法初始化為 null 的變量

* 局部變數 let //離開範圍即無效（ex：if 語句、迴圈或函數）

let 變數名 = 初始值;

//用let宣告的變數不能重複聲明，這有助於避免變數覆蓋問題。

#在js裡,宣告變數時,型別會自動定義

(ex: var i=1;//int ;; var i=＂ ”;//char)

運算子:

+ 加 – 減 \* 乘 / 除 % 餘

++ 遞增 -- 遞減

指派運算子:

+= (ex: a += b 即 a = a + b)

-= (ex: a -= b 即 a = a - b)

\*= (ex: a \*= b 即 a = a \* b)

/= (ex: a /= b 即 a = a / b)

%= (ex: a %= b 即 a = a % b)

布林值:

True真(數值型態為1,布林值為 true)

false假(數值型態為0,布林值為 false)

//布林值轉數值資料時,true轉為1,false轉為0

//數值資料轉布林值時,非0數值轉為true,0轉為false

比較運算子:

> 大於

< 小於

>= 大於等於

<= 小於等於

== 等於

=== 等於且型別相同

!= 不等於

!== 值不等於或型別不同

位元運算子:

& and

| or

^ xor

~ not

<< 位元左移

>> 位元右移

>>> 位元無號位移

運算子優先順序:

|  |  |
| --- | --- |
| () | 括號 |
| () new | 函數呼叫、建立物件 |
| ++ -- ! | 遞增、遞減、not |
| \* / % \*\* | 乘、除、餘數、指數 |
| + - | 加、減 |
| << >> >>> | 比較運算子的等於或不等於(含型別) |
| && | 邏輯運算子 and |
| || | 邏輯運算子 or |
| = += ... >>= >>== | 所有指派運算子 |

for敘述:

解一:

for(statement1;statement2;statement3){ //貼近C

系列 statement;

}

statement1:設定迴圈敘述指標

statement2:測試是否要離開迴圈

statement3:更新迴圈敘述指標

ex:

for(var i=0;i < n;i++){ console.log(“Hello World!”)}

解二:

for(var i in 範圍){ //貼近python

系列 statement;

}

if敘述:

if(condition1){

A系列 statements; //所有條件皆是ture則執行

}

else if(condition2){

B系列 statements; //所有條件皆是ture則執行

}

……

else{

C系列 statements; //所有條件皆是falue則執行

}

簡化版:

變數 =( 變數 > n ) ? “內容”: “內容”;

ex: ex2.js

array 陣列:

var 陣列名 = [data1,data2,...datan];

ex: var numlist=[8,7,9];

//numlist[0]=8,numlist[1]=7,numlist[2]=9 ;; 索引從0開始

//可隨意建立陣列大小,ex: var numlist=[[1,2.5,7],’Hi’,[2,9],5];

建立字串列:

ex: ex3.html

計算陣列長度:

(储存用)變數名 = (要計算的)陣列名.leghth;

var numlist=[1,1,2,6,5];

var x=numlist.length; //x=5

陣列排序(用函式):

陣列名.sort();

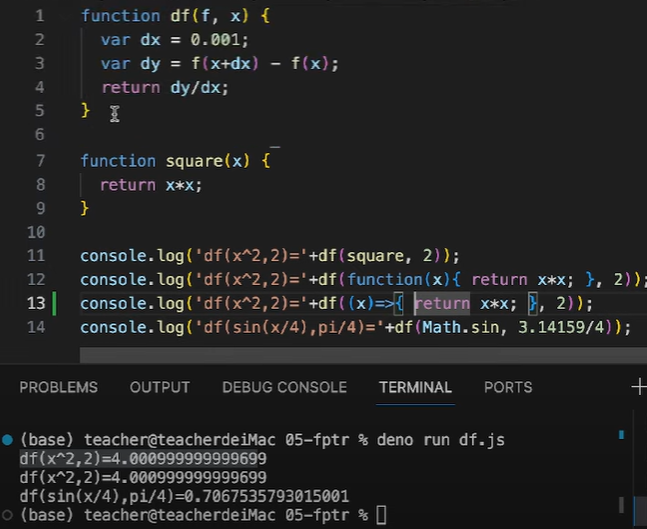
函式:

function funname(parameter1,…,parameterN){

系列 statements;

return 回傳值;

}



函式的不同呼叫方式,圖例為微分

物件:

var 物件名={

屬性名1 :“”,

屬性名2 :“”,

id:1234,

屬性名3:function(){

return this.屬性名1 +“ ”+ this.屬性名2;

}

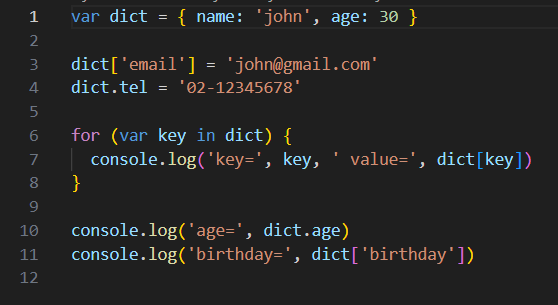
};

document.getElementById(“ID”).innerHTML= 物件名.屬性名3();

//缺少()會輸出函式的定義

ex:





BOW(Browser Object Mobel)

BOW是Javascript和瀏覽器的橋樑。