



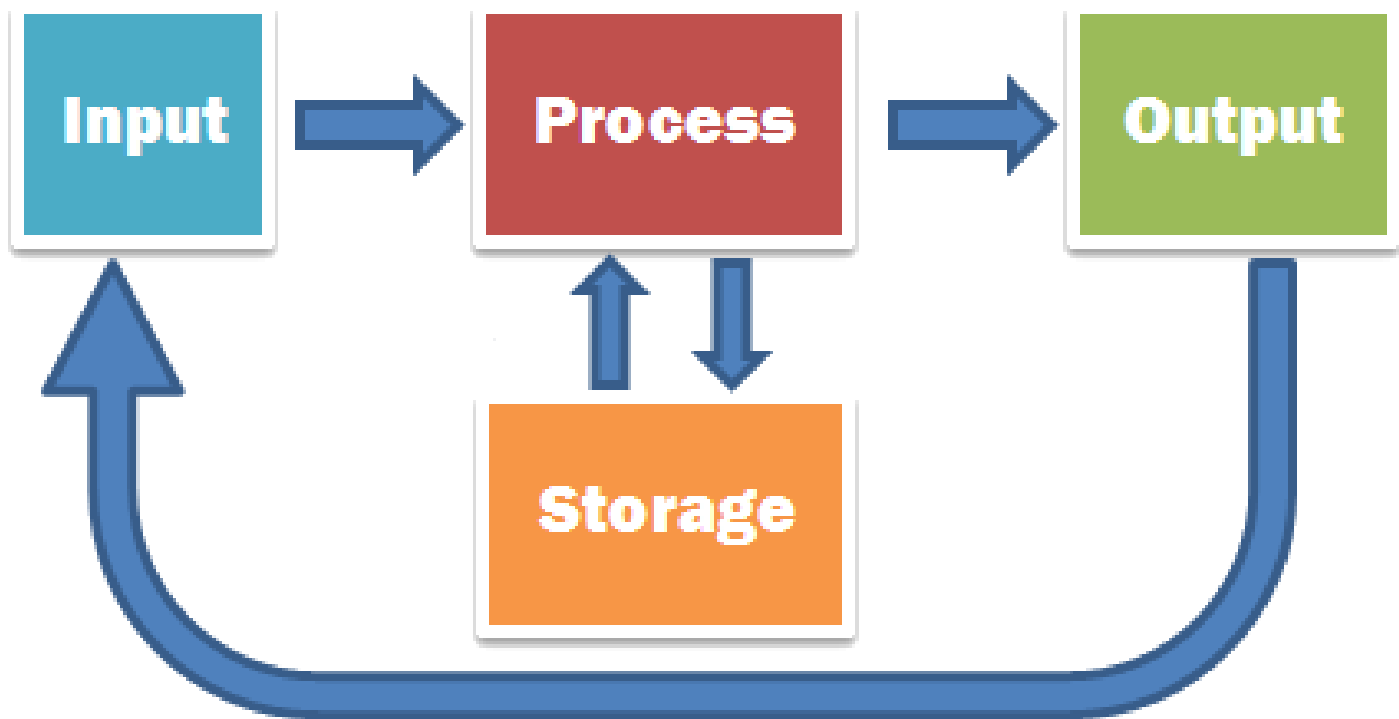
Javascript入門

Created By : 張繼振/Scott Chang

Javascript語法簡介

- 變數
- 流程控制
 - 條件控制：if/else ; case/switch;
 - 迴圈控制：for ; while; do while;
- 陣列
- 函數

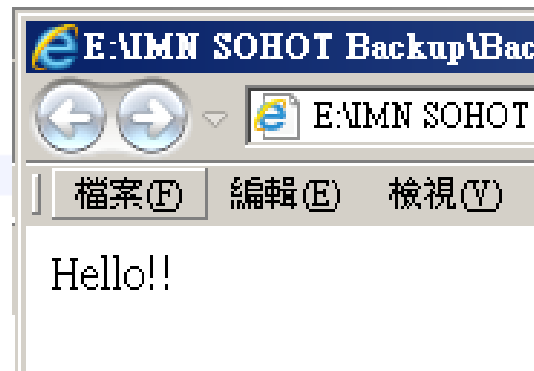
資訊系統



Lab 網頁製作 - output

- JavaScript 程式撰寫位置一

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>JavaScript Template</title>
</head>
<body>
  <script>
    document.write("Hello!!");
  </script>
</body>
</html>
```

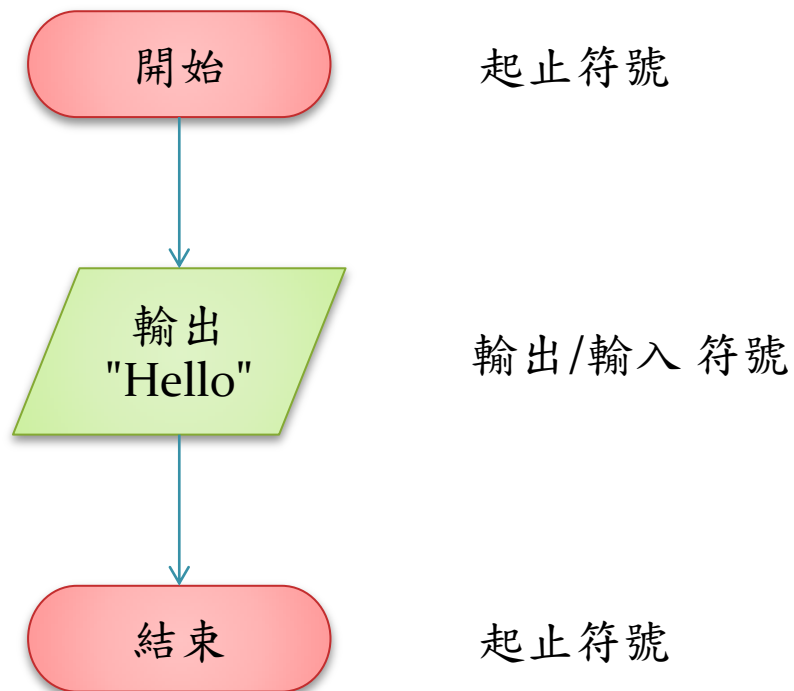


- document.write

```
document.write("Hi!");    //輸出字串:Hi!
document.write(1+1);      //數字相加,輸出:2
document.write("A" + "B"); //字串相加,輸出:AB
```

(練習 : js1-01 output.html)

流程圖



字串、數字及註解

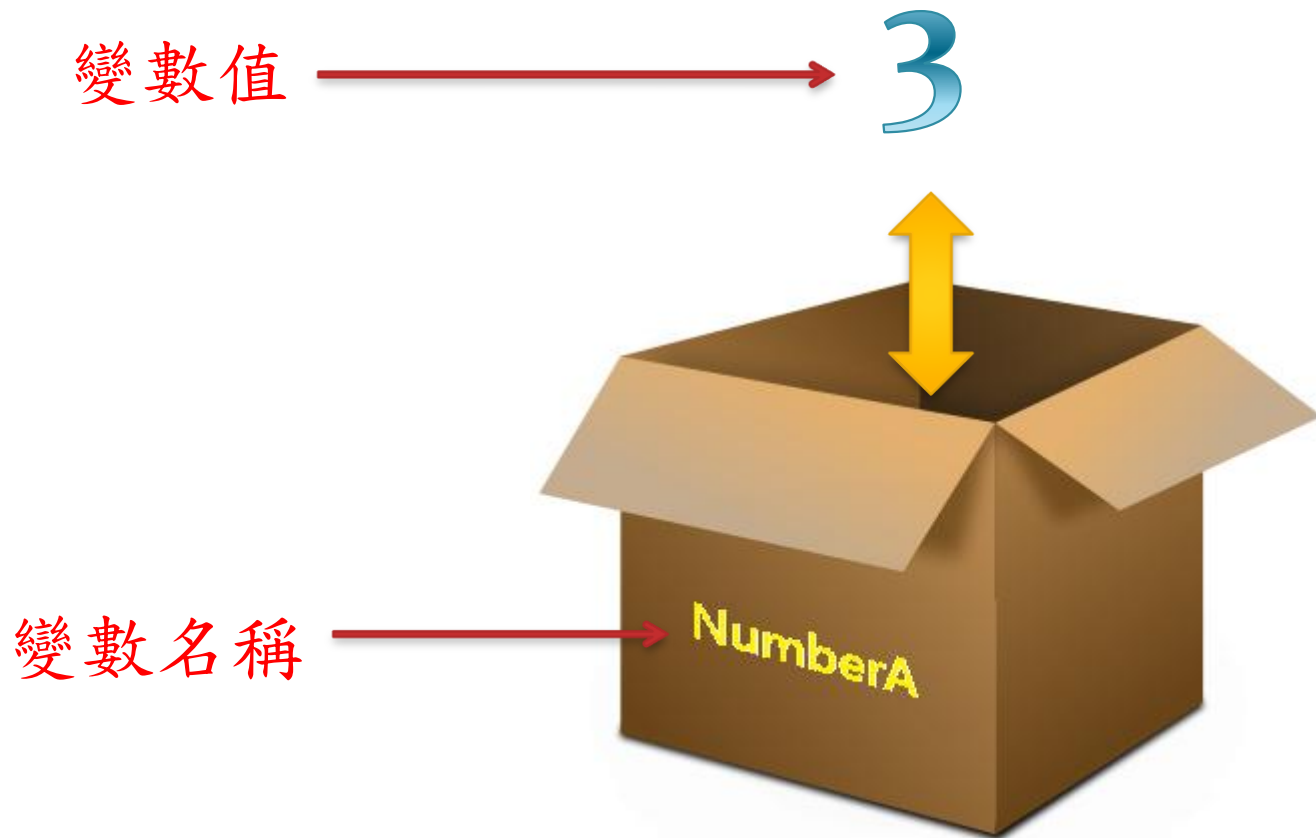
- 字串：置於單引號或雙引號之間
 - "Hi"
 - '12345'
 - "It's a book!"
- 數字：
 - 100
 - 0.01
- 註解
 - //這是單行註解
 - /* 這是多行註解 */



變數

變數

- 儲存資料的容器
- 程式中使用 **變數名稱** 取得或指定 **變數值**



變數的取得與指定

- 程式中使用變數名稱**取得(Get)**變數值
`document.write (NumberA) ;`
- 程式中使用變數名稱**指定(Assign)**變數值

`NumberA = 5;`

`NumberB = 4;`

`NumberA = NumberB;`

`NumberA = NumberA + 2;`

`NumberA = NumberA + NumberB + 3;`

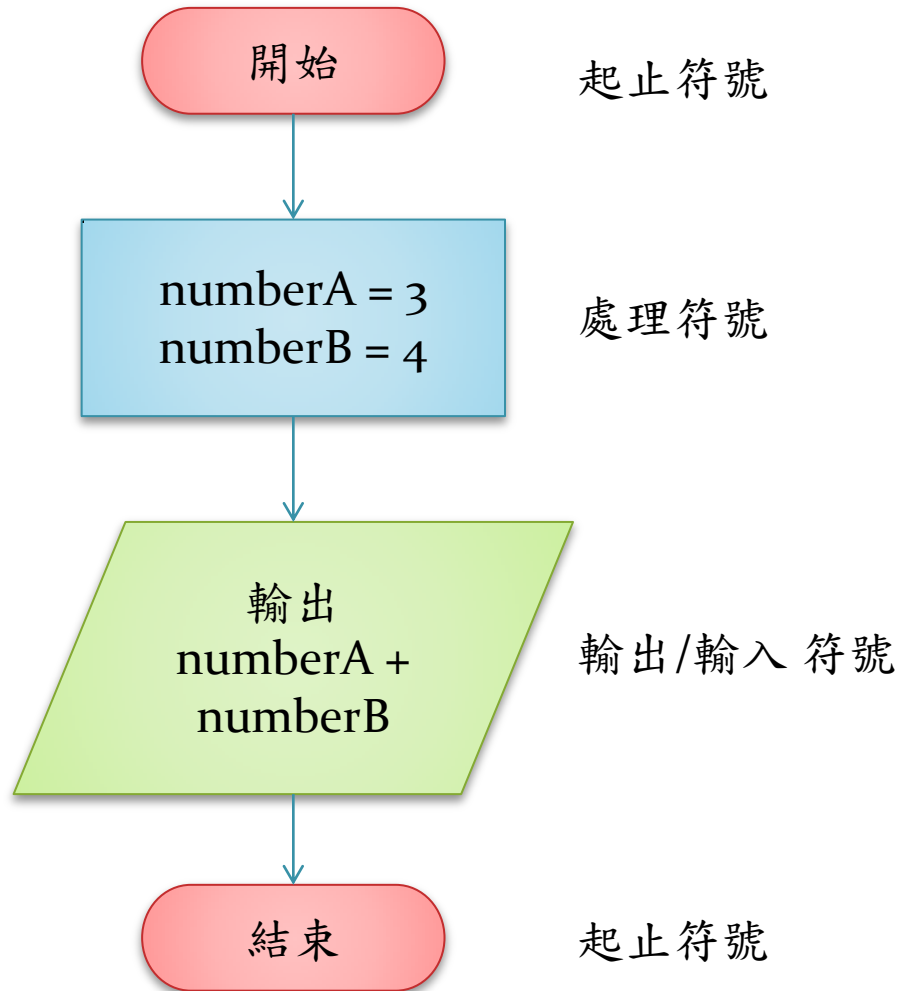
~~`NumberA + NumberB = NumberA + 3;`~~

Lab-變數基本運用

- 指定 `numberA = 3`
- 指定 `numberB = 4`
- 輸出 `"numberA"` //顯示 `numberA`
- 輸出 `numberA` 的值 //顯示 `3`
- 輸出 `numberA = 3`
- 輸出 `numberA + numberB = 7`

(練習 : `js1-02 variable.html`)

流程圖



變數的資料型別(Data Type)

- 如果：Java + Script = JavaScript
那麼：1 + 1 = ?
- Javascript變數的三種基本資料型別

基本資料型別	說明
String	內含一個或多個字元。 使用「'」或「"」括起的字串
Number	整數或浮點數
Boolean	true(真) 和 false(偽)

變數的宣告

- Javascript使用var指令宣告變數

```
//變數宣告
```

```
var strName;
```

```
var strName, intBalance;
```

```
/*
```

```
 * 1. 變數宣告
```

```
 * 2. 同時指定變數值
```

```
*/
```

```
var strName = "Javascript";           //姓名 (String)
```

```
var intBalance = 1000;                 //存款 (Number)
```

```
var blnSex = true;                     //性別 (Boolean)
```

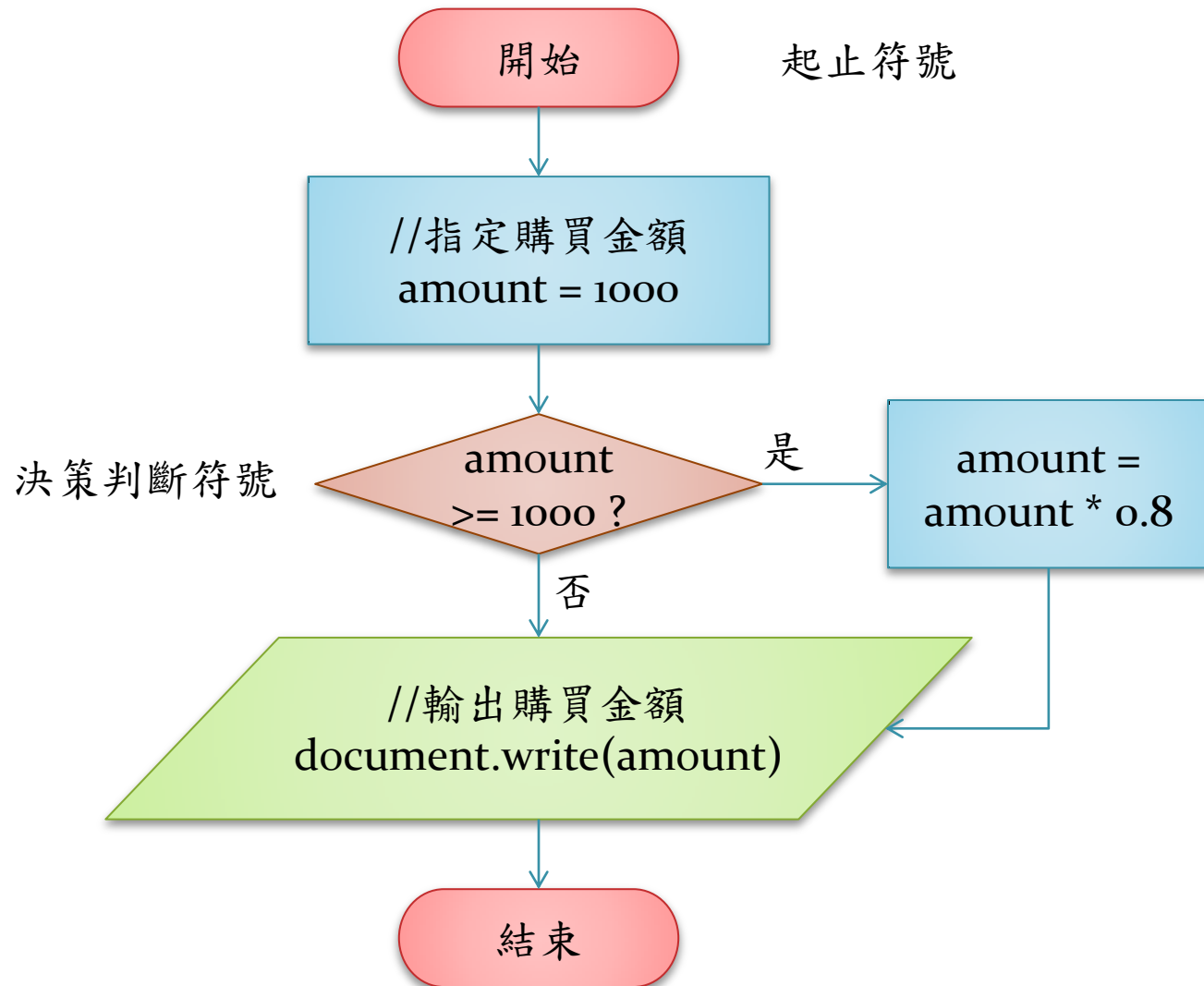


條件控制

問題

- 試寫一程式, 當購買金額超過\$1,000時, 可以打8折

流程圖



If 結構

- if 結構的語法

```
if (條件判斷) {  
    statements to be executed if the condition is true;  
} else {  
    statements  
}
```

- 條件判斷

符號	意義	範例
==	等於	amount == 1000 ; vip == true ; name == "scott"
!=	不等於	amount != 1000 ; vip != false ; name != "scott"
>	大於	amount > 1000
>=	大於等於	amount >= 1000
<	小於	amount < 1000
<=	小於等於	amount <= 1000

問題

- 試寫一程式，當購買金額超過\$1,000時，可以打8折

```
<script>
```

```
var amount = 1200;
```

```
if ( amount >= 1000 ) {
```

```
    amount = amount * 0.8;
```

```
}
```

```
document.write("應付金額 : " + amount);
```

```
</script>
```

Lab

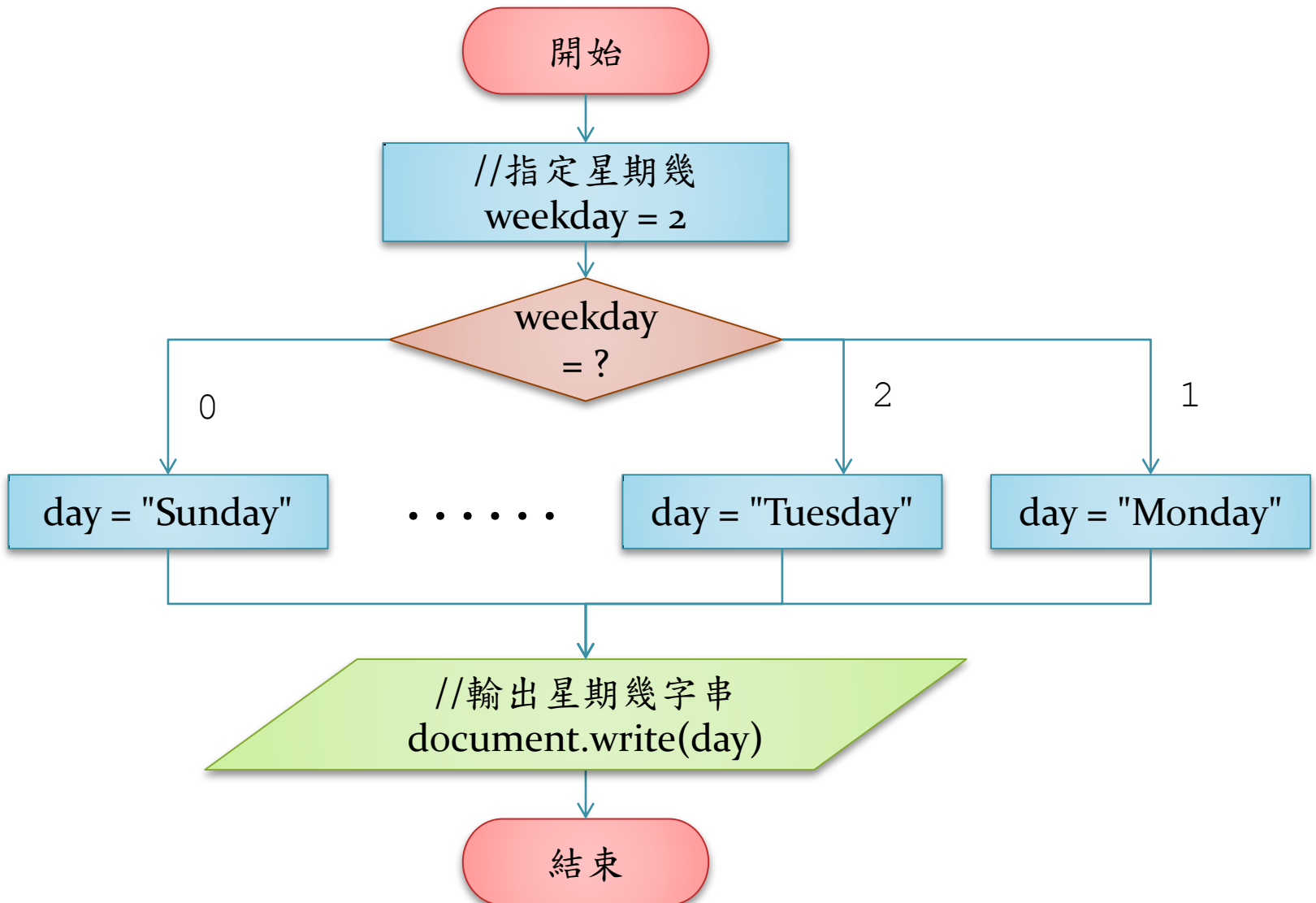
- 試寫一程式, VIP客戶可以打8折, 非VIP客戶只能打9折
- 提示
 - 設定變數 amount 儲存購買金額
 - 設定變數 vip 儲存是否VIP客戶

(練習 : js1-03 if.html)

問題

- 將星期幾的數字改為文字顯示
 - 1 --> Monday
 - 2 --> Tuesday
 - 3 --> Wednesday
 - 4 --> Thursday
 - 5 --> Friday
 - 6 --> Saturday
 - 0 --> Sunday

流程圖



switch...case 結構

- switch...case 結構的語法

```
switch (expr)          //expr通常是一個變數
{
    case value1 :
        code to run if expr equals value1;
        break;
    case value2 :
        code to run if expr equals value1;
        break;
    default :
        default code to run if expr equals no casebreak;
}
```

問題

- 將星期幾的數字改為文字顯示

```
var weekday = 2;

switch (weekday) {
    case 1 :
        day = "Monday";
        break;
    case 2 :
        day = "Tuesday";
        break;
    case 3 :
        .
        .
        .
    case 6 :
        day = "Saturday";
        break;
    default :
        day = "Sunday";
}

document.write(day);
```

Lab – Switch Case

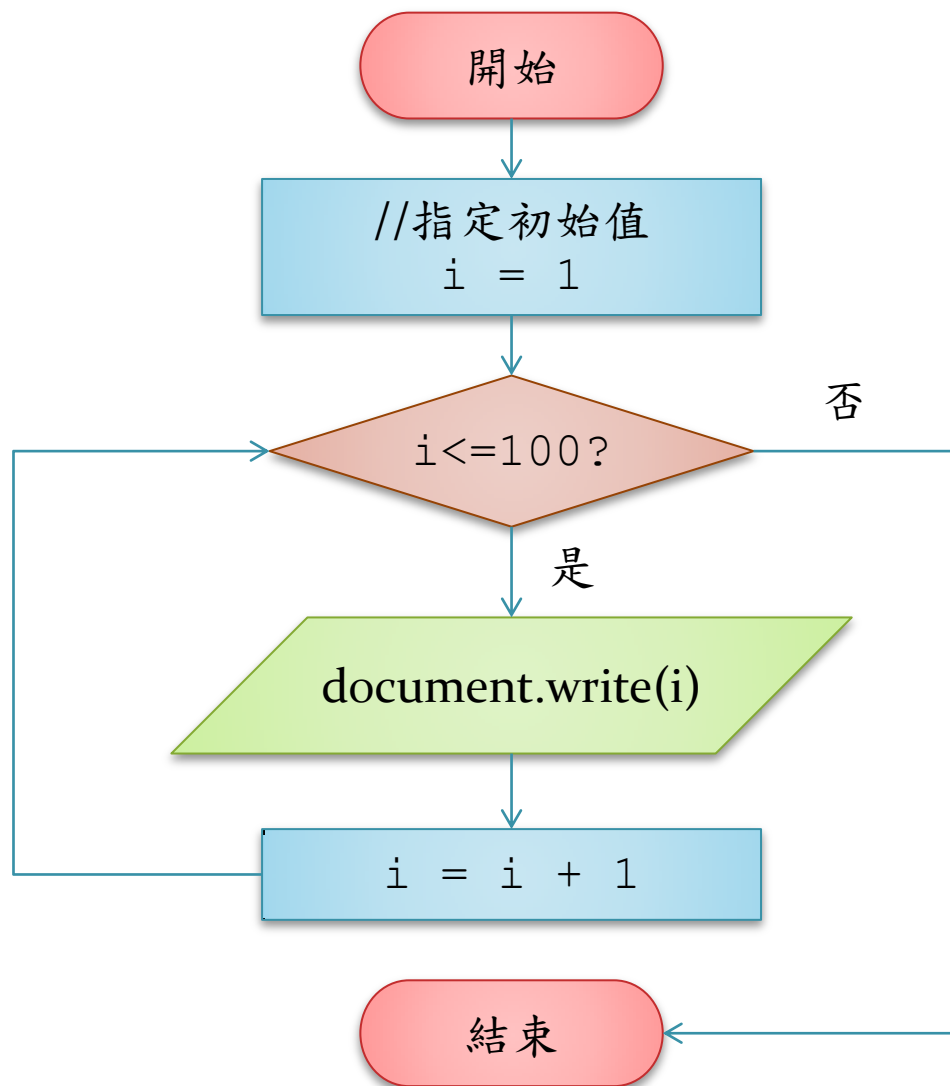
- 設計一程式, 輸入 1 ~ 3 得到票價
 - 1 : 全票 \$500
 - 2 : 優待票 – 全票打8折
 - 3 : 敬老票 – 全票打7折

(練習 : js1-04 switch.html)

問題

- 輸出 1, 2, 3, ... 100

流程圖



while迴圈

while 關鍵字



```
1 while (var<=10)
2 {
3     code to be executed
4 }
```

迴圈條件

要執行的程式碼

問題

- 輸出 1, 2, 3, ... 100

```
var i = 1;

while ( i <= 100 ) {
    document.write(i + "<br>");
    i = i + 1;
}
```

Lab - while

- 使用 while 輸出 $1+2+3+\dots+100$ 的結果

(練習 : js1-05 while.html)

do...while 迴圈

```
1  x=125;  
2  do  
3  {  
4  x=x-100;  
5  }  
6  while (x>100)
```

do 關鍵字

要執行的程式碼

while 關鍵字

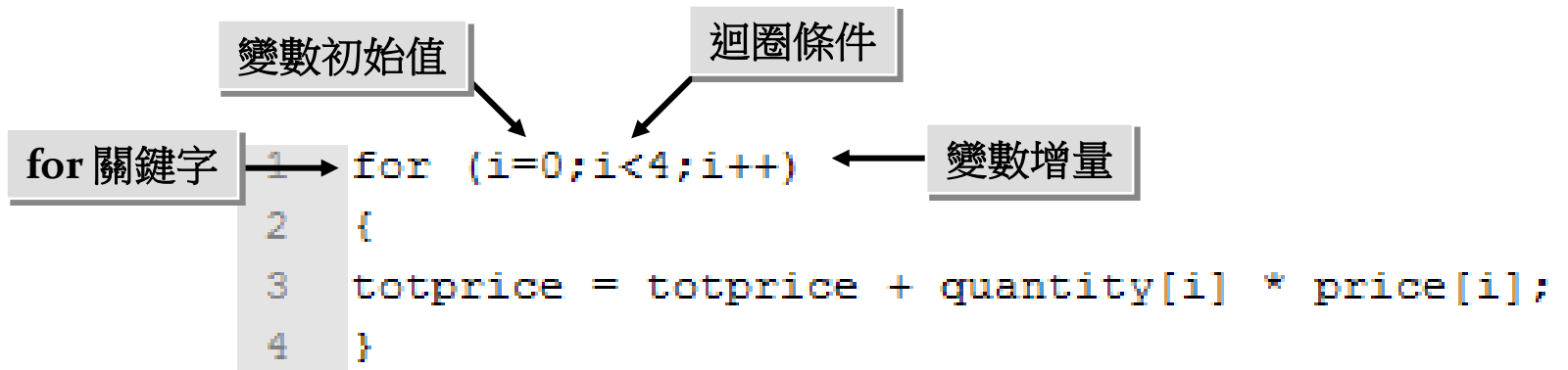
迴圈條件

Lab – do...while

- 使用 do...while 輸出 $1+2+3+\dots+100$ 的結果

(練習 : js1-06 do while.html)

for 迴圈



Lab – for

- 使用for 輸出 $1+2+3+\dots + 100$ 的結果

(練習 : js1-07 for.html)

問題

- 設計一程式, 輸出每位學生的分數, 並計算全班的總分和平均
 - `student_01 = 92`
 - `student_02 = 85`
 - `student_03 = 93`
 - `student_04 = 98`
 - `student_05 = 88`

陣列

- 陣列是一個複合的資料型態，它將相關的資料儲存在一起，並且可以用一個單一的名稱被參照到

變數 fruit

fruit_0

fruit_1

fruit_2

fruit_3

fruit_4

陣列 fruits

fruits[0]

fruits[1]

fruits[2]

fruits[3]

fruits[4]

陣列

- 宣告陣列的語法：

```
var ArrayName = new Array();
```

- 陣列透過陣列索引(index)可以存取每一個陣列元素

The diagram illustrates the process of declaring and accessing an array. It features a code block with five lines of JavaScript code. To the left of the code, a box labeled '陣列元素' (Array Element) has an arrow pointing to the second line of code. To the right, a box labeled '資料' (Data) has an arrow pointing to the fourth line. Below the code, a box labeled '索引值' (Index Value) has an arrow pointing up to the third line. The code itself is as follows:

```
1 var price = new Array();  
2 price[0] = 300.25;  
3 price[1] = 100.15;  
4 price[2] = 125.85;  
5 price[3] = 50.90;
```

問題

- 設計一程式, 輸出每位學生的分數, 並計算全班的總分和平均

```
var student = new Array(); //學生陣列
var sum = 0; //總分

student[0] = 92;
student[1] = 85;
student[2] = 93;
student[3] = 98;
student[4] = 88;

for (i=0; i <= 4; i++) {
    document.write(i + "號學生 : " + student[i] + " 分<br>");
    sum = sum + student[i];
}

document.write("總分:" + sum + "<br>");
document.write("平均:" + sum / 5 + "<br>");
```

Lab – 陣列

- 設計一程式, 計算到超市購買, 一共花了多少錢

品名	單價	數量
泡麵	35	12
可樂	20	6
麵包	40	5
冰淇淋	120	1
餅乾	25	3



總購物金額 : 935

(練習 : js1-08 array.html)

問題

- 分別算出以下結果
 - $1+2+\dots+10$
 - $1+2+\dots+50$
 - $1+2+\dots+100$

問題

- 寫一程式,分別算出以下結果

- $1+2+\dots+10$

```
for (i=1; i <= 10; i++) {  
    sum = sum + i;  
}
```

- $1+2+\dots+50$

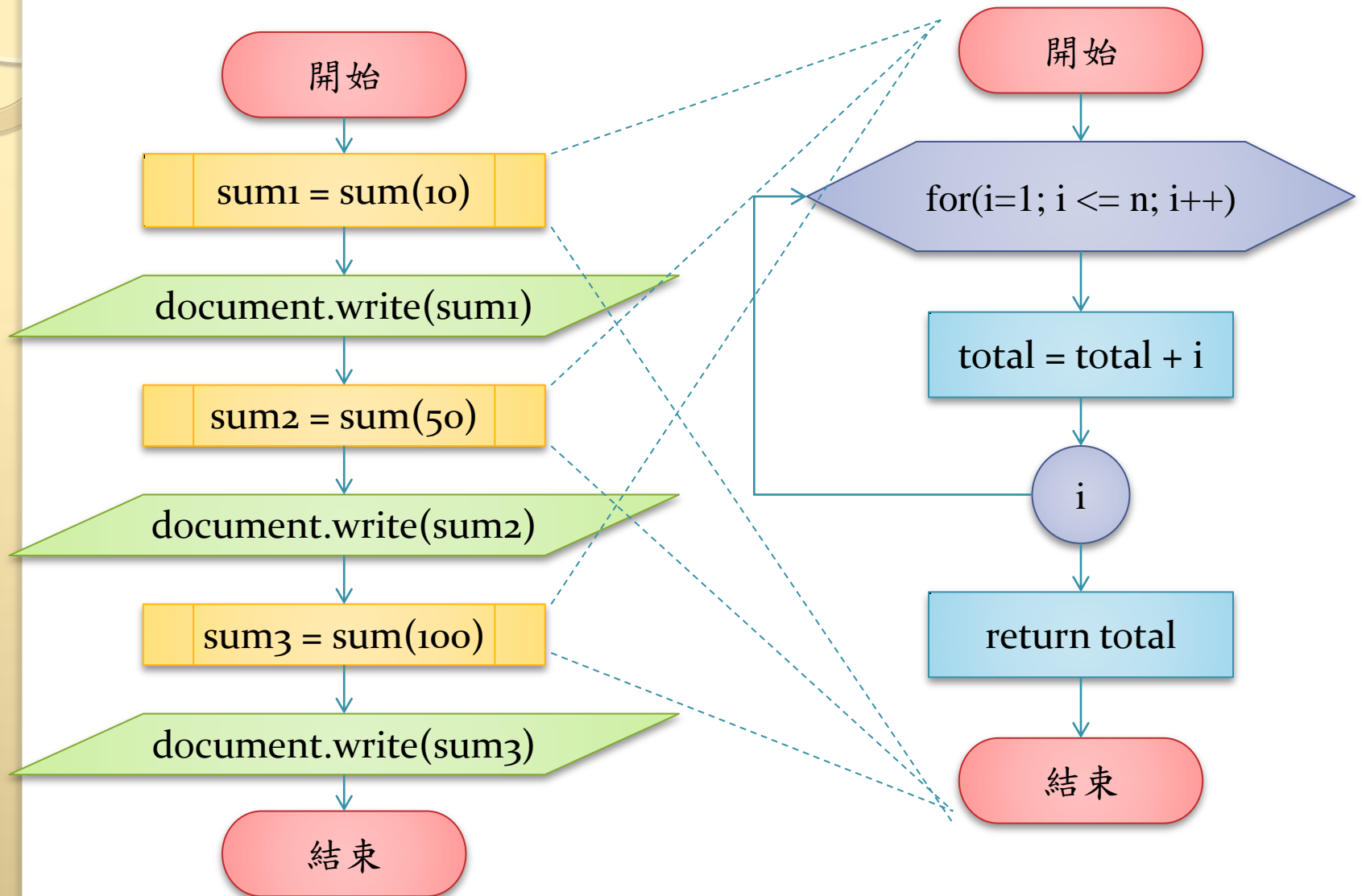
```
for (i=1; i <= 50; i++) {  
    sum = sum + i;  
}
```

- $1+2+\dots+100$

```
for (i=1; i <= 100; i++) {  
    sum = sum + i;  
}
```


流程圖

sum副程式(函數)



建立函數

- 函數 • Functions

- 將編碼集合至有意義的區段, 可被隨時參照而不必重覆撰寫相同的編碼.
- 要定義一個函數, 用關鍵字 `function` 開頭, 加上函數名稱, 小括號, 大括號.
 - 小括號內放置參數 (argument)
 - 大括號內放置程式碼
 - 程式碼中的 `return`, 可將結果回傳給主程式

function 關鍵字



```
function add(num1,num2)
{
    sum=num1+num2;
    return sum;
}
```

問題

- 寫一程式,分別算出以下結果

```
function sum(number) {  
    var total = 0;  
    for (i=1; i <= number; i++) {  
        total = total + i;  
    }  
    return total;  
}
```

```
var sum1 = sum(10);  
document.write("1+2+...+10 = " + sum1 + "<br>");  
  
var sum2 = sum(50);  
document.write("1+2+...+50 = " + sum2 + "<br>");  
  
var sum3 = sum(100);  
document.write("1+2+...+100 = " + sum3 + "<br>");
```

Lab - Function

- 從下列的數對中, 取出較大的數

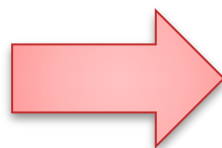
- 23, 55

- 51, 39

- 83, 42

- 27, 61

- 58, 92



$\text{Max}(23, 55) = 55$

$\text{Max}(51, 39) = 51$

$\text{Max}(83, 42) = 83$

$\text{Max}(27, 61) = 61$

$\text{Max}(58, 92) = 92$

(練習 : js1-09 function.html)

JavaScript程式撰寫位置

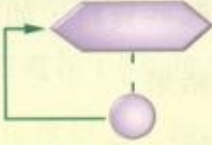
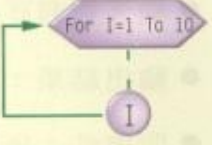





- 全域常數、變數、函數、...等，可於
 - <head>內新增<script>區段：提供該頁面使用，例

```
<head>
  <meta charset="utf-8">
  <title>Function</title>
  <script>
    function sum(number) {
      var total = 0;
      for (i=1; i <= number; i++) {
        total = total + i;
      }
      return total;
    }
  </script>
</head>
```

- <head>內宣告外部 js 檔：提供整個網站使用，例

```
<script src="global.js"></script>
```

名稱	符號	意義	範例
1.起止符號		表示程式的開始或結束	<div>開始</div> <div>結束</div>
2.流程符號		表示流程進行的方向	
3.輸入/輸出符號		表示資料之輸入或結果的輸出	<div>輸入 A 、 B 之值</div> <div>顯示出 總和</div>
4.處理符號		表示執行或處理某些工作	<div>$S = S + A$</div> <div>$A\\$ = "XYZ"$</div>
5.決策判斷符號		表示對某一個條件做判斷	
6.連接符號		用於：1.轉接到另一頁 2.避免流線交叉 3.避免流線太長	

名 稱	符 號	意 義	範 例
7.迴圈符號		表示程式迴圈控制變數初值及終值的假設	
8.副程式符號		表示一群程式步驟或流程，用以說明副程式或其他流程的組合	
9.報表符號		表示以列表機印出報表文件	
10.註解符號		表示對某一流程加以註解	