

JavaScript — 流程控制

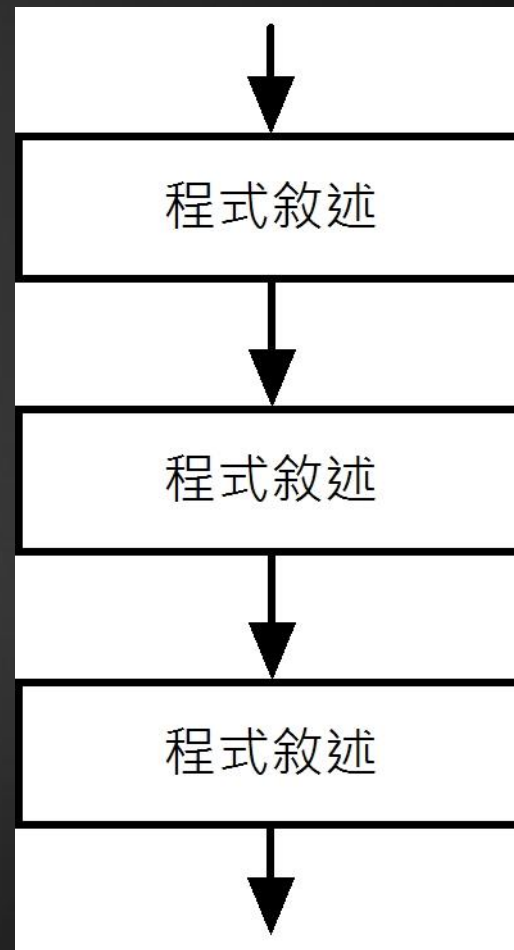
流程控制的基礎

- ▶ 基本上，JavaScript程式碼大部分是一列程式敘述接著一系列程式敘述循序的執行，但是對於複雜的工作，為了達成預期的執行結果，我們需要使用「流程控制結構」（Control Structures）來改變執行的順序。

流程控制的基礎-循序結構

循序結構 (Sequential)

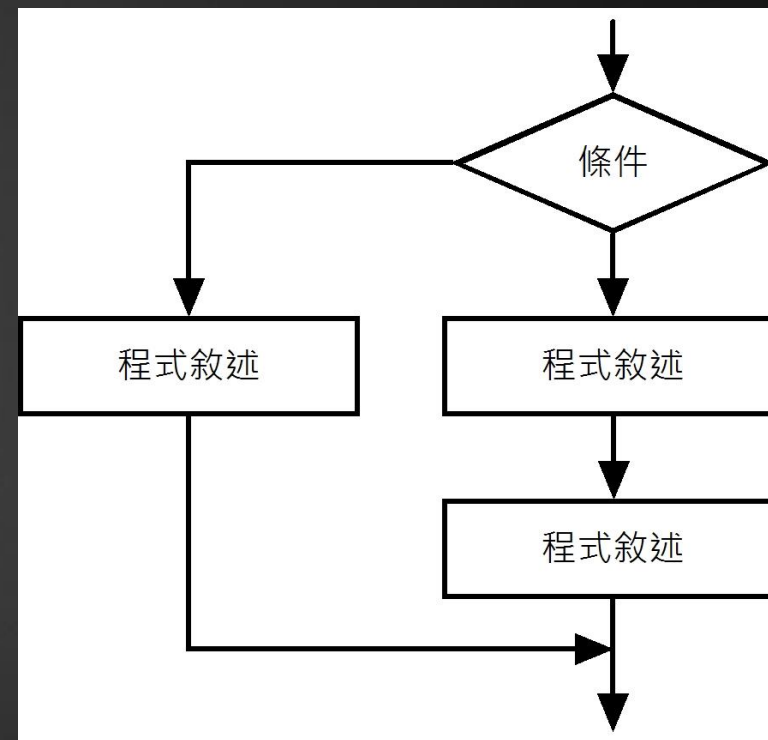
- ▶ 循序結構是程式預設的執行方式，也就是一個程式敘述接著一個程式敘述依序的執行，如右圖所示：



流程控制的基礎-選擇結構

選擇結構 (Selection)

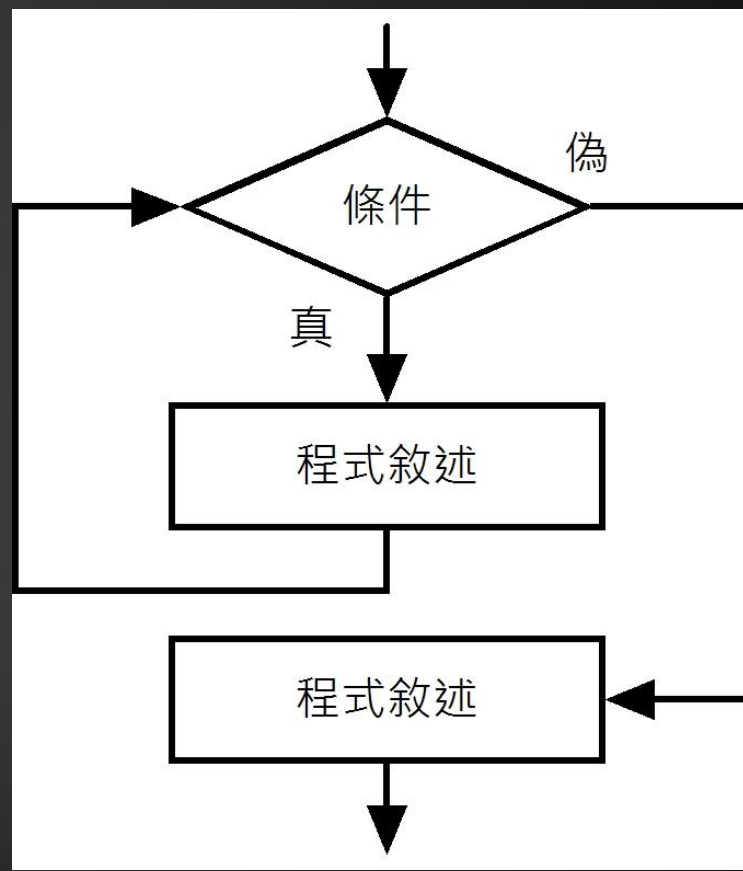
- ▶ 選擇結構是一種條件控制，這是一個選擇題，分為單一選擇、二選一或多選一共三種。程式執行順序是依照邏輯或比較運算式的條件，決定執行哪一個程式區塊的程式碼，如右圖所示：



流程控制的基礎-重複結構1

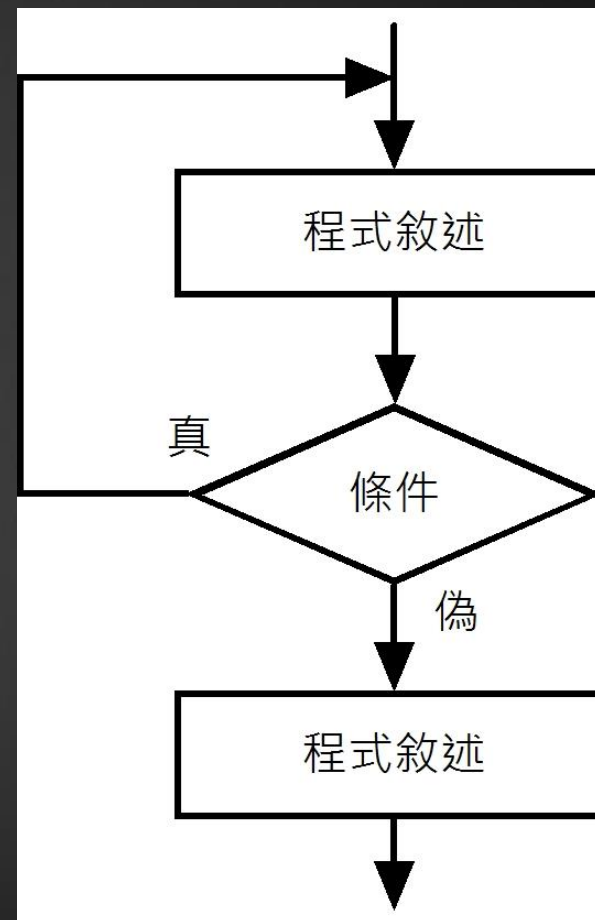
重複結構 (Iteration)

- ▶ 重複結構就是迴圈控制，可以重複執行一個程式區塊的程式碼，提供結束條件結束迴圈的執行，依結束條件測試的位置不同分為兩種，如下所示：
- ▶ 前測式重複結構：測試迴圈結束條件在程式區塊的開頭，需要符合條件，才能執行迴圈中的程式碼，如右圖所示：



流程控制的基礎-重複結構2

- ▶ 後測式重複結構：測試迴圈結束條件在程式區塊的結尾，所以迴圈的程式區塊至少會執行一次，如右圖所示：



JavaScript的流程控制

JavaScript 的流程控制分成下列兩種類型：

- ▶ 判斷結構 (decision structures) :
 - ▶ if
 - ▶ switch
- ▶ 迴圈結構 (loop structures) :
 - ▶ for
 - ▶ while
 - ▶ do
 - ▶ for...in

JavaScript的條件控制

- ▶ if是否選擇條件敘述
- ▶ if/else二選一條件敘述
- ▶ if/else多選一條件敘述
- ▶ switch多選一條件敘述
- ▶ 條件運算子?:

判斷結構 - IF 判斷式

if是否選擇條件敘述

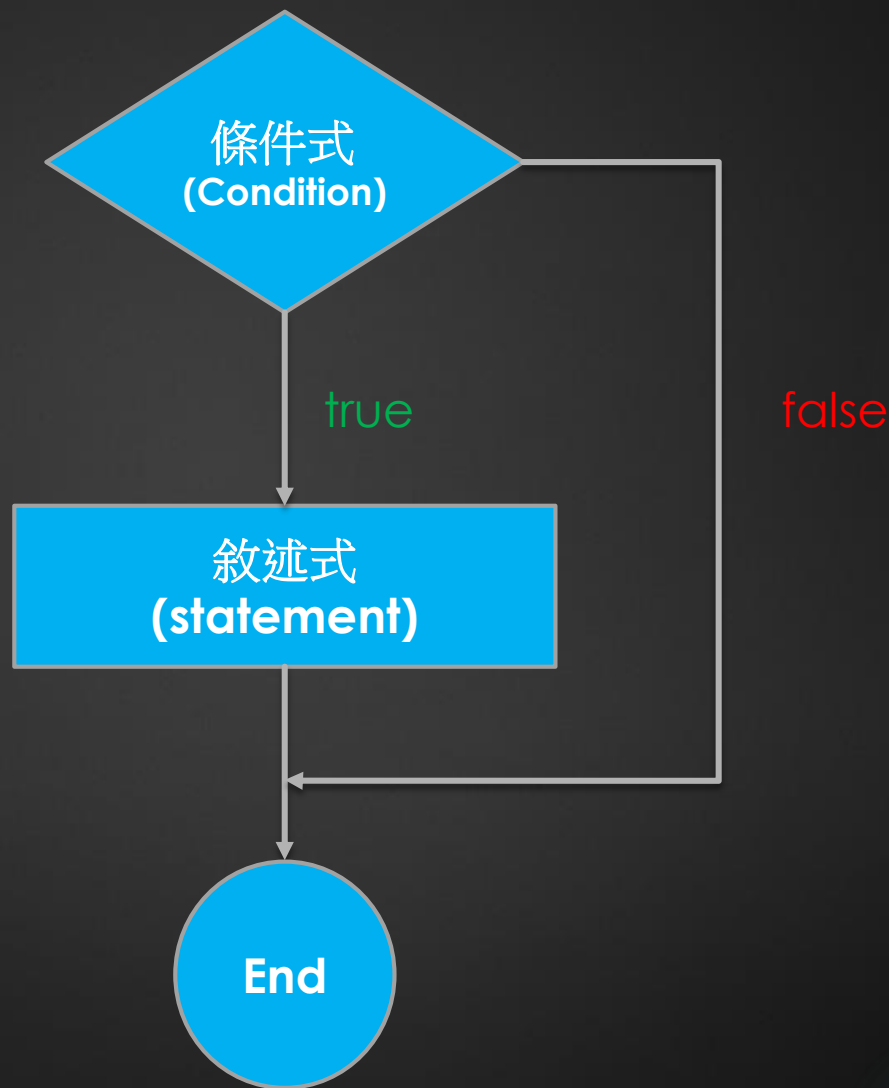
- ▶ if條件敘述是一種是否執行的單選題，可以決定是否執行程式區塊內的程式碼，如果條件運算結果為true，就執行括號之間的程式碼，如下所示：

```
if (strGender == "男"){  
    document.write("男性網友您好!");  
    document.write("歡迎使用JavaScript<br/>");  
}
```

- ▶ if條件如果為true就執行程式區塊的程式碼，如果為false就不執行程式區塊。當程式區塊的程式碼只有一列時，我們也可以省略前後的大括號「{」和「}」。

條件式

- ▶ 判斷條件成立與否，當條件成立時則執行敘述式，當不成立，則執行另一條路
- ▶ 當條件當中只有一行時，不需要加上夸號，但是當條件成立或不成立要執行的敘述式為多行時，請記得加上夸號({ })



if...：若... 就... (單向選擇)

```
if (condition) statement;
```

例如：

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> 流程控制範例 </title>
    <script language="javascript">
      var X = prompt(" 請輸入 0-100 的數字 ", "");
      if (X >= 60) alert(" 及格！ ");
    </script>
  </head>
  <body>
  </body>
</html>
```



本範例中，所使用的 **Prompt** 就是跳出視窗，讓使用者可自行輸入資訊，並將資料存於變數中

if/else二選一條件敘述

- 基本上，if條件敘述只能選擇執行或不執行程式區塊，更進一步，如果擁有兩個程式區塊，而且只能二選一，執行其中一個程式區塊，我們可以改用if條件加上else關鍵字，如果if條件為true，就執行與else之間的程式區塊，如果為false，就執行else之後的程式區塊，如下所示：

```
if (strGender == "男"){  
    document.write("男性網友您好!");  
}  
else{  
    document.write("女性網友您好!");  
}
```

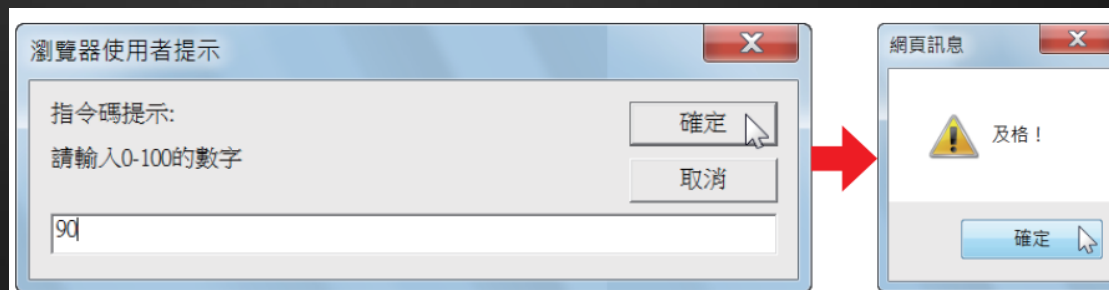
if...else...(二選一)

```
if (condition)
{
    statements1;
}
else
{
    statements2;
}
```

例如：

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> 流程控制範例 </title>
    <script language="javascript">
      var X = prompt(" 請輸入 0-100 的數字 ", "");
      if (X >= 60)
        alert(" 及格 ! ");
      else
        alert(" 不及格 ! ");
    </script>
  </head>
</html>

<\Ch12\prac12-2.html>
```



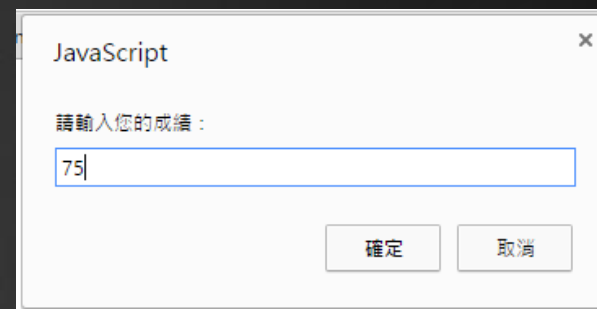
if/else多選一條件敘述

- ▶ 在JavaScript程式如果需要多選一條件敘述，也就是依照一個條件判斷來執行多個不同程式區塊之一。第一種方法是重複使用if/else條件敘述來建立多選一條件敘述，如下所示：

```
if (condition1)
{
    statements1;
}
else if (condition2)
{
    statements2;
}
else if (condition3)
{
    statements3;
}
...
else
{
    statementsN+1;
}
```

練習一 條件判斷練習(if)

- ▶ 請撰寫一個網頁，讓使用者輸入成績，當成績符合以下條件時，請在網頁上顯示對應的文字，且各等級文字的颜色不同
 - ▶ 0- 59：不及格，請再加強!
 - ▶ 60 – 69：及格，但有進步空間!
 - ▶ 70 – 79：及格，程度達到標準!
 - ▶ 80 – 89：及格，接近完美!
 - ▶ 90 – 100：及格，超強，好好保持!
- ▶ 請試著畫出這一個程式的流程圖



A JavaScript dialog box titled "JavaScript" with a close button (X) in the top right corner. The text inside says "請輸入您的成績：" (Please enter your score:). Below this is a text input field containing the number "75". At the bottom right, there are two buttons: "確定" (OK) and "取消" (Cancel).

及格，程度達到標準

switch多選一條件敘述

switch多選一條條件敘述

- ▶ JavaScript程式建立多選一條條件敘述可以使用switch多選一條條件敘述，直接依照符合條件來執行不同程式區塊的程式碼，如下所示：

```
switch(expression)
{
  case value1:
    statements1;
    break;
  case value2:
    statements2;
    break;
  ...
  case valueN:
    statementsN;
    break;
  default:
    statementsN+1;
    break;
}
```

條件運算子?:

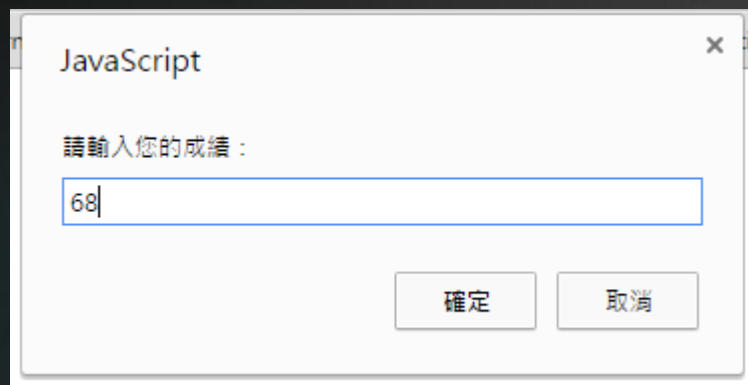
- ▶ JavaScript支援條件運算子?:，這個運算子可以用來指定變數值，條件運算子如同一個if/else條件，使用「?」符號代替if；「:」符號代替else，如下所示：

```
strHours = (dtHour >= 12) ? " PM" : " AM";
```

- ▶ 上述程式碼指定變數strHours的值，使用的是條件運算子，如果條件為true，strHours變數值為PM；false是AM。

練習二-條件判斷練習(Switch)

- ▶ 請將練習一的分數判斷，利用switch case來進行改寫



A JavaScript dialog box titled "JavaScript" with a close button (X) in the top right corner. The text inside says "請輸入您的成績：" (Please enter your score:). Below the text is a text input field containing the number "68". At the bottom of the dialog are two buttons: "確定" (OK) and "取消" (Cancel).



及格，但有進步空間!

解答：score_switch.html

條件運算子?:

- ▶ JavaScript支援條件運算子?:，這個運算子可以用來指定變數值，條件運算子如同一個if/else條件，使用「?」符號代替if；「:」符號代替else，如下所示：

```
strHours = (dtHour >= 12) ? " PM" : " AM";
```

- ▶ 上述程式碼指定變數strHours的值，使用的是條件運算子，如果條件為true，strHours變數值為PM；false是AM。

迴圈控制

JavaScript的迴圈控制

- ▶ for迴圈敘述
- ▶ for/In迴圈敘述
- ▶ while迴圈敘述
- ▶ do/while迴圈敘述

for迴圈敘述

- ▶ for迴圈可以執行固定次數的程式區塊，這個迴圈擁有計數器，計數器每次增加或減少一個值，直到迴圈結束條件成立為止，例如：使用迴圈計算1加到5的總和，每次增加1，如下所示：

```
for (i = 1; i <= 5; i++) {  
    document.write("整數: " + i + "<br/>");  
    intSum += i;  
}
```

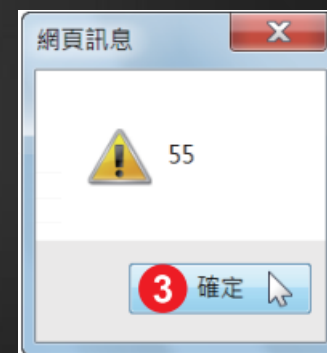
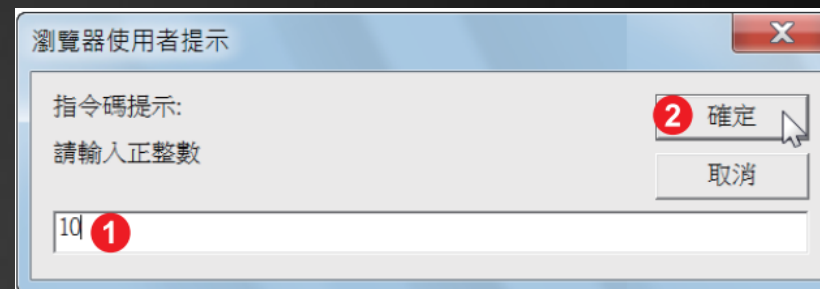
迴圈的程式碼是從1加到5計算總和。

For迴圈

```
for (initializers; expression; iterators)
{
    statements;
    [break;]
    statements;
}
```

例如：

```
<script language="javascript">
    var X = prompt(" 請輸入正整數 ", "");
    var Total = 0;
    for (var i = 1; i <= X; i++)
    {
        Total = Total + i;      // 這行敘述也可以改寫為 Total += i;
    }
    alert(Total);
</script>
```



練習- Multiplicationtable(重要)

- ▶ 請利用 for 迴圈，在網頁上顯示如下的九九乘法表
- ▶ 請利用 CSS 控制表格畫面

九九乘法表

	1	2	3	4	5	6	7	8	9
1	1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2	2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3	3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
4	4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32	4*9=36
5	5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6	6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7	7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8	8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9	9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

while迴圈敘述

- ▶ **while**迴圈敘述需要自行在程式區塊內處理計數器的增減，迴圈是在開頭檢查結束條件，如果條件為true才能夠進入迴圈；false離開迴圈，如下所示：

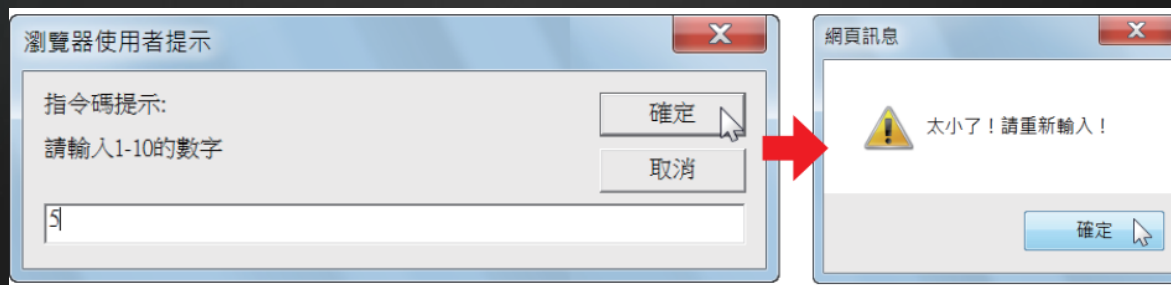
```
while(i <= 6) {  
    document.write("整數: " + i + "<br/>");  
    intSum += i;  
    i++;  
}
```

- ▶ **while**迴圈計算從1加到6的總和，只需符合條件就執行程式區塊，迴圈的結束條件為 $i > 6$ 。

while

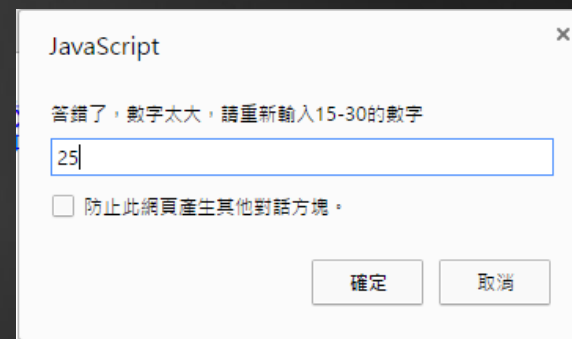
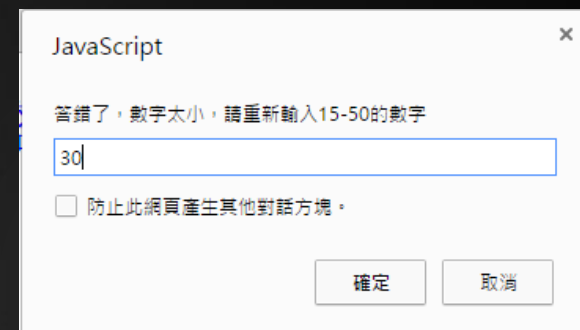
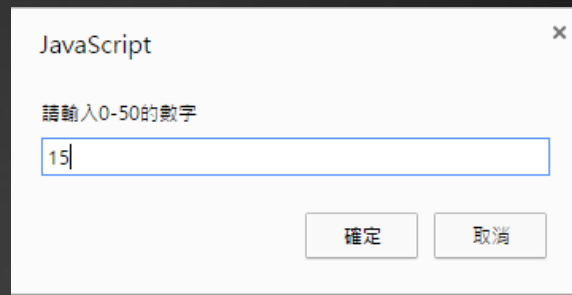
```
while(condition)
{
    statements;
    [break;]
    statements;
}
```

```
<script language="javascript">
    var Number = prompt(" 請輸入 1-10 的數字 ", "");
    while(Number != 6)
    {
        if (Number > 6)                // 太大了繼續作答
        {
            alert(" 太大了！請重新輸入！ ");
            Number = prompt(" 請輸入 1-10 的數字 ", "");
        }
        else if (Number < 6)          // 太小了繼續作答
        {
            alert(" 太小了！請重新輸入！ ");
            Number = prompt(" 請輸入 1-10 的數字 ", "");
        }
    }
    alert(" 答對了！ ");
</script>
```



練習-Guest Number(重要)

- ▶ 網頁開啟時請顯示一個彈出式訊息，讓使用者輸入數字(0 -50)
- ▶ 當使用者輸入的數字小於所猜測的數字時，會出現視窗告知數字太小
- ▶ 當使用者輸入的數字大於所猜測的數字時，會出現視窗告知數字太大
- ▶ 請注意每輸入一次，則提示訊息中的最大最小值會隨之改變
- ▶ 當輸入數字猜到時，則顯示網頁訊息，並統計猜測次數



答對了，您總共猜了3次

do/while迴圈敘述

- ▶ do/while和while迴圈敘述的不同處是在迴圈結尾檢查結束條件，因此，do/while迴圈的程式區塊至少會執行一次，如下所示：

```
do {  
    document.write("整數: " + i + "<br/>");  
    intSum += i;  
    i++;  
} while (i <= 6);
```

- ▶ 迴圈在第一次執行到迴圈結尾，才檢查while條件是否為true，如果是true就繼續執行迴圈，可以計算從1加到6的總和，迴圈的結束條件為 $i > 6$ 。

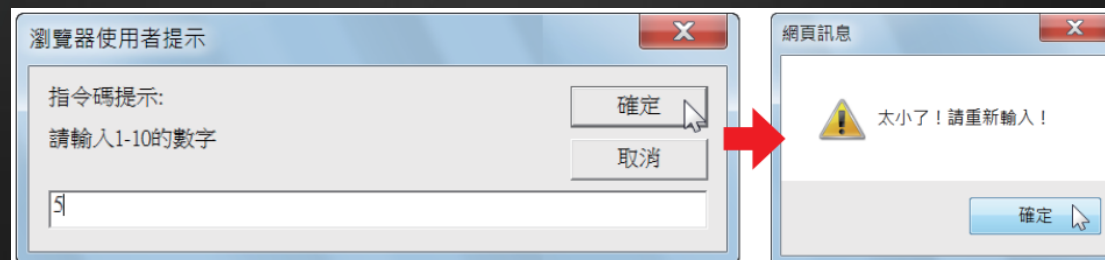
do/while迴圈敘述



```
do
{
    statements;
    [break;]
    statements;
}while(condition);
```

例如：

```
<script language="javascript">
do
{
    Number = prompt(" 請輸入 1-10 的數字 ", "");
    if (Number > 6)          // 太大了繼續做答
    {
        alert(" 太大了！請重新輸入！ ");
    }
    else if (Number < 6)     // 太小了繼續做答
    {
        alert(" 太小了！請重新輸入！ ");
    }
}while(Number != 6)
alert(" 答對了！ ");
</script>
```



for/in迴圈敘述

- ▶ for/in迴圈和for迴圈敘述十分相似，不過for/in迴圈主要是在顯示物件的所有屬性，如下所示：

```
for (prop in objAddress) {  
    document.write("屬性: " + prop + "=" + objAddress[prop] + "<br/>");  
}
```

- ▶ for/in迴圈的程式碼變數objAddress是一個物件，prop可以取得屬性名稱，這個迴圈可以取得物件的所有屬性，詳細JavaScript物件的說明請參閱後續章節說明。

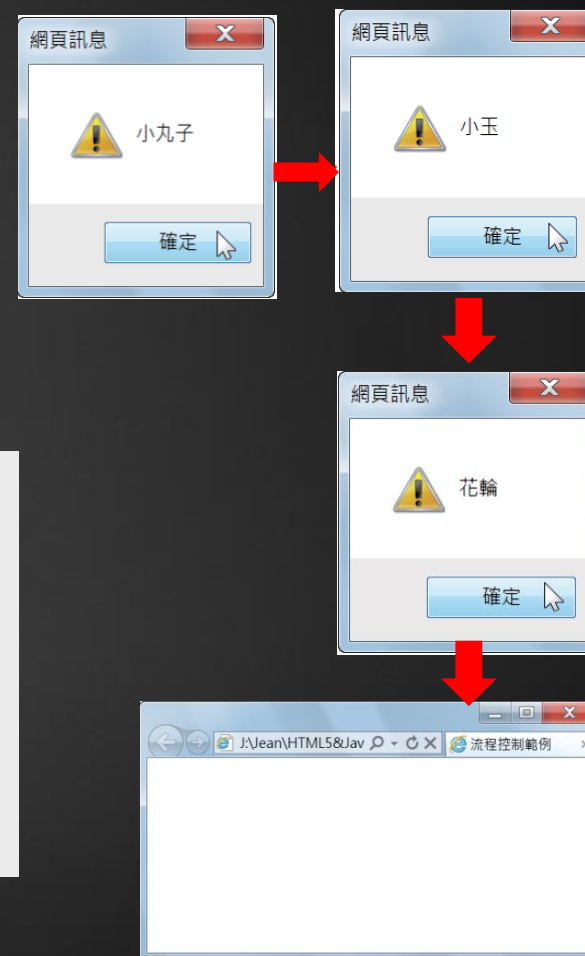
for/In迴圈敘述

[回首頁](#)

```
for (var identifier in expression)
{
    statements;
    [break;]
    statements;
}
```

例如：

```
<script language="javascript">
    // 宣告包含三個元素的陣列
    var Students = new Array("小丸子", "小玉", "花輪");
    for (var i in Students)
    {
        alert(Students[i]);
    }
</script>
```



JavaScript的巢狀迴圈

- ▶ 巢狀迴圈是指在迴圈之中擁有其他迴圈，例如：在for迴圈之中擁有for、while或do/while迴圈，同樣的，while迴圈之中也可以有for、while或do/while迴圈。
- ▶ JavaScript的巢狀迴圈可以有很多層，例如：在for迴圈之中擁有while迴圈，如下所示：

```
for (i=1;i<=9;i++) {  
    .....  
    j = 1;  
    while (j <= 9) {  
        .....  
        j++;  
    }  
}
```

- ▶ 迴圈共有兩層，第一層for迴圈執行9次，第二層的while迴圈也是執行9次，兩層巢狀迴圈共執行81次。

中斷迴圈

繼續和跳出迴圈-跳出

- ▶ JavaScript迴圈如果沒有正常結束，我們可以強迫終止迴圈的執行或繼續執行迴圈，這兩個關鍵字只能在迴圈的程式區塊之中使用，如下所示：
 - ▶ `break`關鍵字：當某些條件成立時，強迫終止迴圈的執行，如同`switch`條件使用`break`關鍵字跳出程式區塊，如下所示：

```
if (number == null || number == target)  
    break;
```

繼續和跳出迴圈

- ▶ `continue`關鍵字：可以馬上繼續下一次迴圈的執行，不過，它並不會執行程式區塊中位在`continue`關鍵字之後的程式碼，如果使用在`for`或`for/in`迴圈，一樣也會自動更新計數器變數，如下所示：

```
if (number != target) {  
    document.write(number + "太小<br/>");  
    continue;  
}
```

- ▶ 換句話說，我們可以使用`do/while`迴圈敘述建立一個無窮迴圈，然後使用`break`和`continue`關鍵字來控制迴圈的執行。

補充：isNaN()

- ▶ 還記得在data type的章節，有使用到parseInt()與parseFloat()兩個函數
- ▶ 這兩個函數可用來判斷是否為數字，若非數字則會傳回NaN的結果
- ▶ 但是如果我們用if(input="NaN")來進行判斷時，就會有判斷錯誤的狀況

```
<script>
  var number;
  var trnumber;

  number=prompt("請輸入數字：");
  trnumber=parseInt(number)

  if(trnumber="NaN")
  {
    document.write("<h1>"+number + "這不是一個數字");
  }
  else{
    document.write("<h1>"+ "<h1>這是一個數字!");
  }
</script>
```

25這不是一個數字

補充：isNaN()

- ▶ 因此在JS當中，有個函數isNaN()，可來應用，因此我們將程式進行修改如下

```
<script>
var number;
var trnumber;

number=prompt("請輸入數字：");
trnumber=parseInt(number)

if(isNaN(trnumber))
{
    document.write("<h1>" + number + "這不是一個數字");
}
else{
    document.write("<h1>" + number + "<h1>這是一個數字!");
}
</script>
```

25

這是一個數字!

綜合練習一-資料判斷練習(重要)

- ▶ 請寫出一個網頁，其內容如下
 - ▶ 網頁開啟時，請分別彈出一個對話視窗，提示使用者輸入數字，提示文字請顯示"請輸入數字A"， "請輸入數字B"
 - ▶ 請在網頁上顯示輸入時的兩個數字"型態"
 - ▶ 請判斷兩個數字的大小，顯示出兩個數字的大小(有三種狀態： $A > B$ ， $B > A$ ， $A = B$)
 - ▶ 當輸入的資料非數字時，請顯示輸入資料錯誤的訊息
 - ▶ 請將這兩個數字分別進行，轉換成整數並分別進行 加、減、乘、除的計算，將結果顯示於網頁上

綜合練習一-資料判斷練習結果畫面

- ▶ 若兩個欄位接為數字，則顯示結果畫面

數字A(25)，資料型態為：string

數字B(50)，資料型態為：string

數字B(50) 大於 數字A(25)

$25 + 50 = 75$

$25 - 50 = -25$

$25 * 50 = 1250$

$25 / 50 = 0.5$

數字A(25)，資料型態為：string

數字B(25)，資料型態為：string

數字B(25) 等於 數字A(25)

$25 + 25 = 50$

$25 - 25 = 0$

$25 * 25 = 625$

$25 / 25 = 1$

- ▶ 當輸入資料有錯誤時顯示的畫面

數字A(25)，資料型態為：string

數字B(aa)，資料型態為：string

數字B輸入錯誤 - aa

數字A(aa)，資料型態為：string

數字B(25)，資料型態為：string

數字A輸入錯誤 - aa

綜合練習二-繳稅計算機

- ▶ 請撰寫一個JavaScript程式，讓使用者輸入所得，並根據使用者所輸入的數字，計算出其所須納稅的總額

所得淨額 (元)	累進稅率
0 ~ 500,000	5%
500,001 ~ 1,130,000	12%
1,130,001 ~ 2,260,000	20%
2,260,001 ~ 4,230,000	30%
4,230,001以上	40%