第 18章 JavaScript的基本語法

資科系
林偉川

變數與常數 - 變數？常數？

- 所謂的變數（Variable）就是程式中變動的資料，在程式中，我們通常以一個名稱來代表某個變數。而常數（Constant）則是程式中不會變動的資料，用於代表一特定值。請看下面的例子。

  X = 1

2

變數與常數 -變數名稱的限制

- 對於變數名稱的使用有以下的限制。
  - 不能與JavaScript所使用的保留字相同，如：您不能宣告一個名為for的變數，因為for是JavaScript中迴圈敘述的保留字。
  - 變數名稱的第一個字元必須是英文字母，或者是下標符號『_』。如：a到z或A到Z。
  - 變數名稱的長度限制為255個字元。
  - 在同一個變數可見範圍中，變數名稱必須是唯一的。
  - 變數中不能有空白出現。

3

變數與常數 - 變數名稱的限制

- 以下是JavaScript中的保留用字。

abstract	boolcan	break	byte	case	catch
char	class	const	continue	default	do
double	else	extends	false	final	finally
float	for	function	goto	if	implements
import	in	instanceof	int	interface	long
native	new	null	package	private	protected
public	return	short	static	super	switch
synchronized	this	throw	throws	transient	true
Try	var	void	while	with	

4

變數與常數 - 變數名稱的限制

- 下面是一些錯誤的變數名稱。
  - 7AB ‘必須用英文字母做為變數名稱的開頭
  - AB* ‘變數名稱中，不可以使用*特殊符號
  - L!Ay ‘變數名稱中，不可以使用!特殊符號
  - var ’var是VBScript的保留字，不可以做為變數名稱

5

變數與常數 -變數的宣告

- 在大部份的程式語言裡，程式中若欲使用某變數時，必須經過變數宣告的動作。在JavaScript裡，宣告變數時，必須使用var敘述來宣告，語法如下：

```
var A=10
//同時完成宣告A變數及指定A變數的值等於10
var A //宣告變數A
A=10 //指定A變數值等於10
```

6

變數與常數 -變數的宣告

- 還可以在同一行做多變數的宣告，而每個變數之間只要用逗號(,)隔開即可，如下例所示：

```
var A,B,C
var A,B,C=10
```

7

變數的資料型態

- 在JavaScript內比較常見的資料型態，有以下五種：
  - 字串
  - 數值
  - 布林值
  - Null值
  - 未定義值

8

變數的資料型態

- string 字串

字串就是由一連串的字元所組合而成，舉凡英文字母、中文字、數字以及符號，只要是顯示在網頁上的字，都可以算是字串。在JavaScript中，字串值的前後必須利用雙引號『“』或單引號『‘』標示，如下所示：

“這是一個字串”或’這是一個字串’

若將一字串值，設定給變數時，其語法如下：

str=“這是一個字串”或 str=’這是一個字串’

9

變數的資料型態

- numbers 數值

整數部份又可分為8進位、10進位與16進位，8進位將以0開頭，如：05、015）；10進位即以數字開頭，如：5、15；16進位則以0X開頭，如：0xff、0xAE。

若具有小數點符號的數值，則為浮點數，如：5.5、24.57、-37；或者以科學浮號E表示（2.3E-2、8E90）。

10

變數的資料型態

- boolean 布林數值

  布林數值共有兩個值，true(真)/false(假)，用於儲存比較資料的結果。欲使用boolean值時，必須注意英文字母的大小寫。

- Null 沒有數值

  這是用一特殊的關鍵字來指定，主要代表空的字串。

- UNDEFINED 未定義值

  在建立變數後（即var一個變數後），未指定變數值之前所提供的值。

11

變數的資料型態

- 特殊字元除了上述資料型態外，在JavaScript中還有一些特殊字元，可供使用者使用。但在進行網頁設計時，比較有用的是『\“』，可在字串中顯示『”』字元。如下所示：

```
document.write("\"大家好\"")
```

上述程式碼將在HTML網頁中輸出如下的字串。

```
"大家好"
```

12

運算子-運算子與算式

- 什麼是運算子

  運算子為程式中用於執行計算動作的符號。在JavaScript中，依照所執行的計算動作，可以將運算子歸類如下：

  ■算術運算子
  ■比較運算子
  ■邏輯運算子
  ■字串運算子
  ■指派運算子

13

運算子-算術運算子

- 算術運算子與運用語法整理如下表：

運算子符號	運算子名稱	使用語法範例	當A=7,B=2時使用語法範例的運算結果
+	相加	A + B	7 + 2 = 9
-	相減	A - B	7 - 2 = 5
*	相乘	A * B	7 * 2 = 14
/	相除	A / B	7 / 2 = 3.5
-	取負數	-A	-7
%	取餘數	A % B	7 % 2 =1
++	遞增	A++	先傳回值，再加1
		++A	先加1，再傳回值
--	遞減	A--	先傳回值，再減1
		--A	先減1，再傳回值

14

運算子-字串連接運算子

- 字串的類型

  在JavaScript中，字串的型態為文字型字串，文字型字串就是一般的字串。在前一小節曾提過『+』運算子，可以運用於數值的相加，但在此它還可以執行字串連結的動作，例如：

```
"ABC" + "DEF"   //結果為ABCDEF
"ABC" + "123"   //結果為ABC123
"123" + 123      //結果為123123
```

15

運算子-比較運算子

- 主要是比較運算子左右兩個運算元之間的關係，若比較結果是正確的則傳回真（true）；若比較結果是錯誤的則傳回假（false）。JavaScript的比較運算子整理如下表：

運算子符號	名　稱	使用語法範例	當A=7,B=2時的計算結果
==	等於	A == B	false
>	大於	A > B	true
<	小於	A < B	false
>=	大於等於	A >= B	true
<=	小於等於	A <= B	false
!=	不等於	A != B	ture

16

運算子-邏輯運算子

•JavaScript中的邏輯運算子整理如下：

運算子符號	名　稱	使用語法範例	說明
&&	且	A && B	當A、B均為真時，結果為真
||	或	A || B	當A、B其中之一為真時為真
!	反	! A	當A為真時為假

17

邏輯運算子

• 下表稱為真值表，T（True）代表『真』，F（False）代表『假』。下表為A、B在各種情形下，各邏輯運算子的運算結果：

A	B	A && B	A || B	! A
T	T	T	T	F
F	T	F	T	T
F	F	F	F	T
T	F	F	T	F

18

運算子-算術指派運算子

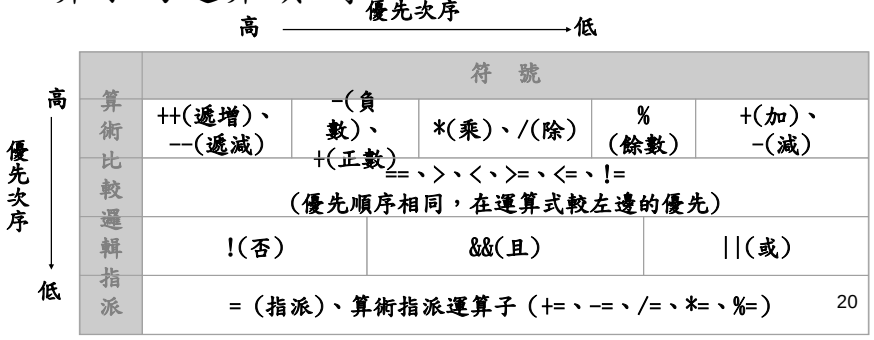
- JavaScript提供了一些由算術運算子與指派運算子結合而成的算術指派運算子，可用於結合運算與指派動作，以簡化運算式的撰寫，以下是各種算術指派運算子的使用範例。

運算子	範例	相當於
+=	A = A + 3	A += 3
-=	A = A - 3	A -= 3
/=	A = A / 3	A /= 3
*=	A = A * 3	A *= 3
%=	A = A % 3	A %= 3

19

運算子-各種運算子的計算順序

- 所謂運算子的計算順序觀念，即若遇到處理順序相同的，則以左邊的運算子優先。這個概念其實就是數學中先乘除後加減的觀念，下表將表達出JavaScript中各類運算子的運算順 序。



	符 號				
算術	++(遞增)、--(遞減)	-(負數)、+(正數)	*(乘)、/(除)	%(餘數)	+(加)、-(減)
比較	==、>、<、>=、<=、!= (優先順序相同，在運算式較左邊的優先)				
邏輯	!(否)	&&(且)		||(或)	
指派	=(指派)、算術指派運算子(+=、-=、/=、*=、%=)				

20

運算子-各種運算子的運算順序

- 當電腦執行到一行運算式時，將會先執行遞增或遞減運算子，然後再處理其他的運算子。若有括號時，則優先計算括號內的算式。以下的例子，將說明運算子的優先處理順序。 ➔ 5*2 > 6 && 7+8 < 8

首先，電腦將先完成上式內算術運算子的運算，計算出5*2=10與7+8=15，如下所示：

10 > 6 && 15 < 8 ➔ true && false

最後得到結果為false

21

各種運算子的計算順序

我們再舉一個例子：

A = 5 *(2 * 2)+ 6 / 2

第一步先計算括號內的算式（2*2=4），得到下面的結果。

A = 5 * 4 + 6 / 2

再計算乘除，得到下面的結果。

A = 20 + 3

最後得到 A = 23

22

流程控制-流程控制的用途與種類

- 流程控制是程式中，用於控制或選擇某一程式區段執行方式的語法。流程控制分為兩類，一是判斷敘述，另一是迴圈。
  - 判斷敘述是利用條件式，進而決定要執行哪一個程式區段。
  - 迴圈則是配合條件式，控制某程式區段的重複執行。

23

流程控制-流程控制的用途與種類

- 這些流程控制的語法中，主要是由條件式與特定保留字所構成。其中條件式用於傳回『真』(true)或『假』(false)值，做為程式決定是否執行某段程式的依據。
- 保留字則將以一定的規則，將條件式與欲控制執行的敘述包含在一起。
- 當程式執行至此處時，依照條件式傳回的『真』、『假』值，來決定接下來該如何執行程式。下面將分別說明判斷敘述與迴圈控制的語法。

24

流程控制-流程控制的用途與種類

- 判斷結構語法

判斷結構最基本類型的語法如下所示：

```
if（條件式）
執行程式敘述;
```

若程式敘述不止一行時，則需用『{…}』，標 示這些 程式敘述。

```
if（條件式）
{ 程式敘述　　…… }
```

25

流程控制-流程控制的用途與種類

- 迴圈控制語法

迴圈控制語法如下所示：

```
while（條件式）
{　程式敘述　　……　　}
```

若程式敘述只有一行時，可省略「{}與『』」符號。當while後的條件式回傳值為true時，則執行程式敘述，直到條件式回傳值為false時，才跳出迴圈。

26

流程控制-if判斷敘述

- if判斷結構依照功能可以分為以下兩類：
  - 單一條件判斷敘述
  - 利用一條件式控制程式是否執行某程式敘述，或由兩程式敘述中擇一執行。此判斷敘述將利用if或if…else…建立。
  - 多條件判斷敘述：
    - 利用多種條件控制程式所執行的敘述，此判斷敘述將以if…else if...else…建立。

27

流程控制-if判斷敘述

- 單一條件判斷敘述有以下兩種語法：

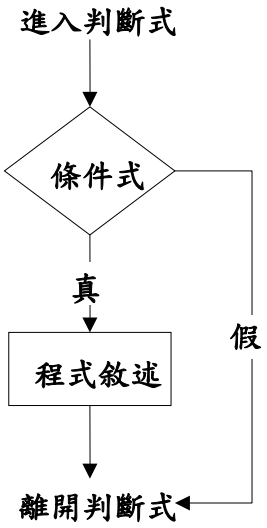
語法一：（控制單一程式敘述）

```
if（條件式）
{
程式敘述
……
  }
```

語法一的意義為，如果（if）條件式為true，則 執行程式敘述。其執行流程圖如下頁：

28

流程控制-if判斷敘述



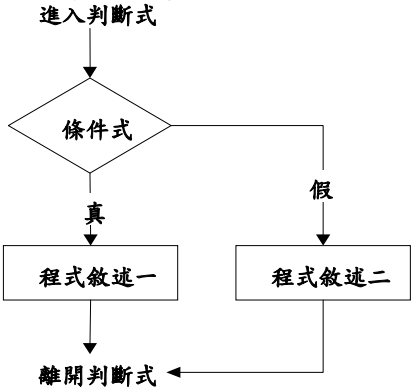
流程控制-if判斷敘述

- 語法二：（兩程式敘述擇一執行）

```
if（條件式）
{
   程式敘述一
    …...
}
else
{
   程式敘述二
   ……
}
```

流程控制-if判斷敘述

- 語法二的意義為，如果（if）條件式為true，則執行程式敘述一，否則（else）執行程式敘述二，其執行流程圖如下：



31

流程控制-if判斷敘述

- 多條件判斷敘述

當控制程式執行的判斷條件不止一個時，就必須利用if…else if…else建立流程控制，語法如下：

```
if（條件式A）
{   程式敘述一   …… }
else if（條件式B）
{
      程式敘述二
      ……
}
```

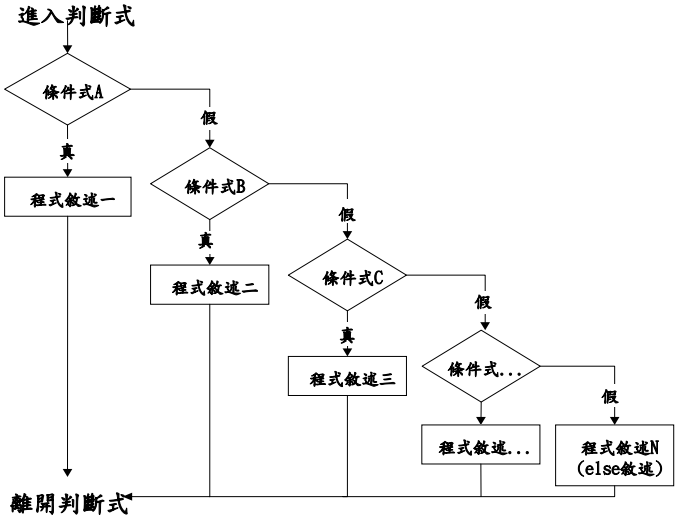
32

流程控制-if判斷敘述

```
else if…..
{
    ……
}
else
{
     程式敘述N
}
```

若都不符合所有條件式，則執行else後的程式敘述N，其執行流程圖如下頁：

流程控制-if判斷敘述



流程控制-if判斷敘述

- 當在判斷式的程式敘述中，又利用一個判斷敘述進行判斷，此時便有兩層判斷敘述，這就是所謂的巢狀判斷敘述。

語法如下：

```
if (條件式A)
{
   if (條件式B)
       程式敘述一;
   else
       程式敘述二;
}
```

35

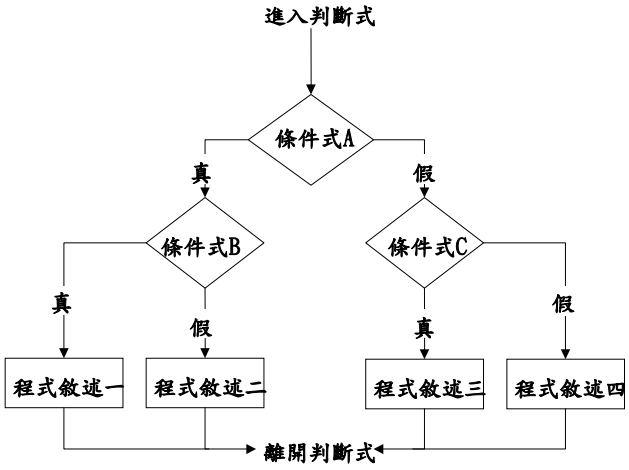
流程控制-if判斷敘述

```
else
{
if (條件式C)
       程式敘述三
else
       程式敘述四;
}
```

下頁是整個巢狀判斷敘述的執行流程。

36

流程控制-if判斷敘述



37

流程控制-switch case判斷敘述

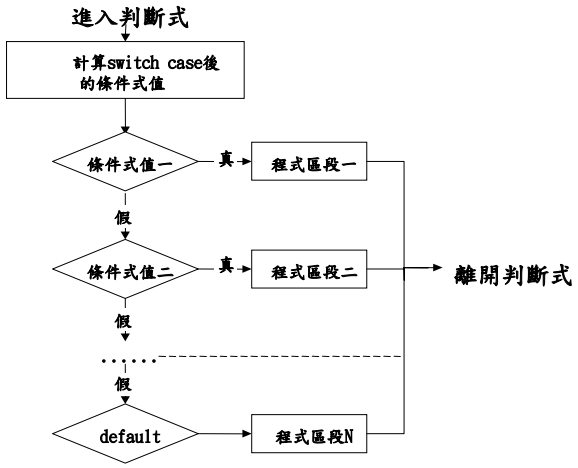
- witch case判斷敘述用於針對某運算式的不同值，進行條件的判斷，然後從多個程式敘述中，選擇執行其中某一程式敘述，語法如下：

```
switch（運算式或變數）{
case 條件式一：程式敘述一; break;
case 條件式二：程式敘述二; break;
case … :
…
break;
default:
        程式敘述N; break;
}
```

38

流程控制-switch case判斷敘述

- switch…case其執行流程如下所示。



流程控制-for迴圈

- 迴圈語法如下：

```
for( 起始運算式; 測試條件式 ; 遞增運算式 )
{
        程式敘述
        ……
}
```

流程控制-for迴圈

- 語法各部份的說明如下：
  - ■起始運算式
  - ■用於設定計次變數起始值，起始運算式只在進 入迴圈的時候執行。
  - ■測試條件式
  - ■計次變數的限制條件式，若為真，則進入迴圈 執行程式敘述，若為假則跳出迴圈。
  - ■遞增運算式
  - ■執行完程式敘述後，利用遞增運算式更改計次變數。

41

流程控制-for迴圈

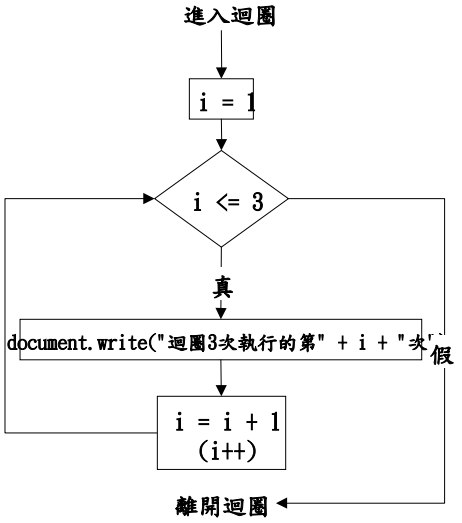
•下面是一個將執行三次程式敘述的for迴圈。

```
for( i = 1; i <= 3; i = i + 1)
{
 document.write("迴圈3次執行的第" + i + "次")
}
```

其中遞增運算式『i = i + 1』，亦可以i++代替。 執行流程如下頁所示。

42

流程控制-for迴圈



流程控制-for迴圈

• 在上述迴圈的執行過程中，我們將各種計次變數值的變化，以表格表示如下：

執行次數	進入時的i值	輸出	執行後的i值
第一次	1	迴圈3次執行的第1次	2
第二次	2	迴圈3次執行的第2次	3
第三次	3	迴圈3次執行的第3次	4
第四次	4		

流程控制-for迴圈

- 巢狀for迴圈

  r迴圈中，還有另一個for迴圈時，就形成了巢 狀 的for 迴圈，如下所示。

```
for( i = 0; i < 10 ; i++ )
{
  for( j = 0; j < 10; j++)
  {
    程式敘述
    ……
  }
}
```

45

流程控制-while迴圈

- while迴圈語法

  while迴圈的語法如下：

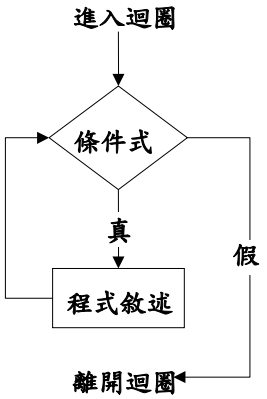
```
while(條件式)
    {
    程式敘述
    ……
 }
```

在進入此迴圈時，將先檢查條件式的值 是否為 true，真時則進入迴圈，false時則跳過此迴圈。

46

流程控制-while迴圈

- while迴圈的執行流程如下圖所示。



47

流程控制-break與continue

- break敘述

  在for迴圈或者while迴圈中，如果您想在某條件下，強迫中止迴圈的執行時，可以利用if敘述，再配合break敘述達到目的。

48

break與continue

- continue敘述

  如果想要中止的不是整個迴圈，而只是想在某條件下，中止某次迴圈的執行。讓該次迴圈的執行中，能跳過continue敘述後的程式敘述，直接進入下一次迴圈的執行。您便可利用if敘述，配合continue敘述，在某條件下，中止某次迴圈的執行。

49