



Chapter 2

JavaScript的變數與運算子

中原大學 資訊管理學系

賴錦慧 老師

chlai@cycu.edu.tw

大綱

- JavaScript的變數
- JavaScript的資料型態
- JavaScript的運算子
- 資料型態的轉換

JavaScript的變數



- 變數命名與宣告
- 指定敘述
- JavaScript的變數是否存在

變數命名



□ JavaScript變數的命名原則

- 有區分英文字母大小寫
- 不能使用JavaScript語法的保留字，即關鍵字。
- 開始字元必須為英文字母的大小寫或「_」字元，而且不能使用數字開頭。
- 變數名稱除開頭字元外，可以是英文字母、數字和「_」符號
- 不能使用句點「.」

變數宣告

- JavaScript程式是使用【var】指令宣告變數

```
var strName;
```

- 如果需要同時宣告多個變數，請使用「,」分隔：

```
var strName, intBalance;
```

- 宣告變數的同時指定初始值

```
var strName = "陳會安";
```

```
var intBalance = 1000;
```

```
var blnSex = true;
```

指定敘述

- 指定敘述「=」等號來指定變數值和資料型態：
`strName = "陳會安";`
`intBalance = 1000;`
- 可以再次使用指定敘述更改變數成其他值
`intBalance = "1000";`
- JavaScript變數只是一個暫存資料的容器，變數宣告只是聲明程式碼需要一個變數的容器
- 使用指定敘述隨時更改其資料型態。

JavaScript的變數是否存在



- JavaScript程式碼的變數需要使用var宣告或指定敘述來隱藏宣告，對於一個變數，程式碼如何知道它是否**存在**
 - "存在" → 指變數擁有值，而不是undefined資料型態
- if條件敘述可檢查變數是否存在
 - JavaScript宣告或使用的變數都屬於Window物件的屬性

if (window.strName)

document.write("strName存在:" + window.strName + "
");

else

document.write("strName不存在:" + window.strName + "
");

資料型態-數值

數值資料型態 (Number Data Type)

□ JavaScript數值資料型態的變數值可以是整數或浮點數

□ 整數值

- 整數值包含0、正整數和負整數
- 使用十進位、八進位和十六進位表示

□ 浮點數值

- 浮點數是整數加上小數
- 範圍最大為 $\pm 1.7976931348623157 \times E308$ ，最小為 $\pm 5 \times E-324$
- 使用「e」或「E」符號代表10為底的指數

資料型態-字串



字串資料型態 (String Data Type)

- 字串可以包含0或多個Unicode字元
- JavaScript程式碼的字串需要使用「 ” 」或「 ’ 」符號括起。
 - "陳會安"是本書的作者'
 - "I'm an author."
- JavaScript不支援單一字元的函數，例如：
Visual Basic或C/C++語言的chr()函數

其他資料型態

布林資料型態 (Boolean Data Type)

- 布林資料型態只有兩個值true和false
- 用在條件和迴圈控制的條件判斷

Null資料型態

- Null資料型態只有一個值null
- 表示變數沒有值或不是一個物件。

Undefined資料型態

- 指一個變數有宣告，但是不曾指定變數值
- 一個物件屬性根本不存在。

Escape 逸出字元

- JavaScript 提供 Escape 逸出字元(「\」符號開頭)
- 在字串資料型態的變數值中顯示無法使用鍵盤輸入的特殊字元

Escape 逸出字元	說明
\b	Backspace , Backspace 鍵
\f	FF , Form feed
\n	LF , Line feed 換行符號
\r	CR , Enter 鍵
\t	Tab 鍵
\'	「'」符號
\"	「"」符號
\\	「\」符號

JavaScript的運算子



- JavaScript指定敘述的運算式都是由運算子和運算元組成

$a + b - 1$

$a \geq b$

$a > b \ \&\& \ a > 1$

- 變數a、b和數值1是運算元

- 「+」、「-」、「>=」、「>」和「&&」是運算子。

運算子的優先順序

- 同一個運算式允許使用多種運算子，為了讓運算式能夠得到相同的運算結果
- 「先乘除後加減」

$a + b * 2$

運算子的優先順序

運算子	說明
()	括號
!、-、++、--	邏輯運算子NOT、算數運算子負號、遞增和遞減
*、/、%	算術運算子的乘、除法和餘數
+、-	算術運算子加和減法
<<、>>、>>>	位元運算子左移、右移和無符號右移
>、>=、<、<=	比較運算子大於、大於等於、小於和小於等於
==、!=	比較運算子等於和不等於
&	位元運算子AND
^	位元運算子XOR
	位元運算子OR
&&	邏輯運算子AND
	邏輯運算子OR
?:	條件運算子
=、op=	指定運算子

算術運算子

- 大部分運算元是數值
- 加法運算子可以連接兩個字串變數。
- 變數`a=10`

運算子	說明	運算式範例
-	負號	-7
++	遞增運算	a++ = 11
--	遞減運算	a-- = 9
*	乘法	5 * 6 = 30
/	除法	7 / 2 = 3.5
%	餘數	7 % 2 = 1
+	加法或字串連接	4 + 3 = 7
-	減法	4 - 3 = 1

算術運算子-遞增和遞減運算



- 遞增和遞減運算++和--可以置於變數之前或之後

```
x++;
```

```
--yy;
```

- X= ? Y=?

```
x = 10;
```

```
y = 10;
```

```
document.write("x++ = " +x+++":x = " + x + "<br/>");
```

```
document.write("--y = " +--y+":y = " + y + "<br/>");
```


比較運算子

- 使用於迴圈和條件敘述的判斷條件
- `true` 為真；`false` 為假。

運算子	說明	運算式範例	運算結果
<code>==</code>	等於	<code>6 = 3</code>	<code>false</code>
<code>!=</code>	不等於	<code>6 != 3</code>	<code>true</code>
<code><</code>	小於	<code>6 < 3</code>	<code>false</code>
<code>></code>	大於	<code>6 > 3</code>	<code>true</code>
<code><=</code>	小於等於	<code>6 <= 3</code>	<code>false</code>
<code>>=</code>	大於等於	<code>6 >= 3</code>	<code>true</code>

邏輯運算子



- 如果條件不只一個，就需要使用邏輯運算子連接各比較運算式

運算子	說明
!	NOT非，傳回運算元相反的值，true成false，false成true
&&	AND且，連接的兩個運算元都為true，運算式為true
	OR或，連接的兩個運算元，任一個為ture，運算式為true

位元運算子



運算子	A	B	C	D	範例	結果	說明
~	1(01)				~A	-2(10)	NOT運算
<<			3(11)		C<< 2	12(1100)	左移運算
>>		2(10)			B >> 1	1(1)	右移運算
>>>				16(1000)	D >>> 1	8(0100)	無符號右移
&	1(01)		3(11)		A & C	1(01)	AND運算
^	1(01)	2(10)			A ^ B	3(11)	XOR運算
	1(01)	2(10)			A B	3(11)	OR運算

指定運算子



運算子	範例	相當的運算式	說明
=	x = y	N/A	指定敘述
+=	x += y	x = x + y	數值相加或字串連接
-=	x -= y	x = x - y	減法
*=	x *= y	x = x * y	乘法
/=	x /= y	x = x / y	除法
%=	x %= y	x = x % y	餘數
<<=	x <<= y	x = x << y	位元左移y位元
>>=	x >>= y	x = x >> y	位元右移y位元
>>>=	x >>>= y	x = x >>> y	無符號右移y位元
&=	x &= y	x = x & y	位元AND運算
=	x = y	x = x y	位元OR運算
^=	x ^= y	x = x ^ y	位元XOR運算



資料型態的強制轉換

- JavaScript 運算式的運算元需要相同型態
- JavaScript 在執行運算時會自動進行資料型態的強迫轉換

運算式	型態強制轉換的處理
數值和字串相加	數值會強制轉換成字串
布林和字串相加	布林會強制轉換成字串
布林和數值相加	布林會強制轉換成數值

資料型態的轉換函數-1



parseInt()函數

- 將字串變數值開頭的數值轉換成整數
- 如果字串沒有數值，就傳回NaN(Not a number)
- 在轉換時可以指定十六、十和八進位。
- Ex: `parseInt("3 pages")` → 3
`parseInt("2.3")` → 2

parseFloat()函數

- 將字串變數值開頭的浮點數轉換成浮點數
- 如果字串沒有數值，就傳回NaN(Not a number)。
- Ex: `parseFloat("3.2")` → 3.2

資料型態的轉換函數-2



eval()函數

- 將運算式的字串參數當作運算式，函數可以傳回運算式的計算結果。
- Ex: `eval("20+4*5")` → 40
`eval("5>4")` → true

typeof()運算子

- `typeof()`運算子可以取得變數的資料型態
 - string、number、boolean、undefined和object等資料型態。

Question

□ 列印的結果為？

```
<script>
  var strName1= "CYCU";
  var strName2=null;
  document.write(typeof(strName)+ " <br/>");
  document.write(typeof(strName2)+ " <br/>");
</script>
```


練習1

- 請建立JS程式,使用表格標籤顯示NOT、AND、OR和XOR位元運算的各種結果,運算元分別為1和0 (參考p2-28, p2-20)

a	b	NOT a	NOT b	a AND b	a OR b	a XOR b
1	1	0	0	1	1	0
1	0	0	1	0	1	1
0	1	1	0	0	1	1
0	0	1	1	0	0	0