

型別、變數與運算子

10-1 資料型別

- 數值 (Number)
 - Infinity
 - -Infinity
 - NaN
- 布林 (Boolean)
- 字串 (String)

逸脫字元	說明	逸脫字元	說明
\"	雙引號	\'	單引號
\\	反斜線	\b	BackSpace
\f	換行 (Form Feed)	\n	換行 (New Line)
\r	換行 (Carriage Return)	\t	Tab
\xNN	Latin-1字元 (NN為十六進位表示法)		
\uNNNN	Unicode字元 (NNNN為十六進位表示法)		

10-2 變數

10-2-1 變數的命名規則

- 第一個字元可以為英文字母或底線 (_)，其它字元可以為英文字母、阿拉伯數字、ISO-8859-1字元、Unicode字元、底線 (_) 或錢字符號 (\$)，英文字母要區分大小寫。
- Unicode字元的格式為 \uNNNN 。
- 不能使用JavaScript關鍵字、內建函式的名稱、內建物件的名稱。
- 每個單字的開頭建議以大寫字母表示，例如 UserName 。
- 避免在內部範圍使用與外部範圍相同的名稱。

10-2-2 變數的宣告方式

- 使用var關鍵字來宣告變數，有需要的話，還可以指派初始值，例如：

```
var MyName;
```

```
MyName = "小丸子";
```

```
var MyName = "小丸子";
```

10-3 運算子

- 單元運算子：只有一個運算元，使用前置記法 (例如 $-x$) 或後置記法 (例如 $x++$)。
- 二元運算子：有兩個運算元，使用中置記法 (例如 $x + y$)。
- 三元運算子：有三個運算元，使用中置記法 (例如 $c ? x : y$)。

10-3-1 算術運算子

運算子	意義	例子	結果	運算子	意義
+	加	$12 + 3$	15	++	遞增1
-	減	$12 - 3$	9	--	遞減1
*	乘	$12 * 3$	36	註：加法運算子也可以用來表示正值，例如 +5表示正整數5；減法運算子也可以用來表示負值，例如 -5表示負整數5。	
/	除	$12 / 3$	4		
%	餘數	$20 \% 3$	2		

- 加法運算子 (+)
 - + 運算元
 - 運算元1 + 運算元2
- 減法運算子 (-)
 - - 運算元
 - 運算元1 - 運算元2

- 乘法運算子 (*)
 - 運算元1 * 運算元2
- 除法運算子 (/)
 - 運算元1 / 運算元2

- 餘數運算子 (%)
 - 運算元1 % 運算元2
- 遞增運算子 (++)
 - ++運算元
 - 運算元++
 - `var X = 10;`
 - `alert(++X);`
 - `alert(X);`
 - `var Y = 5;`
 - `alert(Y++);`
 - `alert(Y);`

- 遞減運算子 (--)

- --運算元

- 運算元—

- `var X = 10;`

- `alert(--X);`

- `alert(X);`

- `var Y = 5;`

- `alert(Y--);`

- `alert(Y);`

10-3-2 比較運算子

符號	說明	例子	傳回值
==	等於	$21 + 5 == 18 + 8$	true
		"abc" == "ABC"	false (大小寫視為不同)
		1 == "1"	true
		1 == true	true
		"1" == true	true
		false == 0	true
		false == 1	false
!=	不等於	$21 + 5 != 18 + 8$	false
		"abc" != "ABC"	true (大小寫視為不同)
<	小於	$18 + 3 < 18$	false
>	大於	$18 + 3 > 18$	true

符號	說明	例子	傳回值
<=	小於等於	18 + 3 <= 21	true
>=	大於等於	18 + 3 >= 21	true
===	等於且相同型別	1 === "1"	false
		1 === true	false
		"1" === true	false
!==	不等於且/或不同型別	1 !== "1"	true
		1 !== true	true
		"1" !== true	true

10-3-3 邏輯運算子

- ! 運算子 (邏輯否定Not)
 - !布林運算式
 - !(50 > 40)
 - !(50 < 40)

- **&& 運算子 (最短路徑邏輯交集)**

- 布林運算式1 && 布林運算式2

- `!(50 < 40) && !(30 < 20)`

- `!(50 < 40) && !(10 < 20)`

- `!(10 < 40) && !(30 < 20)`

- `true && "Script"`

- `1 && "Script"`

- `"Java" && "Script"`

- `0 && "Script"`

- `false && "Script"`

- `null && "Script"`

- || 運算子 (最短路徑邏輯聯集)

- 布林運算式1 || 布林運算式2

- `!(50 < 40) || !(30 < 20)`

- `!(50 < 40) || !(10 < 20)`

- `!(10 < 40) || !(30 < 20)`

- `true || "Script"`

- `1 || "Script"`

- `"Java" || "Script"`

- `0 || "Script"`

- `false || "Script"`

- `null || "Script"`

10-3-4 位元運算子

- ~ 運算子 (位元NOT)
 - ~ 數值運算式
 - ~10會得到 -11，因為10的二進位值是1010，~10的二進位值是0101，而0101在2's補數表示法中就是 -11。

- & 運算子 (位元AND)

- 數值運算式1 & 數值運算式2

- 10 & 6會得到2，因為10的二進位值是1010，6的二進位值是0110，而1010 & 0110會得到0010，也就是2

- | 運算子 (位元OR)

- 數值運算式1 | 數值運算式2

- 10 | 6會得到14，因為10的二進位值是1010，6的二進位值是0110，而1010 | 0110會得到1110，也就是14

- \wedge 運算子 (位元XOR)

- 數值運算式1 \wedge 數值運算式2

- $10 \wedge 6$ 會得到12，因為10的二進位值是1010，6的二進位值是0110，而 $1010 \wedge 0110$ 會得到1100，也就是12

- << 運算子 (向左移位)
 - 數值運算式1 << 數值運算式2
 - $1 \ll 1$ 表示向左移位1個位元，會得到2
- >> 運算子 (向右有號移位)
 - 數值運算式1 >> 數值運算式2
 - $16 \gg 1$ 表示向右有號移位1個位元，會得到8
- >>> 運算子 (向右無號移位)
 - 數值運算式1 >>> 數值運算式2
 - $16 \ggg 1$ 表示向右無號移位1個位元，會得到8

10-3-5 指派運算子

運算子	例子	運算結果
=	X = 3;	將 = 右邊的值或運算式指派給 = 左邊的變數。
+=	X += 3; X += "c";	這兩個敘述相當於 X = X + 3; 和 X = X + "c";。
-=	X -= 3;	這個敘述相當於 X = X - 3;。
*=	X *= 3;	這個敘述相當於 X = X * 3;。
/=	X /= 3;	這個敘述相當於 X = X / 3;。
%=	X %= 3;	這個敘述相當於 X = X % 3;。
&=	X &= 3;	這個敘述相當於 X = X & 3;。
=	X = 3;	這個敘述相當於 X = X 3;。
^=	X ^= 3;	這個敘述相當於 X = X ^ 3;。
<<=	X <<= 3;	這個敘述相當於 X = X << 3;。
>>=	X >>= 3;	這個敘述相當於 X = X >> 3;。
>>>=	X >>>= 3;	這個敘述相當於 X = X >>> 3;。

10-3-6 條件運算子

- 條件運算式? 運算式1 : 運算式2
 - $10 > 2$? "Yes" : "No"
 - false? $10 + 2$: $10 - 2$

10-3-7 運算子的優先順序

分類	運算子
物件成員存取運算子	. []
函式呼叫、建立物件	() new
單元運算子	! ~ - + ++ --
乘除餘數運算子	* / %
加減運算子	+ -
移位運算子	<< >> >>>
比較運算子	< > <= >=
等於運算子	== != === !==
位元AND運算子	&
位元XOR運算子	^
位元OR運算子	
邏輯AND運算子	&&
邏輯OR運算子	
條件運算子	? :
指派運算子	= *= /= %= += -= <<= >>= >>>= &= ^= =