

JavaScript 概述

JavaScript 概述

- ▶ JavaScript 是一種直譯式語言
- ▶ JavaScript 主要應用在瀏覽器端的Script
- ▶ 它的直譯器被稱為**JavaScript引擎**，為瀏覽器的一部分，**廣泛用於客戶端的腳本語言**
- ▶ JavaScript並非 Java請別搞混了。
- ▶ JS相關網頁
 - ▶ <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
 - ▶ <https://developer.mozilla.org/en-US/docs/Web/JavaScript>



JavaScript 的官方Logo

<http://zh.wikipedia.org/wiki/JavaScript>

JavaScript 概述

- ▶ JavaScript、HTML、CSS為網頁設計的黃金組合
 - ▶ HTML：定義網頁的內容
 - ▶ CSS:定義網頁的外觀
 - ▶ JavaScript:定義網頁的行為 (ex:顯示歡迎訊息、或點選按鈕的互動)

JavaScript 概述

- ▶ 一般來說，完整的JavaScript包括以下幾個部分：
 - ▶ ECMAScript，描述了該語言的語法和基本物件
 - ▶ 文件物件模型（**DOM**），描述處理網頁內容的方法和介面
 - ▶ 瀏覽器物件模型（**BOM**），描述與瀏覽器進行互動的方法和介面

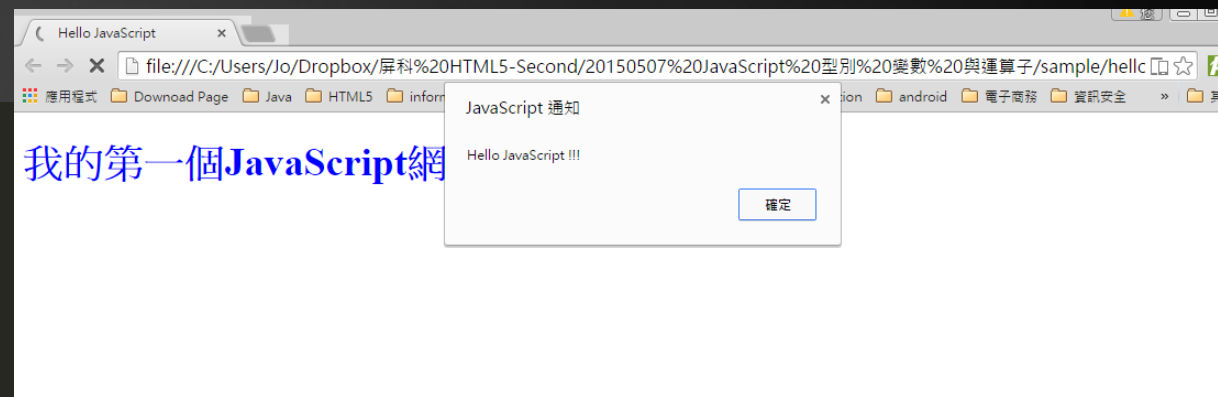
JavaScript 概述

- ▶ 它的基本特點如下：
 - ▶ 是一種解釋性程式語言（代碼不進行預編譯）。
 - ▶ 主要用來向HTML頁面添加**互動行為**。
 - ▶ 可以直接嵌入HTML頁面，但寫成單獨的js檔案有利於結構和行為的分離。
- ▶ JavaScript常用來完成以下任務：
 - ▶ 嵌入動態文字於HTML頁面
 - ▶ 對瀏覽器事件作出響應
 - ▶ 讀寫HTML元素
 - ▶ 在資料被提交到伺服器之前驗證資料
 - ▶ 檢測訪客的瀏覽器資訊
 - ▶ 控制cookies，包括建立和修改等

撰寫第一個JavaScript程式-1

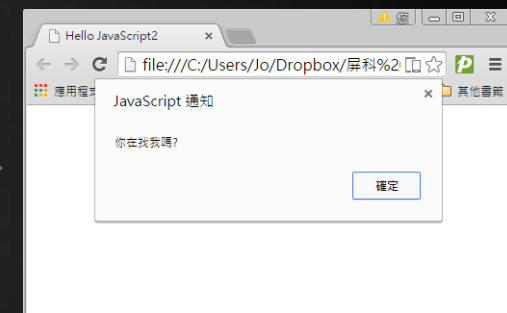
- ▶ JavaScript可放在<script>元素當中，或是透過外部Link的JavaScript檔案進行外部匯入
- ▶ 這是一個內嵌的範例：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Hello JavaScript</title>
  <script language="JavaScript">
    alert("Hello JavaScript !!!");
  </script>
  <style>
    h1{font-size: 36px;font-weight: bolder;color:blue;text-align: center;}
  </style>
</head>
<body>
  <h1>我的第一個JavaScript網頁 !</h1>
</body>
</html>
```



11

- ```
<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <title>Hello JavaScript2</title>
 <style>
 h1{font-size: 36px;font-weight: bolder;color:blue;text-align: center;}
 </style>
</head>
<body>
 <h1 onclick="javascript:alert('你在找我嗎?');">按我! 按我 !</h1>
</body>
</html>
```



# 撰寫第一個JavaScript程式-3

- ▶ 使用外嵌JavaScript時，須將檔名取名為 **.js**
- ▶ 在HTML當中使用 `<script src="外部javascript.js"></script>`進行呼叫

```
<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <title>Hello JavaScript3</title>
 <style>
 h1{font-size: 36px;font-weight: bolder;color:blue;text-align: center;}
 </style>

 <script src="js/hellojs.js"></script>

</head>
<body>
 <h1>呼叫外部的JavaScript!</h1>
</body>
</html>
```

```
alert("Hello JavaScript!");
```

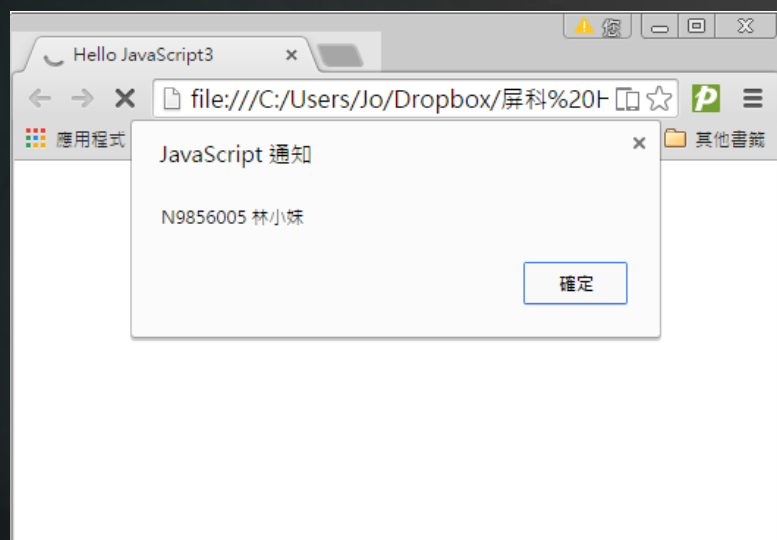
hellojs.js

Hello3.html



# 練習一 hello JS

- ▶ 請建立一個資料夾，在資料夾當中建立一個javascript檔案
- ▶ 使用外部呼叫的方式，呼叫外部的js檔案。
- ▶ 當網頁打開時，請顯示訊息方塊，訊息方塊內容請顯示學號 + 姓名
- ▶ 當訊息關掉後，請顯示網頁內容



# JavaScript的寫作風格

- ▶ JavaScript是由程式敘述市所組成，一系列程式如同英文的一個句子內含多個運算式、運算子或JavaScript關鍵字
- ▶ 每一行js的內容，是以；為結束

```
var intBalance=1000;
intBalance +=100;
document.write('我的第一份JavaScript 程式
');
```

- ▶ 每一段程式區塊，是由多個敘述市所組成，並且使用{ 與 } 包圍

```
function showmessage()
{
 alert('hello JavaScript!!!');
}
```

# JavaScript的寫作風格

- ▶ 對於太長的程式碼，JS允許適度的換行

```
document.write
('hello JavaScript!!!'); //跳出視窗內容
```

- ▶ 程式的註解

- ▶ 單行註解是以 **//** 包圍
- ▶ 多行註解 **/\*** 與 **\*/** 包圍

```
/* 這是一段顯示
跳出視窗的範例
*/

function showmessage()
{
 alert('hello JavaScript!!!'); //跳出視窗內容
}
```

# JavaScript的寫作風格

- ▶ 對於空白字元，JS會自動忽略，以下這個例子來說，所有的變數宣告都將會是5

```
var a=5;
var b = 5;
var c = 5 ;
```

- ▶ 為了讓程式易於閱讀辨識，在撰寫時，請適度的應用縮排

```
var sum=0;
var i=0;
for(i=0 ; i<=5;i++)
{
 document.write("數字 = " + i + "
");
 sum += i;
}
```

# JavaScript變數宣告

# JavaScript 變數

- ▶ 變數 (variable) 是在程式中宣告一個名稱 (識別字)，電腦會提供一個預留的記憶體空間給這個名稱，然後程式設計人員可以利用它來存放數值、字串、布林、陣列、物件等資料。
- ▶ JavaScript變數名稱區分英文字母大小寫，counter、Counter和COUNTER是不同的變數，變數名稱長度並沒有限制。

# JavaScript的變數命名原則

- ▶ 變數名稱的開始字元必須為英文字母的大小寫(要區分大小寫)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「.」，句點是保留給物件使用的運算子。
- ▶ 不能使用javascript的關鍵字(保留字)、內建函數或內建物件名稱
- ▶ 若是由多個單字組成，第二個單字自首要大寫，例如:var myName
- ▶ 函式的開頭建議以動詞表示，例如:initializeComponent
- ▶ 事件處理程式名稱的結尾，建議以EventHandler表示，例如:mouseEventHandler

Js的保留字：<http://pydoing.blogspot.tw/2010/12/javascript-reserved.html>

# 變數命名與宣告-宣告

- ▶ JavaScript程式是使用【var】指令宣告變數，如下所示：

```
var strName;
```

- ▶ 程式碼宣告一個字串變數strName，如果需要同時宣告多個變數，請使用「,」分隔，如下所示：

```
var strName, intBalance;
```

- ▶ 程式碼在同一個var指令宣告兩個變數，一為整數，一為字串。JavaScript可以在宣告變數的同時指定變數值，如下所示：

```
var strName = "陳會安";
var intBalance = 1000;
var blnSex = true;
```



# 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，所謂存在是指變數擁有值，而不是undefined資料型態。
- ▶ 筆者準備使用第3章的if條件敘述來檢查變數是否存在，因為JavaScript宣告或使用的變數都屬於Window物件的屬性，我們只需檢查Window物件的屬性，就可以知道變數是否存在，如下所示：

```
if (window.strName)
```

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

```
else
```

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

# JavaScript的保留字

- 變數名稱不能使用 JavaScript 語法的保留字，即關鍵字，如下表所示：

break	case	catch	continue	debugger
default	delete	do	else	false
finally	for	function	if	in
instanceof	new	null	return	switch
this	throw	true	try	typeof
var	void	while	with	

# 變數的宣告

- ▶ 我們可以使用**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 ";
```

或

```
var myName = " 小丸子 ";
```

- ▶ 程式碼宣告一個字串變數strName，如果需要同時宣告多個變數，請使用「,」分隔，如下所示：

```
var strName, intBalance;
```

# 變數的宣告

- ▶ 我們可以使用指定敘述「=」等號來指定變數值，在指定變數值的同時，也指定了變數的資料型態，如下所示：

```
strName = "林小妹";
intNumber= 1000;
```

- ▶ 程式碼指定變數值，字串變數strName為「林小妹」，整數變數intNumber為1000，不只如此，我們還可以再次使用指定敘述更改變數成其他值，如下所示：

```
intNumber = "1000";
```

變數intNumber的資料型態也隨之成為字串，

- ▶ 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，至於變數的資料型態，我們可以使用指定敘述隨時更改其資料型態。

# 變數的宣告

```
var a=10;
var b=20;
var c=a+b;
var myName="Jo";
var logic=a>b;

document.write('<h1> a+b =>' + a + b);
document.write('<h1> c=>' + c);
document.write('<h1> myName=>' + myName);
document.write('<h1> logic=>' + logic);
```

**a+b =>1020**

**c=>30**

**myName=>Jo**

**logic=>>false**

# JavaScript 資料型別

# JavaScript 資料型別

- ▶ JavaScript的資料型別屬於弱型別(weakly typed)，又稱為動態型別，也就是說使用前無需事先宣告，而在執行期間可動態改變型別  
舉例來說，當javascript當中使用 `1+2` 時，他會視為是數字，所以結果會顯示 3；但是當我們使用 `1+ "2"` 時，其輸出結果就會顯示為 12

```
<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <title>JavaScript 型別</title>
 <style>
 h1{font-size: 36px;font-weight: bolder;color:blue;text-align: center;}
 </style>

 <script>
 document.write('<h1> Number = ');
 document.write(1+2 + '
');
 document.write('<h1> Number = '+ 1+2);
 </script>
</head>
<body>
 <div>
</body>
</html>
```

**Number = 3**

**Number = 12**

typeJS.html

# JavaScript 型別

- ▶ JavaScript 的型別分為下列兩種類型：
  - ▶ 基本型別 (primitive type)：包括數值 (number)、字串 (string)、布林 (boolean)。
  - ▶ 物件型別 (object type)：包括陣列 (array)、函式 (function)、物件 (object)。



# 數值 (number)

- ▶ 我們可以隨意使用諸如123、-456、78.9、-12345.6789 等十進位數值，其型別均為number，注意不要超過  $-2^{1024} \sim 2^{1024}$  ( $-10^{307} \sim 10^{307}$ ) 的範圍即可。
- ▶ JavaScript 提供了下列幾個特殊的數值：
  - ▶ NaN(Not a number)：不當的數值運算(ex:將數值除以字串)
  - ▶ Infinity：正無限大，表示超過JavaScript所能表示的最大範圍(ex:正整數除以0)
  - ▶ - Infinity：負無限大，表示超過JavaScript所能表示的最小範圍(ex:負數除以0)

# 字串 (string)

- ▶ JavaScript 針對字串提供了string 型別，並規定字串的前後必須加上雙引號 (") 或單引號 (')，但兩者不可混用，例如 “生日”、'birthday'。
- ▶ 為了方便區隔，建議以' '作為字串變數的宣告，較不易混淆

```
<script>
 document.write('<h1> 我是林小妹，今年 '+ 18 + '歲');
</script>
```

我是林小妹，今年 18歲

# 字串-溢出字元

- ▶ 當在撰寫網頁時，有時必須顯示" "或' '，但是這幾個符號都是JavaScript中定義字串用的，因此為了要讓網頁也能顯示這類的符號，因此JS提供了規則，也就是加入\符號，告訴瀏覽器，這符號後面是一個特殊符號，讓瀏覽器能順利顯示。

跳脫字元	說明	跳脫字元	說明
\"	雙引號	\'	單引號
\\	反斜線	\b	BackSpace
\f	換行(Form Feed)	\n	換行
\r	換行(Carriage Return)	\t	Tab

# 其他資料型別

- ▶ 布林資料型態 ( Boolean Data Type )
  - ▶ 布林資料型態只有兩個值true和false，主要是用在第3章條件和迴圈控制的條件判斷，以便決定繼續執行哪一個程式區塊的程式碼，或是判斷迴圈是否結束。
- ▶ Null資料型態
  - ▶ Null資料型態只有一個值null，null是一個關鍵字並不是0，如果變數值為null，表示變數沒有值或不是一個物件。
- ▶ Undefined資料型態
  - ▶ Undefined資料型態指的是一個變數有宣告，但是不曾指定變數值，或是一個物件屬性根本不存在。

## 練習二 變數與顯示資料

- ▶ 請宣告以下幾個變數
  - ▶ `a=10;b=20; myName="林小妹";age=18`
- ▶ **`document.write ()`**的指令，就是在撰寫網頁內容
- ▶ 請應用`document.write ()`指令，寫出如下圖的顯示畫面(可參閱前面的幾張投影片)

我是'林小妹'，今年 18歲

數字： $10 + 20 = 30$

字串： $10 + 20 = 1020$

# 練習三 document.write

- ▶ 請應用document.write() 畫出以下的表格

Hello Document		
	A	B
1	A1	B1
2	A2	B2

# JavaScript的運算子

# 運算子

- ▶ 算術運算子：+、-、\*、/、%
- ▶ 遞增/ 遞減運算子：++、--
- ▶ 邏輯運算子：&& ( 邏輯AND)、|| ( 邏輯OR)、! ( 邏輯NOT)
- ▶ 比較運算子：!=、==、!==(、===、>、<、>=、<=
- ▶ 位元運算子：& ( 位元AND)、| ( 位元OR)、^ ( 位元XOR)、~ ( 位元NOT)、<< ( 向左移位)、>> ( 向右移位)、>>> ( 向右無號移位)
- ▶ 指派運算子：=、+=、-=、\*=、/=、%=、&=、|=、^=、<<=、>>=、>>>=
- ▶ 條件運算子：?:
- ▶ 型別運算子：typeof



# 算術運算子-種類

- ▶ JavaScript算術運算子擁有常用的數學運算子，大部分運算元是數值，不過，加法運算子可以連接兩個字串變數。各種算術運算子的說明與範例（變數a的值為10），如下表所示：

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

# 遞增 / 遞減運算子

- ▶ 遞增運算子 (++) 可以用來將運算元的值加1，例如：

```
var X = 10; // 宣告一個名稱為 x、初始值為 10 的變數
alert(++X); // 先將變數 x 的值遞增 1，之後再顯示出來而得到 11
alert(X); // 變數 x 的值在前一個敘述中被遞增為 11，因而顯示 11
var Y = 5; // 宣告一個名稱為 y、初始值為 5 的變數
alert(Y++); // 會先顯示變數 y 的值為 5，之後再將變數 y 的值遞增 1
alert(Y); // 變數 y 的值在前一個敘述中被遞增為 6，因而顯示 6
```

- ▶ 遞減運算子 (--) 可以用來將運算元的值減1，例如：

```
var X = 10; // 宣告一個名稱為 x、初始值為 10 的變數
alert(--X); // 先將變數 x 的值遞減 1，之後再顯示出來而得到 9
alert(X); // 變數 x 的值在前一個敘述中被遞減為 9，因而顯示 9
var Y = 5; // 宣告一個名稱為 y、初始值為 5 的變數
alert(Y--); // 會先顯示變數 y 的值為 5，之後再將變數 y 的值遞減 1
alert(Y); // 變數 y 的值在前一個敘述中被遞減為 4，因而顯示 4
```

# 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，在變數之前，變數值會立刻改變；之後則是在執行運算式後才改變，如下所示：

```
x++;
```

```
--y;
```

- ▶ 變數x和y的初始值為10，後面二列程式碼的第一列是x++，運算子在後所以之後才改變，第一個x++值仍然為10；第二個x為11，最後一列的--y運算子在前，所以第一個為9；第二個也是9，如下所示：

```
x = 10;
```

```
y = 10;
```

```
document.write("x++ = " + x++ + ":x = " + x + "
");
```

```
document.write("--y = " + --y + ":y = " + y + "
");
```

# 邏輯與比較運算子-比較

- ▶ 邏輯與比較運算子主要是使用在迴圈和條件敘述的判斷條件，`true`為真；`false`為假，例如`3 < 10` 會傳回 `true`，而`3 > 10` 會傳回`false`。
- ▶ 比較運算子的說明和範例，如下表所示：

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 &lt;&gt; 3</code>	<code>true</code>
<code>&lt;</code>	小於	<code>6 &lt; 3</code>	<code>false</code>
<code>&gt;</code>	大於	<code>6 &gt; 3</code>	<code>true</code>
<code>&lt;=</code>	小於等於	<code>6 &lt;= 3</code>	<code>false</code>
<code>&gt;=</code>	大於等於	<code>6 &gt;= 3</code>	<code>true</code>

# 邏輯運算子

- ▶ 邏輯運算子可以用來進行AND、OR、NOT等邏輯運算。
  - ▶ && (AND)：例如  $(5 > 4) \ \&\& \ (3 > 2)$  會傳回true， $(5 > 4) \ \&\& \ (3 < 2)$  會傳回false。
  - ▶ || (OR)：例如  $(5 > 4) \ || \ (3 < 2)$  會傳回true， $(5 < 4) \ || \ (3 < 2)$  會傳回false。
  - ▶ !(NOT)：例如  $!(50 > 40)$  會傳回false， $!(50 < 40)$  會傳回true。

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

- ▶ 位元運算子可以用來進行AND、OR、XOR、NOT、SHIFT等位元運算，

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

# 指派運算子

- ▶ 指派運算子可以用來指派值給變數，其語法如下：

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

# 條件運算子

► 條件運算子的語法：

條件運算式 ? 運算式 1 : 運算式 2

例如  $10 > 2$  "Yes" : "No" 會傳回 "Yes"，`false ? 10 + 2 : 10 - 2` 會傳回 8。



# 型別運算子

- ▶ JavaScript 提供的typeof 型別運算子可以傳回資料的型別，例如 typeof("生日")、typeof(-35.789)、typeof(true) 會傳回 "string"、"number"、"boolean"。

# 運算子的優先順序

- ▶ 當運算式中有多種運算子時，JavaScript 會依照如下的優先順序執行運算子，優先順序高者先執行，相同者則按出現的順序由左到右依序執行。

高	類型	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	?:
低	指派運算子	= *= /= %= += -= <=> >>= &= ^=  =

# 練習四 – 運算子

▶ 請在網頁上顯示下列各算式的結果

- ▶ `0xFFF-0xFAB`
- ▶ `2 / 3.0`
- ▶ `True / 0`
- ▶ `False / 0`
- ▶ `12.3 * 10 % 5`
- ▶ `'a' > 'Z'`
- ▶ `~0x0005`
- ▶ `"ABC"+88 == "abc"+88`
- ▶ `50 > 30 ^ 70 > 100`
- ▶ `3 | 5`
- ▶ `8 > 7 true or false`
- ▶ `128 >> 3`
- ▶ `X=5 ; alert(++X)`
- ▶ `Y=10 ; alert (Y--)`

**`0xFFF-0xFAB = 84`**

**`2/3.0 = 0.6666666666666666`**

**`true/0 = Infinity`**

**`12.3 * 10 % 5 = 3`**

**`'a' > 'Z' = true`**

**`~0x0005 = -6`**

**`X=5 , ++X = 6`**

**`Y=10 , Y-- = 9`**

**`"ABC"+88 == "abc"+88 = false`**

**`50 > 30 ^ 70 > 100 = 1`**

**`3 | 5 = 7`**

**`8 > 7 true or false = true`**

**`128 >> 3 = 16`**

# 資料型態的轉換

## 2-4-1 資料型態的強制轉換

- ▶ JavaScript是一種鬆散型態的程式語言，所謂變數的資料型態是指變數值的資料型態，基本上，JavaScript運算式會強制進行型態轉換，因為JavaScript運算式的運算元需要相同型態，資料型態的強制轉換方式，如下表所示：

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

## 2-4-2 資料型態的轉換函數-1

- ▶ JavaScript雖然在執行運算時會自動進行資料型態的強迫轉換，不過，JavaScript仍然提供數個函數和運算子來進行資料型態的轉換。
- ▶ **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。
- ▶ **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)。

## 2-4-2 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。

- ▶ typeof()運算子：

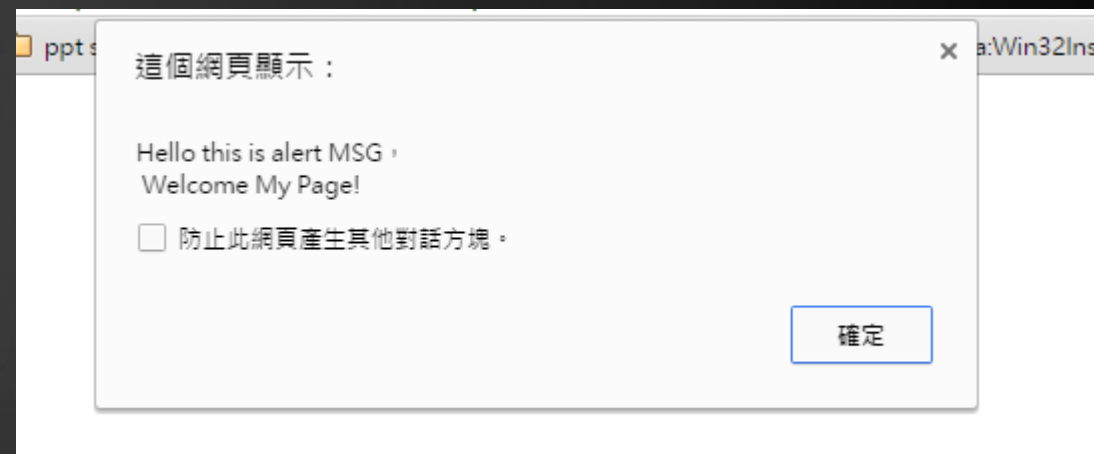
typeof()運算子可以取得變數的資料型態，也就是string、number、boolean、undefined和object等資料型態。

# 補充:windows物件-alert()

- ▶ 在有些時候，讀取網頁時，會需要顯示一些特殊訊息，此時我們可採用alert來彈出訊息窗
- ▶ 請注意在這個範例當中，我們採用"\n"來替文字進行換行。

```
<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <title>Windows 物件-alert</title>
 <script>
 alert("Hello this is alert MSG , \n Welcome My Page!");
 </script>
</head>
<body>

</body>
</html>
```



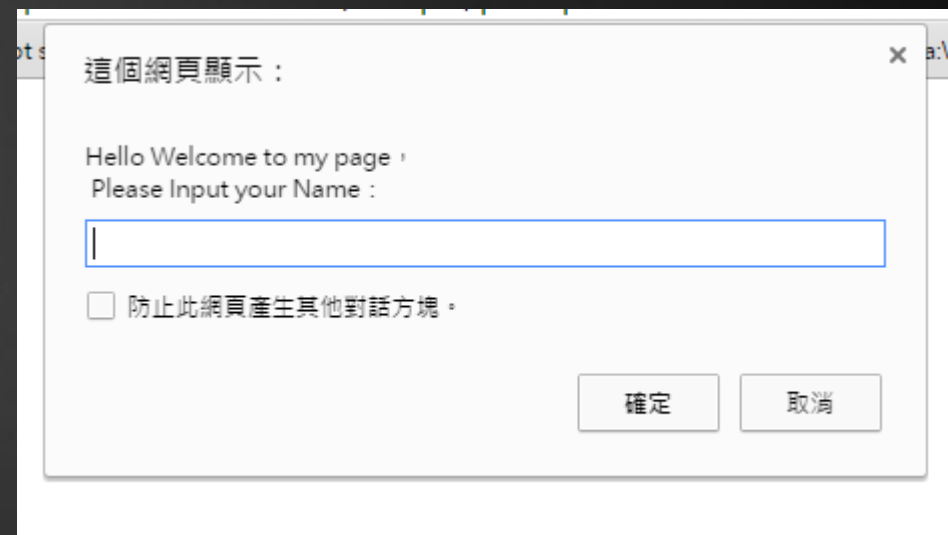


# 補充:windows物件-prompt()

- ▶ 在網頁視窗的元件當中，有另一種可以讓使用者輸入資訊的方法，就是採用prompt()

```
<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <title>Windws 物件-prompt</title>
 <script>
 prompt("Hello Welcome to my page , \n Please Input your Name : ");
 </script>
</head>
<body>

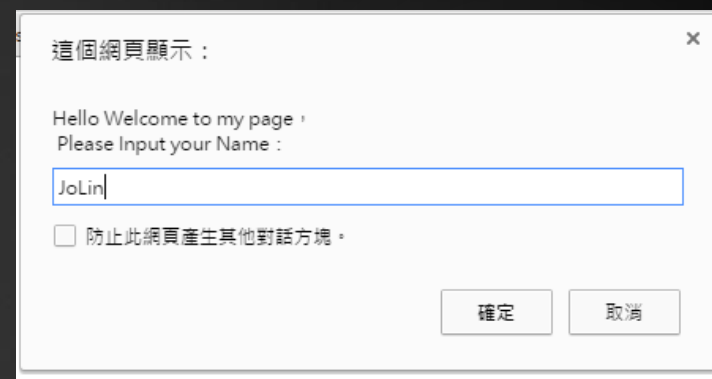
</body>
</html>
```



# 補充:windows物件-prompt()

- ▶ 我們若要取得輸入的內容，其寫法如範例所示

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <title>Windows 物件-prompt_getInfo</title>
7 <script>
8 var name;
9 name=prompt("Hello Welcome to my page , \n Please Input your Name : ");
10 document.write("<h1>Hello "+name);
11 </script>
12 </head>
13 <body>
14
15 </body>
16 </html>
```



**Hello JoLin**

- 第 8 行 先宣告一個變數name
- 第 9 行 將取得的內容，存入變數name當中
- 第10行 在網頁上印出 hello字串，並接上人名

# 補充:String or Int

- ▶ 我們將上個例子進行修改，提示使用者輸入數字，利用使用者輸入，取出使用者輸入的型態，並將型態顯示在網頁上，但我們發現顯示的是String，這並不是錯誤，而是透過prompt取得的內容都會是字串
- ▶ 若要將輸入的數字進行應用，則必須將輸入的內容進行轉型，這就是parseInt()或是parseFloat()的應用實例

```
<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <title> Data Type</title>
 <script>
 var bnumber;
 var anumber;
 anumber=prompt("請輸入一個數字：");
 bnumber=parseInt(anumber);
 document.write("<h1>Before transfer data Type("+anumber+") : " +typeof(anumber)+"
");
 document.write("<h1>After transfer data Type("+bnumber+") : " +typeof(bnumber)+"<p>");
 </script>
</head>
<body>

</body>
</html>
```

**Before transfer data Type(100) : string**

**After transfer data Type(100) : number**

# 練習五 資料型別轉換(重要練習)

- ▶ 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，如圖式

請輸入數字A：

請輸入數字B：

數字A(20)，資料型態為：**string**

數字B(6)，資料型態為：**string**

$$20 + 6 = 26$$

$$20 - 6 = 14$$

$$20 * 6 = 120$$

$$20 / 6 = 3.3333333333333333$$