

JavaScript 語法變數資料型別

【JavaScript 程式碼的位置】

⇒ javascript 程式碼應置於何處？

- ✓ 《參考網址》http://www.w3schools.com/js/js_where.asp
- ✓ 放在 `<head>` `</head>` 區段內
 - 程式碼必須被寫在 `<script>` 和 `</script>` 區段之內
 - 接著，`<script>` 區段再被放在 `<head>` `</head>` 區段內
- ✓ 放在 `<body>` `</body>` 區段內
 - 程式碼必須被寫在 `<script>` 和 `</script>` 區段之內
 - 接著，`<script>` 區段再被放在 `<body>` `</body>` 區段內
- ✓ 放在外部的 javascript 檔案中
 - 必須將此外部 javascript 檔案的副檔名命名為 `.js`
 - 接著，再利用`<script>`去引用該 javascript 檔案
 - ✧ 例如，引用 `myHello.js`，寫成 `<script src="myHello.js"></script>`

⇒ 【範例展示】javascript 程式碼置於不同位置的範例

- ✓ javascript 被放在 `<head>` 區段內
 - 參考程式碼：PutInHead.html
- ✓ javascript 被放在 `<body>` 區段內
 - 參考程式碼：PutInBody.html
- ✓ javascript 被放在外部的 `.js` 檔案中
 - 參考程式碼：PutInFile.html

【JavaScript 常見輸出方式】

⇒ 【範例展示】javascript 常見的輸出方式？

- ✓ 《參考網址》http://www.w3schools.com/js/js_output.asp
- ✓ 利用 `window.alert()`
 - 參考程式碼：UsingAlert.html
- ✓ 利用 `document.write()` 或 `document.writeln()`
 - 此方式通常是被使用來測試執行是否正確！
 - 若要讓 `writeln()` 有換行的效果，則必須將 `<script>` 整個區段放在 `<pre>` 和 `</pre>` 之間

【程式語言：JavaScript】

- 【註】若是 javascript 程式碼被執行的時機是在，「頁面內容全部顯示完成，使用者按下按鈕 or 輸入某些資料後」才被執行，則執行這些程式碼可能會清除頁面中原來的所有內容！
- 參考程式碼：UsingWrite_1.html, UsingWrite_2.html, UsingWrite_3.html
- ✓ 利用 html5 元素的 **innerHTML 屬性**
 - 此方式最為普遍
 - 參考程式碼：UsinginnerHTML.html
 - 《註》試試把 script 區段移至 div 元素之前，會有何效果？

【JavaScript 語法介紹】

⇒ 【陳述式 (statement)】 javascript 的基本語法介紹

- ✓ 《參考網址 1》http://www.w3schools.com/js/js_syntax.asp
- ✓ 《參考網址 2》http://www.w3schools.com/js/js_statements.asp
- ✓ 《參考網址 4》http://www.w3schools.com/js/js_comments.asp
- ✓ 一般來說，javascript 程式碼的陳述式可能包含
 - 文字 (**literal**) 或值 (**value**)。例如
7, 8.8, 0.05, "Hello", 'kech', true, ['Willy', 'Tim']
 - 變數 (**variable**)：用來存放資料/文字/值。例如，
 - ◇ 《範例 1》var x;
 - ◇ 《範例 2》var age = 45;
 - 運算子 (**operator**)。例如
=, +, -, *, /
 - ◇ 《範例 1》var myScore = 99;
 - ◇ 《範例 2》var myScore = 80*0.5 + 90/5 - 3;
 - 運算式 (**expression**)：例如
 - ◇ 《範例 1》5 + 8
 - ◇ 《範例 2》3*(2+8)-5
 - ◇ 《範例 3》x+y-3
 - ◇ 《範例 4》'Hello' + 'cju'
 - ◇ 《範例 5》多重運算式：單行陳述式中可放置多重運算式。例如
var myScore; var myFavor = 'Basketball';
var a, b, c;
a=88; b='Hello'; c=true;
var myScore = 88, myFavor = 'Basketball', myClass = 2;

【程式語言：JavaScript】

- 關鍵字 (keyword) : 例如
var, for, if, else, do, while, switch,
- ✧ 《範例 1》**var** x = 6+9
- ✧ 《範例 2》**for** (**var** i=0; i<10; i++)
- 註解 (comment)
 - ✧ 單行註解：利用「//」進行註解的標示。例如

```
var score;           // 此變數用來記錄成績
var studNum;         // 此變數用來記錄學生人數
```
- 多行註解：以「/*」開始，到「*/」結束。例如

```
/* 底下的迴圈計算總和 */
for (var i=0; i<10; i++) { }

/* 底下共宣告二個變數
   變數 score 是用來記錄成績
   變數 studNum 是用來記錄學生人數
*/
var score;
var studNum;
```
- 【註】：也可以對於目前暫時不執行的程式碼進行註解。例如

```
// var score;
// var myScore = 99;
// document.getElementById("myDiv").innerHTML = "Hello";
```

【JavaScript 變數 (Variable)】

- ⇒ 《參考網址》http://www.w3schools.com/js/js_variables.asp
- ⇒ 變數的用途：用來儲存資料 (data) 或值 (value)
- ⇒ 變數的宣告：必須使用 **var** 進行宣告。例如

```
var no = 8;
var name = "kech";
```
- ⇒ 變數的命名方式：常見的有兩種
 - ✓ 「駱駝峯命名法 (Camel Case)」：例如，firstName, myClass, myAge
 - ✓ 「底線命名法 (Underscore)」：例如，first_name, my_class, my_age
 - ✓ 《註》變數的大小寫，會被視為不同的變數
 - 例如，myAge 和 MyAge 是二個不同的變數
- ⇒ 【註】javascript 的關鍵字「不可以」當變數使用

【程式語言：JavaScript】

表：javascript 程式碼中常見的關鍵字/保留字

關鍵字	意義/作用
break	離開/終止 switch 條件結構或迴圈結構
continue	離開目前迴圈，重頭開始
do.....while	當條件判斷為 true 時，重覆地執行程式區塊
for	當條件判斷為 true 時，重覆地執行程式區塊
function	宣告一個函數
if.....else	依條件判斷式是 true 或 false，決定要執行那一個程式區塊
return	回傳函數的執行結果，同時離開函數
switch	依條件判斷式是何種值，來決定要執行那一個程式區塊
try.....catch	定義在 error 發生的情況下，該執行那一個程式區塊
var	宣告一個變數

⇒ 底下列出更多的關鍵字

⇒ 《常見》

var, for, if, else, do, while, switch, case, break, continue, function, return, try, catch, throw, this, debugger

⇒ 《其他》

abstract arguments boolean byte
char class const continue
debugger default delete double
enum eval export extends
false final finally float goto
implements import in instanceof int interface
let long native new null
package private protected public
short static super synchronized
throws transient true typeof
void volatile with yield

【JavaScript 資料型別 (Data Type)】

⇒ 《參考網址》http://www.w3schools.com/js/js_datatypes.asp

⇒ **數字 (Number)**：例如，8， -3， 5.5， -9.9

✓ 《範例 1》var x = 8;

✓ 《範例 2》var age = 33;

⇒ **字串 (String)**：例如，"Hello" 或 'kech'

✓ 《範例 1》var str = "Hello";

【程式語言：JavaScript】

- ✓ 《範例 2》`var name = 'kech';`
- ⇒ **布林 (Boolean)**：例如 · `true` 或 `false`
 - ✓ 《範例 1》`var open = true;`
 - ✓ 《範例 2》`var flag = false;`
- ⇒ **陣列 (Array)**：例如 · `[2, 4, 6]` 或 `["kech", "Brian", "Tim"]`
 - ✓ **【注意】**是使用**中刮號 []**
 - ✓ 《範例 1》`var num = [2, 4, 6];`
 - ✓ 《範例 2》`var names = ["kech", "Brian", "Tim"];`
- ⇒ **物件 (Object)**：例如 · `{name:"kech", sex:'M', age:45}`
 - ✓ **【注意】**是使用**大刮號 {}**
 - ✓ 《範例 1》`var personInfo = {name:"kech", sex:'M', age:45};`
- ⇒ **《註》空字串的寫法 (經常使用)**
 - ✓ 《範例 1》`var str = "";`
 - ✓ 《範例 2》`var htmlStr = "";`
- ⇒ **《常出錯的範例》**
 - ✓ 《範例 1》`var my = 45 + "kech";` → 執行結果：my 是 45kech
 - ✓ 《範例 2》`var my = 40 + 8 + "kech";` → 執行結果：my 是 48kech
 - ✓ 《範例 3》`var my = "kech" + 40 + 8;` → 執行結果：my 是 kech408
 - ✓ 《範例 4》`var sum = 40 + 8;` → 執行結果：sum 是 48
 - ✓ 《範例 5》`var sum = '40' + '8';` → 執行結果：sum 是 408
 - ✓ 《範例 6》`var sum = '40' + 8;` → 執行結果：sum 是 408
 - ✓ 《範例 7》`var sum = 40 + '8';` → 執行結果：sum 是 408

javascript 資料型別的匯整表

資料型別	範例
字串 (string)	<code>var name = "kech";</code>
數字 (number)	<code>var score = 88;</code>
布林 (boolean)	<code>var pass = true;</code>
陣列 (array)	<code>var names = ["kech", "brian", "Tim"];</code> // 建議用此寫法 <code>var names = new Array("kech", "brian", "Tim");</code>
物件 (object)	<code>var teacher = {firstName:"Chih-Horng", lastName:"kech", age:46};</code>

- ⇒ 實際上 · 陣列 (array) 通常被視為一物件 (object)

【程式語言：JavaScript】

⇒ 【說明】陣列 (array) 與物件 (object) 的區別

- ✓ 陣列 (array) : 使用數字(number)作為索引值，例如 names[1]
- ✓ 物件 (object) : 使用名字(name)作為索引值，例如 teacher.age

⇒ 《範例》顯示下列各種值的資料型別

- ✓ 程式碼

```
<script>
  document.getElementById("demo").innerHTML =
    typeof "kech" + "<br>" +
    typeof 888.88 + "<br>" +
    typeof true + "<br>" +
    typeof [11, 22, 33, 44] + "<br>" +
    typeof {name:'kech', sex: 'M', age:45};
</script>
```

- ✓ 執行結果

```
string
number
boolean
object
object
```