

# RWD網頁設計



# JavaScript

- JavaScript 是一種程式語言，它使你能夠創建動態更新內容、控制多媒體，動畫圖片等等幾乎所有事。
- JavaScript 原始碼執行前不須經過編譯，程式碼會直接透過瀏覽器解譯執行。
- JavaScript 因瀏覽器支援，因此具有跨平台特性。目前JS引擎及框架技術成熟，再開發伺服器端程式也能看到JS相關程式碼。



# JavaScript基本寫法

- 屬性:位於HTML標籤內(通常是事件)

```
<button type="button" class="btn btn-danger" onclick="reset_value();">
```

- script tag:頁面利用<script> </script>包裹JS程式碼。
- script src files:透過html標籤引用的方式引用js檔。

```
<script src="<?=base_url()?>js/jquery.min.js"></script>  
<script src="<?=base_url()?>js/ui.js"></script>
```

# JS程式碼範例

```
<script>
$(function () {
    $('#tb_users .a_link').editable({
        url: function (params) {
            var col = $(this).attr("name");
            var id = $(this).attr("data-pk");
            var val = params.value;
            var id_vol_record = $(this).attr("data-pk2");
            var id_vol_update = $(this).attr("data-pk");
            var table = $(this).attr("data-table");
            $.ajax({
                type: 'POST',
                url: "{site_url}learning/update_video_url/"+id+"/"+table+"/volunteer_update",
                data: {col:col,val:val,id_vol_record:id_vol_record,id_vol_update:id_vol_update},
                dataType: 'html',
                success: function (data, textStatus, jqXHR) {
                    toastr.info("保存成功!");
                    if(data == 'insert'){
                        window.location.href="{site_url}learning/volunteer_list/<?=$this->session->user_session?>";
                    }
                },
                error: function () { alert("error");}
            });
        },
        type: 'text'
    });
});
</script>
```

# 建立基本JS標 籤及輸出

```
<script>  
    document.write("I'm JavaScript");  
</script>
```

# JS語法規定

- 英文大小寫不同
- 忽略空白字元
- 註解方法1.`/**/`(多行註解) 2.`///`(單行註解)
- 結尾加分號
- 保留字元表

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

# JS變數使用

- 變數：表示會隨著程式的執行而改變其內容
- 變數命名規則
  - 大小寫字母區別
  - 不可使用保留字
  - 起始字元僅能使用英文字母及符號\_
  - 除第一字元外可以英文字母、數字、符號\_
- 宣告變數
  - var 變數名稱

# JS變數使用

- 練習1：使用變數輸出一段文字
- 練習2：將練習1之變數，透過取代方式重新設定值後輸出



# JS變數型態

型態名稱	意義	範例
字串	可以為字母、數字、文字、符號等混和組成，需用單引號或雙引號包裹值	<pre>var str='字串'; var str='字串123'; var str='123';</pre>
數字	可以為整數或是小數	<pre>var num=123; var num=0.5 var num=-1;</pre>
布林值	true/false，不需用單引號或雙引號包裹值	<pre>var boo=true; var boo=false;</pre>

註:字串型數字可透過\*1轉換為數字

```
var a='123'*1;//字串123  
var b='123'*1;//數字123
```

# JS串聯符號

- 在JS中，若要將自串串連起來，可使用+符號;

- 範例1:

```
var str="我叫";  
var name="王順陞";  
document.write(str+name); //輸出:我叫王順陞
```

- 範例2:

```
var str="我叫";  
document.write(str+"王順陞"); //輸出:我叫王順陞
```

- 範例3:

```
var str="123";  
document.write(str+"456"); //輸出:123456
```

# JS變數練習

- 練習：利用變數設定下圖紅字的部分後，輸出如下圖文字

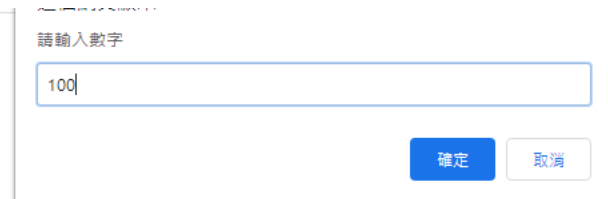
大家好，我是ASheng  
身高:160公分，體重45公斤  
今年25歲

- 註:換行可利用html的<br>來達成

# JS輸入函數

- 在JS程式語言中，可以透過輸入函數取得使用者輸入的值進行程式運算。
- 輸入函數：prompt("顯示訊息","回傳預設值")
- 範例：

100

A screenshot of a JavaScript prompt dialog box. The title bar says "JavaScript 對話框". The main text inside the box is "請輸入數字". Below the text is a text input field containing the number "100". At the bottom right of the box are two buttons: "確定" (OK) and "取消" (Cancel).

```
<script type="text/javascript">  
  var a=prompt("請輸入數字",0);  
  document.write(a);  
</script>
```

# JS運算子

- 運算式：在程式設計中，利用邏輯判斷或演算法運算取得所需的結果，過程稱為運算式。運算式是由運算元與運算子構成

# JS算術運算子

- 算數運算子：可以對程式進行加減乘除等數學運算

符號	說明	範例	運算結果
+	加法	6+4	10
-	減法	6-4	2
*	乘法	6*4	24
/	除法	12/4	3
%	餘數	5%3	2

# JS指派運算子

- 指派運算子：在JS程式設計中「=」不代表數學的等於，而是將數值指派給變數，即需要指定值給變數時就得使用指派運算子。

符號	說明	範例
=	指派	a=1; //a的值為1
+=	複合指派	a=1; a+=1; //a=a+1 a結果為2 a='a'; a+='b'; //a='a'+ 'b' a結果為ab

# JS算術運算子 練習

- 練習:利用輸入函數取得使用者輸入的兩個數字後，在畫面輸出兩數的和。
- 註:由於輸入函數會將使用者輸入訊息轉為字串，因此可以透過將該變數\*1後轉為數字。

```
var a=prompt("請輸入文字",0)*1;
```

- 練習2:利用輸入函數取得使用者輸入的兩個字串後，在畫面輸出兩變數串連後的結果。



# JS邏輯運算子

- 邏輯運算子：邏輯運算子會將運算式兩邊比較後，再將結果以布林值回傳。
- `var a=5;var b=3;`

符號	說明	範例	運算結果
<code>==</code>	相等	<code>a==b</code>	false
<code>===</code>	變數型態及值必須完全相等	<code>a===b</code>	false
<code>!=</code>	不相等	<code>a!=b</code>	true
<code>!==</code>	變數型態或值不相等	<code>a!==b</code>	true
<code>&lt;</code>	小於	<code>a&lt;b</code>	false
<code>&gt;</code>	大於	<code>a&gt;b</code>	true
<code>&lt;=</code>	小於等於	<code>a&lt;=b</code>	false
<code>&gt;=</code>	大於等於	<code>a&gt;=b</code>	true

# JS邏輯運算子

- `var a=true;var b=false;`
- `var c=5;var d=-6;`

符號	說明	範例	運算結果
&&	同為true	<code>a&amp;&amp;b</code>	false
	其中之一為true	<code>a  b</code>	true
!	相反符號	<code>!a</code>	false
&&	同為true	<code>c&gt;o &amp;&amp; d&gt;o</code>	false
	其中之一為true	<code>c&gt;o    d&gt;o</code>	true
!	相反符號	<code>!(c&gt;o &amp;&amp; d&gt;o)</code>	true

# JS條件控制

- 由於程式是由上往下逐行執行，為了使程式能靈活運用(如跳過某幾行程式不執行)，就必須使用條件控制。
- 在JS中，條件控制有if、if/else、switch

```
<script type="text/javascript">
  var a=5;
  var b=3;
  if (a>b) {
    console.log("a>b");
  }
</script>
```

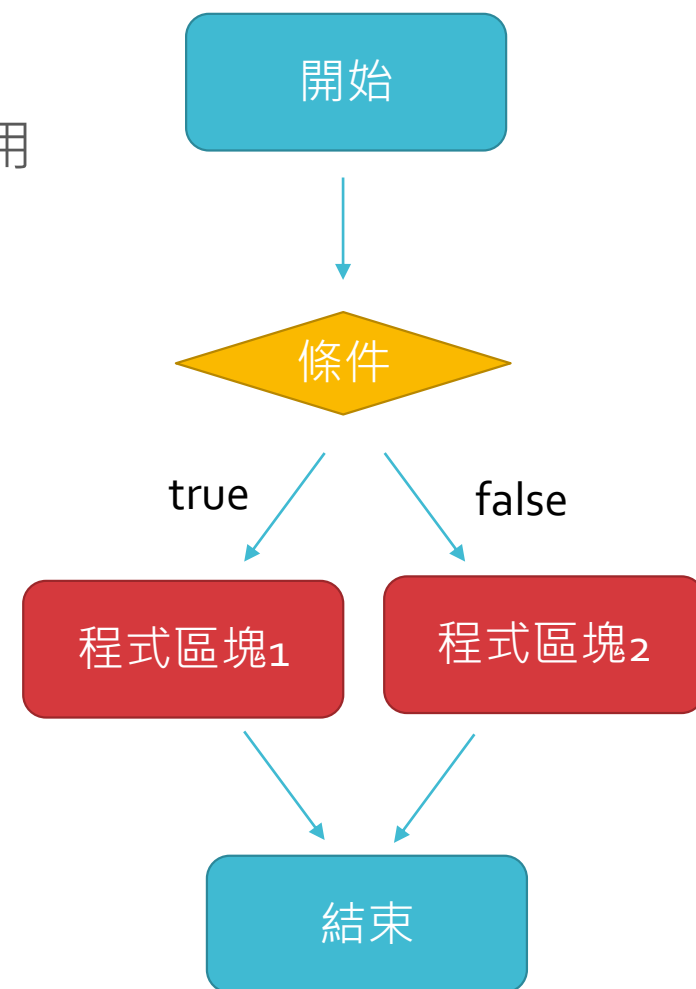
# JS if/else條件控制

- 在js程式語言中if條件式可以單獨使用

```
if (條件式) {  
    //條件式結果為true時執行的內容  
}
```

- 也能搭配else聯合使用

```
if (條件式) {  
    //條件式結果為true時執行的內容  
}else{  
    //條件式結果為false時執行的內容  
}
```



# JS if/else條件控制

- 練習：由使用者輸入一數字後，判斷是否大於100，若結果為true時輸出該數值大於100；若結果為false時輸出該數值小於等於100。
- 註:輸入方法可使用prompt函數