**JavaScript**

**基本觀念**

* W3schools.com , 所有API都在這邊找
* 物件基礎，但不是物件導向。
* 內嵌html , 瀏覽器解譯器執行程式碼。
* 只能讀寫cookie，無法讀取用戶端硬碟或檔案(安全性)。
* 不用宣告變數型態，會自動轉換。
* 分大小寫，script預設是js。

**JavaApplet ?**

Server compile bytecode , using JVM runs on client.

**特殊標籤**

* <pre>印出js的原始效果(忽略html排版)。
* <xmp>可印出html的標籤。
* <audio>標籤，背景音樂。

**輕鬆處理html控制項**

* 透過設定id即可直接操作其屬性。
* <audio id ="player" src="1.mp3" autoplay="ture" > </audio>

<input type=button value="change music" onClick="changeMusic()" />

<script>

function changeMusic(){

player.src="2.mp3";

}

</script>

**Js嵌入html之基本應用**

* 直接透過javascript:指定該html屬性為js處理。

<a href ="javascript:alert('fuck')"> fuck </a>

* 透過onclick事件，可直接透過js處理。
* <a href ="www.ntu.edu.tw" onclick="return(confirm('really'))"> go ntu</a>
* <a href = "" onClick="alert('afffdssdfsdf')"> 點我</a>

超連結的連結設為空，將會透過onClick去做連結的處理。

**基本視窗應用**

* alert 🡪 警示視窗
* confirm 🡪 確認視窗 , 回傳ture , false
* prompt 🡪 輸入視窗，回傳輸入值

**表單**

* 可設定表單名稱做後續處理 , action代表要把表單資料送到哪裡處理
* Button ,，submit送出表單，透過onClick事件，達到點擊要做的事。

<form name="theForm" >

<input type=text name="theString" value="test">

<input type=button value="gogo" onClick="document.title=document.theForm.theString.value">

</form>

**動態操作控制項**

* 可以透過name或者id來指定該控制項。
* eval ，執行字串指令，常用於使用迴圈建立變數。

command = "x"+i+"="+i\*i\*i;

eval(command);

**物件**

Student = new Object();

student.name="Andy";

student.age=3;

for(prop in student)

**Script管理**

透過 <script src= xxx.js> 把js分開寫，方便管理。

不一定要只把function拉出，其他也可以。

**常用物件**

* 字典物件 , 透過ActiveXObject(“Scripting.Dictionary”)建構 ，

可取出所有鍵值, 等相關字典應用。

* 檔案物件 ， fso。

**預約程式碼的執行**

* setTimeout，設定幾秒後執行。
* setInterval，設定幾秒執行一次，如時鐘。
* 可不用input 標籤顯示，透過區塊標籤，去改變其innerHTML即可。

**DOM**

* Document Object Models。
* 所有的物件都有階層關係。
* Window > document > form > ….。
* 透過三種方式來取得物件性質。
  + objectName.propertyName
  + objectName[“propertyName”] (可帶入字串變數)
  + objectName[index]

**動態下拉式選單**

* select標籤。
* list.options , 取得選單陣列。
* list.options[i]=new Option(text,value)，建立新的value。
* list.options.length等屬性。

**表單驗證**

* 基本驗證
* 正規表示法驗證

**事件**

* ondblclick：雙點
* onchange：改變即做。
* onFocus：得到焦點
* onLoad：載入網頁
* onMousedown/up/over：滑鼠事件
* onBeforeUnload：離開前
* onScroll：捲動
* onSelect：選項改變
* 簡言之，透過DOM聆聽事件，接著透過event取得事件資訊做處理

**鍵盤事件**

* 鍵盤與滑鼠事件可透過DOM來處理，注意的是function直接呼叫，不用丟入參數。

function getKey(){

showMsg.innerHTML = event.keyCode;

document.onkeydown= getKey;

* document .onkeydown 🡪 事件監控 ，設成false，則該按鍵失效。
* event物件，取得事件發生時的資訊。
* window.event.keycode 🡪 取得鍵盤代碼。
* onKeyDown (鍵盤按下)
* onKeypress (鍵盤按下狀態), onKeyUp(釋放鍵盤)。
* altKey,ctrlKey,shiftKey

**滑鼠事件**

* event.clientX ,event.clientY (滑鼠位置)。
* event.button 🡪 取得點擊的滑鼠 , 1左2右4 中。
* onMouseDown/OverClick/onDblClick

**Cookie**

* 送出表單時，會把cookie送到伺服器端。
* 伺服器回傳時，則會回傳封包，裡面表頭就包含user cookie。

**正規表示法**

* re = new RegExp(pattern , flag)。
* re.test()，是否符合。
* string.search (re) , string.match (re) , string.replace (re,newStr) ,

**資料保護**

* 網頁內容保護 🡪 封鎖功能。
* 程式內容保護 🡪 escape() , unescape()。
* 先編碼，接著把編碼後的source code放入程式中,用eval執行反編碼。

**實作**

**99乘法表**

* 透過insertRow插入row，在透過其row插入cell，

(需要注意插入都要指定index)，透過innerHTML設定表格內容。

**日期**

* 透過Date物件，動態取得該月天數，以及目前的時間資訊。
* 動態產生select個數與內容。
* 透過event.target取得驅動事件之資訊。

**星星評價**

* 透過createElement動態建立物件，接著藉由此物鍵動態設定屬性，然後append到body。
* onclick事件比較特殊，需透過function直接設定，不能直接寫function名稱。
* 了解直接使用id與getElementById的使用時機，直接使用id必須為英文，較不彈性，透過getElemetByID，可對於一些索引處理較簡潔，視時機而用。

**圖片輪撥**

* 事件聆聽，物件.事件 = 要做的事情。

如imageControl.onmousedown= checkButton;需注意的是，JS裡面有些函數需要寫()，有些不用，如這邊寫checkButton()，則會有問題。

* 透過document.write，直接輸出標籤，建立所需內容。
* 透過setInterval去驅動執行時間。
* oncontextmenu="window.event.returnValue=false;" 關閉右鍵功能。

**表單檢查**

* reset按鈕為清除表單，控制項必須在表單內。
* reAccount = /^09[0-9]{2}-[0-9]{6}$/ ，基本的正規表示法。

**一些事件驅動function的注意事項**

* Onclick = onclick="checkInformation();
* imageBtn.onclick = function(){showStarCount();};
* imageControl.onmousedown= checkButton;

注意以上的差別，

* 第一個是控制項點擊的處理，需透過完整的呼叫函數字串，包含參數。
* 第二個是動態建立控制項後，我們要動態的設定onclick事件，寫法如上例，需寫出function(){完整函數}。
* 第三個是監聽事件，當事件發生時要交給哪個函式處理，這邊很特別，只要寫函數名稱，而不用代入參數，代入反而會有錯誤。

**清除內容(Reset)**

* 透過把innerHTML設為空白即可，如ninenine.innerHTML="";