--- < JAVA SCRIPT > ---

تتم كتابة تعليمات لغة الجافا سكريبت ضمن ملفات الـ HTML ضمن وسم <script> وذلك:

١- ضمن وسم <head> :

```
<head>
    <script >
    ……….
    </script>
</head>
```

٢- في أي مكان ضمن وسم <body> .

٣- يمكن كتابة التعليمات في ملف خارجي بشرط أن نحفظ الملف بامتداد ".js" مثلاً java1.js ، ثم يجب ربط هذا الملف بملف الصفحة:

```
<script  src="اسم الملف.js" > </script>
```

نستخدم الطريقة الثالثة عندما يكون حجم التعليمات كبيراً جداً من أجل سرعة تحميل الصفحة، كما أنها طريقة جيدة عندما نعرّف توابع و نريد الاستفادة منها في أكثر من صفحة واحدة ، أما عندما نريد التصريح عن توابع ضمن ملف الصفحة نفسها يفضل كتابتها في الـ head و ذلك لسهولة القراءة و التطوير.

مثال: الكود التالي يقوم بطباعة عبارة على الصفحة

```
<script >
   document.write("My first code in Javascript!<br>");
</script>
```

ملاحظة: نلاحظ من المثال السابق أننا تستطيع كتابة أي وسم بلغة HTML و لكن بشرط أن يكون ضمن علامات تنصيص " ".

مثال: طباعة عبارة على الصفحة بتنسيقات خط معينة ثم رسم خط أفقي

```
<script >
  document.write("<P><b><u> My second code in Javascript !</u></b> <P><hr/>");
</script>
```

المتحولات:

يمكن التصريح عن المتحولات في أي مكان و ذلك بذكر اسم المتحول فقط دون تحديد نمطه، فعندما نقوم بإسناد قيمة له يقوم بحجز المساحة المناسبة حسب نوع القيمة.

ملاحظة: هذه اللغة حساسة بالنسبة لحالة الأحرف أي يوجد فرق بين الأحرف الصغيرة و الكبيرة.

من أهم أنواع المتحولات لدينا : int , float , string , boolean , null , object حيث أن null تعني أنه لم يتم اسناد قيمة لهذا المتحول.

ملاحظة: من أجل التعليقات نكتب إما // أو /* */ .

عندما نجمع متحولين من نفس النمط تكون النتيجة أيضاً من نفس النمط ، أما إذا جمعنا عدد مع سلسلة محارف يكون الناتج سلسلة محارف.

مثال:

```
<script >
        var  x = 5;
        var  y = 2.5;
        var  z = "kg";
        var  b = true;
        document.write( "x = ", x ,"<br/>", "  y = " , y ,"<br/>", "  z = " , z ,"<br/>");
        document.write( "b=  ", b ,  "<br/>");
        var   a= y+z;
        document.write("y + z = ", a , "<br/>");
</script>
```

العمليات المطبقة على المتحولات:

+ , - , * , / , % باقي القسمة , += , -= , *= , /= , %= , ++ , --

معاملات المقارنة:

== , != , < , > , <= , >=

المعاملات المنطقية:

&& , || , !

بنى التحكم:

توجد نفس بنى التحكم المعروفة بلغة C# مثل: حلقات for ، while ، شرط … if () else ، …

المصفوفات:

يتم التصريح عنها كما يلي: a = new Array(10);

وهنا عرفنا مصفوفة اسمها a و مؤلفة من ١٠ عناصر، حيث أن عناصرها هي:

a[0] , a[1], …. , a[9] = a[n-1]

التوابع:

يمكن التصريح عن التوابع في أي مكان من الصفحة و استدعاؤها من أي مكان ، بحيث نكتب ما يلي:

```
function اسم(البارمترات اللازمة)
{
   ……….
   Return …;                //  عندما لا يعيد التابع قيمة لا داعي لهذه التعليمة
}
```

وذلك ضمن وسم < Script >

مثال: استدعاء تابع يقوم بإسناد قيم لمصفوفة و جمعها

```
<script >
     function  sum_of_array()
     {
        b=new Array(20);
        s=0;
        for ( i=0 ; i<20 ; i++ )
            {
               b[i] =  i ;
               s += b[i] ;
             }
        return s ;
     }
      document.write("sum = ",sum_of_array(),"<hr/>");
</script>
```

مثال: جمع و طرح عددين مدخلين من قبل المستخدم

```
<html>
 <head>
 </head>
 <body>
    <form name=f1>
        Number1: <input type="text"  name="n1"  value=0>    
        <input type="button"  name="add"  value="+" onclick="calc('+');"><br>
        Number2: <input type="text"  name="n2"  value=0>    
        <input type="button"  name="sub"  value="-"  onclick="calc('-');"><br>   
         Result =  <input type="text"   name="result"   value=0>   
        <input  type="reset"   name="clear"  value="C"><br>
    </form>


    <script >
        function calc(x)
        {
            num1 = f1.n1.value*1;
            num2 = f1.n2.value*1;
           if (x == '+')   f1.result.value =  num1+num2;
           else              f1.result.value = num1-num2;
        }
    </script>
</body>

</html>
```

تغيير لون خلفية الصفحة:

document.bgColor = " اسم لون أو لون محدد بالنظام الست عشري "

document.fgColor = " اسم لون أو لون محدد بالنظام الست عشري "

مثال: لنصمم فورم يحوي قائمتين منسدلتين، واحدة نختار منها لون الخلفية و الأخرى نختار منها لون النص:

```
<script>
    function bg(usercolor)
    {
        document.bgColor=usercolor;
    }
    function fg(usercolor)
    {
        document.fgColor=usercolor;
    }
</script>
<form name=f1>
Choose BgColor : <select name=bgc onchange="bg(bgc.value);">
                    <option value="blue">Blue
                    <option value="green">Green
                    <option value="red">Red
                    <option value="#ffffff">White
                </select>
<br>
Choose FgColor : <select name=fgc onchange="fg(fgc.value);">
                    <option value="blue">Blue
                    <option value="green">Green
                    <option value="red">Red
                    <option value="#ffffff">White
                </select>
</form>
```

مثال: لنغير لون الخلفية من خلال الضغط على صورة:

<img src="pic.jpg" width="100" height="100" onclick="bg('blue');">

توابع تحويل السلاسل المحرفية:

parseInt() : يحول سلسلة محرفية إلى عدد صحيح.

parseFloat() : يحول سلسلة محرفية إلى عدد حقيقي.

eval() : يحسب قيمة سلسلة محرفية تحتوي على تعبير رياضي (evaluation = تقييم).

- **رسالة نصية:**

من أجل إظهار نافذة صغيرة تحوي رسالة ما و مع زر "Ok" نكتب: alert(message);

مثال:

```
<script>
        alert (" welcome in my Site! ");
</script>
```

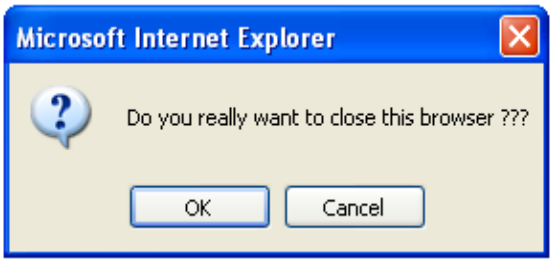
- **التابع "confirm()" – نافذة التأكيد:**

يقوم هذا التابع بإظهار رسالة تأكيد (نص ما) للمستخدم، بحيث يجب عليه أن يختار إما . "Ok" أو "إلغاء الأمر".

يعيد هذا التابع قيمة بوليانية (true/false) و نستطيع حسب اختيار المستخدم أن نقوم بتنفيذ تعليمات مناسبة. يأخذ هذا التابع بارامتر واحد وهو عبارة عن النص الذي نريده أن يظهر للمستخدم.

مثال: عند النقر على رابط تشعبي وظيفته إغلاق نافذة المستعرض ، تظهر رسالة تأكيد أولاً ، فإذا ضغط المستخدم Ok تغلق النافذة فعلياً.

```
<a href = "javascript:exit();" > Close this browser ! </a>
<script >
        function exit()
        {
          var c=confirm("Do you really want to close this browser ???");
          if (c)            // c = = true
            {
              alert("Good Bay , see you soon !!!");
              close();
             }
}
</script>
```



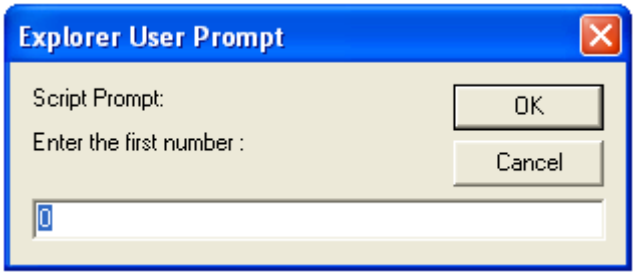
- **التابع "prompt()" – نافذة الإدخال:**

يقوم هذا التابع بإظهار رسالة تطلب من المستخدم إدخال قيمة معينة ثم يعيد القيمة المدخلة و نستطيع معالجتها كما نريد. يأخذ هذا التابع بارامترين: نص و قيمة افتراضية .

prompt(نص , قيمة افتراضية);

مثال: لنقوم بطلب إدخال عددين من قبل المستخدم، ثم نظهر ناتج مجموعهما في رسالة alert();.

```
<script >
n1=prompt("Enter the first number : ",0);
n2=prompt("Enter the second number : ",0);
if (n1&&n2)
  {
    sum=parseInt(n1)+parseInt(n2);
```



```
    alert(n1+" + "+n2+" = "+sum);
  }
else
     alert("No solution! Because You've canceled the summation of the 2 numbers !");
</script>
```

- **الغرض Date() :**

يعيد التاريخ و الوقت الحالي، حيث يمكن الحصول على تفاصيل التاريخ و الوقت باستدعاء التوابع الخاصة بها كما يلي:

```
<script >
 var d=new Date();
 document.write("Date= ", d.getDate(), " - ", d.getMonth()+1, " - ",d.getYear(), "<br>");
 document.write("Time= ", d.getHours(), " : ", d.getMinutes()," : ", d.getSeconds(),"<br>");
</script>
```

Date = 14-5-2015
Time = 08:18:40

ملاحظة: إن التابع getMonth() يعيد رقم الشهر ضمن المجال (٠، ١١) ، و يجب علينا إضافة ١ للحصول على رقم الشهر الحالي.

مثال: لنطبع التاريخ و الوقت في مربعي نص بحيث أن يكون الوقت بشكل متغير (أي كالساعة)، علماً أننا نكتب تابعين، واحد لتوليد التاريخ و الآخر لتوليد الوقت و يتم استدعاءهما تلقائياً عند تحميل الصفحة.

```
<head>
        <script >
                function date()
                {
                   var d=new Date();
                   f1.dat.value=d.getDate()+" : "+(d.getMonth()+1)+" : "+d.getYear();
                   setTimeout("date()",100000);
                }
                function time()
                {
                   var d=new Date();
                   f1.tim.value=d.getHours()+" : "+d.getMinutes()+" : "+d.getSeconds();
                   setTimeout("time()",1000);
                }
        </script>
</head>
<body onload="time(); date();">
        <form name=f1>
         Date = <input type=text name=dat><br>
        Time = <input type=text name=tim><br>
```

</form>

ملاحظة: إن التابع setTimeout("functionname()" , time); يقوم باستدعاء تابع ما و ذلك بعد انتظار فترة زمنية معينة مقاسة بالميلي ثانية ، أي تعادل القيمة ١٠٠٠ ثانية واحدة.

أغراض الجافاسكريبت – Javascript Objects :

يوجد في جافا سكريبت عدد كبير من الأغراض الجاهزة بحيث أنه توجد لكل غرض مجموعة من الصفات و مجموعة من التوابع. لقد تعرفنا على عدة أغراض مثل غرض المصفوفة Array و Date و ... ، لنرى بعض الأغراض الأخرى:

ملاحظة هامة: يجب الانتباه إلى حالة الأحرف (صغيرة/كبيرة) !!!

الغرض document	
بعض الصفات:	
document.bgColor=""	لون الخلفية
document.fgColor=""	لون الكتابة
document.linkColor=""	لون الرابط غير المضغوط
document.alinkColor=""	لون الرابط الفعال
document.vlinkColor=""	لون الرابط الذي تمت زيارته
document.title=""	عنوان النافذة
document.URL=""	اسم الصفحة المحملة
document.fileSize	يعطي حجم ملف الصفحة
document.lastModified	يعطي تاريخ و وقت آخر تعديل تم على ملف الصفحة
document.fileCreatedDate	يعطي تاريخ إنشاء ملف الصفحة
document.location	يعطي المسار الكامل لملف الصفحة
بعض التوابع:	
document.write()	للطباعة على الصفحة
document.getElementByID()	من أجل الوصول إلى عناصر الصفحة من خلال الـ id
document.getElementsByName()	من أجل الوصول إلى عناصر الصفحة من خلال الـ name
document.getElementsByTagName()	من أجل الوصول إلى عناصر الصفحة من خلال اسم الوسم
document.createAttribute()	لإضافة صفة إلى الصفحة
document.createElement()	لإضافة كائن جديد إلى الصفحة

أمثلة:

❖ إن الكود التالي يقوم بتغيير عنوان الصفحة و ذلك بعد النقر على زر:

```
<head><title>  الكلية التطبيقية  </title></head>
<body>
    <form>
    <input type="button"  value="تغيير العنوان"  onclick="document.title='!!!أهلاً و سهلاً بكم' ">
    </form>
</body>
```

❖ إن الكود التالي يقوم بتحميل صفحة جديدة بنفس النافذة:

```
<form>
<input type=button value="Google" onclick="document.URL='http://www.google.com' ">
</form>
```

❖ إن الكود التالي يقوم بإضافة إشارة التحقق للكائن "checkbox" و ذلك بتمرير اسم الكائن كمتحول لتابع :

```
<form name="form1">
<input type=checkbox name="ch" value="1">First Check<br>
<input type=checkbox name="ch" value="2">Second Check <br>
<input type=checkbox name="ch" value="3">Third Check <br>
<input type=button name="but1" value="Check the checkboxes !" onclick="check('ch');">
</form>

<script >
function check(x)
{
  document.getElementsByName(x)[0].checked=true;
  document.getElementsByName(x)[1].checked=true;
  document.getElementsByName(x)[2].checked=true;
}
</script>
```

<table>
<tr><th colspan="3">الغرض window</th></tr>
<tr><td colspan="3">بعض الصفات:</td></tr>
<tr><td>window.status</td><td colspan="2">قيمة النص المعروض في شريط الحالة (الشريط بأسفل نافذة المستعرض)</td></tr>
<tr><td>window.defaultStatus</td><td colspan="2">قيمة النص الافتراضي المعروض في شريط الحالة</td></tr>
<tr><td>window.name</td><td colspan="2">الاسم البرمجي للنافذة</td></tr>
<tr><td colspan="3">بعض التوابع:</td></tr>
<tr><td>window.close()</td><td colspan="2">لإغلاق نافذة المستعرض</td></tr>
<tr><td>window.open()</td><td colspan="2">لفتح نافذة جديدة مع تحديد مواصفات النافذة، علماً أنه يجب تمرير ٣ بارامترات أساسية بالشكل التالي:
window.open("اسم الصفحة" , "اسم برمجي" , "مجموعة من المواصفات");
من مواصفات النافذة لدينا:
<table>
<tr><th>الصفة</th><th>الوظيفة</th></tr>
<tr><td>width = * pixel</td><td>عرض النافذة</td></tr>
<tr><td>height = * pixel</td><td>ارتفاع النافذة</td></tr>
<tr><td>left = * pixel</td><td>المسافة الفاصلة ما بين حد الشاشة الأيسر و حافة النافذة اليسرى</td></tr>
<tr><td>top = * pixel</td><td>المسافة الفاصلة ما بين حد الشاشة الأعلى و حافة النافذة العلوية</td></tr>
<tr><td>menubar = yes | no</td><td>إظهار شريط القوائم</td></tr>
<tr><td>toolbar = yes | no</td><td>إظهار شريط الأدوات</td></tr>
<tr><td>scrollbar = yes | no</td><td>إظهار أشرطة التمرير</td></tr>
<tr><td>status = yes | no</td><td>إظهار شريط الحالة</td></tr>
<tr><td>resizable = yes | no</td><td>إمكانية تغيير حجم النافذة</td></tr>
<tr><td>directories = yes | no</td><td>إظهار شريط أدوات خاص</td></tr>
<tr><td>location = yes | no</td><td>إظهار شريط العنوان</td></tr>
</table></td></tr>
<tr><td>window.confirm()</td><td colspan="2">إظهار نافذة للتأكيد</td></tr>
<tr><td>window.prompt()</td><td colspan="2">إظهار نافذة للإدخال</td></tr>
<tr><td>window.alert()</td><td colspan="2">إظهار نافذة رسالة</td></tr>
<tr><td>.....</td><td colspan="2"></td></tr>
</table>

بعض التوابع:	
window.moveTo(x,y)	نقل النافذة إلى موقع جديد على الشاشة (مقاسة بالبكسل)
window.resizeTo(x,y)	تغيير حجم النافذة (بالبكسل)
window.scrollTo(x,y)	الانتقال إلى إحداثيات معينة ضمن النافذة

ملاحظة: عند استدعاء إحدى توابع الغرض window ليس من بالضرورة أن نذكر اسم الغرض أولاً إذا كنا بنفس النافذة! أي إن التعليمتان التاليتان متطابقتان:

```
window.alert("Hello Students!");
alert("Hello Students!");
```

الغرض String	
ليكن لدينا متحول من نوع String ، مثلاً: s="Hello Students";	
s.length	تعطي طول السلسة المحرفية (أي عدد المحارف) وإذا استخدمناها مع المصفوفات فتعطي عدد عناصر مصفوفة
s.substring(i,j)	يقوم بنسخ جزء من السلسة بدءاً من الدليل الأول و حتى الدليل الثاني
s.toUpperCase()	يعيد السلسة بأحرف كبيرة
s.toLowerCase()	يعيد السلسلة بأحرف صغيرة
s.charAt(i)	يعيد المحرف الواقع في الدليل الذي نمرره
s.indexOf(s2)	يعيد دليل بداية سلسلة جزئية من سلسلة أساسية
s.lastIndexOf(s2)	يعيد دليل بداية آخر نسخة عن سلسلة جزئية من سلسلة أساسية (أي لما تكون السلسة الجزئية مكررة في السلسلة الأساسية، فتهمنا بداية آخر نسخة)
s.fontsize(i)	تغيير حجم خط السلسة
s.italics()	جعل خط السلسة مائلاً
s.bold()	جعل خط السلسة عريضاً
s.small()	جعل خط السلسة أصغر بدرجة
s.big()	جعل خط السلسة أكبر بدرجة
s.sup()	جعل خط السلسة مرتفعاً
s.sub()	جعل خط السلسة منخفضاً
s.strike()	جعل خط السلسة مخططة من الوسط

أمثلة: بفرض أن s="Hello Students , Hello World"; يصبح لدينا ما يلي:

```
document.write("<br>",s.fontsize(14));   → Hello Students , Hello World
document.write("<br>",s.italics());      → Hello Students , Hello World
document.write("<br>",s.bold());         → Hello Students , Hello World
document.write("<br>",s.strike());       → Hello Students , Hello World
document.write("<br>",s.small());        → Hello Students , Hello World
document.write("<br>",s.big());          → Hello Students , Hello World
document.write("<br>",s.charAt(4));      → o
document.write("<br>",s.length);         → 28
document.write("<br>X ",s.sup());        → X Hello Students , Hello World
document.write("<br>X ",s.sub());        → X Hello Students , Hello World
document.write("<br>",s.toLowerCase());  → hello students , hello world
document.write("<br>",s.toUpperCase());  → HELLO STUDENTS , HELLO WORLD
document.write("<br>",s.indexOf('o s')); → 4
document.write("<br>",s.lastIndexOf('hel')); → 17
document.write("<br>",s.indexOf('wrong')); → -1
document.write("<br>",s.substring(6,13)); → Students
```

Math الغرض	
بعض التوابع:	
Math.abs(x)	يعطي القيمة المطلقة للعدد
Math.pow(x,n)	يحسب نتيجة رفع العدد الأول لقوة ما
Math.sqrt(x)	يحسب الجذر التربيعي للعدد
Math.ceil(x)	يدور العدد إلى أقرب عدد صحيح أكبر منه
Math.floor(x)	يدور العدد إلى أقرب عدد صحيح أصغر منه
Math.round(x)	يدور العدد إلى أقرب عدد صحيح
Math.random()	يعيد رقماً عشوائياً محصور ما بين ٠ و ١
Math.sin(x)	يحسب جيب عدد ما
Math.cos(x)	يحسب تجب عدد ما
Math.tan(x)	يحسب ظل عدد ما
Math.asin(x)	يحسب الجيب العكسي لعدد ما بالراديان
Math.acos(x)	يحسب التجب العكسي لعدد ما بالراديان
Math.atan(x)	يحسب الظل العكسي لعدد ما بالراديان
Math.log(x)	يعيد اللوغاريتم الطبيعي للعدد
Math.exp()	يحسب القيمة "" (أساس اللوغاريتمات الطبيعية) مرفوعة إلى قوة

Math.max(x,y)	يعيد العدد الأكبر ما بين عددين
Math.min(x,y)	يعيد العدد الأصغر ما بين عددين
بعض الثوابت:	
Math.E	يعطي أساس اللوغاريتمات الطبيعية
Math.LN2	يعطي اللوغاريتم الطبيعي للعدد ٢
Math.LN10	يعطي اللوغاريتم الطبيعي للعدد ١٠
Math.LOG2E	يعطي اللوغاريتم ذو الأساس ٢ للعدد النيبريE
Math.PI	يعطي القيمة ٣.١٤١٥٩ و التي عبارة عن نسبة محيط الدائرة إلى قطرها
Math.SQRT1_2	يعطي الجذر التربيعي للنصف
Math.SQRT2	يعطي الجذر التربيعي للعدد ٢

أمثلة:

```
document.write("<br>",Math.E);          → 2.718281828459045
document.write("<br>",Math.LN2);        → 0.6931471805599453
document.write("<br>",Math.LN10);       → 2.302585092994046
document.write("<br>",Math.LOG2E);      → 1.4426950408889633
document.write("<br>",Math.PI);         → 3.141592653589793
document.write("<br>",Math.SQRT1_2);    → 0.7071067811865476
document.write("<br>",Math.SQRT2);      → 1.4142135623730951
document.write("<br>",Math.abs(-34));   → 34
document.write("<br>",Math.pow(2,3));   → 8
document.write("<br>",Math.sqrt(25));   → 5
document.write("<br>",Math.ceil(2.6));  → 3
document.write("<br>",Math.floor(2.6)); → 2
document.write("<br>",Math.round(2.6)); → 3
document.write("<br>",Math.random());   → 0.37958163862329386
document.write("<br>",Math.sin(90));    → 0.8939966636005578
document.write("<br>",Math.cos(90));    → -0.4480736161291701
document.write("<br>",Math.tan(45));    → 1.6197751905438615
document.write("<br>",Math.asin(0));    → 0
document.write("<br>",Math.acos(90));   → 1.5707963267948965
document.write("<br>",Math.atan(45));   → 1.5485777614681775
document.write("<br>",Math.log(10));    → 2.302585092994046
document.write("<br>",Math.exp(10));    → 22026.465794806718
document.write("<br>",Math.max(5,7));   → 7
document.write("<br>",Math.min(5,7));   → 5
```