

بسم الله الرحمن الرحيم

درس تصميم معرض إحترافي للصور بإستخدام الجافا سكربت

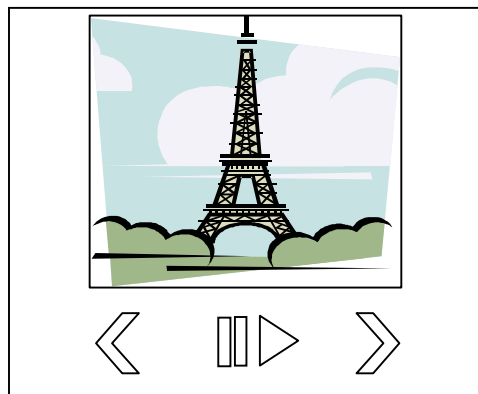
في هذا الدرس سوف أشرح عملية إنشاء معرض تفاعلي للصور بإستخدام الجافا سكربت.

لن أقوم في بشرح أكواد HTML كما أنني لن أقوم بشرح أكواد JavaScript ولكن ما سوف أقوم بشرحة هو كيفية الإستفادة من و إستغلال ال HTML و الجافا سكربت من أجل الوصول إلى أداء مهمة معينة.

- سنقوم أولاً بإنشاء معرض بسيطة (الوظيفة الأساسية)
- بعدها سنقوم بعمل بعض الإضافات و التأثيرات على المعرض
- ثالثاً سوف نقوم بعمل بعض التحسينات على شكل المعرض (Makeup)

1- الشكل البسيط:

في البداية سوف نرسم الشكل العام للمعرض.



كما نرى فإن هنالك 3 أزرار: التالي – السابق – تشغيل/إيقاف. إضافة إلى الصورة المعروضة. طبعاً وظيفة معرض الصور هو عرض مجموعة من الصور في مكان واحد على صفحة إنترنت بدون الحاجة إلى إعادة تحميل الصفحة من أجل التنقل بين الصور.

سنبدأ بكود ال HTML التالي لصفحة إنترنت فارغة، حيث سوف نقوم مبدئياً بوضع الكود في وسم <Script> في داخل قسم ال Head من الصفحة.

```
<html>
<head>
<script type="text/javascript">
//Here Our Code Will Be
</script>
</head>
<body>

</body>
</html>
```

بعد ذلك سوف نضيف بعض ال HTML كود من أجل وضع زررين مبدئيين (التالي، السابق)، إضافة إلى كائن صورة لكي يتم عرض الصورة في داخله، سيتم ترتيب هذه المكونات في جدول <Table> من أجل الترتيب.

ضع الكود التالي داخل ال <Body>:

```
<table align="center">
```

```

<tr>
<td ><img id="img" src="" /></td>
</tr>
<tr>

<td align="center">
<input type="button" value="Previous" />
<input type="button" value="Next" />
</td>
</tr>

</table>

```

شئنا في الكود السابق سيكون من المهم لك التعرف عليهما، الأول عليك ملاحظة أن الضغط على الزر Previous يتم استدعاء الدالة prev() من كود الجافا سكريبت، كذلك الأمر مع زر التالي يتم استدعاء الدالة next()، الأمر الآخر الذي يلزمك ملاحظته في وسم ال حيث أننا قمنا بإعطاء الـ id باسم "img" حيث سيكون بإمكاننا الوصول إلى هذا الـ وسم من داخل السكريبت.

الآن نحن بحاجة إلى شيء (مصفوفة) من أجل تخزين روابط الصور التي سوف يتم عرضها في المعرض، لذلك سوف نقوم بتعريف مصفوفة في أعلى الـ وسم الخاص بالسكريبت في ال <Head>:

```
var Imageslist =new Array();
```

بعدها يجب علينا عمل دالة حيث يتم تعبئة المصفوفة بروابط الصورة داخل هذه الدالة، طبعاً من يقوم بتعبئة المصفوفة هو مصمم صفحة الإنترنت من خلال التعديل على السكريبت، أو من خلال عمل سكريبت إضافي. وهذه الدالة يجب استدعاؤها عند تحميل الصفحة.

```

function LoadImages() {
    addImage ('1.jpg');
    addImage ('2.jpg');
    addImage ('3.jpg');
    addImage ('4.jpg');
    addImage ('5.png');

    setImage();
}

```

تقوم هذه الدالة بإستدعاء دالة أخرى تسمى addImage() و ترسل إليه رابط للصورة، في كل مرة يتم استدعاء الدالة addImage() يتم إرسال رابط لأي صورة، لاحظ أن الصورة قد تكون في أي موقع وقد تكون تحمل أي اسم؛ في حالتنا هنا لدينا 5 صور موجوة في نفس المجلد الذي تتواجد فيه الصفحة. لاحظ أيضاً أن الصورة رقم 5 تحمل الإمتداد png. بعد إضافة روابط الصور إلى المصفوفة يتم استدعاء الدالة setImage() حيث سنتحدث عنها لاحقاً.

```

function addImage(url) {

    Imageslist[count]=url;
    count++;
}

```

الدالة addImage() تستقبل متغير يحمل رابط الصورة كما قلنا سابقة و تضيفه إلى نهاية المصفوفة. طبعاً من الواضح أننا سوف نحتاج لتعريف المتغير count في بداية كود السكريبت (اسفل تعريف المصفوفة).

```
var count=0;
```

كما قلنا سابقا يلزمنا إستدعاء الدالة Loadimages() عند تحميل الصفحة من أجل تجهيز المصفوفة وتعبئتها بالروابط، إذن يجب علينا التعديل على الوسم <Body> الخاص بالصفحة ووضعه بالشكل التالي:

```
<body onload="LoadImages () ">
```

بعد ذلك سوف نقوم ببرمجة الأزرار بحيث عند الضغط على زر التالي أو السابق يتم الإنتقال إلى الصورة التالية أو السابقة، يتم عمل ذلك من خلال إستدعاء كل من الدالتين next() و prev() عند الضغط على الأزرار، إذن يلزمنا تعديل وسوم الأزرار كالتالي:

```
<input type="button" value="Previous" onclick="prev ()"/>
<input type="button" value="Next" onclick="next ()"/>
```

أيضا شكل كل من الدالتين سيكون كالتالي:

```
function next()
{
    if(current >= (Imageslist.length-1))
        current=0;
    else
        current ++;

    setImage ();
}

function prev()
{
    if(current <= 0)
        current = Imageslist.length-1;
    else
        current --;

    setImage ();
}
```

لدينا متغير بإسم current يمثل رقم index الصورة المعروضة من المصفوفة. حيث يتم في داخل الدالة next فحص قيمة المتغير current إذا كانت اكبر أو مساوية من عدد الصور الموجودة في المصفوفة يتم إعطائها القيمة 0 (ما يعني عند الوصول إلى الصورة الأخيرة يتم الإنتقال إلى الصورة الأولى)، إذا لم تكن قيمة ال current اكبر أو يساوي عدد الصور في المصفوفة يتم زيادة قيمة المتغير بواحد من أجل الإنتقال إلى الصورة التالي. نفس الأمر مع الدالة prev ولكن يتم فحص فيما إذا كانت قيمة المتغير تساوي أو اقل من صفر يتم إعطاء المتغير ال index الخاص بالصورة الأخيرة (ما يعني إذا كنا نقف على أول صورة و تم الضغط على السابق يتم الإنتقال إلى الصورة الأخيرة). إذا لم تكن قيمة المتغير مساوية لصفر يتم طرح واحد منه. طبعا من الواضح أنه يلزمنا تعريف متغير بإسم current في أعلى كود السكريبت (تحت تعريف المتغير count).

```
var current=0;
```

الآن نعود إلى الدالة setImage(), كما لاحظنا فإن الدالة next و الدالة prev تقوم أيضا بإستدعاء الدالة setImage(). ربما تكون قد تكهنت بشكل هذه الدالة حيث تتمثل الدالة بالكود البسيط التالي:

```
function setImage()
{
    document.getElementById("img").src=Imageslist[current];
}
```

يتم في هذه الدالة تغيير قيمة خاصية ال src الذي يمثل رابط الصورة الخاص بالوسم الذي يحمل ال id المسمى img، حيث يتم إسناد القيمة الموجودة في الفهرس current في المصفوفة الخاصة بروابط الصور. طبعا يتم إستدعاء هذه الدالة من ثلاث أماكن كما

لاحظنا، حيث يتم استدعائها من داخل الدالتين next() و prev() بحيث يتم تغيير قيمة المتغير current قبل استدعاء هذه الدالة. أيضا عندما يتم استدعاء هذه الدالة من داخل الدالة LoadImages() يكون المتغير current يحمل القيمة صفر؛ ما يعني أنه سوف يتم عرض الصورة الأولى.

في هذه اللحظة بإمكانك حفظ الكود في ملف html و فتحه في المتصفح، ستلاحظ أننا قد إنتهينا عمليا من الوظيفة الأساسية لمعرض الصور، حيث يتم عرض الصور مع إتاحة زررين للتنقل بين الصور.

2- إضافات و تأثيرات:

أول الإضافات التي سوف نقوم بعملها هي زر لتشغيل العرض التقديمي و إيقافه، حيث يبدأ عند الضغط على زر Play العرض (ويتحول الزر إلى Pause) وعند الضغط على الزر Pause يتم إيقاف تشغيل العرض. في العرض يتم الإنتقال إلى الصورة التالية بعد 3 ثواني.

لعمل ذلك سوف نحتاج أول لتعريف زر بين زرري التالي و السابق، سيكون تعريف هذا الزر بالشكل التالي:

```
<input type="button" value="Play" id="plypus" onclick="PlayPause()" />
```

كما نلاحظ أنه عند الضغط على الزر يتم استدعاء الدالة PlayPause()، أيضا الزر يحمل ال id = plypus مما سوف يمكننا من الوصول إليه من داخل السكريبت (حيث سوف نقوم بتغيير قيمة من Play إلى Pause و العكس).

الدالة PlayPause() :

```
function PlayPause()
{
    if(play==0)
    {
        play=1;
        document.getElementById("plypus").value="Pause";
        pt=setTimeout("playSlideShow()", 3000);
    }else
    {
        play=0;
        document.getElementById("plypus").value="Play";
        clearTimeout(pt);
    }
}
```

في هذه الدالة يتم فحص قيمة متغير باسم play إذا كان يحمل القيمة 0 (أي أن العرض متوقف) يتم إعطاء هذا المتغير القيمة 1 و بعدها يتم تعديل النص الظاهر في الزر الخاص بالتشغيل و كتابة "Pause" في داخله، بعد ذلك يتم استخدام الدالة (المضمنة في الجافا سكريبت) setTimeout() حيث تقوم هذه الدالة باستدعاء الدالة playSlideShow() بعد 3 ثواني (3000 ملي ثانية = 3 ثواني). -هنا أفترض ان لديك معرفة مسبقة بطريقة عمل ال setTimeout و ال clearTimeout إذا لم يكن لديك هذه المعرفة عليك بالرباط [التالي](#)-. إذا كان المتغير play يحمل القيمة 1 يتم تحويل قيمته إلى 0 و يتم تغير النص في الزر، و بعدها يتم إيقاف المؤقت الذي يستدعي الدالة playSlideShow() بشكل متكرر.

من الواضح أننا بحاجة لتعريف المتغيرين play و pt و ذلك في أعلى كود السكريبت (تحت تعريف ال current):

```
var play=0;
var pt;
```

بالنسبة للدالة playSlideShow() فهي بسيطة جدا و تحمل الشكل التالي:

```
function playSlideShow()
{
    pt=setTimeout("playSlideShow()",3000);
    next();
}
```

حيث تقوم الدالة بإستدعاء نفسها كل 3 ثواني، و بعدها يتم إستدعاء الدالة next() من أجل الإنتقال إلى الصورة التالية.

الإضافة التالية التي سوف نقوم بعملها، هي إضافة تأثير Fade-in (تلاش الدخول) عند الإنتقال من صورة إلى أخرى، مما سوف يكسب المعرض جانب جمالي كبير.

أول شيء سوف نحتاج لتطبيق هذه الفكرة هو دالة تقوم بتغيير قيمة شفافية اشباع الصورة "Opacity"، لذلك حصلنا على هذه الدالة التي تستقبل كائن، إضافة إلى قيمة بين ال 0 و ال100 و تقوم بإعطاء الكائن شفافية بناء على هذه القيمة (0 تعني شفافية تامة، الجسم مخفي). حصلنا على هذا الكود من [هنا](#).

```
function setOpacity(obj, opacity) {

    opacity = (opacity == 100)?99.999:opacity;

    // IE/Win
    obj.style.filter = "alpha(opacity:"+opacity+")";

    // Safari<1.2, Konqueror
    obj.style.KHTMLOpacity = opacity/100;

    // Older Mozilla and Firefox
    obj.style.MozOpacity = opacity/100;

    // Safari 1.2, newer Firefox and Mozilla, CSS3
    obj.style.opacity = opacity/100;

}
```

من الواضح أن الكود يقوم بوضع الشفافية للعنصر بناء على المتصفح الذي يستخدمه المستخدم.

سوف نحتاج أيضا إلى دالة أخرى بحيث تقوم هذه الدالة بتقليل شفافية الصورة على مدى زمن معين، ما يعني أن الصورة تبدأ بشفافية تامة و تنتهي بصورة بدون أي شفافية، سوف نحتاج إلى الدالة ImgOpen() لعمل ذلك:

```
function ImgOpen ()
{
    op++;

    img=document.getElementById("img");
    setOpacity(img,op);
    t=setTimeout("ImgOpen()",10);

    if(op >=100)
    {
        op=0;
        clearTimeout(t);
    }
}
```

سنحتاج لتعريف المتغير op الذي يمثل الشفافية، و المتغير t لازم لعمل دالة التوقيت، إضافة إلى المتغير loadT الذي سوف يتم استخدامه لاحقا، في أعلى الكود (تحت تعريف المتغير pt و play):

```
var t,loadT,op=0;
```

نعود للدالة السابقة (ImgOpen) حيث يتم زيادة قيمة الشفافية في كل مره يتم فيها دخول هذه الدالة، مع العلم بعدها يتم إسناد قيمة الشفافية الحالية الممثلة بالمتغير op إلى الصورة المعروضة حاليا من خلال الدالة (setOpacity)، بعد ذلك تقوم الدالة بإستدعاء نفسها كل 10 ملي ثانية، عندما تصبح قيمة الإشباع تساوي أو اكبر من 100 يتم إيقاف المؤقت الذي يستدعي الدالة و تتوقف عن التنفيذ، ويتم إسناد القيمة 0 لمتغير الإشباع\الشفافية.

الآن علينا إستدعاء هذه الدالة من داخل الدالة (setImage) حيث يتم تطبيق مؤثر الشفافية في كل مرة تتغير فيها الصورة. قبل عمل ذلك دعنا نفكر بالسيناريو التالي: لو فرضنا أن هنالك صورة معروضة، وقام المستخدم بالضغط على زر التالي وكانت الصورة التالية كبيرة الحجم، إذن سوف يؤدي ذلك إلى تطبيق تأثير التلاشي على الصورة المعروضة وهذا شيء خاطئ!! لأننا نريد تطبيق المؤثر على الصورة التالية أثناء ظهورها.

لحل هذه المشكلة سوف نقوم بتعريف وسم صورة مؤقت بحيث يتم تحميل الصورة التالية إليه ولكن بدون إظهارها، و عندما نتأكد أن الصورة تم تحميلها يتم إظهارها في وسم الصورة الأساسي و تطبيق تأثير التلاشي عليها، لذلك دعنا نبدأ بإضافة السطر التالي إلى جانب وسم الصورة السابق داخل كود ال HTML، لاحظ أننا أعطينا خاصية العرض لهذه الصورة القيمة 0 ما يعني أنها لن تظهر على الشاشة، كما أننا أعطيناها ال id يحمل القيمة tmpImg.

```
<img id="tmpImg" src="" width="0"/>
```

بعد ذلك توجه إلى الدالة (setImage) وقم بتعديلها بحيث تصبح بالشكل التالي:

```
function setImage()  
{  
    document.getElementById("tmpImg").src=Imageslist[current];  
    var tmpImg=document.getElementById("tmpImg");  
    if(tmpImg.complete && !(op >0 && op < 100))  
    {  
        document.getElementById("img").src=Imageslist[current];  
        ImgOpen();  
        clearTimeout(loadT);  
    }else{  
        loadT=setTimeout("setImage()",10);  
    }  
}
```

في السطر الأول نقوم بتحميل الصورة الحالية من المصفوفة إلى الصورة المؤقتة، بعدها في السطر الثاني و الثالث نفحص إذا كان قد تم الإنتهاء من تحميل الصورة أمل لا (من خلال الشرط tmpImg.complete) قمنا بإضافة إلى هذا الشرط أيضا شرط يفحص إذا ما كانت قيمة متغير الإشباع أكبر من صفر أو أقل من 100 ما يعني أن لن يكون بالإمكان الإنتقال إلى الصورة التالية إذا كانت الصورة الحالية في مرحلة الظهور. إذا تحقق كل من الشرطين السابقين يتم تحميل الصورة إلى وسم الصورة الأساسي و يتم تطبيق تأثير تلاشي الدخول على تلك الصورة (من خلال إستدعاء الدالة imgOpen) كما يتم إيقاف مؤقت التكرار. إذا لم يتحقق الشرطين السابقين تقوم الدالة بإستدعاء نفسها بعد 10 ملي ثانية.

بهذا نكون قد حصلنا على معرض للصور مع خيار إضافي لتشغيل العرض التلقائي، إضافة إلى المؤثر الجميل الخاص بتلاشي الدخول للصور.

Makeup -3

سنقوم في هذه المرحلة بعمل بعض التحسينات الإضافية التي تهدف إلى زيادة جمالية المعرض و ترتيبه.

اول هذه التحسينات، سوف نقوم بعرض صورة صغيرة متحركة لتشير أنه جاري تحميل الصورة أثناء فتحها يمكنك الحصول على هذه الصورة من [هنا](#) الرابط. إذن قم بإضافة خلية ثانية للسطر الثاني في الجدول، وذلك من خلال إضافة الكود التالي بعد ال `</td>` الموجود مباشرة بعد الوسم الخاص بزر "التالي" في ال HTML:

```
<td width="35" height="40">
  
</td>
```

هنا نفترض أن الصورة موجودة في مجلد "Stuff" أيضا قم بإضافة الخاصية التالية إلى وسم أول خلية في الجدول `<td>` بحيث تصبح كالتالي، وذلك لكي يظهر الجدول بشكل جيد:

```
<td colspan="2" >
```

بعدها إتجه إلى الدالة `setImage()` وقم بإضافة السطر التالي داخل الشرط مباشرة بعد `ImgOpen()`:

```
document.getElementById("loadImg").width="0";
```

ثم في نفس الدالة ضع السطر التالي داخل الجزء الغير متحقق من الشرط بعد جملة `...loadT=setTimeout`

```
document.getElementById("loadImg").width="35";
```

في السطرين السابقين إذا كانت الصورة لم يتم تحميلها بعد يتم إظهار صورة مؤشر التحميل من خلال إعطائه العرض 35، ولكن إذا كانت الصورة قد إنتهى تحميلها يتم إخفاء المؤشر من خلال تحديد العرض 0 لمؤشر التحميل و بالتالي لن يظهر.

ثاني التحسينات التي سوف نجريها هي قيامنا باستخدام صور جميلة بدل من الأزرار التي نستعملها، يمكنك الحصول على صور الاسهم من [هنا](#) كما يمكنك الحصول على صور تشغيل\إيقاف من [هنا](#). لإستخدام صورهِ بدل زر "next". قم بإستبدال وسم الزر `next` بالكود التالي:

```
<a href="javascript:next()"></img></a>
```

وكذلك الأمر مع زر السابق "Previous":

```
<a href="javascript:prev()"></img></a>
```

بالنسبة لزر تشغيل\إيقاف "Play/Puase"، إستخدم الكود التالي:

```
<a href="javascript:PlayPause()"></img></a>
```

لاحظ أننا وضعنا ال `id` داخل وسم ال `<img>` لكي يتم تبديل الصور بصور الإيقاف عندما يتم الضغط تشغيل و العكس صحيح، لعمل ذلك إتجه إلى الدالة `PlayPuase()` وقم بإستبدال السطر الذي يقوم بتغيير نص الزر إلى `Pause` بالسطر التالي:

```
document.getElementById("plypus").src="stuff/puase.png";
```

وقم باستبدال السطر الذي يقوم بتغيير نص الزر إلى Play:

```
document.getElementById("plypus").src="stuff/Play.png";
```

جميل حاول التشغيل الآن، ☺ نتيجة جميلة.

الآن سوف نقوم بعمل توهج حول الصور بحيث يظهر هذا التوهج عند تمرير الماوس فوق قم بتعديل صور أزرار التالي و السابق و إضافة وهج حول كل من الأزرار وقم بتسمية الزر بإسمه و أضف كلمة over قبل إسم الملف، مثل "overleft.png". إستخدم برنامج مثل الفوتوشوب لتعديل الصور. أو بإمكانك الصور الجديد من الملفات المرفقة. بعدها سنقوم بإضافة دالتين إلى الكود MouseOver MousOut كالتالي، حيث سوف تقوم الدالة الأولى بعرض الصورة ذات الوهج بدل الصورة الأصلية و ستقوم الدالة الثانية بإعادة عرض الصورة الأصلية. تحتوي هذه الدوال أيضا على إستثناء حيث إذا كان الid يحمل القيمة "plypus" يتم فحص حالة التشغيل و الإيقاف للعرض للتقديمي، و من ثم يتم إسناد الصورة الصحيحة إلى الزر حسب حالة العرض:

```
function MouseOver(id)
{
    if(id=="plypus" )
    {
        if(play==0)
            document.getElementById(id).src= "stuff/overplay.png";
        else
            document.getElementById(id).src= "stuff/overpause.png";
    }else{
        document.getElementById(id).src= "stuff/over" + id + ".png";
    }
}
function MouseOut(id)
{
    if(id=="plypus" )
    {
        if(play==0)
            document.getElementById(id).src= "stuff/play.png";
        else
            document.getElementById(id).src= "stuff/pause.png";
    }else{
        document.getElementById(id).src= "stuff/" + id + ".png";
    }
}
```

بعد ذلك قم بتعديل وسوم صور أزرار "التالي" و "السابق" و "تشغيل\إيقاف" بحيث تصبح بالشكل التالي:

```

```

```

```

```

```

لاحظ أننا قمنا بإضافة ال id من أجل الوصول إلى الصورة من داخل الدوال MouseOver و MouseOut، كذلك لاحظ أننا قمنا باستدعاء كل من الدالتين في حال حصل حدث مرور الماوس فوق الصورة، وفي حال خروجه من فوق الصورة. أيضا أرسلنا إلى الدالة قيمة نصية تعبر عن ال id الخاص بالصورة.

التحسين الثالث و الأخير بسيط جدا ألا وهو وضع السكربت في ملف منفصل إذا قم بقص كل الكود بين كل من وسمي البداية و النهاية للسكربت <script> (لا تقص وسم البداية و النهاية للسكربت).

قم ب لصق الكود في ملف باسم GallScript.js وضعه في نفس مجلد الذي توجد فيها الصفحة، بعدها قم بتعديل وسم السكربت بالتالي:

```
<script type="text/javascript" src="GallScript.js"/>
```

ضع خلفية رمادية للجدول:

```
<table border=0 align="center" bgColor="#222222">
```

شغل الكود، وأخيرا قد حصلنا على معرض جميل للصور D-: .



أرجو أن أكون قد وفقت في إيصال و لو بعض الأفكار الجديدة و الجميلة إليكم، و الآن أصبح لدينا معرض للصور مفتوح المصدر للإستخدام.

ملاحظة: الكود مجرب ويعمل بشكل جيد على كل من Google Chrome و Mozilla Firefox 3.

تقبلو تحياتي: خلدون خالد، الفريق العربي للبرمجة.