

AGASHE

الطريق لاحتراف تطوير المواقع

ج 5



تأليف

م / محمد يوسف

الفهرس

- 1 - المقدمة 2
- 2 - الفصل الأول / الصيغة العامة 4
- 3 - الفصل الثاني / تحديد العناصر 7
- 4 - الفصل الثالث / الأحداث 11
- 5 - الفصل الرابع / المؤثرات 13
- 6 - الفصل الخامس / دالة الإرجاع و طريقة السلسلة 22
- 7 - الفصل السادس / ربط jQuery مع CSS و التعامل مع خصائص العناصر 24
- 8 - الفصل السابع / التعامل مع العناصر عن طريق jQuery 27
- 9 - الفصل الثامن / دالة noconflict 32
- 10- ملاحظات 34
- 11 - الخاتمة 35

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

السلام عليكم و رحمة الله تعالى و بركاته

نعود مرة أخرى في جزء جديد من سلسلة " الطريق لاحتراف تطوير المواقع " حيث نتعرف معا على تقنية جديدة و رائعة ، لديها الكثير لتقدمه للمبرمج من السرعة و القوة ألا و هي

مكتبة jQuery

هذه المكتبة و التي نقصد بها ملف يوجد بداخله مجموعة كبيرة من الأكواد و المكتوبة بلغة Javascript التي يمكن للمبرمج أن يستخدمها مباشرة بدل أن يضع الكثير من الوقت في كتابتها .

في هذا الجزء سوف نقسم جميع الدوال إلى مجموعة من الفصول في كل منهم نشرح جزء إلى أن تكتمل الصورة في النهاية .

طيب ، كالعادة فأنا لا أجيد المقدمات 😊 لكن نبدأ بالسؤال المعتاد قبل تعلم أي شيء جديد و هو " لماذا نتعلم ما نتعلمه الآن؟؟"

فأجيبك بأن jQuery قدمت حلول مختصرة لكثير من المواضيع مثل : عمل معرض للصور أو عمل تأثير على الهيدر بحيث نجعله يتحرك ، و الكثير من الأمور التي تحتاج فقط منك أن تطلق العنان لإبداعك .

ماذا أحتاج للبدأ؟؟

- أن تكون مجيد لكل من HTML , CSS , Javascript
- محرر أكواد (Atom , Sublime , Vim , Notepad++ الخ)
- متصفح (Fire Fox , Chrome , Safare , Obera الخ)

إذا توافرت لديك كل الشروط السابقة فهيا بنا نبدأ مع jQuery .

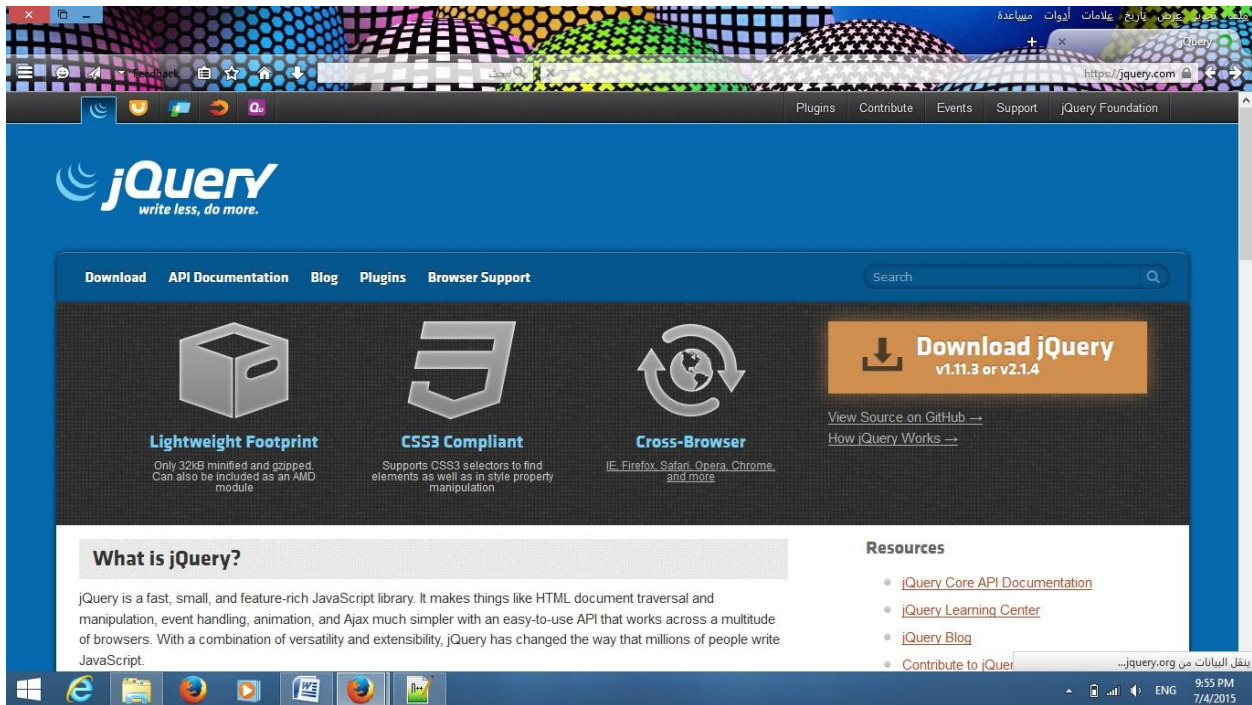
الفصل الأول / الصيغة العامة :

كما درسنا سابقا في **Javascript** أن الأكواد تكتب بين وسمي **<script>** أو أن تكتب في ملف منفصل مع الإشارة إلى مكانه ، بالنسبة لملف **jQuery** يوجد لدينا طريقتين :

```
<script src = "file url" type="text/javascript">
// Code
</script>
```

الأولى و هي بتحميل المكتبة من الرابط التالي : www.jquery.com

بنفس الطريقة سوف نربط مكتبة **jQuery** بالصفحة فقط بوضع إمتداد الملف (الذي حفظت فيه المكتبة) داخل الخاصية **src** و بعدها يمكنك استعمال كافة المميزات و الدوال الموجودة داخلها ، كأنك تماما تتعامل مع ملف منفصل كتبته بنفسك .



الثانية هي باستعمال طريقة **CDN** و هي طريقة مختصرة حيث تكون المكتبة مرفوعة على خادم (server) يمكنك منه استعمال المكتبة مباشرة دون الحاجة لتحميلها .

```
<head>
```

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
```

```
</head>
```

بعد أن انتهينا من موضوع تضمين المكتبة بنجاح ننتقل الآن إلى كيفية كتابة الاكواد .
أي كود **jQuery** يتكون من ثلاث عناصر مرتبين بهذا الشكل :

`$("Selector").Action(Values);`

ما هذا الكلام ؟؟؟؟

أجيبك بكل هدوء و بساطة :

Selector - المحدد --< هو العنصر الذي اخترته في الصفحة : `div , p, h3 ... الخ` و سوف نتعلم المزيد عن تحديد العناصر في الدروس القادمة .

Action - الحدث --< و هو من اسمه الأمر الذي تريد تطبيقه على العنصر الذي اخترته و سوف نتعلم الكثير من هذه الأحداث في الدروس القادمة أيضا .

Values - القيم --< و هي في الغالب الزمن الذي سيقع فيه التغيير .

نظرا لأننا نتعامل مع الكثير من المستخدمين فيجب أن نضع في الإعتبار أن سرعات النت تتفاوت بالتأكد من مستخدم للأخر ، لذا لحماية الموقع من المشكلات و أن نجعل شكل الموقع سخي ، نستعمل كود للتأكد من تمام تحميل الصفحة .

ملحوظة : `$` و `;` و `.` هذه كلها ثوابت يجب أن تضعها في كل سطر .

```
$(document).ready(function( ) {
```

```
//jQuery Codes
```

```
});
```

شرح الكود السابق :

- Document** <-- هو selector و هنا هو كلمة محجوزة تعني تحديد كامل الصفحة .
- Ready** <-- هو action و هنا من الترجمة معناه حدث الاستعداد أو اكتمال التحميل .
- Function** <-- و هنا نعتبرها بمثابة الدالة الرئيسية للأكواد و التي منها يبدأ تنفيذ الكود .

نكتب الآن مثال 1 - 1

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("p").hide(500);
});
</script>
```

اعتقد تقريبا أن كل شيء واضح ، عنصر p اضفنا له حدث الاختفاء hide بعد 1000 مل ثانية ، لا تنسى هنا أن هذا الكود يكتب داخل صفحة **html** يستفضل أن يكون بين وسمي **<head>** في بداية الصفحة و لا تنسى أن تضع عنصر p تكتب بداخله بعض الكلمات لكي يطبق التأثير .

الفصل الثاني / تحديد العناصر :

تعرفنا في الفصل الأول على الصيغة العامة لكتابة أكواد **jQuery** في هذا الفصل نتعرف أكثر على أول جزء ألا وهو المحدد (selector) و في هذا الجزء سوف نستعرض 8 طرق مختلفة لكتابة المحدد .

1 - اختيار كافة العناصر :

```
$(".*).action();
```

في هذه الطريقة نختار كافة العناصر داخل الوسم **<body>** فقط على عكس **document** فهي كانت تشمل جميع مكونات الصفحة .

2 - تحديد العناصر من نوع واحد :

```
$(".element").action();
```

هنا نحدد جميع العناصر (elements) من نفس النوع مثلا : **div , p , a , h4 ,img ... الخ**

3 - تحديد كافة العناصر المتبوعة بتصنيف (class) معين :

```
$(".element.class").action();
```

مثال جميع عناصر **a** المرتبطة بتصنيف **button** .

4 - تحديد العنصر المرتبط برقم تعريف (id) معين :

```
$("#element#id").action();
```

كما نلاحظ فهنا **id** غالبا ما يستخدم مع عنصر واحد و لا يتكرر على عكس **class** فهو يمكن تطبيقه مع كافة العناصر التي ترتبط به .

ملحوظة : يمكنك الاستغناء عن كتابة element قبل id و class لكنها زيادة للتذكير و التوضيح يعني الطريقتين صحيحتين .

5 – تحديد أول أو آخر عنصر من نوع معين :

```
$("#element:first").action();
```

```
$("#element:last").action();
```

هذه الطريقة مفيدة خاصة مع القوائم أو الجداول .

6 – تحديد العناصر عن طريق الاشتراك في خاصية :

```
$("#[property]").action();
```

بمعنى أنك تملك مجموعة من العناصر قد تكون مختلفة النوع ، المهم أنها تشترك في خاصية مثل href , src , type , font: ... الخ ، عندها يمكنك تحديدها و تطبيق التأثير عليها .

7 – اختيار العناصر الزوجية و الفردية :

```
$("#element : even").action();
```

```
$("#element : odd").action();
```

طبعا هذه الطريقة مفيدة جدا خاصة مع الجداول ، في حالة عرض مثلا أسعار بضائع أو نتائج اختبار ما ، فيمكنك استخدامها لتزيين السطور و جعلها ملونة بدل أن تكون جميع السطور بنفس اللون ، و يجب أن ننتبه أن إلى ان عد العناصر يبدأ من الصفر و ليس الواحد !!

8 – الوصول للعناصر داخل عناصر أخرى :

```
$("#element1 element2").action();
```

هذه النقطة سوف تشرح بمزيد من التفصيل في الفصول القادمة لكن حاليا يكفي أن نعرف بأنه لو لدينا مثلا ul و نريد تحديد كافة عناصر li داخلها فيمكننا استخدام كود مشابه :


```
$(“ul li”).action();
```

مثال 5 - 2

```
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("[font]").hide(500);
});
</script>
</head>
<body>
<p font="tahoma">Hello World 1</p>
<a>Link1</a>
<p>Hello World 2</p>
<p font="tahoma">Hello World 3</p>
<a font="tahoma">Link2</a>
<p font="tahoma">Hello World 4</p>
<p>Hello World 5</p>
</body>
</html>
```

الشرح :

- اولا ربطنا المكتبة داخل وسم head .
- دخل الوسم body وضعنا بعض العناصر من النوعين p و a و جعلنا بعضهم يحملون الخاصية font .
- عدنا بعد ذلك للوسم head و كتبنا الأكواد بين وسمي script
- في البداية كود التحقق من تحميل المكتبة
- أخيرا كتبنا كود لتطبيق تأثير الاختفاء و جعلناه يعمل بعد 1000 مل ثانية أي ما يعادل ثانية واحدة ، و كما نلاحظ أننا ربطنا التأثير بجميع العناصر التي تحمل الخاصية font

طبعا باقي طرق التحديد أتركها لك لتجربها و تشاهد النتيجة و كما تعلم هناك الكثير من الأمثلة المرفقة مع الكتاب يمكنك الرجوع لها في حالة واجهتك أي مشاكل .

قد تلاحظ هنا أنني لم اكتب type = "text/javascript" داخل وسم script و السبب في ذلك يرجع إلى أن معظم إن لم يكن جميع المتصفحات الحديثة تعتمد javascript كلغة سكربتية أولى ((باستثناء طبعا internet explorer حتى في أحدث إصداراته ، في الغالب يعتمد على نفسه ☺))

الفصل الثالث / الأحداث :

كما شاهدنا في الأمثلة حتى الآن أن **jQuery** حتى الآن تطبق مباشرة بمجرد تحميل الصفحة ، لكن هذا عمليا غير مفيد لأن المفترض أن الموقع يكون فيه تفاعل مع المستخدم مثل الضغط على زر تمرير مؤشر الفأرة على مساحة ما في الصفحة ، كل هذه الأمور سنتعلم استخدامها بعد الفاصل مباشر..ة .. اقصد الآن 😊

الصيغة العامة :

```
$("#element").event(function(){
//code
});
```

كم كتبنا دالة التأكد من تحميل الصفحة ، هنا أيضا نكتب التأثيرات داخل الدالة كنوع من التحديد ، و كما تلاحظ أن الحدث (event) يكتب بعد النقطة مباشرة كأنه تأثير أنواع الأحداث :

- الضغط - click
- الضغط مرتين - dblclick
- تحريك مؤشر الفأرة داخل حيز العنصر - mouseenter
- تحريك مؤشر الفأرة خارج حيز العنصر - mouseleave

ملحوظة : يرجى كتابة الأحداث كما في الأعلى لأننا كما نعلم JS حساسة لحالة الأحرف ف mouseleave حدث أم MouseLeave فلا معنى لها .

هنا في هذا الكتاب و ربما حتى خارج هذا الكتاب تقريبا لن نتعامل سوى مع click و هذا هو مثال 1 - 3 عليها :

```
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
```

```
$(document).ready(function){  
  $("#a").click(function(){  
    $("#a").hide(500);  
  });  
};  
  
</script>  
</head>  
<body>  
<p id="txt">Hello World </p>  
</body>  
</html>
```

الشرح :

- فالبدائية وضعنا عنصر p و خصصنا له id باسم txt
- انتقلنا لكتابة التأثير و هنا استخدمنا الأحداث
- وضعنا داخل حدث الضغط على العنصر يتم اخفائه خلال نصف ثانية

باقي الأمثلة ستجدها في ملف الأمثلة ، و هي عموما لا تختلف عن هذا .

الفصل الرابع / المؤثرات :

درسنا في الفصول السابقة كيفية تحديد العناصر و تعرفنا كيفية التعامل مع الأحداث في هذا الفصل نتعرف على المؤثرات في **jQuery** ، سوف نقوم بتقسيم المؤثرات إلى فئات :

أ / الإظهار و الإخفاء :

و هنا نحن نستطيع إظهار أو إخفاء أي عنصر داخل الصفحة دون التأثير على باقي العناصر .

- لإخفاء عنصر :

```
$("#element").hide( time , callback );
```

هو المؤثر الذي كنا نعمل عليه منذ البداية .

- لإظهار عنصر :

```
$("#element").show( time , callback );
```

- لتطبيق كلا التأثيرين :

```
$("#element").toggle( time , callback );
```

يعمل هذا التأثير بشكل عكسي ، بمعنى لو كان العنصر مخفي نظره و لو كان ظاهر نخفيه و هكذا .

ملحوظة : قد تلاحظ وجود **callback** هذه هي دالة إرجاع سوف يتم شرحها في الفصول القادمة ، و هي موجودة في جميع التأثيرات .

مثال 1 - 4 :

في هذا المثال سوف نضع شيء أشبه بصندوق تنبيهات يتم إظهاره و إخفائه من قبل المستخدم و سوف نستعمل هنا طريقتين مختلفتين

a - استخدام toggle :

```
<html>
<head>
<style>
#btn{
height:20px;
width:200px;
background-color:red;
text-align:center;
}
#note{
padding-top:20px;
height:80px;
width:200px;
background-color:blue;
text-align:center;
}
</style>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("#btn").click(function(){
$("#note").toggle( 500 );
});
});
```

```

</script>
</head>
<body>
<div id="btn"><p>Click</p></div>
<div id="note"><p>Notafication</p></div>
</body>
</html>

```

الشرح :

- لدينا عنصرين من النوع div أحدهما كزر btn و الآخر كشاشة لإظهار التنبيه note
- أكواد CSS أعتقد أنها واضحة و لا تحتاج لشرح فقط بعض الألوان و الأبعاد
- بالنسبة لكواد jQuery فهو يعتمد على حدث الضغط على btn و هنا يختفي note و العكس طبعاً ، طبيعي جداً أن تستخدم أحداث عناصر للتحكم في عناصر أخرى حتى تعطي للمستخدم المزيد من السيطرة على الموقف... أقصد الموقع 😊

b – استخدام الشرط :

```

var clk = 0;
$(document).ready(function(){
$("#btn").click(function(){
if(clk == 0){
$("#note").hide( 500 );
clk = 1;
}else{
$("#note").show( 500 );

```

```

clk = 0;
}
});
});

```

هنا نحن استخدمنا نفس العناصر في الطريقة السابقة (غالباً سنستخدمهم في جميع الأمثلة القادمة) لذا سنشرح فقط كود **jQuery** .

- كما نعلم مكتبة **jQuery** مبنية بلغة **javascript** لذا نستطيع استخدام جميع مميزات و دوال **javascript** داخل الكود .
 - في البداية عرفنا متغير **clk** يحمل قيمة صفر و في مثالنا تعني أن التنبيه ظاهر
 - قمنا بعمل شرط **if** ، و للتذكير فالشرط إذا تحقق ما بين القوسين () ينفذ الكود داخل القوسين { }
 - في حالة أن المتغير **clk** يساوي 0 عندها فأننا نطبق تأثير الاختفاء على التنبيه و بعدها نحول قيمة **clk** إلى 1 لكي نعرف الكود بأن التنبيه مخفي
 - في حالة أن المستخدم ضغط على الزر و كان المتغير **clk** يساوي 1 حينها سيفهم الكود أن التنبيه مخفي و سينتقل لتنفيذ الأكواد بعد **else** و في حالتنا هذه سوف نظهر التنبيه بالتأثير **show** و نغير قيمة **clk** إلى 0
- و هذه هي فكرة عمل **toggle** فهو يستخدم نفس الفكرة .
- أعتقد أن الموضوع واضح حتى هذا الوقت .

ب / التلاشي (fade) :

نقصد بالتلاشي هو عملية اختفاء لكن في اتجاه معين حيث يعطي تأثير جميل على العناصر و هو شبيه بالاختفاء إلى حد ما .

- عمل تلاشي للداخل (إخفاء) :

```
$("#element").fadeIn( time , callback );
```

- عمل تلاشي للخارج (إظهار) :

```
$("#element").fadeOut( time , callback );
```

- تطبيق كلا التأثيرين معا :

```
$("#element").fadeToggle( time , callback );
```

- التحكم بدرجة الوضوح (الشفافية) :

```
$("#element").fadeTo( time , opacity , callback );
```

هنا نجد معامل جديد و هو الشفافية opacity و هو يأخذ قيم من 0 حتى 1 .

ملحوظة : يمكن في time استخدام كلمات مثل "slow" , "fast" للتحكم بزمن التأثير .

المثال 2 – 4

هنا سنستعمل نفس العناصر في المثال السابق حيث لدينا عنصران div أحدهما btn و الآخر note و سوف نقوم بتطبيق الكود عليهما

a – باستخدام fadeToggle :

```
$(document).ready(function(){
$("#btn").click(function(){
$("#note").fadeToggle( 500 );
});
});
```

b – استعمال fadeTo :

```
$(document).ready(function(){
$("#note").hide();
$("#btn").click(function(){
$("#note").fadeTo( 3000,0.5 );
});
});
```

الشرح :

في البداية نقوم بإخفاء عنصر note (طبعا يمكنك فعل ذلك أيضا باستعمال CSS) ، بعدها

داخل حدث الضغط على الزر btn نقوم بتطبيق تأثير fadeTo و كما تلاحظ فإن درجة الشفافية هنا قلت للنصف ؛ أيضا يمكنك استعمال هذا التأثير لتقليل الشفافية لأي عنصر دون الحاجة لعمل التأثير كله ، كما فعلنا مع تأثير hide فلا نحدد زمن التنفيذ و نكتب هكذا مباشرة :

```
$(document).ready(function(){$("#btn").click(function(){
$("#note").fadeTo( 0 , 0.5 );
}); });
```

ج / الشرائح (slide) :

هو تأثير يعمل مثل ستائر منسدلة لأعلى و لأسفل .

- عمل تحريك للشريحة لأسفل :

```
$("#element").slideDown( time , callback );
```

- عمل تحريك للشريحة للأعلى :

```
$("#element").slideUp( time , callback );
```

- عمل كلا التأثيرين :

```
$("#element").slideToggle( time , callback );
```

المثال 3 - 4

```
$(document).ready(function(){  
  $("#btn").click(function(){  
    $("#note").slideToggle( 1000 );  
  });  
});
```

كما في الأمثلة السابقة تمام لا جديد يذكر .

د / التحريك (animate) :

التحريك هو التأثير الشامل حيث تستطيع هنا أخذ أي عنصر و أن تطبق عليه مجموعة من التغيرات ، معظمها عبارة عن خصائص موجودة في لغة **CSS** مثل `width , height , left`

- لإضافة تحريك :

```
$("#element").animate( { Parameters } , time , callback );
```

نلاحظ هنا وجود معامل جديد و هو الخصائص التي سنعدل عليها ، و أرجوا أن ننتبه للأقواس .

- لإيقاف التحريك قبل الزمن المحدد :

```
$("#element").stop( );
```

يوجد معاملات لهذا التأثير لكنها لا تهتمنا المهم هو إيقاف التحريك .

المثال 4 - 4

نضيف هنا عنصر div آخر نسميه stop و نعدل click ليصبح start .

html :

```
<div id="btn-start"><p>Start</p></div>
<div id="btn-stop"><p>Stop</p></div>
<div id="note"><p>Notafication</p></div>
```

يمكن طبعا التعديل على **CSS** بالطريقة التي تناسبك لكن المهم الآن التركيز على القادم .

في مثالنا سوف نعطي مثال نغير فيه كل من width و height و يمكن طبعا للمستخدم أن يتوقف متى أراد .

jQuery :

```
$(document).ready(function(){
$("#btn-start").click(function(){
$("#note").animate( {width:'+=250px' , height:'+=250'} , 3000 );
});
$("#btn-stop").click(function(){
$("#note").stop();
});
});
```

الشرح :

- داخل حدث الضغط على العنصر btn-start و ضعنا تأثير التحريك ، طبعا استعملنا عدد من خصائص **CSS**
- في هذه الطريقة نحن نقوم بزيادة العرض و الارتفاع في كل مرة يضغط فيها المستخدم على الزر
- نكتب اسم الخاصية بدون أي إضافات فقط بين قوسين { } و بالنسبة للقيم فهي توضع داخل علامتي تنصيص منفرد (') ، لا تنسى أيضا الفصل بين الخصائص المختلفة بعلامة (,) .
- في القيم يمكن كتابة القيمة مباشرة 250 و 1000 أو أن تستعمل تعابير رياضية مثل +=250 , -=250 و هي تعني الزيادة و النقصان ، و لا تنسى رجاء وحدة القياس px

ملحوظة : تأثير التحريك (animate) لا يتعامل مع الألوان هو فقط يؤثر على الأبعاد

الفصل الخامس / دالة الإرجاع و قاعدة السلسلة :

1 – دالة الإرجاع (callback) :

تستخدم دالة الإرجاع كمنبه للمستخدم عادة بعد إتمام الإجراء ، حيث نظهر له رسالة تعلمه بتمام التحميل ، أيضا يمكن إستخدامها لتخزين مثلا بيانات أو جلب بيانات مثلا لو فرضنا أن المستخدم قام بغلاق نافذة تحيث بيانات العضوية عندها نقوم باتعمال أكواد لتخزين بياناته في قاعدة البيانات و الكثير من الأفكار الرائعة التي من الممكن أن نستعمل فيها دالة الإرجاع .

مثال 1 – 5

a – نريد عند إخفاء صندوق التنبيهات أن نرسل للمستخدم برسالة مفادها أنه تم إخفاء صندوق التنبيهات .

```
$(document).ready(function(){
    $("#btn").click(function(){
        $("#note").hide( 500 , alert("your notf. box is hidden !") );
    });
});
```

كما تلاحظ سوف تظهر الرسالة بمجرد الضغط على click و بالنسبة للدالة alert فهي دالة إظهار الرسائل أعتقد أنها مرت عليك .

b – نفس المثال السابق لكن هنا سوف نستعمل مفهوم الدوال .

```
function msg(){
    alert("your notf. box is hidden !");
}
$(document).ready(function(){
    $("#btn").click(function(){
```

```
$("#note").hide( 500 , msg( ) );
});
});
```

كل ما فعلناه هنا هو أننا قمنا بعمل دالة msg تظهر نفس الرسالة ، كلني أريد أن أريك أن **jQuery** هي في النهاية أكواد javascript عادية تماما .

2 – طريقة السلسلة (chain rule) :

تمكننا هذه الطريقة من وضع أكثر من تأثير على نفس العنصر ، مما يتيح أكبر قدر ممكن من التأثيرات على العنصر .

```
$("#element").action1( ).action2( ).action3( ) ..... actionM( );
```

المثال 2 – 5

```
$(document).ready(function(){
$("#btn").click(function(){
$("#note").hide( 500 ).slideDown("fast");
});
});
```

في هذا المثال طبقنا تأثيرين الأول و هو إخفاء العنصر note ثم أظهارناه مرة أخرى باستعمال تأثير slideDown .

الفصل السادس / ربط jQuery مع CSS و التعامل مع خصائص العناصر:

1 - التعامل مع خصائص CSS :

تمكننا jQuery من التعامل مع خصائص CSS بمنتهى البساطة عن طريق استعمال الآتي

```
$("#element").css("property1","value1");
```

أو في حالة لو أردنا وضع أكثر من خاصية يمكن استعمال الصيغة التالية :

```
$("#element").css({"property1" : "value1" , "property2" : "value2" , ..... });
```

هكذا أصبح بإمكاننا التحكم بأي خاصية دون الحاجة لاستعمال ملف CSS خارجي حتى .

المثال 1 - 6

```
$(document).ready(function(){
$("#btn").click(function(){
$("#note").css({"background-color" : "yellow" , "color" : "red"});
});
});
```

قمنا بتعديل خصائص كل من لون الخلفية و لون الخط لعنصر div و الذي نسميه في جميع أمثالتنا note و هذا طبعا عند وقوع حدث الضغط على btn و هو أيضا من النوع div أرجوا أن لا نكون نسينا .

2 - التعامل مع خصائص العناصر :

في هذا الموضوع سوف نتحدث عن أحدث التأثيرات المفيدة ، وهو تأثير التحكم بقيمة عنصر أثناء عمل الصفحة على سبيل المثال : name , value , id يعني أمور قد تخص المبرمج أحيانا و قد لا يراها المستخدم .

الصيغة العامة :

```
$("#element").attr("property1","value1");
```

أو مجموعة خصائص :

```
$("#element").attr({"property1":"value1","property2":"value2", .....});
```

قد تصبح الفكرة واضحة بالنسبة لنا ننتقل لهذا المثال الجميل :

المثال 2 – 6

```
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("#btn").click(function(){
$("#a").attr("href","www.abc.com");
alert($("#a").attr("href"));
}});
</script>
</head>
<body>
<button id="btn">Click</button>
<a href=""> WebSite Link </a>
</body>
</html>
```

الشرح :

- في هذا المثال سنجد لدينا اخيرا تجديد 😊 فلقد وضعنا عنصرين الاول و هو زر button وضعنا له id يساوي btn ، أما الآخر فهو عبارة عن رابط a لم نضع أي قيم في الخاصية href .
- قمنا بعد ذلك بعمل شيء لطيف و هو وضع حدث الضغط على الزر كالعادة لا جديد
- داخل حدث الضغط يوجد أمران الأول و هو التحكم بالخاصية href و جعلها تساوي الرابط الذي فرضناه في مثالنا www.abc.com
- الأمر الثاني و هو الجديد كتابة attr("href") ، هذه الطريقة دون وضع قيمة تعني طلب القيمة الحالية فهنا نحن أرسلنا رسالة للمستخدم نوضح فيها الرابط الذي سينقل له ، بنفس الطريقة يمكنك استعمالها مع تأثير CSS لتحصل مثلا على لون أو تعرف ارتفاع أو عرض عنصر .

الفصل السابع / التعامل مع العناصر عن طريق jQuery:

في هذا الفصل نتعلم سويا كيفية التعامل مع العناصر التي بداخل الصفحة عن طريق أكواد jQuery فننتعرف على كيفية إضافة و حذف العناصر .

- إضافة عنصر في النهاية :

```
$("#element").append("new element");
```

- إضافة عنصر في البداية :

```
$("#element").prepend("new element");
```

- إضافة عنصر بعد آخر :

```
$("#element").after("new element");
```

- إضافة عنصر قبل آخر :

```
$("#element").before("new element");
```

كل ما تفعله هو تحديد العنصر الذي سنضيف عنده أو داخله و بعدها نعين نوع العنصر الجديد الذي سنضيفه .

ننظر معا للمثال التالي :

المثال 1 – 7

Html :

```
<button id="btn">Add element</button>
<ul>
<li>new element in list</li>
</ul>
```

jQuery :

```
$(document).ready(function(){
$("#btn").click(function(){
$("#ul").append("<li>new element in list</li>");
});
});
```

الشرح :

- وضعنا في البداية زر btn و وضعنا قائمة ul تحتوي على عنصر واحد li
- في الكود حددنا حدث الضغط على الزر طبعا بعد اكتمال تحميل الصفحة
- عند وقوع حدث الضغط حددنا القائمة ul يمكن أن تكون أي عنصر آخر a , div , p أو حتى بالنسبة لجميع العناصر التي تشترك في تصنيف معين (class) أو أن تكتب " * " و كما قلنا فهي تعني كامل الصفحة بالتالي سوف تكون الإضافة للصفحة كلها و ليس بالنسبة لعنصر بعينه
- أخيرا استعملنا تأثير append لنضيف عنصر li جديد في نهاية القائمة و لا تنسى كتابته بين علامتي التنصيص " "

المثال 2 – 7

نأخذ مثال آخر على الإضافة هذه المرة سوف تكون باستعمال before .

Html :

```
<button id="btn">Add text</button>
<p class="txt">some text</p>
<a class="txt" href="#">some text</a>
<h3 class="txt">some text</h3>
```

CSS:

```
.txt{
color:red;
}
```

jQuery :

```
$(document).ready(function(){
var element = "<p>new p element</p>";
$("#btn").click(function(){
$("#.txt").before( element );
});
});
```

الشرح :

- في هذا المثال لدينا صفحة تحتوي على الزر المعتاد btn عنصر p و آخر a و ثالث h3
- العناصر الثلاث تشترك في تصنيف واحد (class) بإسم txt و هو مجرد تعيين للون الخط الأحمر
- بالنسبة للكود فما هو element؟؟؟ في هذا المثال أردت أن أريك عزيزي القارئ طريقة أخرى لإضافة عناصر و هي عن طريق متغير فبدل أن تكتب كل شيء بشكل أوامر يمكنك عمل متغير و فقط تمريره للدالة و سوف يتم إضافته
- لاحظ أيضا أننا استعملنا txt. و بالتالي أي عنصر مهما كان نوعه سوف يتم إضافة عنصر p جديد قبله

ننتقل للحذف ...

● حذف عنصر :

```
$("#element").remove(" class " );
```

```
$("#element").empty( );
```

كلا الطريقتين صحيحة لكن empty تحذف جميع العناصر أيضا التابعة لذلك العنصر كما في حالات القوائم ul و الحاويات div .

أيضا هذه الدوال تعمل دون معاملات فقط حدد العنصر ثم قم بحذفه و لكن remove ربما تسمح بتمرير اسم التصنيف الذي تريده لكي تحدد العنصر بدقة .

المثال 3- 7

Html :

```
<ul>
<li>some text</li>
<li>some text</li>
<li>some text</li>
<li>some text</li>
</ul>
<p id="txt">some text </p>
<p>another text </p>
```

jQuery :

```
$(document).ready(function(){
```

```
$("#btn1").click(function(){  
$("#p").remove("#txt");  
});  
$("#btn2").click(function(){  
$("#ul").empty();  
});  
});
```

الشرح :

- لدينا زران btn1 , btn2
- أيضا أضفنا قائمة ul تحتوي على أربع عناصر li
- عنصرين p أحدهما يحمل id يساوي txt
- في الكود وضعنا حدث الضغط لكلا الزرين أحدهما لحذف القائمة و الآخر لحذف النص المحدد بال id
- في النهاية وضعنا الدوال في حالة remove مررنا txt كعامل لتحديد النص الذي نريد حذفه

الفصل الثامن / دالة `noConflict`:

عند العمل على مشروع كبير به عدد كثير من اللغات و أطر العمل ، ستجد أن علامة `$` ربما تؤدي إلى بعض الارتباك داخل الكود فربما هناك إطار عمل آخر يستعملها ، فعند تجربة الكود لا يعمل بسبب التداخل بين عناصر أخرى داخل الكود ، من هنا وجدت هذه الدالة التي تسمح لك باستبدال `$` بأي كلمة أخرى و ليس رمز فلا يمكن تسمية متغير برمز `😊` .

```
var newSym = $.noConflict( );
```

هذه هي الدالة كل ما عليك هو استبدال `newSym` بما يناسبك ، و كل ما تفعله المكتبة هنا هو استبدال اسم المتغير الذي وضعته بعلامة `$` أثناء التنفيذ .

المثال 1 - 8

Html :

```
<button id="btn">hide text</button>
<p>some text</p>
```

jQuery :

```
var jq = $.noConflict();
jq(document).ready(function(){
jq("#btn").click(function(){
jq("p").hide( 1000 );
});
});
```


الشرح :

- في هذا المثال أضفنا للصفحة عنصري `button` , `p` و وضعنا `id` يساوي `btn` لعنصر `button`
- في الكود بدأنا باستعمال دالة `noconflict` و فرضنا حرفي `jq` كبديل أو اختصار لعلامة `$`
- كتبنا داخل حدث الضغط على الزر تأثير الاختفاء على النص

أعتقد أنه أول مثال شرحناه في الكتاب لكن الفرق فقط في دالة `noconflict` .

ملاحظات في غاية الخطورة

1 – قد تجد في هذا الكتاب الكثير من اللغة العربية مثل : تأثير و تلاشي الخ ، هذه كلها اجتهادات شخصية أو معلومات سمعتها من مبرمجين آخرين ربما تكون صحيحة أو خاطئة عموما كي تريح أعصابك من مشكلة المصطلحات العربية استعمل دائما المصطلحات الانجليزية إلى أن يظهر مرجع أو أي شكل من أشكال الاتفاق حول موضوع المصطلحات و الاصطلاح و اعتقد أنني تقريبا وضعت أغلب المصطلحات الانجليزية التي قد تمر عليك .

2 - في هذا الكتاب اعتمدت على طريقة CDN في تضمين مكتبة **jQuery** داخل الصفحات إذا كنت تواجه مشاكل مع الاتصال بالانترنت فلا تقلق لأن المكتبة مرفقة مع الكتاب فيمكنك إضافتها بكل سهولة من مجلد "jq lib" .

3 – في الأمثلة جميعها التركيز على أكواد **jQuery** فبالنظرة لو لم تفهم أكواد **html** أو **css** فانصحك بمراجعة ما لم تفهمه من الأجزاء السابقة من السلسلة أو من أي مصدر آخر إذا كان متوافر لديك .

4 - **jQuery** ليست لغة برمجة بحد ذاتها ، هي مجرد مكتبة توفر علينا الكثير من العمل لكن هذا لا يعني أنك تستطيع عمل مكتبة خاصة بك .

5 – الرموز في هذه اللغة كثيرة لذا أرجوا أن تكون حذر في كتابتها .

6 – انتبه جيدا لحالة الأحرف و عموما **javascript** تستعمل غالبا بطريقة **camel** و هي طريقة كتابة بحيث أن الكلمات متلاصقة و أول حرف من الكلمة الثانية يكون كبير : **slideToggle** , **fadeTo** الخ .

7 – أخيرا هذا الكتاب قد لا يكون المرجع الأكثر شمولية و إبداعا و..... الخ ، لكن يمكنك اعتباره كمقدمة إذا أردت لما بعد ذلك من مستويات ، فكل يوم يظهر شيء جديد و بالتأكيد **jQuery** سوف تطور لذا كن دائما مواكب لكل ما هو جديد .

إلى هنا ينتهي هذا الجزء من سلسلة الطريق ، ممكن نتوقف في القرية التالية ☺ إلى أن نكمل رحلتنا في هذا الطريق الطويل إن شاء الله .

و كالعادة في حالة وجود أي استفسارات أي أمور أو نقاط غير واضحة ، أو ملاحظات فأنا موجود في الخدمة يمكنك مراسلتي على :

Modi401@hotmail.com

أو

<https://www.facebook.com/mohamed.yossef.583>

AGASHE

2015 – 2010