

( )

[الأساسيات] [الألوان] [الخطوط] [الفقرات والقوائم] [الصور والرسومات] [الوصلات التشعبية]  
[الجدول ١] [الجدول ٢] [الإطارات ١] [الإطارات ٢] [الإطارات ٣] [وسوم من هنا وهناك]  
[الخرائط الصورية] [النماذج ١] [النماذج ٢] [الوسوم الخاصة] [الويب واللغة العربية] [وسوم Meta]

أهلاً وسهلاً بك إلى الدرس الرابع عشر من دروس HTML والذي سيكون الأول من درسين سنتحدث فيهما عن النماذج وكيفية تضمينها في صفحات الويب.

مع أن النماذج تعتبر من المواضيع المتقدمة (وغير السهلة) نوعاً ما في لغة HTML إلا أن معظم مواقع الويب تكاد لا تخلو من وجودها، وذلك لعدة أسباب لعل من أهمها إيجاد إمكانية للتفاعل بين الموقع وصاحبه من جهة والزوار من جهة أخرى... أحياناً قد تحتاج كمصمم لموقع ويب أن تعرف آراء زوار موقعك في مسائل معينة وقد تكتفي برسائل البريد الإلكتروني التي يرسلوها لك، لكن عندما تريد معرفة أشياء محددة بذاتها فإن النماذج هي الخيار الأفضل لك. بالإضافة إلى إمكانية تنظيم البيانات المدخلة من خلالها وسهولة وسرعة استخدامها من قبل زوار الموقع. ومن أبرز الأمثلة على النماذج في مواقع الويب هي دفاتر الزوار وصفحات البحث عن الكلمات أو العبارات داخل المواقع.

لا تكمن صعوبة التعامل مع النماذج في كونها معقدة بحد ذاتها، كلا... فهي إحدى العناصر التي تدعمها لغة HTML وهي مجرد وسوم عادية مثلها مثل الوسوم التي تعاملنا معها في جميع الدروس السابقة. وبإمكانك إنشاء النماذج في موقعك بنفس السهولة التي تدرج فيها جدولاً أو إطاراً (هذا بالطبع إذا كنت تعتقد أن الجداول والإطارات سهلة) لكن التداخل بينها (وأعني النماذج) وبين لغات البرمجة المتقدمة في الويب مثل JavaScript, CGI هي ما يجعلها تختلف عن سابقتها من الوسوم أو العناصر الأخرى. خاصة إذا احتجت إلى بعض المقاطع البرمجية من هذه اللغات ضمن نماذجك. أما إذا اكتفيت بالإمكانات المتواضعة التي توفرها HTML بالنسبة للنماذج. فما من مشكلة... لأنه سيكون بإمكانك التعامل معها بكل بساطة. وفي هذا الدرس لن نتطرق بالطبع إلى أي من اللغات سوى HTML.

قبل أن نبدأ، ما رأيك بزيارة صفحة [\(دفتر الزوار\)](#) في هذا الموقع للإطلاع على مثال للنماذج، (وربما تود أيضاً التوقيع فيه).  
كم شكلاً من أشكال إدخال البيانات يوجد في هذا الدفتر؟ الحقيقة أنه يوجد ستة أشكال هي كالتالي: أعلى النموذج

Text

Option 1

3 2 1

رسل

وهي الأشكال الموجودة في الدفتر فقط. وأود أن ألفت نظرك إلى وجود أشكال أخرى سوف يتم التعامل معها من خلال هذا الدرس.

## والآن إلى العمل

مع أن الأشكال السابقة تختلف عن بعضها البعض من حيث المبدأ والمظهر (وطريقة التعريف أيضاً) إلا أنها يجب أن تدرج جميعاً ضمن وسمين أساسيين للنماذج هما:

</FORM> ... <FORM>

وكما جرت العادة نحتاج لتحديد بعض الخصائص التي تتعلق بطبيعة هذا النموذج. ولدينا هنا ثلاث خصائص:

### ACTION

تحدد العنوان الذي سيتم إرسال بيانات النموذج إليه لتتم معالجتها بالصورة المطلوبة. وعادة يكون هذا عنواناً لبريد إلكتروني Email سوف يتم إرسال بيانات النموذج إليه. أو قد يكون عنواناً لبرنامج CGI موجود على الكمبيوتر الخادم Server الذي تتواجد عليه صفحة الويب، حيث يستقبل هذه البيانات ويعالجها حسب التعليمات الموجودة فيه كأن يضيفها مثلاً إلى إحدى الصفحات (كما يحدث عادة في دفاتر الزوار) أو يتحقق من صحة بعض الحقول المدخلة ومطابقتها لمعايير معينة، أو أن يقوم بالبحث عن كلمة أو عبارة ضمن صفحات الموقع كما في نماذج البحث الموجودة في مواقع الويب.

<FORM ACTION="mailto:someone@domain.com"> ... </FORM>  
<FORM ACTION="name\_and\_address\_of\_CGI\_script"> ... </FORM>

### METHOD

تحدد الطريقة التي سيتم بها التعامل مع العنوان المحدد في الخاصية السابقة ACTION. وهناك قيمتين لهذه الخاصية هما: GET التي تستخدم في حالة كون عملية المعالجة داخلية أي تتم داخل الخادم Server نفسه. ففي مثالنا السابق عندما نستخدم نموذج البحث عن كلمة في الموقع، فإن عملية المعالجة (أي البحث) تجري مباشرة في الموقع. والقيمة الثانية هي Post وتستخدم عندما تكون عملية المعالجة خارجية كأن يتم إرسال البيانات إلى عنوان بريد إلكتروني.

<FORM ACTION="mailto:someone@domain.com" METHOD="post"> ...  
</FORM>  
<FORM ACTION="name\_and\_address\_of\_CGI\_script" METHOD="get"> ...  
</FORM>

### ENCTYPE

هذه الخاصية تحدد طريقة الترميز التي سيتم إرسال البيانات وفقاً لها. وهي تأخذ القيمتين التاليتين: (يجب أن تكتب هذه القيم كما هي نصاً وحرافاً)

- application/x-www-form-urlencoded
- text/plain

وبدون الخوض في الأسباب التقنية التي أدت إلى إيجاد هذين النوعين من طرق الترميز أو في أمور برمجية بعيدة عن موضوعنا، فإن الدافع لإستخدام أي من القيمتين هو طبيعة عملية المعالجة التي ستجرى على البيانات أو طبيعة برنامج البريد الإلكتروني الذي ستستقبل هذه البيانات من خلاله (إذا كان يدعم MIME أم لا، وهي إختصار للعبارة Extentions Mail Internet Multi-purpose وهي من المعايير السائدة في الإنترنت والتي تتعلق بنقل جميع أنواع البيانات من صوت وصورة وليس فقط النصوص من خلال البريد الإلكتروني). وما يعيننا هنا هو الفرق بين الطريقتين من حيث طريقة إرسال واستقبال البيانات. فعند استخدام text/plain ستصل البيانات بالشكل التالي:

NAME=Yahya Al-Sharif  
Address=Nablus , Palestine  
Email=yahya@palnet.com

(الكلمات, Address, Email Name هي أسماء الحقول في النموذج ونقوم نحن بتعريفها أثناء عملية تصميم النموذج أما النصوص الظاهرة بعد إشارة المساواة فهي البيانات المدخلة، وسوف نتحدث عن تعريف أسماء الحقول بعد قليل)

أما عند استخدام application/x-www-form-urlencoded فستصل البيانات بالشكل:

NAME=Yahya+Al-Sharif&Address=Nablus+,+Palestine&Email=yahya@palnet.com

ولك أن تخيل مبلغ الصعوبة في تحليلها إذا احتوت على عشرات الحقول. لذلك تتوفر برامج خاصة تعرف بـ Formaters تقوم بإعادة ترتيب البيانات المرسله من خلال النماذج بشكل مفهوم بحيث تصبح كما لو كانت مرسله بترميز text/plain وإليك أحدها وهو برنامج مجاني يدعى [UrlCook](#). لكن لا تعتقد أن الطريقة الأولى هي الأفضل دائماً فذلك يعتمد كما قلنا على طريقة المعالجة والنقل بالبريد. لذلك لا ضير من أن تجرب الطريقتين لتعرف أيهما أنسب لك.

إذن خلاصة القول: قد تكون أفضل صيغة لتعريف النموذج في حالة أردت استقبال البيانات من موقعك إلى عنوان بريدك الإلكتروني هي:

```
<FORM ACTION="mailto:email@domain.com" METHOD="post"  
ENCTYPE="text/plain">  
...  
</FORM>
```

وبهذا نكون قد إنتهينا من عملية تعريف النموذج وخصائصه، لكن انتظر فما زلنا في بداية الطريق.

نبدأ الآن في عملية تعريف أشكال البيانات في النموذج. ونستخدم الوسم `<INPUT>` لتعريفها والحقيقة أن هذه الأشكال هي مجرد خصائص أو بالأحرى قيم لخصائص تابعة لهذا الوسم. كيف؟ ... لنأخذ مثلاً على ذلك لأوضح لك هذا المفهوم

ملاحظة: إذا كنت تستخدم `Sindbad 3.x` فسيبدو الحقل والنص المجاور له بصورة معكوسة، وهي مشكلة ناتجة عن برنامج `Netscape` الذي يعمل من خلاله

Please enter your address:

< لتعريف هذا الشكل (هذه إتفقنا عليها `INPUT` حسناً، لقد استخدمت الوسم `>` مسبقاً) ومن ثم قمت بإضافة الخاصية `TYPE` لهذا الوسم لتحديد نوع الشكل الذي أريده وأعطيتها القيمة `TEXT` أي

```
<FORM ...>
<INPUT TYPE="text">
</FORM>
```

لينتج لدينا هذا الشكل:

فقرة معترضة:	
إليك جميع الأشكال (القيم) المستخدمة مع الخاصية <code>TYPE</code> وسوف أتركها الآن بدون تعليق لحين مناقشتها لاحقاً بشكل مفصل. مع ملاحظة أن هناك شكلين آخرين ندرجهما بالوسوم <code>&lt;TEXTAREA&gt;</code> ، <code>&lt;SELECT&gt;</code>	
<input type="text"/>	<code>&lt;INPUT TYPE="text"&gt;</code>
<input type="password"/>	<code>&lt;INPUT TYPE="password"&gt;</code>
<input type="hidden"/>	<code>&lt;INPUT TYPE="hidden"&gt;</code>
<input type="radio"/>	<code>&lt;INPUT TYPE="radio"&gt;</code>
<input checked="" type="checkbox"/>	<code>&lt;INPUT TYPE="checkbox"&gt;</code>
<input type="submit" value="إرسال استعلام"/>	<code>&lt;INPUT TYPE="submit"&gt;</code>
<input type="reset" value="إعادة تعيين"/>	<code>&lt;INPUT TYPE="reset"&gt;</code>
<input type="button"/>	<code>&lt;INPUT TYPE="button"&gt;</code>

أرجو أن أكون قد وضحت لك الآن وظيفة الخاصية TYPE وجميع القيم المستخدمة معها

ونعود الآن إلى مثالنا.. الخاصية الثانية المستخدمة مع <INPUT> هي NAME وتستخدم لتسمية حقل البيانات حيث قمت بإعطاء الإسم address لهذا الحقل في المثال. (لك كل الحرية في إعطاء الإسم الذي تريده للحقل). والحقيقة أن هذا الإسم يعرف الحقل في داخل النموذج نفسه، بحيث يمكن استخدامه فيما بعد للحاجات البرمجية وضرورات المعالجة إن وجدت من قبل البرامج التي قد تضيفها كمصمم للموقع. وحتى عندما تريد أن يرسل النموذج إليك بالبريد فإن حقله تعرف بالاسم الذي أدرجته لها من خلال هذه الخاصية. (لاحظ ما قلته سابقاً عن تعريف أسماء الحقول عندما تحدثنا عن الترميز والطرق التي تصل بها محتويات النموذج). وكما ترى لا يوجد (حتى الآن) ما يدل على أن هذا الحقل يختص بإدخال العنوان.

```
<FORM ...>
<INPUT TYPE="text" NAME="address">
</FORM>
```

فهي مجرد عبارة توضيحية أضفتها : Please enter your address أما العبارة ليعرف الزائر ما الذي يجب عليه كتابته وتستطيع صياغة هذه العبارة كما تريد. ففي كل الأحوال ليس لها علاقة بجوهر النموذج نفسه بعكس الخاصية NAME.

```
<FORM ...>
Please enter your address : <INPUT TYPE="text" NAME="address">
</FORM>
```

Please enter your address :

(أو أي عبارة Nablus, Palestine أما بالنسبة للعبارة الظاهرة داخل الحقل أخرى تريدها) وهي بمثابة القيمة الافتراضية التي تريدها للحقل، فبالإمكان إظهارها من خلال الخاصية VALUE. وهذه الخاصية تستخدم في الحالات التي نتوقع فيها كتابة قيمة دارجة أو منكررة من قبل معظم الزوار وللتسهيل عليهم يتم تعيينها كقيمة افتراضية وبالطبع مع توفر إمكانية حذفها وكتابة ما يريدونه بدلاً منها.

```
<FORM ...>
Please enter your address : <INPUT TYPE="text" NAME="address"
VALUE="Nablus, Palestine">
</FORM>
```

Please enter your address :

مع الرقم SIZE قد نحتاج أحياناً إلى تحديد حجم الحقل ولذلك نستخدم الخاصية الذي نريده كحجم للحقل، لنجرب الرقم ٤٠

```
<FORM ...>
Please enter your address : <INPUT TYPE="text" NAME="address"
VALUE="Nablus, Palestine" SIZE="40">
</FORM>
```

Please enter your address :

أو لنجرب الرقم ١٠ أيضاً

Please enter your address :

أي صفة تحكمية بالنسبة لحجم المدخلات التي يمكن SIZE لا يوجد للخاصية للزائر أن يكتبها داخل الحقل. وبعبارة أخرى: صحيح أننا حددنا حجم الحقل لكن ذلك يسري فقط على مظهره على الشاشة. ولا يوجد ما يمنع الزائر من الكتابة بحيث يتجاوز النص حجم الحقل المحدد. وهنا يأتي دور الخاصية MAXLENGTH لتتحكم بالحد الأقصى للنص المدخل.

<FORM ...>

Please enter your address : <INPUT TYPE="text" NAME="address" VALUE="Nablu, Palestine" SIZE="40" MAXLENGTH="30">  
</FORM>

Please enter your address :

حاول الكتابة في هذا الحقل لأكثر من ٣٠ حرفاً وأنظر ماذا سيحدث؟

إنتهينا الآن من خصائص الوسم INPUT فما رأيك بإجمالها مرة أخرى؟ حسناً، هذه هي:

- TYPE: لتحديد نوع (شكل) حقل البيانات.
- NAME: لتعيين اسم لحقل البيانات.
- VALUE: لتعيين قيمة افتراضية (مبدئية) لحقل البيانات.
- SIZE: لتحديد حجم حقل البيانات.
- MAXLENGTH: لتعيين الحد الأقصى لعدد الحروف المدخلة في الحقل.

النوع الثاني من الحقول المستخدمة في النماذج هو حقل password وهو يشبه الحقل text من حيث الخصائص تماماً غير أن مدخلاته تظهر على شكل \*\*\*\*\* مهما كانت، وهو الفرق الوحيد بينهما. وربما تكون قد استنتجت الآن أن هذا النوع من الحقول يستخدم عندما يوجد حاجة لإدخال كلمة سر من قبل الزائر في النموذج

<FORM ...>

Please enter your name :  
<INPUT TYPE="text" NAME="the name" VALUE="" SIZE="40" MAXLENGTH="30">

Please enter your passwod :  
<INPUT TYPE="password" NAME="the password" VALUE="" SIZE="40" MAXLENGTH="30">  
</FORM>

Please enter your name :

Please enter your password :

للحقول، ولذلك تركتها VALUES لاحظ أنني لم أرغب في كتابة قيم افتراضية فارغة وأستطيع أيضاً أن أُلغيتها نهائياً من الشيفرة. وأنا في هذا المثال أردت أن أوضح لك عدم أهمية كتابة قيمة افتراضية للحقول في بعض الحالات.

نأتي الآن إلى النوع الثالث من أنواع الحقول وهو hidden أي الحقل المخفي. وكما نستنتج من اسمه فهو لن يظهر ضمن النموذج. وهذا مثال:

```
<FORM ...>
Please enter your name :
<INPUT TYPE="text" NAME="the name" VALUE="" SIZE="40"
MAXLENGTH="30">

<INPUT TYPE="hidden" NAME="my forms" VALUE="form1">

Please enter your passwod :
<INPUT TYPE="password" NAME="the password" VALUE="" SIZE="40"
MAXLENGTH="30">
</FORM>
```

Please enter your name :

Please enter your passwod :

لاحظ هنا أن وجود هذا الحقل مثل عدمه بالنسبة لمظهر النموذج، وأن الزائر لن يتعامل معه بل وربما لن يعرف أن هناك حقلاً مخفياً. والسؤال هنا: ما الفائدة من وجود شيء مخفي لا إمكانية لاستخدامه؟ ولكي أجيب على هذا السؤال دعني أطرح لك مثلاً أو حالة قد تواجهك كمصمم صفحات ويب... لنفرض أن لديك ثلاث صفحات تتضمن كل منها نموذجاً ما وأن هذه النماذج متشابهة. وتحتوي على نفس الحقول. وعندما تصلك البيانات كيف ستستطيع تمييز أي من هذه النماذج استخدم لإرسال البيانات؟ بإمكانك إضافة هذا الحقل (الوهمي) وإسناد أي اسم وأي قيمة له في كل نموذج.

في النموذج الأول ...

```
<INPUT TYPE="hidden" NAME="my forms" VALUE="form1">
```

في النموذج الثاني ...

```
<INPUT TYPE="hidden" NAME="my forms" VALUE="form2">
```

في النموذج الثالث ...

```
<INPUT TYPE="hidden" NAME="my forms" VALUE="form3">
```

وبذلك عندما تصلك البيانات المرسله من قبل أي زائر استخدم أي من النماذج الثلاثة سيصلك أيضاً حقل إضافي قمت أنت نفسك بتعبئته سلفاً عندما صممت النموذج وذلك بأحد الأشكال التالية:

أو my forms=form1  
أو my forms=form2  
my forms=form3

إن نستطيع القول أن الحقل المخفي هو لاستخدام المصمم وليس الزائر، وأن قيمته تدخل مباشرة عند التصميم. ويستخدم بهدف تعريف قيم ما سيتم إرسالها جنباً إلى جنب ضمن بيانات النموذج التي قام الزائر بتعبئتها

**ملاحظة مهمة** بالنسبة للنماذج بشكل عام. من أجل إظهار النموذج بصورة مرتبة ومنسقة والتحكم بموقع الحقول فيه فمن الأفضل دائماً وضعه داخل جدول مع جعل الجدول بلا حدود.

```
<FORM ...>
<TABLE BORDER="0">
<TR>
<TD>Please enter your name : </TD>
<TD>
<INPUT TYPE="text" NAME="the name" VALUE="" SIZE="40"
MAXLENGTH="30">
</TD>
</TR>

<TR>
<TD>Please enter your password :</TD>
<TD>
<INPUT TYPE="password" NAME="the password" VALUE="" SIZE="40"
MAXLENGTH="30">
</TD>
</TR>
</TABLE>
</FORM>
```

وكما ترى تحتاج إلى القليل من العمل الإضافي لكنك بالمقابل ستحصل على النتيجة التالية

Please enter your name :

Please enter your password :

هكذا أفضل... أليس كذلك؟