

الفصل السادس : التعامل مع النماذج Dealing with HTML forms

عند التعامل مع البرامج المعقدة، تكون الحاجة ملحة للتفاعل مع البيانات المدخلة عن طريق المستخدم. و أكثر الطرق شيوعا عند التعامل مع تلك البيانات هي النماذج HTML forms. بداية سنقوم بإنشاء نموذج HTML كالتالي:

```
<form name="contact " method="post" action="contact.php">
  <p>Name:
    <input type="text" name="name" />
  </p>
  <p>Email:
    <input type="text" name="email" />
  </p>
  <p>Message:
    <textarea name="message" cols="45" rows="5"></textarea>
  </p>
  <p>
    <input type="submit" name="send" value="Send" />
    <input type="reset" name="reset" value="Reset" />
  </p>
</form>
```

من خلال الكود السابق، قمنا بإنشاء نموذج يحتوي على حقل الاسم من نوع نص ، حقل البريد الإلكتروني من نوع نص كذلك، بالإضافة الى مساحة نص Text area . أخيرا لدينا زر لإرسال البيانات و زر لمسح قيم الحقول. كما يجب الإنتباه الى انه عند الرغبة في التعامل مع بيانات حقل معين، يجب تحديد اسم ذلك الحقل عن طريق الخاصية name لنتمكن من استخدامها لاحقا . بقي أن نشير أن الوسم form يحتوي على خاصيتين مهمتين الأولى method و تستخدم لتحديد كيفية إرسال البيانات. و الخاصية الأخرى method و تستخدم لتحديد الصفحة التي سوف نرسل البيانات لها.

الفرق بين استخدام GET و POST

عند ارسال نموذج HTML بالطرق العادية، نقوم بتحديد طريقة إرسال البيانات إما بواسطة POST أو GET . في حال كانت الطريقة المستخدمة في ارسال البيانات هي GET، سترسل كافة محتويات النموذج كاستعلام نصي query string . بمعنى أن متصفح الإنترنت سيقوم بتجميع كل محتويات الحقول التي تتبع لنموذج HTML و يضعها في سلسلة نصية طويلة Long strings و من ثم ارسالها الى الصفحة المحددة. أول مشكلة قد تواجهنا عند استخدام GET هو طول البيانات الممررة. لأن البيانات سوف تَجَمع في سلسلة نصية و من ثم تمرر الى متصفح الانترنت و الذي تتفاوت طول البيانات التي يدعمها من نسخة إلى أخرى. لذلك لا تعتبر GET كافية للتعامل مع التطبيقات الكبيرة. أيضا يعيب GET أنها ترسل البيانات بطريقة غير آمنة قد تجعلها عرضة للعبث. وفقا لذلك، كثيرا من مطوري الويب يقومون باستخدام POST لأنها أكثر أمانا و تقلل من فرص اعتراض البيانات من قبل المتطفلين.

سنقوم الآن بإنشاء الصفحة التي سوف تستقبل البيانات المرسلة عن طريق النموذج السابق. و ستكون كالتالي:

```

<?php

$name = $_POST['name'];
$email = $_POST['email'];
$message = $_POST['message'];

echo "name = $name<br/>";
echo "email = $email<br/>";
echo "message = $message<br/>";

?>

```

كما هو ملاحظ في المثال السابق. سنقوم بطباعة قيم الحقول المدخلة من قبل المستخدم في النموذج السابق و التي تم تخزينها داخل المصفوفة العامة POST و التي تم تحديدها مسبقا في النموذج السابق بواسطة الخاصية method.

التعامل مع القوائم المنسدلة Menu/List

قد نحتاج في بعض الاحيان الى بناء قائمة منسدلة تحتوي على اختيارات المتعددة لتتيح للمستخدم الاختيار من بينها. المثال يوضح كيفية ارسال نموذج HTML يحتوي على خيارات قائمة ذات خيارات متعددة:

```

<form action="greeting.php" method="post">
<table border="0" cellspacing="1" cellpadding="3"
bgcolor="#353535" align="center">
<tr>
<td bgcolor="#FFFFFF" width="50%">Name</td>
<td bgcolor="#FFFFFF" width="50%">
<input type="text" name="Name">
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">Greetings</td>
<td bgcolor="#FFFFFF">
<select name="Greeting">
<option value="Hello">Hello</option>
<option value="Hola">Hola</option>
<option value="مرحبا">مرحبا</option>
</select>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" colspan=2 align="center">
<input type="submit" name="SUBMIT" value="Submit">
</td>
</tr>
</table>
</form>

```

سنقوم الآن بإنشاء ملف آخر سيقوم بالتعامل مع البيانات المدخلة في النموذج السابق. الصفحة التالية هي greeting.php والتي سوف تتعامل مع النموذج السابق:

```
<?php  
  
$name = $_POST['Name'];  
$greeting = $_POST['Greeting'];  
echo "<i>$greeting</i> <b>$name</b>";  
?>
```

كما هو واضح ، كل ما قمنا بعمله هو تعريف متغيرين، الأول \$name والذي سيحتوي على قيمة الحقل name في النموذج السابق عن طريق استدعاء قيمته من المصفوفة \$_POST كالتالي:

`$_POST['Name']`

أما المتغير الثاني فهو \$greeting والذي سيحتوي على قيمة القائمة المنسدلة List . مع ملاحظة أن القيمة ستكون عبارة عن القيمة المختارة لأياً من الحقول option والتي تتبع للقائمة المنسدلة List. و يمكن الحصول على قيمة الحقل المختار كالتالي:

`$_POST['Greeting']`

التعامل مع مربعات الاختيار Checkbox

مربعات الاختيار checkbox تمثل احتمالين فقط. الأولى أن تكون مختارة checked ، ففي هذه الحالة يتم تمرير القيمة on الى المصفوفة \$_POST أو \$_GET . بخلاف ذلك لن يتم تمرير أي قيمة. المثال التالي يوضح كيفية التعامل مع حقول الاختيار:

```
<form action="test.php" method="post">  
<table border="0" cellspacing="1" cellpadding="3"  
bgcolor="#353535" align="center">  
<tr>  
<td bgcolor="#FFFFFF">موافق:<input type="checkbox" name="agree"/></td>  
<td bgcolor="#FFFFFF">غير موافق:<input type="checkbox" name="not_agree"/></td>  
</tr>  
<td bgcolor="#FFFFFF" colspan=2 align="center">  
<input type="submit" name="SUBMIT" value="Submit">  
</td>  
</tr>  
</table>  
</form>
```

و للتعامل مع النموذج السابق سنقوم بإنشاء صفحة الـ PHP التالية:

```
<?php

$agree = $_POST['agree'];
$not_agree = $_POST['not_agree'];
echo "<div dir='rtl'>";
echo "<b>موافق:$agree</b><br/>";
echo "<b>غير موافق:$not_agree</b>";
echo "</div>";
```

```
>
```

التعامل مع أزرار الاختيار radio

أزرار الاختيار radio من العناصر البسيطة و الفعالة عند التعامل مع نماذج HTML. في الوضع الافتراضي، نقوم بتحديد أحد أزرار الاختيار radio ليكون هو القيمة الافتراضية وذلك عن طريق إعطاءه الخاصية checked للدلالة على أن هذا الحقل هو القيمة الافتراضية. بمعنى أنه، إذا لم يقم المستخدم باختيار أي من أزرار الاختيار، فسيتم إرسال قيمة الحقل المحدد مسبقاً كقيمة افتراضية إلى أي من المصفوفتين \$_POST أو \$_GET عند إرسال النموذج. في حال كانت قيم أزرار الاختيار radio مرتبطة ببعضها، مثل لو أردنا تحديد ما إذا كان المستخدم ذكر أم أنثى. ففي هذه الحالة يجب أن نعطي جميع حقول الاختيار نفس الاسم. في هذه الحالة نقوم بإخبار متصفح الإنترنت بأنه في حال اختيار أحد الحقول فيعني هذه تجاهل بقية الحقول الأخرى.

```
<form action="test.php" method="post">
<table border="0" cellspacing="1" cellpadding="3"
bgcolor="#353535" align="center">
<tr>
<tr><td bgcolor="#FFFFFF">ذكر:<input type="radio" name="gender" value="ذكر"
checked/></td>
<td bgcolor="#FFFFFF">أنثى:<input type="radio" name="gender"
value="أنثى"/></td>
</tr>
<td bgcolor="#FFFFFF" colspan=2 align="center">
<input type="submit" name="SUBMIT" value="Submit">
</td>
</tr>
</table>
</form>
```

أخيراً، للتعامل مع النموذج السابق سنقوم بكتابة شفرة PHP التالية:

```
<?php

$gender = $_POST['gender'];
echo "<div dir='rtl'>";
echo "<b>$gender</b><br/>";
echo "</div>";
```

```
>
```

💡 إذا لم نقم باختبار زر الاختيار فإن القيمة المُرسلة مع النموذج تكون on في حالة اختيار الزر و ستكون فارغة null في حال لم يتم اختيار الزر.

استخدام الدالة (isset) مع النماذج

تُستخدم الدالة (isset) للتحقق من استخدام متغير ما (أي تم إعطاءه قيمة) وخصوصا عند التعامل مع النماذج للتحقق من أن المستخدم قام بإرسال النموذج (أي قام بالضغط على زر Submit).

المثال التالي يوضح كيفية استخدام هذه الدالة للتأكد من أن المستخدم قام بإرسال النموذج.
الصفحة :login.html

```
<div align="center" dir="rtl">
<form method="post" action="test.php">
<table border="1" bgcolor="#99CC66">
  <tr><td>الاسم: </td>
  <td> <input type="text" name="name" id="name"></td></tr>
  <tr><td> كلمة المرور : </td>
  <td><input type="password" name="password" id="password"></td></tr>
  <tr><td colspan="2"> <input type="submit" name="login" id="login"
value="دخول"></td></tr>
</table>
</form>
</div>
```

الآن سنقوم بإنشاء الصفحة :login.php

```

<div align="center" dir="rtl">
<?php

if(!isset($_POST['login']))
{
    echo"<font color='red'><h4>ليس لديك صلاحية الوصول الى هذه
    الصفحة.</h4></font>";
}
else
{
    $name = $_POST['name'];
    $password = $_POST['password'];

    if(empty($name))
    {
        echo "<font color='red'><h4>حقل اسم المستخدم مطلوب";
        echo "</h4></font>";
        echo " <input type='button' value='عودة' onclick='history.go(-1);> ";
    }
    elseif(empty($password))
    {
        echo "<font color='red'><h4>حقل كلمة المرور مطلوب";
        echo "</h4></font>";
        echo " <input type='button' value='عودة' onclick='history.go(-1);> ";
    }
    else
    {
        echo "<b>شكرا لتسجيل دخولك يا ".$name."</b>";
    }
}
?>
</div>

```

بداية استخدمنا الدالة (isset) وذلك للتحقق من أن المستخدم قام بالضغط على زر دخول كالتالي:

`isset($_POST['login'])`

تقوم الدالة (isset) بإرجاع true في حال قام المستخدم بالضغط على زر دخول، لذلك كل ما قمنا بعمله هو ، أنه في حال لم يتم المستخدم بالضغط على زر الدخول و حاول الوصول لصفحة PHP عن طريق كتابة عنوان الصفحة : `http://localhost/login.php` (أي وصل بطريقة غير شرعية)، حينها نقوم بطباعة الرسالة المناسبة في هذه الحالة.

في حال وصل المستخدم بالطريقة الافتراضية أي عن طريق الضغط على زر الدخول في الصفحة `login.html`، ففي هذه الحالة نقوم بالتأكد من أن المستخدم قام بإدخال اسم المستخدم و كلمة المرور و ذلك عن طريق استخدام الدالة (empty) في حال نسي المستخدم ادخال البيانات لأيا من الحقليين، قمنا بطباعة رسالة مناسبة و انشاء زر للعودة الى الصفحة السابقة. علما بأننا استخدمنا شفرة جافاسكربت مع زر العودة للعودة صفحة واحدة للخلف عند الضغط على زر عودة.

أخيرا الدالة (empty) ستقوم بإرجاع القيمة true في حال لم يتم المستخدم بإدخال أيا من الحقليين ، اسم المستخدم أو كلمة المرور. في حال قام المستخدم بإدخال البيانات السابقة بشكل صحيح قمنا بطباعة رسالة مناسبة كما هو واضح داخل جملة `else`.

استخدام الدالة trim()

تعتبر الدالة trim() من دوال PHP المبنية مسبقا وتستخدم لإزالة المسافات الفارغة في بداية و نهاية النص string . و تعتبر من الدوال التي يشيع استخدامها من نماذج HTML و ذلك لإزالة أي مساحات فارغة في القيم المدخلة من قبل المستخدم قبل إجراء أي عملية على تلك البيانات كتخزينها في قواعد البيانات مثلا. المثال التالي يوضح كيفية التعامل مع الدالة trim() :

```
<?php
// remove leading and trailing whitespace
// output 'a b c'
$str = ' a b c ';
echo trim($str);
?>
```

التحقق من أزرار الاختيار Validating checkbox and radio buttons

في أحيان كثيرة تحتوي نماذج HTML على أكثر من حقل ، و من أمثلة هذه الحقول أزرار الاختيار radio و checkbox . المثال التالي يوضح كيفية التحقق من أزرار الاختيار :

```
<div align="center" dir="rtl">
<h3>نموذج تسجيل مستخدم جديد</h3>
<form method="post" action="test2.php">
<table border="1" bgcolor="#99CC66">
<tr><td>الاسم:</td>
<td><input type="text" name="name" id="name"></td></tr>
<tr><td>البريد الإلكتروني:</td>
<td><input type="text" name="email"/></td></tr>
<tr><td>الجنس:</td>
<td><input type="radio" name="gender" value="ذكر"/>ذكر<br/>
<input type="radio" name="gender" value="أنثى"/>أنثى</td></tr>
<tr><td>شروط التسجيل:</td>
<td><input type="checkbox" name="agree" value="agree"/>موافق</td></tr>
<tr><td colspan="2"><input type="submit" name="register"
value="تسجيل"></td></tr>
</table>
</form>
</div>
```

كما هو واضح، الشفرة السابقة عبارة نموذج HTML يحتوي على أكثر من حقل من ضمنها حقول الاختيار. و يمكن ملاحظة، أننا قمنا بإعطاء حقول الاختيار radio نفس الاسم لكي نضمن التعامل مع اختيار واحد فقط للمستخدم و هو تحديد ما إذا كان جنسه ذكر أم أنثى. الآن سنقوم بإنشاء صفحة PHP و التي ستقوم بالتحقق من النموذج السابق كالتالي:

```

<div align="center" dir="rtl">
<?php

if(!isset($_POST['register']))
{
    echo"<font color='red'><h4>ليس لديك صلاحية الوصول الى هذه
الصفحة.</h4></font>";
}
else
{
    $name = $_POST['name'];
    $email = $_POST['email'];
    $gender = $_POST['gender'];
    $agree = $_POST['agree'];

    if(trim($name == null))
    {
        echo "<font color='red'><h4>حقل اسم المستخدم مطلوب";
        echo "</h4></font>";
        echo " الرجاء العودة و كتابة اسم المستخدم";
        echo " <input type='button' value='عودة' onclick='history.go(-1);'> ";
    }
    elseif(trim(empty($email)))
    {
        echo "<font color='red'><h4>حقل البريد الالكتروني مطلوب";
        echo "</h4></font>";
        echo " الرجاء العودة و كتابة البريد الالكتروني";
        echo " <input type='button' value='عودة' onclick='history.go(-1);'> ";
    }
    elseif($gender == "")
    {
        echo "<font color='red'><h4>حقل الجنس مطلوب";
        echo "</h4></font>";
        echo " الرجاء العودة و تحديد جنسك";
        echo " <input type='button' value='عودة' onclick='history.go(-1);'> ";
    }

    elseif($agree != "on")
    {
        echo "<font color='red'><h4>الرجاء الموافقة على شروط التسجيل";
        echo " <input type='button' value='عودة' onclick='history.go(-1);'> ";
    }

    else
    {
        echo "<h3>لقد اتممت عملية التسجيل </h3>". $name. " <h3>شكرا لك يا
بنجاح</h3>";
    }
}
}

```


أسئلة الفصل

١. ما الفرق بين كلا من : POST و GET .
٢. ما وظيفة كلا من الدوال التالية:
 - empty()
 - isset()
 - trim()
٣. كيف يمكن التحقق من أزرار الاختيار radio في حال كانت تتبع لمجموعة واحدة.
٤. أين سيتم تخزين قيم الحقول التي تتبع لنموذج HTML .