

جامعة طرابلس  
كلية تقنية المعلومات  
قسم نظم المعلومات

المقرر الدراسي:

تطوير التطبيقات

Application development

ITIS311

المحاضرة ( 6 )

اعداد: أ. فاطمة بشير القاضي  
F.alqadhi@uot.edu.ly

# التعامل مع نماذج HTML كمدخلات PHP

أكثر الأشياء أهمية عند التعامل مع نماذج HTML هو أن محتوى أي عنصر من النماذج في صفحات HTML سيكون متوفرة تلقائياً في أكواد PHP. تستخدم المتغيرات `$_GET` و `$_POST` لاسترجاع معلومات من النماذج مثل اسم المستخدم وكلمة المرور وغيرها.

**استخدام الطريقة POST:**

مثال:

يحتوي المثال التالي على نموذج HTML يحتوي على حقلين ادخال و زر إرسال:

```
<html>
<body>
  <form action="welcome.php" method="post">
    Name: <input type="text" name="fname"><br>
    E-mail: <input type="text" name="email"><br>
    <input type="submit" value="Send">
  </form>
</body>
</html>
```

# تابع التعامل مع النماذج

عند تعبئة الحقول في المثال السابق والضغط على الزر ارسال سيتم ارسال البيانات الى ملف `welcome.php` و المسمى `PHP` يجب أن يكون ملف `welcome.php` يحتوي على الكود التالي:

```
<html>
<body>

Welcome <?php echo $_POST["fname"]; ?> <br>
Your email address is: <?php echo $_POST["email"]; ?>

</body>
</html>
```

عند استخدام الطريقة `POST` داخل النموذج عندها نستخدم المتغير `$_POST` لاستقبال البيانات منه. كما نستخدم الخاصية `name` لعناصر النموذج كمعرف (`ID`) للمتغير لتحديد العنصر الذي يتعامل معه المتغير.

# تابع التعامل مع النماذج

## استخدام الطريقة GET

تختلف الطريقة GET عن الطريقة POST من حيث أن المعلومات ترسل عبر النماذج من خلال الطريقة GET بشكل مرئي للجميع أي تظهر المعلومات على رابط المتصفح كما تحتوي على كمية محدودة من المعلومات المرسله، على عكس الطريقة POST التي تخفي المعلومات المرسله باستخدامها الى حين الحاجة الى استخدامها، كما يمكن باستخدامها ارسال كمية كبيرة من المعلومات.

مثال:

```
<html>
<body>
<form action="welcome_get.php" method="get">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit" value="Send">
</form>
</body>
</html>
```

## تابع التعامل مع النماذج

عند تعبئة الحقول في المثال السابق والضغط على الزر ارسال سيتم ارسال البيانات الى ملف `welcome_get.php` و المسمى `welcome_get.php` يجب أن يكون ملف `welcome_get.php` يحتوي على الكود التالي:

```
<html>  
<body>
```

```
Welcome <?php echo $_GET["name"]; ?><br>  
Your email address is: <?php echo $_GET["email"]; ?>
```

```
</body>  
</html>
```

عند استخدام الطريقة `GET` داخل النموذج عندها نستخدم المتغير `$_GET` لاستقبال البيانات منه. كما نستخدم الخاصية `name` لعناصر النموذج كـ معرف (`ID`) للمتغير لتحديد العنصر الذي يتعامل معه المتغير.

## تابع التعامل مع النماذج

### متى نستخدم الطريقة GET؟

- لا يجب استخدام هذه الطريقة لإرسال بيانات هامة مثل كلمة المرور أو أي معلومات حساسة، و على أي حال لأن المتغيرات يتم عرضها على رابط المتصفح يمكن استخدام bookmark علامة للصفحة و التي يمكن أن تكون مفيدة في بعض الحالات .
- الطريقة get غير مناسبة عند ارسال كميات كبيرة من البيانات و أيضاً لا يجب استخدامها عندما تتجاوز الرموز 2000 حرف .

عند الضغط على رز الارسال سيظهر على المتصفح المعلومات التي قمت بإدخالها في حقول النموذج .

<http://www.test.com/welcome.php?name=Peter&email=peter@uog.edu.com>

# متى نستخدم الطريقة GET؟

## المتغير \$\_REQUEST

يمكن أن يستخدم المتغير \$\_REQUEST لجمع بيانات النموذج و استقباله لبيانات كلاً من الطريقتين GET و POST

ملاحظة:

لا يؤثر المتغير \$\_REQUEST في خاصية الاخفاء أو العرض وخاصية حجم البيانات المرسله في الطريقتين GET و POST

```
<html>
<body>
Welcome <?php echo $_REQUEST["name"]; ?><br>
Your email address is: <?php echo $_REQUEST["email"]; ?>
</body>
</html>
```

# التحقق من المعلومات المرسله باستخدام النموذج

يجب استخدام خاصية التأكد من صحة المعلومات المرسله باستخدام النماذج، وخاصةً ان كان سيتم إرسال المعلومات الموجودة بالنموذج الى قاعدة البيانات، وافضل طريقة لإنشاء تفعيل تحقق للنموذج على السيرفر هو ادراجه في النموذج نفس الصفحة المحتوية على النموذج بدلاً من الذهاب الى صفحة أخرى و عندها سيحصل المستخدم على رسائل الخطأ في نفس الصفحة و بتلك الطريقة يمكن استكشاف الأخطاء بسرعة و سهولة.

**المتغير \$\_SERVER :**

هو متغير عام وظيفته تخزين معلومات عن الخادم وبيئة التنفيذ.

**المتغير \$\_SERVER [ "PHP\_SELF" ] :**

هو متغير عام يقوم بإرجاع اسم الملف الذي ينفذ حالياً.

**\$\_SERVER["REQUEST\_METHOD"] :**

هو متغير عام يقوم بإرجاع طريقة الطلب المستخدمة للوصول إلى الصفحة (مثل POST).



# التحقق من المعلومات المرسلّة باستخدام النموذج

```
<html>
<body>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if ($_POST["fname"] == null)
        echo "the first name is requmeint <br> ";
    if ($_POST["email"] == null)
        echo "the email is requmeint ";
    } ?>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
    Name: <input type="text" name="fname"><br>
    E-mail: <input type="text" name="email"><br>
        <input type="submit" value="Send">
</form>
</body>
</html>
```

## التعامل مع عدة ازرار إرسال

عادةً ما يحتوي نموذج HTML على زر إرسال (Submit) واحد فقط ولكن هناك حالات نحتاج فيها إلى استخدام أكثر من زر إرسال واحد، و هنا نحتاج من PHP التحقق من الزر الذي تم الضغط عليه والإجراء الذي سيتم تنفيذه وفقاً للزر الذي تم الضغط عليه. إن وجود أزرار إرسال متعددة والتعامل معها من خلال PHP هو مجرد مسألة التحقق من اسم الزر مع القيمة المقابلة للزر باستخدام أحد العبارات الشرطية. عادتاً نستخدم أكثر من زر إرسال واحد بطريقتين:

الاولى : نموذج يُمكن المستخدم من إختيار أحد ازرار الارسال فقط والضغط عليه لتنفيذ مهمة محددة، وهنا نحن بحاجة الى استخدام الخاصيتين name و value للتفريق بين هذه الازرار داخل PHP .

الثانية : نموذج يُمكن المستخدم من إختيار عدة ازرار ارسال والضغط عليه لتنفيذ مهمة محددة، وهنا نحن بحاجة الى استخدام الخاصية name فقط للتفريق بين هذه الازرار داخل PHP .

## مثال على الطريقة الأولى

```
<html>
<body>
<?php
switch
($_REQUEST['btn_submit']) {
case "Button 1":
print "You pressed Button 1";
break;
case "Button 2":
print "You pressed Button 2";
break;
case "Button 3":
print "You pressed Button 3";
break;
} ?>
```

```
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
  <input type="submit" name="btn_submit" value="Button 1" />
  <input type="submit" name="btn_submit" value="Button 2" />
  <input type="submit" name="btn_submit" value="Button 3" />
</form>
</body>
</html>
```

## مثال على الطريقة الثانية

```
<html>
<body>
<form action = "<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
  Enter a number: <input type="text" name="number" size="3"> <br>
  <input type="submit" name="add" value="Add 10">
  <input type="submit" name="subtract" value="Subtract 10">
</form>
<?php
if ($_POST['add']) {
echo $_POST['number'] . " + 10 = " . ($_POST['number']+10); }
  else if ($_POST['subtract']) {
echo $_POST['number'] . " - 10 = " . ($_POST['number']-10); }
?>
</body>
</html>
```

## مثال 1: على التحقق من المعلومات المرسله باستخدام النموذج

يوضح هذا المثال كيفية التحقق من المدخلات المطلوبة قبل إرسالها باستخدام النموذج، وقواعد التحقق من صحة النموذج ادناه هي كما يلي:

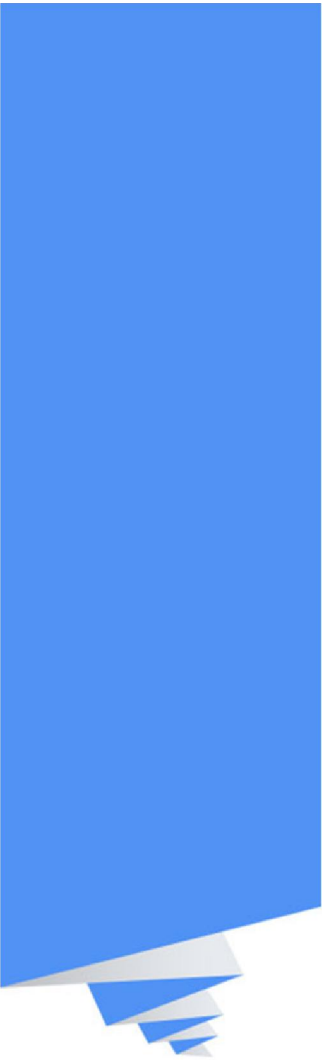
Name : يجب ملئ هذه الخانة لأنها مطلوبة.

Email : يجب ملئ هذه الخانة لأنها مطلوبة.

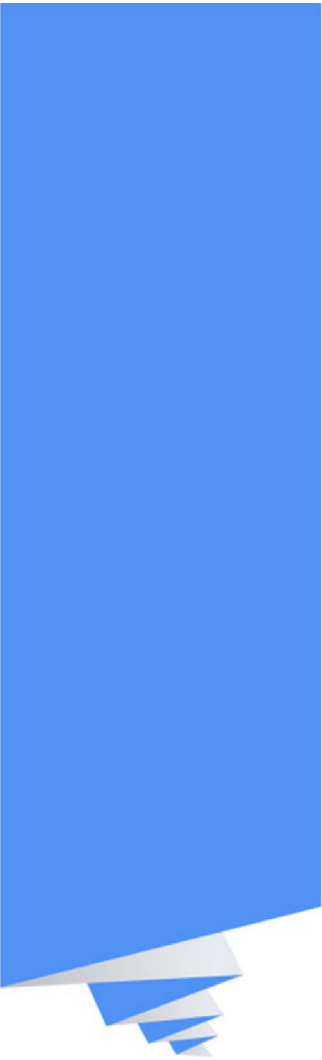
Website : لا تحمل هذه الخانة اي شرط تحقق لأنها غير مطلوبة يمكن للمستخدم ملئها او تركها فارغة.

Comment : لا تحمل هذه الخانة اي شرط تحقق لأنها غير مطلوبة يمكن للمستخدم ملئها او تركها فارغة.

Gender : يجب أن يتم اختيار نوع الجنس اما ذكر أو انثى.



```
<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>
<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = "";
$name = $email = $gender = $comment = $website = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = $_POST["name"];
    }
    if (empty($_POST["email"])) {
        $emailErr = "Email is required";
    }
}
```



```
else {
    $email = $_POST["email"];
}
if (empty($_POST["website"])) {
    $website = "";
} else {
    $website = $_POST["website"];
}
if (empty($_POST["comment"])) {
    $comment = "";
} else {
    $commen = $_POST["comment"];
}
if (empty($_POST["gender"])) {
    $genderErr = "Gender is required";
} else {
    $gender = $_POST["gender"];
}
}
```

## <h2>PHP Form Validation Example</h2>

<p><span class="error">\* required field</span></p>

<form method="post" action="<?php echo \$\_SERVER["PHP\_SELF"];?>">

Name: <input type="text" name="name">

<span class="error">\* <?php echo \$nameErr;?></span>

<br><br>

E-mail: <input type="text" name="email">

<span class="error">\* <?php echo \$emailErr;?></span>

<br><br>

Website: <input type="text" name="website">

<br><br>

Comment: <textarea name="comment" rows="5" cols="40"></textarea>

<br><br>

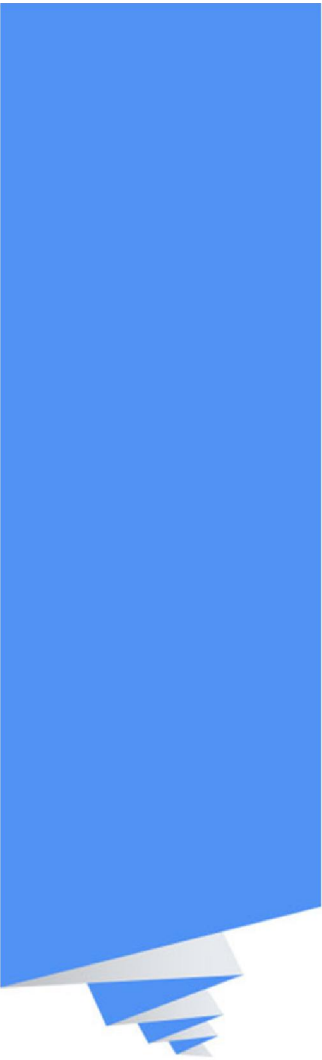
Gender:

<input type="radio" name="gender" value="female">Female

<input type="radio" name="gender" value="male">Male

<span class="error">\* <?php echo \$genderErr;?></span>





```
<br><br>
  <input type="submit" name="send" value="send">
</form>
<?php
echo "<h2>Your Input:</h2>";
echo "The name is :" . $name;
echo "<br>";
echo $email;
echo "<br>";
echo $website;
echo "<br>";
echo $comment;
echo "<br>";
echo $gender;
?>
</body>
</html>
```

## مثال 2: على التحقق من المعلومات المرسله باستخدام النموذج

يوضح هذا المثال كيفية التحقق من المدخلات المطلوبة قبل إرسالها باستخدام النموذج، وقواعد التحقق من صحة النموذج ادناه هي كما يلي:

**Name** : يجب ملئ هذه الخانة لأنها مطلوبة، إضافة للتحقق مما اذا كانت تحتوي فقط على حروف و مسافة فارغة .

**Email** : يجب ملئ هذه الخانة لأنها مطلوبة، إضافة للتحقق مما اذا كان عنوان البريد الإلكتروني منسقًا بشكل جيد أم لا.

**Website** : لا تحمل هذه الخانة اي شرط تحقق من ملئها لأنها غير مطلوبة يمكن للمستخدم ملئها او تركها فارغة، ولكن اذا قام المستخدم بملئها يتم التحقق مما إذا كانت بنية عنوان URL صالحة أم لا.

**Comment** : لا تحمل هذه الخانة اي شرط تحقق لأنها غير مطلوبة يمكن للمستخدم ملئها او تركها فارغة.

**Gender** : يجب أن يتم اختيار نوع الجنس اما ذكر أو انثى.

# الدالة preg\_match() و الدالة filter\_var()

**الدالة preg\_match():** تبحث هذه الدالة عن نمط معين في سلسلة نصية، ترجع True إذا كان النمط في حالة التطابق، و False في حالة عدم التطابق. شكلها العام:

`preg_match (pattern, input, matches, flags, offset)`

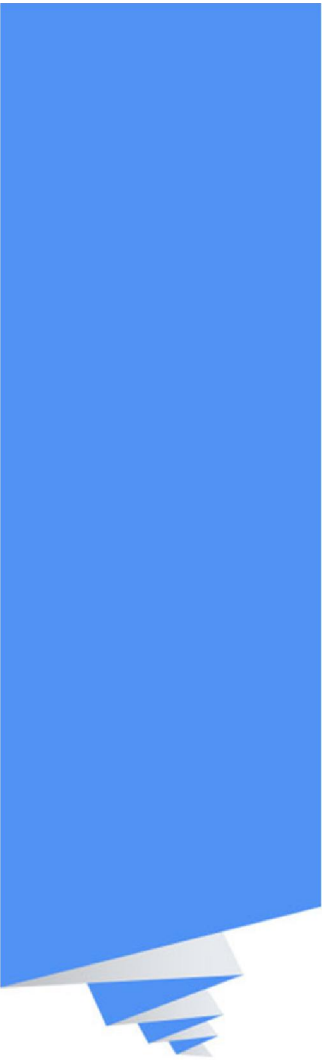
**Pattern:** يحتوي على تعبير يشير إلى ما يجب البحث عنه وهو مطلوب.  
**input:** السلسلة التي سيتم إجراء البحث فيها.

**الدالة filter\_var():** تقوم هذه الدالة بتصفية متغير باستخدام عامل التصفية المحدد. شكلها العام:

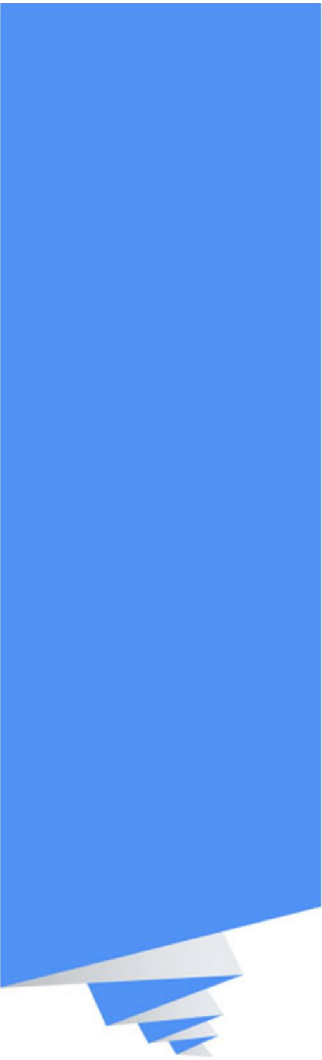
`filter_var(var, filtername, options)`

**Var:** المتغير المراد تصفيته وهو مطلوب.

**Filtername:** يحدد معرف أو اسم عامل التصفية المطلوب استخدامه وهو اختياري. الافتراضي هو `FILTER_DEFAULT`، مما يؤدي إلى عدم وجود تصفية.

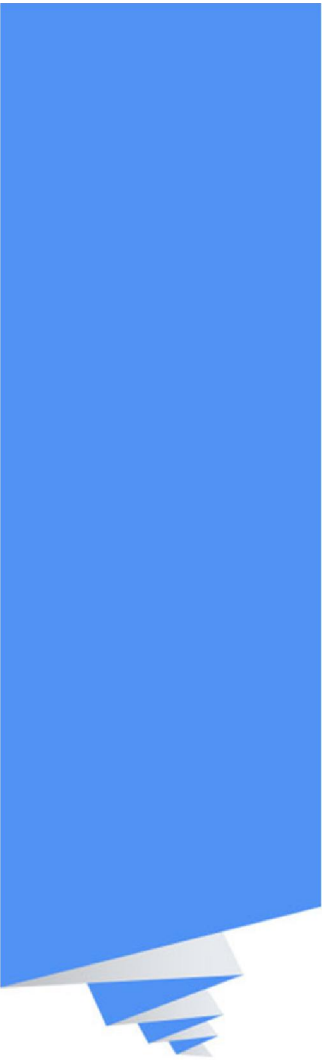


```
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>
<?php
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = $_POST["name"];
        // check if name only contains letters and whitespace
        if (!preg_match("/^[a-zA-Z- ]*$/", $name)) {
            $nameErr = "Only letters and white space allowed";
        }
    }
}
```



```
if (empty($_POST["email"])) {
    $emailErr = "Email is required";
} else {
    $email = $_POST["email"];
    // check if e-mail address is well-formed
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $emailErr = "Invalid email format";
    }
}
```

```
if (empty($_POST["website"])) {
    $website = "";
} else {
    $website = $_POST["website"];
    // check if URL address syntax is valid
    if (!preg_match("/^b(?:(:https?|ftp):\\W|www\\.)[-a-z0-9+&@#V%?=?~_!|:,;]*[-a-z0-9+&@#V%?=?~_!|:,;]/i",$website)) {
        $websiteErr = "Invalid URL";
    }
}
```



```
if (empty($_POST["comment"])) {
    $comment = "";
} else {
    $comment = $_POST["comment"];
}
if (empty($_POST["gender"])) {
    $genderErr = "Gender is required";
} else {
    $gender = $_POST["gender"];
}
} ?>
```

```
<h2>PHP Form Validation Example</h2>
```

```
<p><span class="error">* required field</span></p>
```

```
<form method="post" action="<?php echo $_SERVER["PHP_SELF"];?>">
```

```
  Name: <input type="text" name="name" value="<?php echo $name;?>">
```

```
  <span class="error">* <?php echo $nameErr;?></span>
```

```
  <br><br>
```

```
  E-mail: <input type="text" name="email" value="<?php echo $email;?>">
```

```
  <span class="error">* <?php echo $emailErr;?></span>
```

```
  <br><br>
```

```
Website: <input type="text" name="website" value="<?php echo $website;?>">
  <span class="error"><?php echo $websiteErr;?></span>
  <br><br>
  Comment:<textarea name="comment" rows="5" cols="40" > <?php echo $commen
?></textarea>
  <br><br>
  Gender:
  <input type="radio" name="gender" value="female">Female
  <input type="radio" name="gender" value="male">Male
  <span class="error">* <?php echo $genderErr;?></span>
  <br><br>
  <input type="submit" name="submit" value="Submit">
</form>
<?php
echo "<h2>Your Input:</h2>";
echo $name . "<br>";
echo $email . "<br>";
echo $website . "<br>";
echo $comment . "<br>";
echo $gender;
?> </body> </html>
```

# الانتقال الى الصفحة الجديدة بعد التحقق من المعلومات

يمكن الانتقال الى الصفحة الجديدة باستخدام PHP فقط، ولكن يُفضل إظهار رسالة للمستخدم توضح له أن عملية تعبئة النموذج قد تمت بنجاح، ويمكننا فعل ذلك باستخدام javascript .

## الدالة alert():

تعرض الدالة alert() في javascript مربع تنبيه مع رسالة وزر موافق، يتم استخدامها عندما نريد توضيح معلومات إلى المستخدم.

## الكائن window.location:

يمكن استخدام الكائن window.location للحصول على عنوان الصفحة الحالية (URL) أو إعادة توجيه المتصفح إلى صفحة جديدة.

```
if (empty($nameErr) && empty($emailErr) && empty($genderErr) && empty($websiteErr)){  
    echo "<script type='text/javascript'>  
    alert('All data were filed successfully. ')  
    window.location = 'ok.html';  
</script>";}
```