

جامعة طرابلس  
كلية تقنية المعلومات  
قسم نظم المعلومات

المقرر الدراسي:

تطوير التطبيقات

Application development

ITIS311

المحاضرة ( 5 )

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

# الدوال Functions

توفر PHP عددا كبيرا من الدوال يتجاوز عددها ألف دالة , ناهيك عن العدد الضخم من المكتبات التي تقوم بعدد لا بأس به من العمليات , لكن بشكل أو باخر ستحتاج الى دالة جديدة تقوم بمهمة معينة لبرنامجك . الشكل العام لتعريف الدالة هو :

```
function functionName(/*وسائط الدالة*/)
{
/*
جسم الدالة
*/
}
```

**وسائط الدالة:** يمكنها أن تقبل أي نوع من البيانات، وكذلك يمكنها إعادة أي نوع من البيانات أو عدم الرجوع بأي قيمة .

# تابع الدوال

الدالة التالية دالة لا تأخذ أي وسائط ولا تعيد أي قيمة فقط تطبع جملة على المتصفح، ولتشغيل هذه الدالة علينا استدعائها بكتابة اسمها و من ثم قوسين ( ) كما يلي:

```
<?php
function f_echo()
{
echo "<h1>PHP:hypertext processor</h1>";
}
f_echo();
?>
```

# تابع الدوال

اما الدالة التالية فهي تأخذ وسيطا لتقوم بطباعته ضمن وسمي `h1`، لاحظ ان المتغير `$in` هو متغير محلي معرف داخل الدالة فقط ولا علاقة له بالمتغير `$in` خارج الدالة.

```
<?php
$in = 'username1';
function f_echo($in)
{
echo "<h1>$in</h1>";
}
f_echo(10);
f_echo(12.332);oooo
f_echo('username2');
f_echo(true);
?>
```

# تابع الدوال

الدالة التالية تأخذ وسيطين و تُعيد حاصل الجمع

```
<?php
function sum($var1, $var2)
{ return $var1 + $var2; }
echo sum(10, 20);
?>
```

كما يمكن تمرير الدوال لبعضها البعض

```
<?php
function f_echo($in)
{ echo "<h1>$in</h1>"; }
function sum($var1,$var2)
{ return $var1+$var2; }
f_echo(sum(10,20));
?>
```

# تابع الدوال

## القيم الافتراضية للوسائط

في بعض الاحيان يكون للدوال وسائط اختيارية حيث يتم وازع قيمة افتراضية لها، فإذا لم يتم تحديد قيمة الوسيط , فسيتم اخذ القيمة الافتراضية بدل عنه، ويجب ان تكون جميع الوسائط بعد الوسيط الافتراضي افتراضية، اي لا يجوز ان تعريف الدالة بالشكل التالي :

```
<?php  
function function_name ($var1 = 'value', $var2)  
{ } ?>
```

مثال عن الاستخدام الصحيح :

```
<?php  
function f_echo($in = "text")  
{  
    echo "<h1>$in</h1>";  
}  
f_echo();  
?>
```

# تابع الدوال

إعادة أكثر من قيمة من الدالة

لا يجوز أن تعيد الدالة الواحدة أكثر من قيمة، لكن يمكن تجاوز هذه المشكلة باستخدام المصفوفات حيث يتم إعادة مصفوفة تكون عناصرها هي القيم المطلوبة.

```
<?php
function math($x)
{
return array($x * $x, log($x));
}
print_r(math(23));
?>
```

# تابع الدوال

## تمرير الوسائط بمرجعياتها

في بعض الاحيان , نحتاج الى تعديل قيمة الوسيط مباشرة في الدالة عوضا عن ارجاع قيمة منها واسنادها الى متغير، لجعل الوسائط تُمرر الى دالة بمرجعيتها (By Reference) يجب استخدام الرمز & قبل اسم الوسيط عند تعريف الدالة

```
<?php
function sum(&$var1, $var2)
{
    $var1 = $var1 + $var2; //the same as $var1+= $var2;
}
$num1 = 10;
$num2 = 15;
echo $num1; //outputs 10
echo '<br>';
sum($num1, $num2);
echo $num1; //outputs 25
?>
```

# التعامل مع الملفات

تعد معالجة الملفات جزءًا مهمًا من أي تطبيق ويب، وغالبًا ما تحتاج إلى فتح ملف ومعالجته لمهام مختلفة. لدى PHP العديد من الوظائف لإنشاء الملفات وقراءتها وتحميلها وتعديلها.

## قراءة الملفات

تستخدم الدالة `readfile()` لقراءة الملف وكتابته إلى المخزن المؤقت للإخراج، كما ترجع الدالة `readfile()` عدد البايتات المقروءة عند النجاح.

```
<?php  
echo readfile("welcome.txt"); ?>
```

تعتبر دالة `readfile()` مفيدة إذا كان كل ما تريد فعله هو فتح ملف وقراءة محتوياته.

# تابع التعامل مع الملفات

## فتح الملفات

أفضل طريقة لفتح الملفات هي استخدام الدالة `fopen()`، تمنحك هذه الدالة خيارات أكثر من وظيفة `readfile()`. تعتبر الدالة `fopen()` مسؤولة عن فتح الملفات في لغة PHP، أول وسيط داخل الدالة يستخدم لإدراج اسم الملف المراد فتحه أما الوسيط الثاني يحدد حالة الملف بعد فتحه :

```
<?php  
$file=fopen("welcome.txt","r"); ?>
```

## ملاحظات :

- اذا كانت الوظيفة `fopen()` غير قادرة على فتح ملف معين عندها ستعود القيمة 0.
  - لا يمكنك قراءة الملفات المفتوحة من خلال `w` و `a` و `x`
- في المثال التالي سيتم اظهار رسالة اذا كانت الوظيفة `fopen()` غير قادرة على فتح الملف .

```
<?php  
$file=fopen("welcome.txt","r") or exit("Unable to open file!"); ?>
```

# تابع التعامل مع الملفات

يمكن تحديد حالة الملف من خلال الحالات التالية :

الحالة	الشرح
r	للقراءة فقط وتبدأ مع بداية فتح الملف .
+r	قراءة / كتابة وتبدأ مع بداية فتح الملف .
w	كتابة فقط . فتح الملف ومسح محتويات الملف أو إنشاء ملف جديد إن لم يكن موجوداً .
+w	كتابة و قراءة . فتح الملف ومسح محتويات الملف أو إنشاء ملف جديد إن لم يكن موجوداً .
a	ملحق . يفتح و يكتب الملف حتى نهايته أو إنشاء ملف جديد اذا لم يكن موجوداً .
+a	ملحق . يقدم محتوى الملف من خلال كتابة الملف من بدايته لنهايته .
x	كتابة فقط . إنشاء ملف جديد . يعيد نتيجة خاطئة او خطأ اذا كان الملف موجود بالفعل .
+x	كتابة و قراءة . إنشاء ملف جديد . يعيد نتيجة خاطئة او خطأ اذا كان الملف موجود بالفعل .

# تابع التعامل مع الملفات

## إغلاق الملفات

تستخدم الوظيفة `fclose()` لإغلاق ملف معين .

Ex:

```
<?php
$file = fopen("test.txt","r");
//some code to be executed
fclose($file); ?>
```

## فحص آخر الملف

تفحص الوظيفة `feof()` نهاية الملف أي في حالة تم قراءة الملف بشكل كامل، وتعتبر مثالية في استخدام الحلقات عبر البيانات الغير معروفة الطول .

Ex:

```
<?php
if (feof($file))
echo "End of file"; ?>
```

# تابع التعامل مع الملفات

## قراءة الملف سطر بسطر

- تستخدم الوظيفة `fgets()` لقراءة سطر معين من الملف .
- ملاحظة : بعد استخدام الوظيفة `fgets()` سينتقل المؤشر للسطر الثاني تلقائياً .

Ex:

```
<?php
```

```
$file = fopen("welcome.txt", "r") or exit("Unable to open file!");
```

```
//Output a line of the file until the end is reached
```

```
while(!feof($file)) {
```

```
echo fgets($file). "<br />"; }
```

```
fclose($file); ?>
```

# تابع التعامل مع الملفات

## قراءة الملف حرف بحرف

- تستخدم الوظيفة `fgetc()` لقراءة حرف واحد من الملف .
- ملاحظة : بعد استخدام هذه الوظيفة سينتقل المؤشر للحرف الثاني تلقائياً .

Ex:

```
<?php
$file=fopen("welcome.txt","r") or exit("Unable to open file!");
while (!feof($file))
{
echo fgetc($file);
}
fclose($file); ?>
```

# تابع التعامل مع الملفات

## التحقق من وجود الملف أو المجلد

تستخدم الوظيفة `file_exists()` لتنفيذ عملية تحقق بسيطة سواء كان الملف أو المجلد موجودًا أو لا قبل محاولة الوصول إليه، ويفضل استخدامها دائمًا لتجنب رسائل الخطأ والتحذير.

Ex:

```
<?php  
echo file_exists ("welcome.txt"); ?>
```

## معرفة حجم الملف بالبايت

تستخدم الوظيفة `filesize()` لإرجاع حجم الملف بالبايت

Ex:

```
<?php  
echo filesize ("welcome.txt"); ?>
```

# تابع التعامل مع الملفات

## قراءة الملفات

يحتوي PHP على العديد من الوظائف لقراءة البيانات من ملف، يمكنك القراءة ابتداءً من حرف واحد فقط إلى الملف بأكمله من خلال الوظيفة `fread()`.

`fread(file handle, length in bytes)`

## قراءة عدد ثابت من الحروف

المثال التالي يقرأ 20 بايت من ملف "welcom.txt" بما في ذلك المسافات.

```
<?php
$file = fopen("welcom.txt","r");
fread ($file,"20");
fclose ($file);
?>
```

# تابع التعامل مع الملفات

## قراءة المحتويات الكاملة للملف

يمكن استخدام الوظيفة `fread()` بالاقتران مع الوظيفة `filesize()` لقراءة الملف بأكمله مرة واحدة

Ex:

```
<?php
$file = fopen("welcome.txt","r");
fread ($file,filesize("welcome.txt"));
fclose ($file); ?>
```

وأسهل طريقة لقراءة محتويات الملف بالكامل في PHP هي باستخدام الوظيفة `readfile()` تتيح هذه الوظيفة قراءة محتويات الملف دون الحاجة إلى فتحه. سيولد المثال التالي نفس نتائج المثال أعلاه:

Ex:

```
<?php
readfile ("welcome.txt"); ?>
```

# تابع التعامل مع الملفات

## الكتابة داخل الملفات

في PHP يمكنك كتابة البيانات إلى ملف أو إلحاقها إلى ملف موجود باستخدام وظيفة `fwrite()`.

`Fwrite (file handle, string)`

Ex:

```
<?php
$file = fopen("test.txt","w");
fwrite($file,"Hello World. Testing!");
fclose($file);
?>
```

في المثال أعلاه لم يكن الملف "test.txt" موجوداً ، فستقوم PHP تلقائياً بإنشائه وكتابة البيانات بداخله، لكن إذا كان الملف موجوداً بالفعل فستقوم PHP بمسح محتويات هذا الملف قبل كتابة البيانات الجديدة ، ومع ذلك إذا أردنا فقط إلحاق الملف والحفاظ على المحتويات الموجودة فاستخدم المعرف "a" بدلاً من "w" داخل الوظيفة `fopen` .

# تابع التعامل مع الملفات

## إعادة تسمية الملفات

باستخدام الوظيفة `rename()` يمكنك إعادة تسمية ملف أو مجلد في PHP

```
rename (old_name, new_name, context_optional_)
```

Ex:

```
<?php
```

```
rename ("test.php","test1.php");
```

```
rename ("images","pictures");
```

```
rename ("/test/file1.txt","/home/docs/my_file.txt"); ?>
```

## ملاحظات:

- الوظيفة `rename()` ترجع `True` عند النجاح و `False` عند الفشل.
- يمكن ان نستخدم هذه الوظيفة ايضاً في تغيير مسار الملف او المجلد كما موضح في المثال اعلاه.

# تابع التعامل مع الملفات

## إزالة الملفات

يمكنك باستخدام الوظيفة `unlink()` حذف الملفات أو المجلدات

```
unlink(filename, context_optional_)
```

Ex:

```
<?php
$file = fopen("test.txt","w");
echo fwrite($file,"Hello World. Testing!");
fclose($file);
unlink("test.txt");
?>
```