مقدمة عن لغة ال PHP :

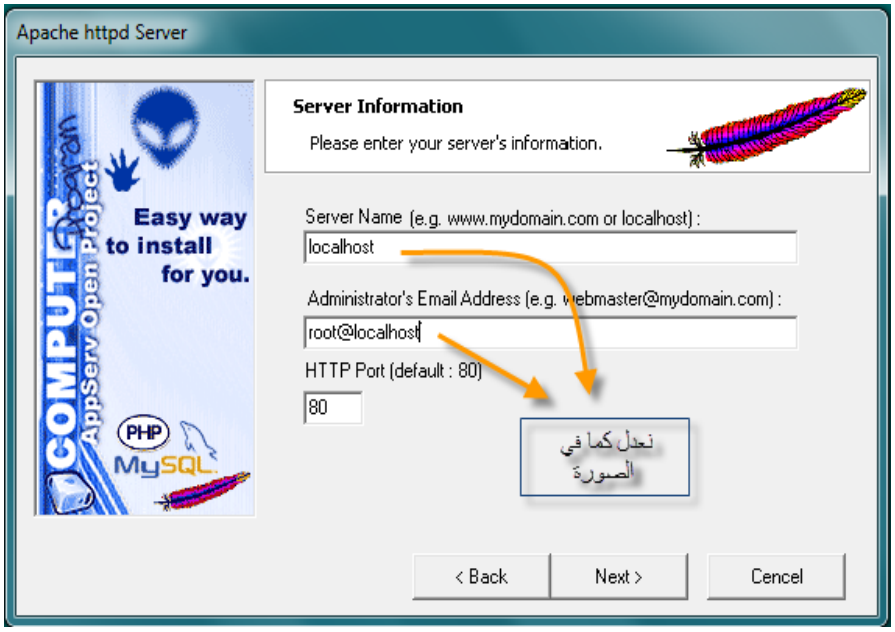
لغة ال PHP هي من أكثر لغات برمجة المواقع الاكترونية استخداما على الانترنت و هي لغة برمجة متخصصة بتصميم المواقع لا يمكن استخدامها في برمجة البرامج تعتبر لغة ال PHP مفتوحة المصدر open source أي يمكن للشركات أن تستخدم هذه اللغة بشكل مجاني كما أنها لغة نصية تبرمج من طرف المخدم أي لا يمكن للمستخدم رؤية الكود البرمجي و تتميز أنها تتعامل مع جميع أنظمة التشغيل الويندوز و اللينكس (اليونكس) و الماكنتوش بكفاءة عالية و كما تتعامل مع قواعد البيانات و خاصة قاعدة بيانات MYSQL و هي لغة غرضية التوجه أي أنها تعتمد على الأغراض في البرمجة كما في لغات البرمجة المتقدمة مثل لغة الجافا و السي .

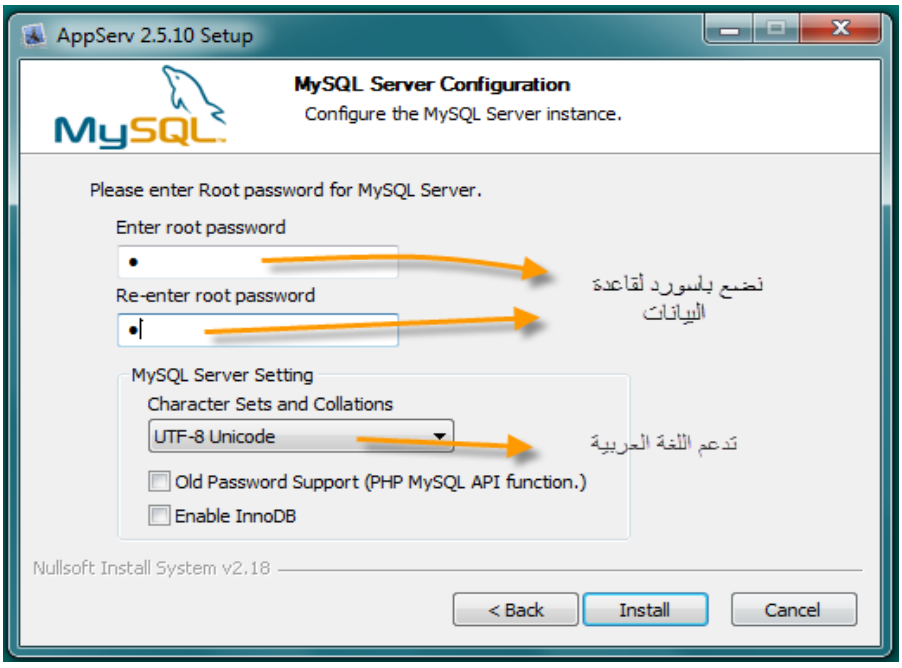
يقوم المستخدم بإعطاء أمر طلب صفحة ما HTTP-request من قبل المتصفح عن طريق البروتوكل http وهو بروتوكول نقل الصفحات التشعبية hyper-text transfer protocol و عندما يجد السيرفر هذه الصفحة المطلوبة في السيرفر يرسل HTTP-response حيث يشكل صفحة HTML و يرسلها ولا يعرف المستخدم ماهية الكود البرمجي المكتوب .

- ما نحتاجه للبرمجة بهذه اللغة هو متصفح انترنت و محرر نصوص يفضل استخدام الدريم ويفر و مخدم (سيرفر) يمكن استعمال برنامج ما مثل برنامج appserv الذي يحول جهاز الكومبيوتر الى سيرفر و هو الذي يعالج أكواد ال PHP .

تنصيب الأباتشي :

بعد فتح ملف appserv المرفق نقوم بتنصيبه كما في أي برنامج لكن بالتعديل كما في الصورتين :





By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

للتأكد من أنه تم التنصيب بنجاح نقوم بفتح المتصفح و نطبع الرابط التالي http://localhost و اذا أعطانا إشارة خطأ فهذا يعني أنه هناك خطأ ما في التنصيب .

يجب قبل تعلم لغة ال PHP أخذ و لو فكرة بسيطة عن لغة HTML التي هي لغة توصيف أكثر من كونها لغة برمجة بحد ذاتها أي أنها مسؤولة عن مظهر الصفحة و تتولى منظة W3 (www.W3.org) وضع المواصفات المعيارية لل HTML و هي مؤلفة من وسوم tags و التي تتكون بشكل عام من الشكل التالي :

```
<html>
<head><title></title></head>
<body>
<?php
// هنا نضع كود PHP
?>
</body>
</html>
```

لمزيد من المعلومات عن ال HTML اطلع على HTML basics

بدايتك للبرمجة في لغة ال PHP

المجلد الواجب وضع ملفات ال PHP فيه يكون على المسار C:/appserv/www و يتم عرضه عبر المتصفح فاذا كان اسم الملف مثلا example.php فإن المسار على المتصفح http://localhost/example.php

مثال لبرنامج صغير في لغة ال PHP اكتبه باستخدام notepad أو dreamweaver أو أي محرر PHP و احفظ الملف باسم test.php مثلا و ضعه في المجلد www و اعرضه في المتصفح و انظر النتيجة و عند التعديل على البرنامج يمكنك استخدام ctrl+F5 لإعادة تحميل الصفحة في المتصفح لأن المتصفح قد يخزن الصفحة ضمن الكاش

```
<html>
<head><title>my first pro</title></head>
<body>
<?php
echo "hani";
?>
</body></html>
```

عند كتابة أي كود PHP نبدأ عادة ب <?php و ننتهي ب ?> و هي الطريقة الأكثر شيوعا لتعريف بدء و نهاية كود ال PHP

تستخدم echo لطباعة عبارة ما أو حساب عملية حسابية و هي تحتاج الى زواج من عبارات الاقتباس "" وننسى وضع ; في النهاية

عدل كود echo الى الكود التالي و انظر النتيجة

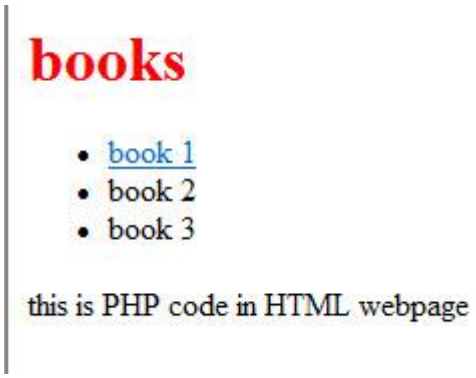
```
Echo "the result is ",5+4;
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

مثال برنامج صغير آخر (تضمين أكواد ال PHP داخل كود ال HTML) نحفظ البرنامج باسم ما بامتداد php. و سيقوم السيرفر بتفسير كود ال php و عرضه ككود html

```
<html>
<head>
<meta name=keywords content="computer,books,technology,learning" />
<title>books</title>
</head>
<body>
<font color="#FF0000">
<h1>books</h1></font>
<ul>
<li><a href="link1.html">book 1</a></li>
<li>book 2</li>
<li>book 3</li>
</ul>
<?php
Echo "this is PHP code in HTML webpage" ;
?>
</body>
</html>
```

و تكون النتيجة كالتالي



By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

الأمر print يعمل نفس الأمر echo تقريبا ، يمكن إدخال وسوم ال HTML داخل كود ال PHP و ذلك كما في المثال التالي :

```
<html>
<head><title>my first pro</title></head>
<body>
<?php
print ('<a href=test.html>PHP</a>');
?>
</body></html>
```

انتبه أنه لا حاجة الى إشارتين التنصيص "" التي نضعها مع الروابط التشعبية في ال HTML حيث كنا نضع في ال HTML

```
<a href="test.html">PHP</a>
```

التعليمة التي تعطينا معلومات واسعة عن بيئة ال PHP هي ()phpinfo

طبق المثال و انظر النتائج

```
<html>
<body>
<?php
phpinfo();
?>
</body></html>
```

الاستدعاء

و هي عملية استدعاء أو تضمين كود موجود في ملف معين ليصبح في ملف آخر هناك أربع طرق للاستدعاء و هي

```
Include ('path1');
```

اذا كان هناك أي خطأ في الكود أو المسار خطأ يعطي تحذير و يتابع العمل

```
Require();
```

اذا كان هناك أي خطأ يتوقف معه كود ال بي اتش بي

```
Include_once();
Require_once();
```

نفس include و require لكنها تقوم بعملية الاستدعاء لمرة واحدة فقط و هو مفيد في حال الاستدعاء لملفات حاوية على function (نتعرف عليها لاحقا) حيث لا يجب تكرار تصريح الدالات أكثر من مرة

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

مثال للاستدعاء

اكتب الكود التالي في صفحة جديدة و احفظها في في المجلد www باسم new.php مثلا

```
<html>
<body>
<?php
echo "hello world" ;
?>
</body></html>
```

افتح صفحة جديدة باسم ما و ليكن example.php و خزنها في نفس المسار و اكتب الكود التالي

```
<html>
<body>
<?php
echo "this is php code";
echo '<br>' ;
include('new.php');
?>
</body></html>
```

التعليقات في لغة ال PHP

نستخدم التعليقات لتوضيح بعض أسطر الكود لأجل الصيانة أو لأجل مستخدمين معينين أو لإبطال عملية برمجية معينة

يمكن وضع إشارة // قبل العملية الحسابية و لإبطال هذه العملية و تصبح عبارة عن تعليق كما يمكن استخدام رمز # قبل السطر البرمجي لإبطاله و يمكن أيضا أن نضع */ في بداية الكتلة البرمجية و /* في نهاية الكتلة البرمجية المراد إبطالها أو تحويلها لتعليق .

```
<html>
<body>
<?php
echo "this is php code"; //this is comment
echo '<br>' ; # this is another comment
/* this is
Another comment */
?>
</body></html>
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

المتحولات في لغة ال PHP

هي أي سلسلة حرفية لا تحوي على فراغ و لا تحتوي أي رمز خاص ك ؟ أو + أو – و يجب أن تبدأ بإشارة $ تأخذ المتحولات قيمها عن طريق المساواة

مثال صغير عن المتحولات

```
<html>
<body>
<?php
$number1=10;
echo "number 1 is ", $number1;
$number2= 30;
$number3=$number1*$number2;
echo '<br>',"number 3 is ",$number3;

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

أنواع المعطيات (البيانات) في لغة PHP

تكون على أربعة أنواع

١- Integer يشمل الأرقام الموجبة و السالبة و الصفر
٢- Bolean يتضمن قيمتين فقط true أو false
٣- String أي رموز تكتب ما بين إشارتي الإقتباس ""
٤- Float الأرقام الحاوية على فواصل عشرية
٥- Null يشير الى لا شيء و يستخدم لتهيئة المتحولات

الأنواع الرقمية (الأنظمة العددية المستخدمة للتعامل مع البيانات):

١- Decimal النظام العشري من ٠ حتى ٩ أي تحوي على عشرة أرقام
٢- Octal النظام الثماني يحتوي على ثمانية أرقام من ٠ حتى ٧ و يتم تمييزها عن النظام العشري بوضع ٠ قبل القيمة الثمانية مثلا ٠٦٥
٣- Hexa-decimal يتضمن ١٦ قيمة من ٠ حتى ٩ و ٦ أحرف من a حتى f تمثل الأرقام من ١٠ حتى ١٥ و يتم تمييزها بوضع علامة 0X قبل الصيغة الست عشرية مثلا 0X9CF

للتحويل من القيمة الثمانية أو الست عشرية الى العشرية نقوم كما في المثال التالي

065 = 5*(8^0)+6*(8^1) = 53

حيث وضعنا رقم ٨ لأنها في النظام الثماني نريد تحويلها للنظام العشرية و رقم ٠ لأن ترتيب الخمسة ٠ و ترتيب ال ٦ هو ١ و هكذا

0X9CF=F*(16^0)+C(16^1)+9*(16^2)

حيث قيمة f هي ١٥ و قيمة c هي ١٢

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

تحتوي لغة ال PHP على أنواع معطيات أخرى

array أو المصفوفة و هي عبارة عن سلسلة من الأرقام أو سلاسل حرفية توضع بين قوسين و تفصل بينهما فواصل

Objects و هي عبارة عن أنواع معطيات مركبة تستخدم فيها البرمجة غرضية التوجه مثل ال class نتكلم عنه في فصل خاص

null أو الا شيء حيث يستخدم لتهيئة المتحولات قبل إسناد القيم الرقمية أو الحرفية اليها

Resource و هي كلمة محجوزة في لغة PHP و المتحول اذا كان من هذا النواع فهذا يعني أنه يمثل متحولا للملف file handle أو أنه يمثل قيمة ثنائية مثل صورة أو وسائط متعددة كالصوت والفيديو

Exponential و هو نوع المعطيات الأسية نكتب رقم الأساس بعد ذلك نكتب e ثم رقم الأس مثلا 2e10

متحولات المتحولات :

نستطيع إنشاء متحولات من المتحولات التي نقوم بإشائها سابقا بتكرار إشارة الدولار قبل اسم المتحول و بذلك نشير الى متحول جديد باستخدام محتوى متحول سابق

أي اذا كان لدينا متحول $user_name="admin" فعندما نضع $$user_name="khaled" فإنه يتشكل لدينا متحول جديد $admin يأخذ القيمة khaled طبق المثال و انظر النتيجة

```
<html>

<body>

<?php

$user_name="admin";

$$user_name="khaled";

echo $admin;

?>

</body></html>
```

التعابير في لغة php :

و نقصد بها تجميع للمعاملات مع عمليات الجمع و الطرح و الضرب و القسمة و الأقواس (* / - + و الأقواس) و ذلك أثناء الاسنادات تعطى الأقواس الأولوية الأولى يأتي بعدها الضرب و القسمة و في المرتبة الثالثة الجمع و الطرح ترتب العمليات من اليسار الى اليمين عند التكرار لنفس الأولوية لدينا مثلا 8.5 = 9-5/4*(8+6)

```
<html>

<body>

<?php

$number=($five=5)+($ten=10);

echo $number;

?>

</body></html>
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

الثوابت :

يلتزم بالقيمة التي نسندها اليه في المرة الأولى و ما يميز التمحول عن الثابت هو أن المتحول دائما يسبق بإشارة $ دائما و ما ينطبق على قواعد تسمية المتحولات ينطبق على قواعد تسمية الثوابت يتم التصريح عن الثوابت بالأمر define(constant_name,constant_value)

```
<html>
<body>
<?php
define (pi,3.14);
$a=5;
echo "circle surface= ", pi*$a*$a;
?>
</body></html>
```

معملات حسابية أخرى

١- معامل الموديول أي اذا تمت قسمة رقم على رقم و كان هناك باقي أو فائض فإنه يعرض قيمة هذا الفائض و يرمز له بإشارة % نطبق المثال التالي

```
<html>
<body>
<?php
$number=($num1=21)%($num2=10);
echo $number;
?>
</body></html>
```

ستكون النتيجة في هذا المثال هي ١ لأن ناتج القسمة هو ٢.١ و الفائض هو ١

٢- معامل تكرار الزائد أو تكرار الناقص قبل المتحول يزيد قيمة ١ أو يطرح قيمة ١ لقيمة Sa أي

```
<?php
$a=5;
++$a; //try --$a and try $a++
echo $a;
?>
```

عند وضع إشارتي الزائد بعد المتحول يؤدي تنفيذ هذه العملية الى إبقاء قيمة $a مساوية للواحد و لكن بعد الانتهاء من التفسير ستصبح قيمة $a مساوية الى زيادة بقيمة ١

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

العمليات الممكن تطبيقها على السلاسل الحرفية

الدمج بين السلاسل الحرفية والاستخلاص أي اقتطاع جزء من السلسلة الحرفية

```
<html>
<body>
<?php
$str1="hello";
$str2="php";
$str3=$str1.$str2; //دمج السلاسل الحرفية
echo $str3;
print '<br>';
$str4=substr($str3,2,4); //استخلاص جزء من السلسلة  الحرفية حيث هنا بدأ من الحرف رقم ٢ و أخذنا ٤ أرقام بعده
echo $str4;
?>
</body></html>
```

Hello لها ترتيب الأرقام التالي 01234 حيث تأخذ ال H قيمة 0 و ال O قيمة 4

العمليات المنطقية :

لدينا قيمتين منطقيتين هما 1 و 0 أو true و false و لدينا العمليات المنطقية التالية

١- Not و تمثلها إشارة ~ تعكس القيمة المنطقية
٢- And و تمثلها إشارة & خرجها صفرا اذا كان أحد معاملا صفر ولا يكون خرجها واحد إلا اذا كان المعاملان واحد
٣- Or و تمثلها إشارة | عكس الواحد تكون قيمة الخرج فيها واحد اذا كان أحد المعاملين واحد ولا تكون صفر إلا اذا كان المعاملان صفر
٤- Xor و تمثلها إشارة ^ و تعرف أيضا بعملية الخلاف أي اذا كان هناك خلاف بين المعاملين تكون النتيجة واحد أما اذا كانا متماثلين فيكون الخرج هو القيمة صفر
٥- Bitwise left و تمثلها >> تقوم بضرب القيمة بمقدار اثنين كل مرة حيث تقوم بإزاحة القيمة المنطقية الى اليسار مما يؤدي الى استخدامنا الى القيم الثنائية و تطبق على القيم الثنائية و ليس على المنطقية لكنها تعتبر من العمليات المنطقية
٦- Bitwise right و تمثلها << تقوم بعملية القسمة على اثنين و بذلك تحول القيم الثنائية الى قيم مقسومة على اثنين عن طريق الإزاحة على اليمين أيضا هي تطبق على القيم الثنائية و ليس على المنطقية لكنها تعتبر من العمليات المنطقية

```
<?php
$b1=true;
$b2=false;
$b3=$b1^$b2;
echo $b3;
?>
```

في المثال السابق ستكون النتيجة هي 1 لأنه هناك خلاف بين $b1 و $b2

```
<?php
$b1=1;
$b2=$b1<<4; //=1*2*2*2*2 =16
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
echo $b2;
?>
```

و النتيجة هي 16

```
<?php
$b1=5;
$b2=$b1<<1; //=5/2 =2.5
echo $b2;
?>
```

و النتيجة هي 2 حيث أنه لا يضع فواصل

مفهوم الإسناد السريع :

ليكن لدينا $a=5 و تريد تعديل قيمته وفق $a=$a+4 فيمكن ذلك عن طريق كتابة $a+=4

أو ليكن متحول يحوي قيمة نصية $a="this is " و نريد تعديل قيمة المتحول وفق $a=$a+"quick assignement" فيمكن عمل ذلك عن طريق كتابة $a.="quick assignement"

و هذا هو مفهوم الإسناد السريع الذي يوفر الوقت

معاملات المقارنة :

تستخدم في البنى الشرطية و الحلقات المستخدمة في اللغات البرمجية و يؤدي الشرطالى قيامنا بعمل ما و تنفيذنا لكتلة برمجية أو عدم تنفيذنا

المساواة و رمزه == انتبه أنه = هي للاسنادات

المساواة المطابقة === تطابق القيمة و نوع المعطيات اذا كان لدينا $a=3 و هي كما تعلمنا أنها من نوع integer فإذا سألنا هل تساوي $a القيمة 3.0 فإن القيمة ستكون false لأنها مساوية لها بالقيمة لكنها تحوي فاصلة أي أنها float

عدم المساواة =! عكس المساواة

عدم المساواة المنطقية ==!

لدينا أيضا إشارة الأكبر المعروفة > و إشارة أكبر أو يساوي =>

إشارة الأصغر < و إشارة الأصغر أو تساوي =<

ملاحظة خرج العملية الخاصة بالمقارنة يكون خرجا منطقيا أي إما true أو false

```
<?php
$t=5;
echo $t==5;
?>
```

نتيجة المثال السابق هي 1 و إلا فتكون النتيجة هي null أو لا شيء

```
<?php
$a="c";
$a1="d";
```

```
$b=(int)($a<$a1);
echo $b;
?>
```

النتيجة هي 1 لأن شفرة ال C هي بالفعل أقل من شفرة ال d و ذلك بعد الصيغة النصية الى رقمية عن طريق int

عبارة الشرط If :

الصيغة العامة لعبارة الشرط if

```
If(expression1) {code1 here;}
Elseif(expression2) {code2 here;}
Else {code3 here;}
```

طبق المثال التالي و غير في قيمة $a و انظر النتائج

```
<?php
$a=2;
if($a>3) {
echo "$a is greater than 3";}
elseif($a<3) {
echo "$a is less than 3"; }
else {
echo "$a equals 3"; }
?>
```

عبارة الشرط switch :

تعني التفريع أو التحويل و نستعملها في حال استعمال شرط واحد و مقارنته مع عدة قيم و اذا لم يساوي إحدى هذه القيم فإنه يعرض القيمة الافتراضية و الشكل العام لهذه العبارة

```
Switch(expression) {
Case value1 : {code1;break;}
Case value2: {code2;break;}
.
.
Default {codex; } }
```

طبق المثال و غير قيمة $day و انظر النتائج

```
<?php
$day=5;
switch($day){
case 1: {echo 'monday';break;}
```

```
case 2: {echo 'tuseday';break;}
case 3: {echo 'wednesday';break;}
case 4: {echo 'thursday';break;}
case 5: {echo 'friday';break;}
case 6: {echo 'saturday';break;}
case 7: {echo 'sunday';break;}
default: {echo 'wrong number';}
}
?>
```

بنية التكرار while :

و هي عملية تكرار لعملية ما طالما أن الشرط محقق و لها شكلين رئيسين مع do و من دون do

الشكل الأول While(condition) {code}

الشكل الثاني Do {code} while(condition)

و الفرق بينهما هو أن while تنفذ الكود مرة واحدة على الأقل حتى و إن لم يكن الشرط محققا أما while من دون do لا تنفذ الكود إلا اذا كان الشرط محققا انظر المثال التالي

```
<?php
$x=0;
while($x<=10) {
$x=$x+1;
echo '<br>';
echo $x;
}
?>
```

نلاحظ أننا اذا وضعنا الصيغة الثانية ستكون لنا نفس النتيجة أي افعل مجموعة الأوامر (زيادة قيمة $x و طباعة فراغ و طباعة قيمة $x) طالما أن الشرط محقق و هو أن $x أصغر أو تساوي القيمة 10

بنية التكرار for :

تستخدم لأجل عدد معروف من المرات بناء على تحقق شرط معين و الشكل العام له

```
For(start_with_value;stopping_loop_condition;increasing_value)
{code}
```

مثال على ذلك

```
<?php
$x=0;
for($i=0;$i<=10;$i++) {
$x=$x+1;
```

```
echo $x," ";
}
 ?>
```

الحلقات اللانهائية تستخدم من أجل البرمجيات التي تعمل بشكل لا نهائي مثل السيرفرات

```
While(true) {code}
For(;;) {code}
```

بنى المتابعة و التوقف :

تستخدم مع الحلقات سواء من أجل for أو while و تستخدم دائما مع الشروط أي مع العبارة if

الهدف من استخدام break هو كسر الحلقة عند توفر شرط معين

و نستخدم continue لمتابعة استخدام بعض تكرارات الحلقة في حال توفر شرط معين و في حال عدم التحقق فإننا نتجاهل تلك التكرارات

```
<?php
$x=0;
for($i=0;$i<=10;$i++) {
$x=$x+1;
echo " ";
if($x%2!=0)
continue;
echo $x; }
 ?>
```

في المثال السابق سألنا هل باقي القسمة على 2 لا يساوي الصفر أي هل هو عد فردي اذا كان عدد فردي فتخطى هذا الرقم و تابع الحلقة أي أنه سوف يطبع الأرقام 2 , 4 , 6 , 8 , 10 أما اذا وضعنا break بدل continue فإنه سوف يكسر الحلقة ولا يعود لها نهائيا

```
$i=0;
while(true) {
$i=$i+1;
echo " ";
if($i==10)
break;
echo $i;
}
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
?>
```

هذا المثال سيطبع القيم من 1 الى 9 لأنه عند القيمة 10 سيتم كسر الحلقة و لن يعود لها نهائيا

تذكر أنه while(true) {} هي حلقة التكرار اللانهائية

```
<?php


for($i=0;$i<10;$i++){

if($i>4 && $i<7)  // $$ تعني أنه يجب أن يتحقق كلا الشرطين

continue;

echo $i;

}

?>
```

ستكون النتيجة بالتأكيد هي 01234789 و هنا استخدمنا المتحول $i الموجود داخل حلقة ال for

التوابع functions :

من أهم العناصر الرئيسية في لغة البرمجة و يتوجب على المبرمج إتقان استعمالها و هو عبارة عن كتلة من الأكواد قد تبدأ من تعلمية و حتى آلاف التعليمات و تستدعى في مناطق معينة من النص الرئيسي للبرمجة و في كل مرة نسند إليها متحولات مختلفة للدخل تساعد التوابع على الإقلال من حجم الكتلة البرمجية للبرنامج الرئيسي هناك توابع محفوظة و مبيتة بشكل افتراضي في لغة ال PHP لدينا على سبيل المثال تابع الجذر التربيع sqrt(number) ولدينا أيضا strlen("string") و هو يحسب طول السلسلة الحرفية و غيرها من التوابع ذات العدد الكبير جدا و تتيح لغة ال PHP إمكانية برمجة التابع الخاص بك

التصريح عن التوابع :

يتم عبر الصيغة التالية

```
Function_name() {}
```

مثال على ذلك

```
<?php

function hello() {

echo "hello ";

echo "I'm working here";

}

hello()  //calling function hello

?>
```

نطاق المتحولات الموجودة داخل التابع :

بشكل عام فإن التابع لا يستطيع أن يرى المتغيرات الموجودة خارج التابع و لا يستطيع أن يعدل في قيمة المتغيرات اذا استخدم نفس اسم المتغير

```
<?php

function hello() {

$a="inside function ";

echo $a; }
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
hello() ;
$a="outside of the function";
echo $a;
?>
```

لدينا أيضا النطاق العام و هو الذي يسمح للتابع أن يرى خارجه يتم إخذ قيمة متغير ما كما في المثال

```
<?php
$a="this variable is outside of the function";
function hello() {
echo $GLOBALS['a']; }
hello();
?>
```

لدينا عدد من التوابع المبيتة التي تقوم بتحديد عدد معاملات الدخل و استخراج قيمتها بأسلوب ديناميكي و هي

Func_num_args() يقوم بإعادة عدد المدخلات الموجودة في التابع

Func_get_arg() يقوم بالإستعانة بالرقم الذي سندخله في هذا التابع كمعامل للدخل لإعادة المدخل حسب موقعه فمثلا اذا أدخلنا 0 يعيد أول مدخل و اذا أدخلنا 1 يعيد المدخل الأول و هكذا

Func_get_args() يعيد مصفوفة بكامل المدخلات الموجودة

```
<?php
function hello() {
if(func_get_args()>0) {
$arg=func_get_arg(0);
echo "hello $arg"; } }
hello(ahmed);
?>
```

لدينا هنا برمجة تابع يحصي عدد محارف المدخلات مهما كان عددها

```
<?php
function countall() {
if(func_num_args()==0) {
die ("you need to specify at least one argument"); }
else {
$args =func_get_args();
$count=0;
```

```
foreach ($args as $arg) {

$count = $count + strlen($arg); }}

return $count; }

echo countall("php","SQL");

?>
```

أي اذا كان عدد المعاملات صفر سنقوم باستخدام الأمر die المسؤول عن إنهاء عمل السكربت و طباعة رسالة خطأ و اذا كان لدينا مدخل واحد على الأقل سنأخذ جميع المدخلات على شكل مصفوفة و foreach تأخذ كل عنصر من عناصر المصفوفة (مصفوفة معاملات الدخل) و نقوم بحساب طول كل عنصر من هذه العناصر و يتم جمعها باستخدام $count .

جرب وضع countall() بدل countall("php","SQL") و انظر ماذا سوف يعطيك البرنامج .

تمرير المعاملات عن طريق المراجع :

يمكن جعل التوابع تؤثر على المتحولات التي هي خارج التابع بالطريقة التالية كما في المثال

```
<?php

function countall(&$count) {

if(func_num_args()==0) {

die ("you need to specify at least one argument"); }

else {

$args =func_get_args();

array_shift($args);

foreach ($args as $arg) {

$count = $count + strlen($arg); }} }

$countA=0;

echo cou.ntall($countA,"php","SQL");

echo $countA;

?>
```

وضعنا إشارة & قبل $count للإشارة أنه يمكن لهذا التابع أن يوثر على المتحولات الخارجية تكمن طريقة استدعاء التابع للتأثير على متحول ما و ليكن $X و ذلك حسب كود التابع أو ال function بوضع function_name($X,argument1,argument2,...) لكن يجب حذف العنصر الأول من هذه المصفوفة عن كتابة كود التابع و ذلك بالأمر array_shift($variable_of_arguments) .

الاستدعاء الذاتي :

هو قيام الكود باستدعاء نفسه من خلال كود التابع نفسه و تهدف الى تبديل المدخلات في كل مرة و مع استخدام نفس الكود

```
<?php

function countdown($num_arg) {

if($num_arg>0) {

echo 'counting down from ',$num_arg,'<br>';

countdown($num_arg-1);} }
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
countdown(10);

?>
```

لاحظ في هذا المثال الذي يقوم بالعد التنازلي حسب الرقم المدخل حيث أنه يقوم بالبداية بطباعة الرقم المدخل ثم طبقنا التابع مرة أخرى و لكن بإنقاص قيمة 1 الى أن يتحقق الشرط و يصبح الرقم المدخل يساوي الصفر .

بناء المصفوفات :

- طريقة الاسناد المباشر لدي مثلا متحول $fruits يتم أسناد مصفوفة اليها كما يلي

```
$fruits=array('apple','banana','orange');
```

- طريقة الاسناد عن طريق ذكر دليل المصفوفة كما يلي

```
$fruits[0]='apple';

$fruits[1]='banana';

$fruits[2]='orange';
```

و هكذا نسند كل عنصر من العناصر حيث أن الدليل يبدأ من القيمة 0 و حتى عدد العناصر – 1

- الطريقة الثالثة تتم بنفس الطريقة الثانية مع حذف الدليل أي نكتفي بكتابة الأقواس المربعة دون ذكر الدليل و العنصر الذي تم إدخاله أولا سيكون في الموقع 0 و الذي تم إدخاله ثانيا سيكون في الموقع 1 و هكذا .

اذا أردنا بناء مصفوفة عددية على شكل سلسلة من الأرقام المتعاقبة يتم بالأمر range وهو عبارة عن تابع

```
<?php

$a=array(1,2,3,4,5);

$a=range(1,5,1); //range(low value,high value,step)

Print_r ($a);  //type the array

?>
```

و يمكن عدم كتابة الخطوة اذا كانت 1 لأنه يأخذ هذه القيمة بشكل افتراضي .

طريقة طباعة أنواع المعطيات المركبة كامصفوفات :

- الطريقة الأولى هي طريقة print_r
- الطريقة الثانية هي طريقة var_dump و هي تزيد عن سابقتها بتحديد نوع المعطيات

و var_dump تستطيع طباعة أكثر من مصفوفة مثلا var_dump($a,$b)

المصفوفة متعددة الأبعاد :

اذا كان أحد عناصر مصفوفة أحادية البعد هو عبارة عن مصفوفة أخرى عند ذلك تتحول المصفوفة الى مصفوفة متعددة الأبعاد

```
<?php

$fruits[0]=array('apple','banana');

$fruits[1]=array('orange','pear');

echo $fruits[0][1];
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
?>
```

قمنا في المثال السابق بطباعة العنصر ذو الدليل صفر من العنصر الرئيسي و ذو الدليل رقم 1 من العناصر المتفرعة من ذلك العنصر و ذلك العنصر هو بالتأكيد banana .

فك المصفوفة الى مجموعة من المتحولات و إسناد كل عنصر من عناصر المصفوفة الى متحول :

و يقوم بهذه العملية التابع list و تستخدم list في العادة مع خرج التوابع عندما يكون الخرج عبارة عن مصفوفة لفك المصفوفة و إسنادها الى مجموعة من المتحولات

```
<?php
function clac($n1,$n2) {
$x1=$n1+$n2;
$x2=$n1-$n2;
$x3=$n1*$n2;
$x4=$n1/$n2;
$a=array($x1,$x2,$x3,$x4);
return $a; }
list($a,$b,$c,$d)=clac(4,5) ;
echo $a, '<br>' , $b ,'<br>', $c,'<br>' , $d ;
?>
```

لا تنسى إرجاع قيمة المتحول $a لأنه ما نريده من التابع هو إيجاد قيمة $a كمصفوفة .

مقارنة المصفوفات :

هناك عمليتين رئيسيتين المساواة == و المطابقة === اذا كانت المصفوتان لهما نفس عدد العناصر فهما متساويتان حتى لو كانت العناصر مختلفة في الترتيب أما اذا كانت المصفوفتان لهما نفس العدد من العناصر و تحتوي على نفس القيم و نقس المفاتيح فهما متطابقتان

```
<?php
$a=array(1,2,3);
$b=array(0=>1,1=>2,2=>3);
$c=array(0=>1,1=>2,2=>3);
$d=array(a=>1,b=>2,C=>3);
if ($a==$b) {echo '$a==$b ';}
else {echo '$a !==$b ';}
if ($a==$d) {echo '$a==$d ';}
else {echo '$a !==$d ';}
if ($b===$c) {echo '$b===$c ';}
else {echo '$b !===$c ';}
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
?>
```

ستكون النتيجة في المتصفح هي $a==$b $a !==$d $b===$c

التعليمة أو التابع count :

تستخدم التعليمة count في لغة ال php لإحصاء عدد عناصر المصفوفة طالما أن عملية إدخال و حذف العناصر في المصفوفات هي عملية ديناميكية و ليست ستاتيكية أي أن العناصر تتغير بشكل مستمر لذلك فإننا بحاجة لمعرفة عدد عناصر المصفوفة في كل مرة كما في المثال

```
<?php

$a=array(1,2,4);

$b=array();

$c=10;

echo count($a),'<br>'; //viewing in the browser gives 3

echo count($b),'<br>'; //viewing in the browser gives 0

echo count($c); ////viewing in the browser gives 1

?>
```

التعليمة isset :

من العمليات التي يمكن تطبيقها على المصفوفات الترابطية فقط هي ال isset و دخل هذه العملية هي المصفوفة الترابطية أو المتحول المعبر عن تلك المصفوفة الترابطية مع مفتاح معين حيث سأقوم بالسؤال عن ذلك المفتاح هل هو موجود فعلا في المصفوفة الترابطية و هل يمتلك قيمة ما في تلك المصفوفة الترابطية و النتيجة ستكون نتيجة منطقية إما true أو false و القيمة true أو 1 يعني أن المصفوفة الترابطية تمتلك ذلك المفتاح و ذلك المفتاح له قيمة ما اذا كانت القيمة false فذلك يعني أن ذلك المفتاح لا وجود له في تلك المصفوفة الترابطية تفيد تلك العملية لأنني أبحث دائما في المصفوفات الترابطية عن القيمة المقابلة للمفتاح ولا أستطيع أن أقوم بعملية البحث إلا اذا كان ذلك المفتاح موجودا و إلا سأحصل على خطأ و يتم تعطيل البرنامج في لغة ال php

```
<?php

$a=array('a'=>1,'b'=>2);

echo isset($a['a']);

echo isset($a['c']);

?>
```

و النتيجة في الحالة الأولى هي 1 و في الحالة الثانية 0 وبما أنه لا وجود للصفر في لغة ال php فإنه يعني null أو لا شيء .

الانتماء in_array :

عملية الانتماء أي سنقوم بالسؤال عن قيمة ما هل تنتمي الى المصفوفة عند ذلك سنحصل على true

```
<?php

$a=array(a=>1,c=>2,d=>3);

echo in_array($a, 2); //true

echo in_array($a, 4); //false

?>
```

الحذف من المصفوفة unset:

نقوم بحذف عنصر معين من المصفوفة و ذلك بذكر إما المفتاح اذا كانت ترابطية أو الدليل اذا كانت تسلسلية

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```php
<?php
$a=array("a"=>1,"c"=>2,"d"=>3);
$b=array(3,0,7);
unset($a["a"]);
unset($b[2]);
var_dump($a);
var_dump($b);
?>
```

تعليمة مشابهة لل in_array :

و تستخدم في المصفوفات الترابطية فقط هذه التعليمة هي array_key_exists و هي تسأل عن المفاتيح فقط و النتيجة منطقية إما true أو false

```php
<?php
$tel=array('ahmed'=>548251,'nour'=>548215);
if(array_key_exists('ahmed',$tel) ) {
echo 'phone number of Ahmed is ',$tel['ahmed'];}
else {echo "Ahmed number is not exist"; }
?>
```

عند التعامل مع المصفوفات الترابطية يقع المبرمجون عادة في خطأ ما بين عملية قلب المصفوفة flipping و عكس المصفوفة reversing عملية القلب تتم بالعكس ما بين القيمة و المفتاح حيث أن المفتاح يصبح هو القيمة و القيمة تصبح هي المفتاح أما عملية عكس المصفوفة فهي تجعل آخر عنصر في المصفوفة يصبح أول عنصر و أول عنصر يصبح آخر عنصر انظر المثال .

```php
<?php
$tel=array('ahmed'=>548251,'nour'=>548215);
$a=array_flip($tel);
$b=array_reverse($tel);
var_dump($a);
var_dump($b);
?>
```

المؤشرات في المصفوفات :

المؤشرات هي ما يشير الى دليل المصفوفة يتحرك المؤشر بصورة ديناميكية ليتحرك للعنصر التالي أو السابق أو ليشير للعنصر الحالي و هناك مؤشرات تعدينا الى بداية المصفوفة و مؤشرات تعيدنا الى نهاية المصفوفة و لدينا بشكل عام خمس مؤشرات

- المؤشر reset يعيدنا دوما الى بداية المصفوفة .
- المؤشر end يأخذنا الى نهاية المصفوفة .
- المؤشر current يدل على العنصر الحالي و عن ذلك نستطيع استخدام next و previous .
- المؤشر next يدل على العنصر التالي من دليل العنصر الحالي .
- المؤشر previous يدل على العنصر السابق من دليل العنصر الحالي .

```php
<?php
$tel=array('ahmed'=>548251,'nour'=>548215,'ali'=>541200);
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
function displayarray($array) {

reset ($array);

while(key($array) !== null) {

echo key($array)." : ".current($array),'<br>';

next($array); } }

displayarray($tel);

?>
```

قمنا في المثال السابق ببرمجة تابع يقوم بطباعة مفاتيح متغير مصفوفة ما $tel و طباعة عناصرها حيث قمنا في البداية بجعل المؤشر في بداية المصفوفة عن طريق reset و وضعنا شرط طالما أن مفتاح المصفوفة لا يساوي null أي لا شيء فإننا اذا في المصفوفة و نطبع المفتاح الأول للعنصر الأول و نطبع العنصر الأول ثم ننتقل للعنصر الثاني و يتم طباعته و هكذا .

يمكن كذلك البدء من العنصر الأخير و نرجع للخلف

```
$tel=array('ahmed'=>548251,'nour'=>548215,'ali'=>541200);

end($tel);

while (key($tel)!==null) {

echo key($tel).' : '.current($tel),'<br>';

prev ($tel); }

?>
```

فرز المصفوفات :

من العمليات الشهيرة التي تطبق على المصفوفات الترابطية و التسلسلية حيث ترتب الأرقام ترتيب تصاعدي و بالنسبة للمصفوفات الترابطية فإنها تلغي قيم المفاتيح و تضع مكانها أدلة مرتبة من الصفر و حتى عدد العناصر – 1 و للتخلص من مشكلة القضاء على المفاتيح التي تقوم بها sort هناك التعليمة asort التي ترتب ترتيبا أبجديا بالنسبة لعناصر المصفوفات و ترتيبا تصاعديا اذا كانت هذه العناصر الأرقام مع الإبقاءعلى المفاتيح .

```
<?php

$tel=array('ahmed'=>548251,'nour'=>548215,'ali'=>541200);

sort($tel);

var_dump($tel);

$numbers=array(10,30,20,50,40);

sort($numbers);

var_dump($numbers);

?>
```

لفرز مصفوفة تحوي على أرقام و حروف نقوم بعملية الفرز الطبيعي عن طريق التعليمة natsort التي تبحث عن القيم الرقمية و ترتبها ترتيبا تصاعديا

```
<?php
```

```
$tel=array('20a','30b','5c');

natsort($tel);

var_dump($tel);

?>
```

للقيام بالفرز بناء على قيم المفاتيح هناك التعليمة ksort التي ترتب ترتيبا أبجديا اذا كانت المفاتيح عبارة عن أحرف و ترتيب تصاعدي اذا كانت عبارة عن أرقام

و هناك أيضا عملية الفرز العكسي للمفاتيح عن طريق التعليمة krsort

```
<?php

$tel=array('ahmed'=>548251,'nour'=>548215,'ali'=>541200);

ksort($tel);

var_dump($tel);

?>
```

البعثرة لعناصر المصفوفة :

و هي عملية معاكسة لعملية الفرز تهدف لخلق حالة عشوائية للترتيب الأساسي للمصفوفة و عند كل مرة نستخدم فيها هذا التابع يؤدي الى تجديد لعملية البعثرة و نحصل على ترتيب عشوائي جديد في كل مرة و التابع المسؤول عن عملية البعثرة هو shuffle

```
<?php

$numbers=array(1,2,3,4,5,6,7,8,9,10);

shuffle($numbers);

var_dump($numbers);

?>
```

التحويل من المصفوفة الى المكدس :

التحويل من بنية المصفوفة كبنية بيانات معقدة حيث أنها تضم مجموعة من العناصر و بدليل يميز موقع العنصر في المصفوفة الى المدسات stacks و هو بنية معطيات مركبة أيضا لكن ليس هناك دليل و يتم التخزين باعتبار أن آخر ما يدخله و هو أول ما يخرجه last in first out و بالنسبة للمكدس فإن هناك عمليتين رئيسيتين عملية إدخال العناصر push و عملية إخراج العناصر pop لاحظ المثال و انظر كيف عرفنا المكدس stack1 و أخرجنا آخر عنصر تم إدخاله عبر array_pop و هو العنصر third و عرضناه عن طريق var_dump .

```
<?php

$stack1=array();

array_push($stack1,'first','second','third');

var_dump($stack1);

echo '<br>';

$last_element_in=array_pop($stack1);

var_dump($last_element_in);

?>
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

الأرتال :

تعني أن أول ما يدخل الى الرتل هو أول ما يخرج منه أي first in fist out

لاحظ المثال التالي أن أول من دخل و هو first هو أول من خرج عندما حذفنا عنصر عن طريق array_shift و عندما أدخلنا عنصر جديد فإنه سيدخل في مقدمة الرتل مكان العنصر الذي حذف

```
<?php

$queue=array('first','second','third');

$first_element=array_shift($queue);

var_dump($queue);

echo '<br>';

array_unshift($queue,'new');

var_dump($queue);

?>
```

تابع الاختلاف في المصفوفات :

دخل هذا التابع هو مصفوفتان و عملية المقارنة التي تتم هي عملية مطابقة بين عناصر المصفوفتين و الذي يخرجه هذا التابع هي العناصر المختلفة فقط من المصفوفة الأولى أي أن المصفوفة الأولى هي التي يقارن بها القيم التي ستخرج في المثال التالي هي 15 و 25

```
<?php

$a=array(10,15,20,25);

$b=array(10,14,20,23);

var_dump(array_diff($a,$b));

?>
```

التوابع التي تتعامل مع السلاسل الحرفية :

التابع strtr

من أشهر التوابع و تعني عملية التحويل أو الترجمة في السلاسل الحرفية ما تقوم به هو عملية استبدال مقاطع من السلاسل الحرفية بسلاسل أخرى عند وجود خطأ مطبعي و نريد استبداله بسلسلة حرفية اخرى أو غير ذلك من استبدال بعض المقاطع بمقاطع أكثر فائدة أو أكثر تعبيرا .

```
<?php

echo strtr("hillo php","hi","he");

echo '<br>';

$subst=array('1'=>'one ','2'=>'two ','3'=>'three ');

echo strtr( '123',$subst);

?>
```

في المثال الأول استبدلنا أول متغيرين هما hi بالمتغيرين he .

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

و في المثال الثاني كتبنا مصفوفة ندعوها مصفوفة التحويل أو الترجمة و وضعنا لكل مفتاح العنصر الموافق له عند ذلك يستبدل كل قيمة مفتاحية موجودة في النص المكتوب و هو في حالتنا 123 بقيم عناصر المصفوفات و هي one two three .

السلاسل الحرفية كمصفوفات :

السلاسل الحرفية هي عبارة عن مصفوفات

```
<?php
$str1="abcdefg";
$str2=array('a','b','c','d','e','f','g');
echo $str1[2]." = ".$str2[2];
echo '<br>';
$s='france';
for($i=0;$i<strlen($s);$i++) {
if($s[$i] >'c') {
echo 'letter '.$s[$i].' found','<br>'; } }
?>
```

لاحظ في المثال الأول المتحول $str1 يكافئ $st2 و في المثال الثاني عرفنا متحول $s و أسندنا له قيمة france مثلا و وضعنا حلقة بعدد دورات يساوي عدد حروف المتحول و وضعنا شرط لطباعة الحروف التي هي أكبر من الحرف c حيث أن أكبر الحرف a أصغر من الحرف b و الحرف b أصغر من الحرف c و هكذا .

مقارنة السلاسل الحرفية :

تتم المقارنة بناء على شفرة ال ASCI و هي عبارة عن رقم يعبر عن الحرف أو الرقم أو الإشارة مثلا حرف a يقابله رقم 65 تتم المقارنة عن طريق التابع strcmp الذي يعطي صفر اذا تمت المطابقة بنجاح و ذلك بمطابقة كل حرف من السلسلة الأولى مع حرف من السلسلة المقابلة بمعنى آخر اذا قام هذا التابع بعملية الطرح للقيمتين المعبرتين عن شفرة ال ASCI اذا كان هناك أي اختلاف سيعطي التابع القيمة 1 .

```
<?php
$str="HELLO PHP";
if(strcmp($str,"hello php") ===0) {
echo 'comparing ok'; }
else { echo 'comparing not ok'; }
?>
```

و ستكون نتيجة المثال السابق comparing not ok لأن الحروف الصغيرة تأخذ قيما مختلفة عن الحروف الكبيرة .

عملية البحث في السلاسل الحرفية :

تتم عن طريق التابع strpos و هو يقارن بين سلسلتين حرفيتين الأولى كبيرة و الثانية صغيرة جزئية من الكبيرة حتى تكون النتيجة هي إخراجنا للموقع الموجود فيه هذه السلسلة الجزئية الصغيرة في السلسلة الكبيرة و التابع الثاني هو strstr و هو عملية استخراج سلسلة جزئية من السلسلة الكلية يقوم بإخراج ما تبقى من السلسلة الكلية ابتداء من موقع إيجاد السلسلة الجزئية .

```
<?php
$str1="I'm searching in PHP";
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
$str2='in';

if(strpos($str1,$str2) !==false) {

echo 'found ',strpos($str1,$str2),'<br>',strstr($str1,$str2);}

else { echo "sorry not found"; }

?>
```

قمنا في المثال السابق بوضع سلسلتين حرفيتين كلية كبيرة $str1 و جزئية صغيرة $str2 و وضعنا شرط اذا كانت نتيجة بحث التابع strpos في السلسلة الكلية لا تساوي 0 أي هناك نتيجة للبحث فاطبع لنا موقع تلك السلسة الجزئية في السلسلة الكلية و موقع in هو 12 في السلسلة الكلية في الكلمة searching و اطبع أيضا ما تبقى من السلسة الكلية عن طريق التابع strstr و إلا فاطبع لنا أنه لم يتم العثور على نتيجة للبحث .

عملية البحث عن طريق التابع strspn :

```
<?php

$string='521 is a good number';

$mask='0123456789';

echo strspn($string,$mask);

?>
```

في المثال السابق يدخل التابع strspn متحولين الأول هو السلسة الحرفية و الثاني يتكون من أرقام فقط و تتم عملية البحث و تكون النتيجة هي عدد الأرقام الموجودة في السلسة $str والنتيجة هي 3 .

عملية استبدال أجزاء حرفية بأخرى :

يتم عن طريق التابع str_replace و str_ireplace و الفرق بينهما أن التابع الأول يهتم بحالة الحروف كبيرة صغيرة أما الثاني فلا يهتم

```
<?php

echo str_replace("php","world","hello php"),'<br>';

echo str_ireplace("PHP","world","hello php");

?>
```

ندخل في التابع العبارة المراد البحث عنها و هي هنا php و العبارة المراد الاستبدال بها و هي هنا wold و نص النص الكلي و هو في حالتنا hello php .

عملية استبدال سلسة جزئية بسلسلة أخرى أو فراغ :

```
<?php

$user="hani@way2paradise.com";

$name=substr_replace($user," ",strpos($user,'@'));

echo 'hello '.$name ;

?>
```

التابع substr_replace هو من قام بعملية الاستبدال و كما نرى أدخلنا في التابع أولا العبارة التي سوف يتم استبدال جزء من النص فيها و هي في مثالنا $user و أدخلنا العبارة التي نريد الاستبدال بها و هي فراغ و أخيرا أدخلنا موقع الجزء الموجود فيه الاشارة @ و التي سيتم حذفها و حذف ما بعدها من السلسلة النصية و يمكن إدخال كم عدد المحارف التي نريد الحذف بعد هذه إشارة @

```
$name=substr_replace($user," ",strpos($user,'@'),4);
```

سيتم حذف أربع خانات اعتبارا من الإشارة @ في الحالة السابقة .

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

تنسيق الأرقام في السلاسل الحرفية :

```
$number=1234.56;

$a=number_format($number);

echo $a,'<br>';
```

تكون النتيجة 1,235

نظام فرنسي :

```
$b=number_format($number,2,',',' ');

echo $b,'<br>';
```

الرقم 2 أي نأخذ فاصلتين بعد الرقم و و نأخذ شكل , كفاصلة عشرية أو يمكن اختيار نقطة . و أخيرا يمكن وضع فراغ أو فاصل يفصل بين الألوف و الملايين و المليارات أو يمكن عدم وضعها .

نظام انجليزي :

```
$c=number_format($number,3,'.','');

echo $c;
```

تنسيق العملات :

نستخدم رقما و نضع الإعدادت الإقليمية و الدولية

النماذج أو الأنماط في لغة ال PHP :

هي أسلوب للتعبير عن السلاسل الحرفية أو هي أسلوب لمطابقة السلاسل الحرفية للحصول على سلاسل حرفية مناسبة للإدخال فمثلا اذا طلبنا من المستخدم إدخال بريده الاكتروني و لا نريد أن يحتوي على أرقام أو إشارات معينة هناك مجموعة من الرموز المستخدمة للتعبير عن الأنماط

النقطة . لوصل المحارف

^ تمثل بداية السلسلة الحرفية

$ تعبر عن النهاية

\s تعبر عن مساحة فارغة (الفراغ الموجود بين الكلمات)

\d تعني المطابقة مع أي رقم مهما كان عدد الأرقام و مهما كانت حالة الرقم

\w تعني المطابقة مع أي كلمة (أي مجموعة من الأحرف و الأرقام متقاربة من بعضها من دون أية فراغات)

* تعني أن هذا الحرف يمكن أن يظهر عدة مرات أو لا يظهر أبدا

+ يعني أنه قد يظهر مرة واحدة أو أكثر و من الواجب أن يظهر هذا الحرف أي لا نقبل عدم وجود هذا الحرف

؟ قد يظهر هذا الحرف مرة واحدة فقط أو لا يظهر

{1,5} أي قد يظهر مرة مرة واحدة على الأقل أو خمس مرات على أكثر أي هو تحديد للأكبر و الأصغر

```
<?php

$name="hani lakmoush";

$regex="/[a-zA-Z\s]/";

if (preg_match($regex,$name)) {

echo $name,' is a valid name<br>'; }
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
else {echo $name,' is a invalid name<br>'; }

$regex='/^(\w+)\s(\w+)/';

$matches=array();

if (preg_match($regex,$name,$matches)) {

var_dump($matches); }

?>
```

في المثال السابق لدينا التابع preg_match الذي يطابق ما بين السلسلة الحرفية التي يمثلها $name و القاعدة $regex و في المثال وضعنا أن المسموح لدينا هو حروف كبيرة و صغيرة و فراغات و التابع يعطي قيمة true اذا تحققت المطابقة و نضع عبارة أنه ذلك الاسم صالح للاستعمال .

و في المثال الثاني كانت القاعدة هي '/^(\w+)\s(\w+)/' حيث تمثل ^ بداية السلسة الحرفية و إشارة + تعبر عن وجود حرف واحد على الأقل و w تعني أي كلمة و وضعنا بين الكلمتين فراغ و أضفنا أيضا في هذا المثال الثاني مصفوفة فارغة لتخزين النتائج ضمن هذه المصفوفة و قمنا بطباعة هذه المصفوفة عن طريق var_dump .

```
<?php

$string="aa1c3 bb1 d2";

$regex="#([abc])\d#";

$matches=array();

if (preg_match_all($regex,$string,$matches)) {

var_dump($matches); }

?>
```

لدينا هنا القاعدة ;"#([abc])\d#" و هي تختار السلسلة النصية بناء على قاعدتين الأولى هي وجود أحد الأحرف a أو b أو c و الثانية هي وجود أحد الأحرف a أو b أو c مع رقم لأن \d تعني وجوب وجود رقم و الإشارة # تعني بغض النظر عن بداية و نهاية السلسة الحرفية و تكون النتيجة هي عبارة عن مصفوفتين الأولى تحوي على الأحرف a و c و b و الثانية تحوي على a1 و c3 و b1 .

التعامل مع صفحات الويب المبرمجة من طرف السيرفر(أي السيرفر هو المسؤول عن تفسير كود ال php) :

هناك طريقتان لنقل المعطيات get و post

الطريقة get و هي تقوم بنقل المعطيات نستخدمها عند الضغط على أي لنك أو عند إرسال بيانات عن طريق صناديق البحث و هي أسرع من الطريقة post .

الطريقة post تنقل المعطيات بطريقة سرية مشفرة تستخدم عادة مع مستطيلات إدخال كلمات السر و هي أبطأ من الطريقة post كونها تسلك طريقين في نقل البيانات .

لدينا المثال التالي عن الطريقة get ، نفتح صفحة جديدة نسميها مثلا example.html يمكن حفظها بامتداد php. و نكتب الكود

```
<html>

<body>

which animal do you like ?

<a href="new.php?a=dogs">dogs</a>

<a href="new.php?a=cats">cats</a>
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

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

حيث أننا عند الضغط على الرابط dogs مثلا فإننا نسند للمتحول a قيمة dogs و إشارة الإستفهام تدل على استخدامنا للطريقة get .

نفتح صفحة جديدة نسمها new.php و نكتب الكود

```
<html>
<body>
you said you like <?= $HTTP_GET_VARS['a'] ?>
</body></html>
```

المصفوفة $HTTP_GET_VARS هي المسؤولة عن استقبال البينات بالطريقة get حيث أننا أخذنا قيمة المتحول a من الصفحة السابقة

مثال على الطريقة post

```
<html>
<body>
<form action="new.php" method="post">
username: <input type="text" name="username"><br>
password: <input type="password" name="password"><br>
<input type="submit" value="login">
</form>
</body></html>
```

و هي صفحة HTML عادية و يمكن تخزينها بصيغة PHP أسندنا قيم المتحول الذي سميناه username الى القيمة التي سوف يدخلها المستخدم و أسندنا القيمة password الى القيمة التي سوف يدخلها المستخدم أيضا و ما إن نضغط على الزر loging الذي وضعناه على أنه submit سيقوم النموذج بعمل action التي سوف تنقلنا الى الصفحة new.php و التي فيها الكود

```
<html>
<body>
the username entered is :<br />
 <?= $HTTP_POST_VARS['username'] ?><br />
and the password is :<br />
 <?= $HTTP_POST_VARS['password'] ?>
</body></html>
```

كود ال php <?= $HTTP_POST_VARS['variable'] ?> هو المسؤول عن نقل البيانات بالطريقة post حيث وضعنا قيمة المتحول من الصفحة السابقة .

طريقة الاختيار و الإرسال :

كود صفحة ال HTML

```
<html>
<body>
what's your favourite major :
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
<form action="new.php" method="get">
<input type="radio" name="major" value="english">english<br>
<input type="radio" name="major" value="math">math<br>
<input type="radio" name="major" value="engineering">engineering<br>
<input type="submit" value="submit">
</form>
</body></html>
```

و المتحول هنا هو major أسندنا اليه القيم value حسب اختيار المستخدم و في الصفحة php التي كودها

```
<html>
<body>
you favourite major is :<br>
 <?= $HTTP_GET_VARS['major'] ?>
</body></html>
```

أسلوب رفع الملفات من جهاز المستخدم الى مخدم الويب :

هي عملية دقيقة جدا و يجب على المبرمج تحديد البارامترات الأساسية و من أشهرها تحدبد مساحة كل ملف بالإضافة الى تحديد نوع الملف الذي يمكن للمستخدم رفعه الفورم الخاص برفع الملفات يأخذ الكود التالي :

```
<html>
<body>
<form enctype="multipart/form-data" action="new.php" method="post">
<input type="hidden" name="max_file_size" value="100000">
choose a file to upload <input name="uploaded_file" type="file"><br>
<input type="submit" value="upload file">
</form>
</body></html>
```

نختار نوع التشفيرعادة enctype عند رفع الملفات صيغة multipart/form-data و الطريقة تكون غالبا هي بوست و يتم إضافة عنصر مخفي لا يظهر في النموذج و له وظيفة تحديد الحجم الأعظمي للملف .

أما كود ال php الذي سوف نخزنه في صفحة أخرى نسميها new.php

```
<?php
$target_path="uploads/";
$target_path = $target_path . basename($_FILES['uploaded_file']['name']);
if(move_uploaded_file($_FILES['uploaded_file']['tmp_name'],$target_path)) {
echo "the file ". basename($_FILES['uploaded_file']['name'])." has been uploaded"; }
else { echo "Ooops there is something went wrong"; }
?>
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

نضع متحول سميناه Starget_path و نسند له قيمة نصية هي اسم مجلد نضعة في المجلد www الذي نضع فيه ملفات ال php المراد تجربتها نعدل قيمته لنضيف اليه القيمة (basename($_FILES['uploaded_file']['name'] (و هي قيمة اسم الملف الذي أدخلناه في النموذج و امتداده)عن طريق النقطة التي تدمج بين القيم النصية ليصبح المسار اذا كان اسم الصورة test.jpg هو upload/test.jpg وظيفة التابع basename هي تشكيل اسم الملف و وظيفة التابع move_uploaded_file هو نقل الملفات حيث نذكر الملف و المسار المراد نقل الملف اليه و وضعنا شرط اذا تم نقل الملف ذو الاسم المؤقت الذي يتم وضع الملف فيه بشكل مؤقت الى المسار $target_path فأعطينا رسالة بأنه تم نقل الملف بنجاح و إلا فاطبع رسالة أنه هناك خطأ ما .

إعادة التوجيه :

هي عملية التحويل من صفحة الى صفحة عن طريق إدخال باراميترات معينة و يتم ذلك عن طريق التابع header .

يمكن عمل فورم و أخذ قيمة متغير و ليكن name و ليكن ذلك الفورم هو

```
<html>

<head><title></title></head>

<body>

<form action="test.php" method="get">

pleas enter you name :<br />

<input type="text" name="name" /><br />

<input type="submit" value="submit" />

</form>

</body>

</html>
```

نقوم بفتح صفحة نسمها طبعا test.php و نكتب فيها الكود فإذا كان الاسم المدخل هو ahmad سيتم توجيه ذلك المستخدم تلقيائيا الى صفحته الشخصية و هي http://www.site.com/ahmed.php

```
hello <?= $HTTP_GET_VARS['name'] ?> you will redirected

<?php

if($HTTP_GET_VARS['name']=="ahmed") {

header('location:http://www.site.com/ahmed.php') ; }

?>
```

الكاش أو التخزين المؤقت :

تقوم المتصفحات عادة بعملية الكاش للصفحات التي تمت زيارتها بهدف تسريع تصفحها في حال زيارتها مرة ثانية لكن هذه العملية قد لا تكون مستحبة من قبل بعض المواقع بالنسبة للمبرمجين و خاصة في حال كون الموقع ذو تحديث دائم و مستمر نقوم كذلك أيضا عن طريق header لأجل منع الكاش بشكل دائم يكون الكود

```
header("cashe-control: no-cache ,must-revalidate"); //no caching at all
```

أما منع الكاش لتاريخ معين

```
header("expires: wed, 25 May 2015 00:00:00 GMT");
```

أما لإعادة التحديث بمدة زمينة معينة اعتبار من الزمن الحالي فإن الكود

```
$date=gmdate("D, j M Y H:i:s",time()+2592000);

header("expires: ".$date."UTC");
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
header("cache-control:puplic");

header("pragma:public")
```

حيث يمثل الرقم 2592000 هو عدد الثواني لشهر و يمكن تغيير هذا الرقم حسب ما يريد المبرمج .

الكوكيز أو الكعكات :

تلجأ بعض المواقع الى تسجيل بيانات المستخدم في جهاز المستخدم و هي ملفات صغيرة لا تؤذي أبدا و مهمتها تسهيل العمل للمستخدم لعدم إدخال بياناته في كل مرة يدخل فيها من جهازه و يجب تحديد فترة صلاحية للكوكيز و يتم ذلك عن طريق التابع setcookie و عندما تنتهي صلاحية هذه الكعكات التي يحدد زمنها المبرمج فإنها تحذف تلقائيا .

```
<?php

setcookie('membername','khaled',time()+(60*60*24),"/","www.site.com",1);

$membername=$_COOKIE['membername'];

echo "hello ".$membername;

?>
```

تقوم أولا بكتابة اسم للكعكعة و هي في مثالنا membername و نكتب اسم المستخدم الذي سيخزن في الكوكيز و ليكن khaled ثم نضع الزمن و ليكن يوم واحد بالثواني ابتداءا من الزمن الحالي و نضع المسار و هناك مسار افتراضي في الويندوز للكعكات نكتفي بوضع إشارة / أو يمكن وضع مجلد داخل هذا المجلد للتخزين فيه و نحدد أهمية البيانات اذا كانت بيانات مهمة مثل باسورد نضع 1 أي أن طريقة التخزين هي post و هي آمنة secure .

نقوم بعد ذلك باستدعاء ذلك الكوكي و نطبع أهلا أهلا بك khaled مثلا .

جلسة العمل أو ال session :

البروتوكول HTTP عديم الحالة أي أن العميل اذا أرسل طلب سيقوم السيرفر بإرسال استجابة واحدة أما اذا أرسل مئة الطلب فإنه على السيرفر إعادة مئة استجابة و هذا الأمر غير مستحب لأنه قد يؤدي الى انشغال السيرفر بعميل وحيد قد يقوم بإرسال مئات الطلبات بنفس اللحظة حيث قد يكون هاكر أو مخرب يعمد الى انشغال السيرفر لذلك ننشأ جلسة العمل حتى لا يحق للعميل سوى جلسة واحدة و بانقضاء هذه الجلسة يستطيع العميل افتتاح جلسة أخرى و بشكل عام فإن بيانات الجلسة يتم عن طريق بيانات الكوكيز .

```
<?php

session_start();

?>

<html>

<head><title>greetings</title></head>

<body>

<h2> welcome to our site </h2>

<?php

if (!IsSet($_SESSION['visit_count'])) {

echo "you have just arrived welcome <br>";

$_SESSION['visit_count']=1; }

else { $visit_count=$_SESSION['visit_count']+1;

echo "you visited us ",$visit_count,"times until now <br>";

$_SESSION['visit_count']=$visit_count; }

$self_url=$_SERVER['php_self'];
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
$session_id=SID;
if(isset($session_id && $session_id) {$href="$self_url?session_id";}
else {$href=$self_url ;}
echo '<br> <a href=\"$href\">visit us again</a> dont forget ';
?>
```

مثال آخر :

```
<?
session_start();
session_register("zx");
session_register("co");
$zx=10;
$co++;
echo 'hello visitor'."<br>";
echo "your visit number". $co ;
?>
```

العدادات أو counter :

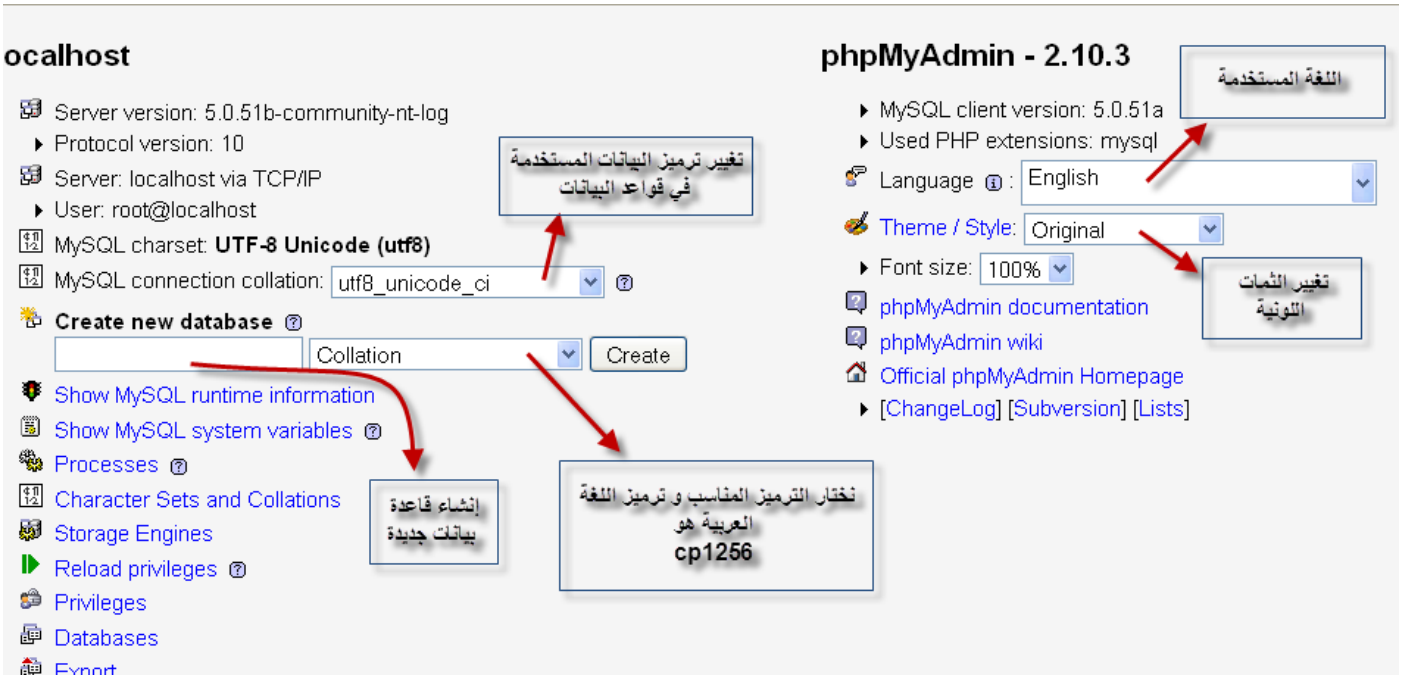
في المثال التالي استعملنا التابع fopen لإنشاء مستند نصي نسمية counter.txt مثلا و +a تعني أخذ المؤشر الى بداية النص و السماح بالحذف و التعديل في هذا الملف ، ثم سألنا هل الملف counter.txt موجود اذا لم يكن موجود أنهي عمل السكربت و أطبع عبارة ما و إلا تأكد هذا الملف النصي حجمه صفر أي ليس به أي حرف عند ذلك صفر قيمة العداد و احصل على الملف عن طريق التابع fgets و اطبع فيه القيمة صفر عن طريق التابع ftruncate ثم أضف قيمة 1 و اكتب في هذا الملف النصي قيمة $counter عن طريق التابع fwrite .

```
<?php
$file=fopen("counter.txt",'a+');
if ($file == false ) {
die ("unable to open/create file") ;}
if (filesize("counter.txt") ==0) {
$counter=0; }
else {
$counter = (int) fgets($file) ;}
ftruncate($file,0);
$counter++ ;
fwrite($file,$counter);
echo "there has been $counter to this site";
?>
```

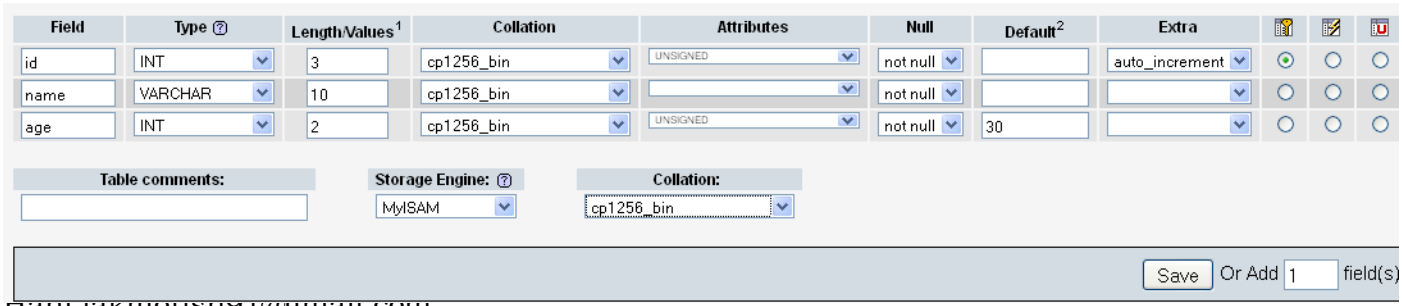
By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

قواعد البيانات mysql :

سنتعلم كيفية التعمل مع قواعد البيانات من نوع mysql و كيفية دمج قواعد البيانات مع صفحات الويب و التفاعل معها بقوة و فعالية مشروع ال appserv الذي قمنا بتنصيبه يحتوي على مخدم ال apatche الذي يقوم بترجمة و تفسير كود ال PHP و تحويله لكود HTML بالإضافة الى أنه يحتوي قواعد البيانات phpmyadmin و التي هي عبارة عن دمج ما بين mysql server أو مخدم قواعد البيانات من نوع mysql بالإضافة الى التعامل المباشر مع صفحات ال PHP كواجهات الويب ندخل الى php database manager عن طريق الدخول الى الرابط http://localhost حيث ننتقل الى صفحة البداية

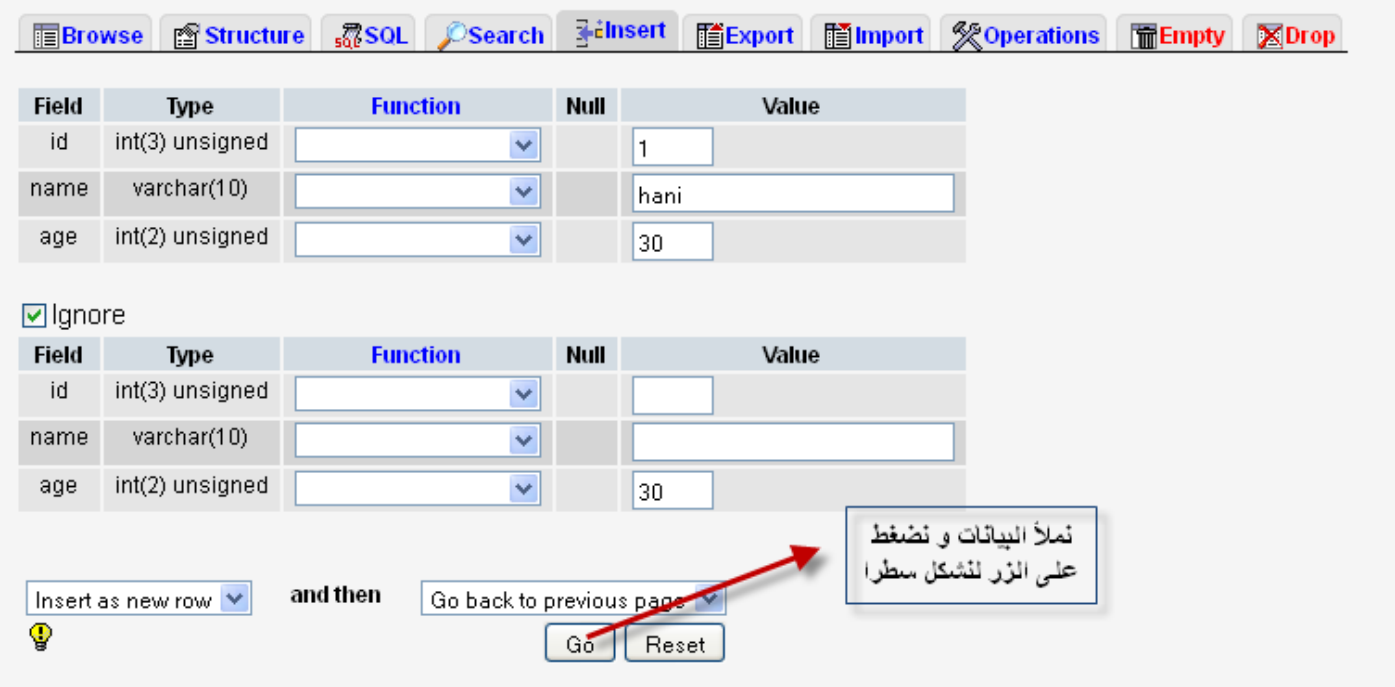


ننشأ قاعدة بيانات جديدة و ليكن اسمها mydb بالترميز العربي إن قواعد البيانات mysql هي قواعد بيانات علائقية أي أنها تحتوي على الجداول و هي العناصر الأساسية في تخزين البيانات و لا بد أن تحتوي قواعد البينات على جدول واحد على الأقل لأنه العنصر الرئيسي فيها نسمي جدول جديد mytable في الحقل create new table in database و نحدد عدد من الحقول المستخدمة فيه و الحقول هي الأعمدة في الجداول نحدد عدد من الحقول و ليكن ثلاثة حقول و نضغط على go ، نسمي الحقول بأسماء مناسبة و يجب أن لا نبتدئ برقم و يجب أن لا تحتوي على فراغات و اذا احتجت الى فواصل يمكن استخدام ال _ نمسي الحقل الأول مثلا id و الحقل الثاني name و الحقل الثالث age نحدد بعد ذلك نوع البيانات المستخدمة في قواعد البيانات لدينا VARCHAR مستخدمة في الأحرف و تحجز مساحة في الحقل متناسبة مع طول البيانات الموجودة في الحقل نستخدمها مع الأسماء و العناوين و لكافة السلاسل الحرفية التي نريد تخزينها و لدينا CHAR هي أيضا مستخدمة في الأحرف لكنها تحجز مساحة ستاتيكية و هذا ما يؤدي الى استهلاك البيانات في قاعدة البيانات و أخذ مساحة كبيرة و يوجد أيضا INT و تعني الرقم الصحيح يمكن استخدامها مع المعرف id و ال age هناك أيضا طول للبيانات نحدد مثلا أن طول ال id لا تتجاوز ثلاث خانات و ال name لا يتجاوز عشر خانات و ال age لا تتجاوز الخانتين ثم نحدد أسلوب الترميز نستعمل الترميز العربي ثم إن لدينا ال attributes و تحديدها قد يكون مهما في بعض الحقول و قد يكون غير مهم في حقول أخرى و ذلك بحسب استخدامنا للحقول اذا كان الخيار unsigned هذا يعني أن الرقم دائما موجب نستخدمه مع ال id و age و اختيارنا ل null يعني أنه يمكن أن يحتوي الحقل على قيم فارغة و not null تعني وجوب قيمة واحدة على الأقل أيضا و يمكن وضع قيمة افتراضية ابتدائية تتكرر كثيرا لدينا أيضا خصائص إضافية مثل auto_increment و تعني التزايد التلقائي نستخدمها مع ال id حيث أنها مفيدة لها ، و لدينا أيضا ال primary key وهو المفتاح الرئيسي و خصائصة أنه not null و unique أي أنه وحيد و لدينا أيضا ال index أو الفهرس الذي يستخدم لتسريع البحث و لدينا أيضا unique و اذا وضعت المفتاح الرئيسي فإنه لا داعي لل unique أخيرا نتجه الى تحديد نوع الجدول و أشهرها هو MyISAM و اختيار الترميز و نضغط حفظ ليعرض لنا sql querry و هو استعلام ال sql الذي يقوم به phpmyadmin دون تعلمنا للغة mysql و نستطيع بعد أن ننتهي بتعبئة الجدول بالبيانات المناسبة و لتعبئة البيانات نحدد الحقول و نضغط على Insert .



Hani.lakhoush91@gmail.com

عند كل عملية تعبئة نشكل record و هو يمثل سطر في الجدول .



الاتصال بقاعدة البيانات :

الهدف من قاعدة البيانات هي إضافة المحتوى الديناميكي لصفحات الويب و بذلك نستطيع تخزين و عرض كمية كبيرة من البيانات دون الحاجة لإنشاء عدد كبير من الصفحات سنتصل بقاعدة البيانات التي قمنا بإنشائها في الفقرة السابقة عن طريق الكود .

```
<?php

define('MYSQL_HOST','localhost');

define('MYSQL_USER','root');

define('MTSQL_PASS','hanilak100');

define('MYSQL_DB','mydb');

if(! mysql_connect(MYSQL_HOST,MYSQL_USER,MTSQL_PASS) ) {

die ("failed to connect to the host ".MYSQL_HOST ); }

else {

echo 'connected to MYSQL server '.MYSQL_HOST.' as user '.MYSQL_USER. '<br>' ; }

mysql_select_db(MYSQL_DB);

?>
```

نحدد مجموعة من الثوابت عن طريق define كما تعلمنا في فقرة الثوابت و التابع المسؤول عن الاتصال هو التابع mysql_connect و هو يحتاج الى بارامترات اسم الهوست و يوزر قاعدة البيانات و باسورد قاعدة البيانات ذا لم يتم الاتصال ننهي عمل السكربت و نطبع عبارة ما و إلا فنكتب أنه تم الاتصال بنجاح و نختار قاعدة البينات التي قمنا بإنشائها في الفقرة السابقة عن طريق التابع mysql_select_db .

طريقة الإدخال الى قاعدة البيانات :

سنقوم الآن بتنفيذ الاستعلام insert الذي يقوم بإضافة سجل الى الجدول حيث أنه سيضيف بيانات الى حقول ذلك الجدول و صيغة هذا الاستعلام هو

INSERT INTO MYTABLE في قاعدة البيانات mydb التي قمنا بإنشائها ثم نضع set و نكتب كل حقل و نسند له قيمة ما يدخلها المستخدم قمنا في الفورم بكتابة كود مخفي لا يظهر للمستخدم يعطي متغير $action له قيمة null عند عدم الضغط على الزر submit و القيمة signup عند الضغط على ذلك الزر فإذا لم يتم الضغط نطلب من الكود أن تعرض لي التابع displayform الذي برمجناه و يحتوي على صناديق المدخلات و الذي يستدعي

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

نفس الصفحة $PHP_SELF و نكتب كود الفورم الذي هو عبارة عن كود HTML خارج وسوم ال PHP اقرأ الكود بتمعن و كرر قراءته إن لزم الأمر حتى تفهمه بشكل جيد .

```
<?php
//constants declarations
define('MYSQL_HOST','localhost');
define('MYSQL_USER','root');
define('MTSQL_PASS','hanilak100');
define('MYSQL_DB','mydb');
//main programme here
if(! isset($action)) {
$action=null; }
switch($action) {
default: {displayform(); break;}
case 'signup': {
signUp($HTTP_POST_VARS);
displaysuccess();
break; } }
function displayform() {
head();
?>
<form action="<?php echo $PHP_SELF ?>" method="post" >
<input type="hidden" name="action" value="signup">
ID: <input type="text" name="id_"><br>
name:<input type="text" name="name_"><br>
age:<input type="text" name="age_"><br>
<input type="submit">
</form>
<?php
foot();
}
//end of displayform
function signUp($input) {
if(! mysql_connect(MYSQL_HOST,MYSQL_USER,MTSQL_PASS) ) {
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
die ("failed to connect to the host ".MYSQL_HOST ); }
else {
echo 'connected to MYSQL server '.MYSQL_HOST.' as user '.MYSQL_USER. '<br>' ; }
mysql_select_db(MYSQL_DB);
mysql_query("INSERT INTO mytable SET id='{$input['id_']}',name='{$input['name_']}', age='{$input['age_']}' "); }
//end of signUp function
function displaysuccess() {
head() ;
?>
your submission has been completed
<?php
foot();
}
// end of display success
function head() {
echo '<html><body>' ; }
function foot() {
echo '</body></html>' ; }
?>
```

سحب البيانات من قاعدة البيانات :

الاستعلام المسؤول عن اختيار بيانات من الحقول الموجودة في جدول ما ضمن قاعدة بيانات ما هو الاستعلام select و اذا كتبنا select * mytable فهذا يعني اختر كل شيء موجودة ضمن هذا الجدول أي أن * تعبر عن كامل الحقول الموجودة فيها أو وضع أسماء الحقول بدل من * تفصل بينهما فواصل (,) .

```
<?php
define('MYSQL_HOST','localhost');
define('MYSQL_USER','root');
define('MTSQL_PASS','hanilak100');
define('MYSQL_DB','mydb');
if(! mysql_connect(MYSQL_HOST,MYSQL_USER,MTSQL_PASS) ) {
die ("failed to connect to the host ".MYSQL_HOST ); }
else {
echo 'connected to MYSQL server '.MYSQL_HOST.' as user '.MYSQL_USER. '<br>' ;
}
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
mysql_select_db(MYSQL_DB);

$result=mysql_query('SELECT * FROM mytable');

echo "<br>";

while ($row=mysql_fetch_array($result)) {

echo "{$row['id']} "."{$row['name']} "."{$row['age']} ",'<br>'; }

?>
```

عملية التحقق :

تستخدم عادة عند عملية الدخول الى صفحة المستخدم للتحقق من هوية المستخدم و تتضمن التحقق من تطابق اسم المستخدم مع كلمة سره لهذا السبب نمنع المستخدمين من تكرار اسم المستخدم و نسمح لهم بتكرار الباسورد حيث أننا لا نعلم ما هي كلمة السر التي نقوم بإدخالها نقوم بإنشاء جدول جديد كما تعلمنا سابقا في قاعدة البيانات mydb و ليكن اسم هذا الجدول users و فيه حقلين واحد لليوزر و الثاني للباس و نقوم بإدخال مستخدمين مثلا لكل مستخدم كلمة سر معينة و عند الطباعة الخاطئة لاسم المستخدم أو كلمة السر فإننا سنقوم بطباعة عبارة خطأ و الانتقال تلقائيا الى صفحة التسجيل أما اذا كانت الكلمة صحيحة فإننا سنقوم بطباعة عبارة ما أو الانتقال الى صفحة ما أو ما نرغب بالقيام به نقوم بعمل ملفين php الأول يحوي الفورم له الكود التالي :

```
<html><body>

<form action="dblogin.php" method="post">

username : <input type="text" name="user" />

password :<input type="password" name="pass" />

<input type="submit" value="log in" />

</form>

</body></html>
```

و كود ال PHP :

```
<?php

define('MYSQL_HOST','localhost');

define('MYSQL_USER','root');

define('MYSQL_PASS','hanilak100');

define('MYSQL_DB','mydb');

mysql_connect(MYSQL_HOST,MYSQL_USER,MYSQL_PASS);

mysql_select_db(MYSQL_DB);

echo 'connected to MYSQL server '.MYSQL_HOST.' as user '.MYSQL_USER. '<br>' ;

$result=mysql_query("SELECT COUNT(*) AS numfound FROM users WHERE user='{$HTTP_POST_VARS['user']}' AND pass='{$HTTP_POST_VARS['pass']}'");

$result_ar=mysql_fetch_array($result);

if( $result_ar['numfound']<1) {

echo("<script>location.href = 'dbform.php?error=1';</script>");

}
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
else {echo 'logged in successfully'; }

?>
```

للدواعي الأمنية يمكن منع المستخدم من تكرار عملية تسجيل الدخول الى حسابه و يتم عن طريق تسجيل متحولات جلسة العمل و كود ال PHP يصبح

```
<?php

session_start();

define('MYSQL_HOST','localhost');

define('MYSQL_USER','root');

define('MYSQL_PASS','hanilak100');

define('MYSQL_DB','mydb');

if(empty($HTTP_SESSION_VARS['user'])) {

mysql_connect(MYSQL_HOST,MYSQL_USER,MYSQL_PASS);

mysql_select_db(MYSQL_DB);

$result=mysql_query("SELECT COUNT(*) AS numfound FROM users WHERE user='{$HTTP_POST_VARS['user']}' AND
pass='{$HTTP_POST_VARS['pass']}'");

$result_ar=mysql_fetch_array($result);

if( $result_ar['numfound']<1)

{header ('location: dbform.php?error=1');exit; }

$user=$HTTP_POST_VARS['user'];

session_register('user');

echo 'logged in successfully';

}

else { echo "sorry ", $user," you are logged in" ; }

?>
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

البرمجة غرضية التوجة :

الغاية من هذه البرمجة هي تقليل حجم الكود حيث تعتمد على كتل من الكود تسمى classes و ال class أو الصف يحتوي على مجموعة من المتحولات تسمى بالخصائص و مجموعة من التوابع تسمى طرائق لكن اذا أردنا أن نعدل كتلة معينة من الكود نعدل الكلاس فقط ليتم التعديل في مواقع مختلفة و هذا يوفر الوقت و الجهد و من الخصائص الرئيسية في البرمجة غرضية التوجه هي الوراثة و الوراثة هي انتقال الخصائص و التوابع حيث أننا نقوم عادة بإنشاء كلاس و نضعه ضمن متحول يسمى المثيل الذي يشمل كافة الخصائص و التوابع الموجودة في هذا الكلاس بالإضافة الى يوجد العديد من الكلاسات الجاهزة مثل كلاس قواعد البيانات أو كلاس رفع الملفات و التي يمكن للمبرمج أن يستغلها في برمجياته .

طريقة تعريف الكلاس

```
//class definition

class myclass { // class contents go here

}
```

إنشاء مثيل لهذا الكلاس instance

```
// class instance creation

$myclassintance=new myclass();
```

إنشاء نسخة عن المثيل السابق

```
//instance copy creation

$copyinstance=$myclassinstance();
```

الوراثة لنفهم الوراثة نأخذ هذا التطبيق الصغير التالي ، نعرف الكلاس a و نعرف فيه التوابع test() و func()

نكتب كلاس b الذي يرث من الكلاس a إنه يرث كامل التابع func() لكنه يعدل قيمة التابع test() ليطبع قيمة أخرى

نكتب كلاس c الذي يرث من الكلاس b و يرث التابع test()

نكتب كلاس d الذي يرث من الكلاس c لكن التابع test() يرث قيمته من الكلاس b

ننشأ مثيل instance لكل كلاس

نقوم باستدعاء التوابع عن طر يق ذكر المتغير و وضع إشارة -> ثم ذكر التابع الذي نريد استدعاءه

```
<?php

class a {

function test()

{echo "a::test called ";}

function func() {

echo "a::func called ";} }

class b extends a {

function test() {

echo "b::test called "; } }

class c extends b {

function test() {

parent::test();} }

class d extends c {
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
function test() {
b::test();} }
$a= new a();
$b=new b();
$c=new c();
$d=new d();
$a->test();
$a->func();
echo '<br>';
$b->test();
$b->func();
echo '<br>';
$c->test();
$c->func();
echo '<br>';
$d->test();
$d->func();
?>
```

مثال آخر

```
<?php
class simplemath {
public $num=18;
function math(){
$num2= 2;
$sum = $this->num + $num2;
echo " the sum of $this->num and $num2 is $sum ";
}
}
$usepersonal= new simplemath ;
$usepersonal->math();

class newclass extends simplemath{
function math(){
```

```
echo '<br>';
echo "this is extend";
}
}
$useextend= new newclass();
$useextend->math();
?>
```

المشيدات أو constructors :

هو أول تابع (function) يتم كتابته في الكلاس يتم استدعاءها تلقائيا و تنفيذها بمجرد استخدام الكلاس .

```
<?php
class useconstructor{
function __construct(){
echo "this is the constructor function";
} }
$useconstructor= new useconstructor();
?>
```

ستكون نتيجة المثال السابق مباشرة هي this is the constructor function

تصاريح الاستخدام visibility :

التصريح العام public يصرح لاستخدام الكلاس في أي مكان داخل أو خارج الكلاس .

التصريح المحمي protected يقتصر على استخدام الكلاس من خلال الكلاسات المتوارثة فقط .

التصريح الخاص private يقتصر على استخدام الكلاس نفسه فقط .

```
<?php
class visibile{
public $public3='public ';
protected $protected3='protected ';
private $private3='private ';
function usevisible(){
echo $this->public3;
echo $this->protected3;
echo $this->private3;
}
}
$vis= new visibile();
```

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com

```
echo $vis->public3; //this only works
//echo $vis->protected3;  this won't work
//echo $vis->private;  this won't work
echo "<br>";
echo $vis->usevisible();
?>
```

Scope resolution operator أو ال SRO :

و هي خاصية في المتغيرات تستخدم للإشارة الى متغير أو تابع function و لابد للمتغير أن يكون ثابت و ليس متغير للإشارة اليه .

```
<?php
class myname2 {
public static $thename="M.hani lakmoush";
function declarescope(){
echo self::$thename;
} }
$use= new myname2;
$use->declarescope();
?>
```

الثابت constant :

```
<?php
class myclass{
const somevar="www.way2paradise.com";
function useconstant(){
echo self::somevar;
}
}
echo myclass::useconstant();
echo "<br>";
echo myclass::somevar;
?>
```

تم الانتهاء من القسم الأول من الكتاب PHP basics بعون الله جاري العمل على القسم الثاني PHP advanced لا تنسونا من صالح دعائكم .

By web developer M.hani lakmoush .
Hani.lakmoush91@gmail.com