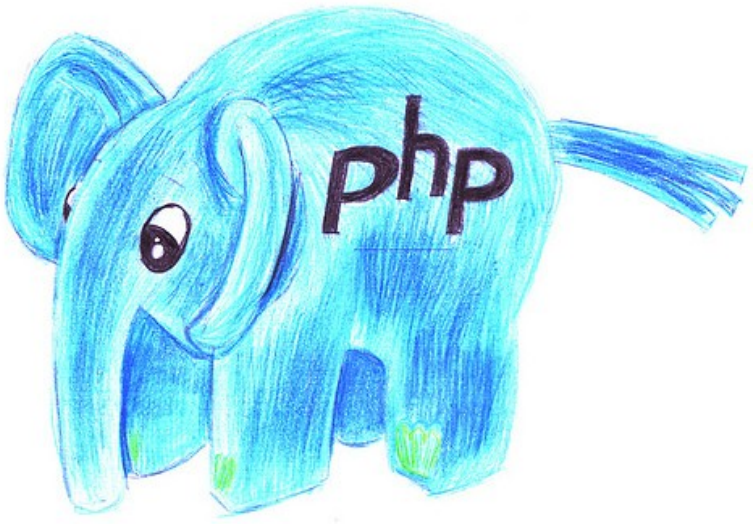
مقدمه مطولة في



(c) flickr.com/afilina

لغة برمجة الويب PHP

كتبها : عبدالرحمن محمد الشويعي

مقدمه :

هذه الكتيب تم جمعه من دروس متفرقه كتبتها في مفيد.كوم في أواخر سنه 2008 وتم تنقيحه في عام 2010.

لم تلقى الدروس ماكنت متوقعا لها من الزياره , ليس لأنها غير جيده فهي من عدة مصادر موثقه بآخر الكتيب , لكن لأن الدروس في منتدى ومتفرقه و يتكاسل الناس في قراءه الدروس التي تكون بهذه الشاكله ..

كتبت الدروس للمبتدئ جداً في اللغه ومع اعطاء امثله بسيطه جداً ليكون مدخل مبسط لهم ..

أيضاً يمكنك ان تكمل بعد قراءه هذه المقدمه في اللغه أي مقال او درس في اللغه لأنك ستكون قابل للتعلم وقادر على قراءه بعض الأكواد ..

حتى لو لم تستفيد منها , يكفي انك تعلمت بعض الأشياء وتعلمت طريقه الاكواد في هذه اللغه .. و لا تيأس وتابع قراءه باقي الكتب التي تجدها في دربك في سبيل التعلم ..

يمكنك طباعه هذه المقدمه و تنسيقها وقراءتها في أيام الراحه و حتى في أيام السفر مثلاً ..

لا تأخذ التعلم على محمل الجد .. فقط أقراها كأنك تقرأ اي شيء اخر كمجله و صحيفه وخلافه ..

التعلم الذي تكون مجبر عليه لن تفهم منه الا 20% فقط .. وهذا لايفيد أبداً في حالتنا ..

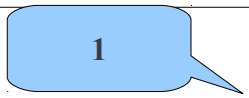
فلتجلب كأسا من الشاي او القهوه .. ولتتابع القراءه ..

عبدالرحمن محمد

صيف عام 2009

الموقع: http://www.idev.me

البريد الالكتروني: saanina@gmail.com



جزء 1 : ماهي الـ php ؟

بالأحرى أنك سمعت بلغة php او قد رأيت بعض الملفات التي تحمل لاحقه "php"
لكن هل فكّرت يومياً , ماهي ال PHP ومالذي أستطيع فعله بها ؟ ومالمميزات التي تجعلني اختارها من بين اللغات الأخرى ؟

لنبدأ..

ماهي الـ PHP ؟

PHP تنطق عربياً "بي إتش بي" وهي إختصار لكلمه **Hypertext Preprocessor** هي لغة لتطوير الويب مفتوحة المصدر ويمكن تضمينها داخل اكواد html

جيد !*لكن ماذا يعني هذا* ؟ إنظر المثال :

CODE:

```
<html>
	<head>
		<title>Example</title>
	</head>
	<body>
		<?php
			echo "أهلا , أنا نص برمجي php";
		?>
	</body>
</html>
```

في PHP لاتحتاج أوامر لعرض ال html فقط قم بتضمينهما معاً وسوف يظهر لك ماتريد !
الكود البرمجي المضمن بالمثال السابق كان بين

CODE:

```
<?php ... ?>
```

وهذا يعني ان اكواد php يجب كتابتهما بين الوسمين هذين لتطبيق النص البرمجي ..
أي نص داخل هذه الوسوم يعتبر نص برمجي php .
المميز في php انها لغة خادم "Server side" أي تنفذ قبل الظهور للعرض بعكس لغة الجافاسكربت التي تعتبر لغة متصفح "Client side" لأنها تنفذ عند ظهور الصفحه. أي ان مصدر الصفحه هو ناتج كود PHP و اي كود PHP لن يظهر في مصدر الصفحه ويكون مخفي على الخادم !.

المميز حقاً في PHP انها مناسبه جداً للمبتدئين و تملك مميزات لاحصر لها للمحترفين !
لاتخاف عندما تقرأ ملف برمجي مليء بالأكواد فغالباً هذه الملفات تم بنائها من سطر واحد وتراكمت السطور . أي انك تستطيع البدء مثل اي مبرمج وتقوم بكتابة سطرك الأول ! وتنطلق ...

أكمل الأجزاء الاخرى لتعّلم أكثر..

جزء 2 : مالذي أستفيده ومالذي أستطيع عمله مع PHP ؟

السؤال يقول "مالذي أستطيع فعله مع php؟" .
والجواب هو " أي شيء ",
اساساً php تركز على تنفيذ الأكواد على الخادم قبل ظهورها للعرض لذا يمكنك فعل الكثير كالإرتباط مع cgi وإنشاء صفحات تفاعليه و إستخدام الكوكيز , و php لا تفعل فقط هذه الأشياء بل أكثر من ذلك .

هناك 3 أماكن يمكن تطبيق أكواد php فيها :

- تنفيذه الأكواد على الخادم وإرسالها للمتصفح وهذا هو الأساس , يتطلب هذا النوع إستخدام معالج php وخادم و المتصفح بالطبع
- تطبيق الأكواد في الأنظمة عن طريق سطر الأوامر
- إنشاء برامج لسطح المكتب مثل إستخدام php-gtk

لا تلقي للنوعين الأخيرين أي بالاً وركّز على النوع الأول وهو الأساس ...

php يمكن إستخدام في أغلب الأنظمه والخوادم في هذه الأيام لذا هي لغة مشهوره جداً , لذا لك حرية إختيار الخادم الذي سوف تعمل عليه مع هذه اللغه .

لك الحريه في php ان تكتب أكواداً نظيفه بسيطه أو تستخدم صيغة المكتبات البرمجيه (او تترجم من البعض الكلاسات و "OOP" إختصار لـ object oriented programming اي البرمجه بإستخدام الكائنات وهو مصطلح مخيف للبعض لذا لاتخف واستخدم مصطلح "مكتبة برمجيه") , وعموماً المكتبات البرمجيه في php4 كانت ضعيفة مقارنة في المكتبات البرمجيه في php5 , المبتدئ لا يجب أن يتعلم هذه الأمور مباشره بل يجب أن يتعلم الدوال الأساسيه قبل كل شيء.

المميز في php أنك لست محدوداً في استخدام طرق للعرض ف php عباره عن لغة خلفيه تعمل بالخلف و تقوم بإسناد مهمه العرض لأي شي أخر كـ html او xml أو حتى ملفات pdf او zip . لذا فقط تحتاج لكتابة كودك الأساسي واستخدام html في العرض مثلاً.

php تملك ميزه قوية جداً , ألا وهي دعم أغلب أنواع قواعد البيانات وأشهر المتعارف عليه هو MySQL حالياً (إلا أنني أفضل أن تقوم بتجربه SQLite للبرامج البسيطه) و هذه الأنواع المدعومه حالياً :

- ✔ dabas D
- ✔ dBase
- ✔ Empress

- ✔ FilePro (قراءه فقط)
- ✔ Hyperwave
- ✔ IBM DB2
- ✔ Informix
- ✔ Ingres
- ✔ nterBase
- ✔ FrontBase
- ✔ mSQL
- ✔ Direct MS-SQL
- ✔ MySQL *
- ✔ ODBC
- ✔ Oracle (OCI7 و OCI8)
- ✔ Ovrimos
- ✔ PostgreSQL
- ✔ SQLite*
- ✔ Solid
- ✔ Sybase
- ✔ Velocis
- ✔ Unix dbm

* : الأشهر حالياً بأغلب البرامج ..

هناك الكثير من المكتبات البرمجيه التي تسهل لك الربط بقواعد البيانات بأكواد بسيطه .. لكن في البدايه ننصح بتعلم الدوال الأساسيه للربط بقواعد البيانات ..

ببساطه , PHP تملك كل ماتريده كلغة خادم ..

جزء 3 : أريد أن أبدأ في php , ماذا أحتاج ؟

تحتاج لمستضيف مفّعل ال php لديه ولو كنت أيضا تريد استخدام قواعد البيانات فلابد ان تكون أحد الانواع مفعله كـ MYSQL .

ماذا لو أردت أن تبدأ في البناء او العمل على جهازك الشخصي ,
أذن تحتاج برنامج يعمل كخادم محلي في جهازك , لذا **قم بإستخدام أحد** هذه البرامج :

لنظام ويندوز :

- خادم الجمل
- wampserver
- xampp
- easyphp

لنظام لينكس

- xampp
- بناء خادم محلي مخصص lamp
- ايضا جرب LampMenu المشروع الذي بدأته منذ فتره.

لنظام ماك

- xampp
- mamp
-

بتثبيتك أحد هذه البرامج تستطيع الآن إنشاء ملفات php وتنفيذها بسهوله (لندع الامر هذا للأجزاء اللاحقه) و تعطيك امكانيه الربط مع قواعد البيانات , إذا جهازك الآن لديه "خادم محلي"local server لاتنسى هذا المصطلح.

جزء 4 : صفحتك الاولى في PHP

قم بإنشاء ملف نصي جديد بإسم مثلاً hello.php ولاحظ أن النهاية .php
وضعه في مجلد العرض في الخادم (غالبا يكون أسمه www او htdocs)
قم بوضع المحتوى التالي بداخله :

CODE:

```
<html>
	<head>
		<title>PHP Test</title>
	</head>
	<body>
		<?php echo '<p>Hello World</p>'; ?>
	</body>
</html>
```

قم بطلب الملف عن طريق المتصفح عن طريق الرابط

http://localhost/hello.php

او

http://127.0.0.1/hello.php

سوق يظهر لك "Hello World" اما لو قمت بعرض المصدر من المتصفح سوف تجد

CODE:

```
<html>
	<head>
		<title>PHP Test</title>
	</head>
	<body>
		<p>Hello World</p>
	</body>
</html>
```

لم يظهر أي كود من أكواد php هذا لأنها لغة خادم , أما أكواد الهتمل فظهرت لأنها لغة العرض , وهذا يعني أن أكواد php منفصلة تماما عن العرض وذلك من مميزاتها ..

صحيح أن الملف الذي قمنا بعمله بسيط لكنه مهم في البدايه لتعرف الأساس .

مثلا عرفنا الآن أن الملف تم إرساله لمعالج php لانه يحمل إمتداد .php
بعدما ذهب للمعالج , قام المعالج بفحص الكود فوجد أنه يحمل وسمي البدايه والنهاية ل php

CODE:

```
<?php ... ?>
```

قام بتنفيذ الأوامر داخلهما ..

الأمر echo أو لنقل الداله , تقوم بعرض مايكتب بعدها ...
قام المعالج بتنفيذ الأمر هذا وعرض الكود الذي بعدها
وأرسل الناتج للمتصفح , الناتج كان نص "Hello World"
أما الكود الباقي فهو كود هتمل عادي لم يمرر للمعالج لانه لم يبتدي و ينتهي بوسمي php

ملاحظات :
- لاتقم بإستخدام محررات النصوص المتقدمه كأوفيس وورد , لذا إستخدم المفكره او برنامج notepad++
- عند حفظ الملف في المفكره تأكد أن تحفظه بالإمتدادا .php وليس .txt لأنه قد يقوم بها أحياناً ..
الآن تستطيع إنشاء الكثير والكثير .. فقد عرفت البدايه .. والباقي .. لن يكون أصعب لو كنت تريد فعلاً ان تتعلم ..

الآن قم بتجربه بعض الأكواد .. مثلا قم بحذف الكود السابق من الملف hello.php ووضع

CODE:

```
<?php phpinfo(); ?>
```

جرب وإنظر ماذا يحدث !.

جزء 5 : أشياء مفيده أخرى ...

لنفعل شي اكثر تطوراً الآن ..

مثلا , لنقم بمعرفة نوع متصفح الزائر وذلك عن طريق user agent له , الذي يحمل نوع المتصفح والنظام وامور اخرى مفيده .

user agent يتم تخزينه في **متغير** (لاحقا سنعرف ماهو المتغير) , والمتغيرات في php تبدأ دائما بعلامة الدولار $, المتغير الذي نتحدث عنه هو $_SERVER['HTTP_USER_AGENT'].

قم بوضع هذا الكود في ملف hello.php ألذي تكلمنا عن في الجزء السابق

CODE:

```
<?php
echo $_SERVER['HTTP_USER_AGENT'];
?>
```

سوف يظهر شي قريب من

CODE:

```
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
```

المتغير هذا ماهو إلا عنصر في متغير $_SERVER الذي هو بدوره عباره عن مصفوفه "array"

المتغير $_SERVER هو عباره عن "متغير سوبر" superglobal وهو ليس الوحيد فيوجد متغيرات سوبر أخرى وهو معرفه مسبقا ومفيده بالطبع .

لفهم ماقمنا بطباعته , عنصر من مصفوفه ..

CODE:

```
$1[2]
```

حيث ان 1 هو اسم المصفوفه و 2 هو العنصر داخل المصفوفه ويمكن أيضا أن يكون العنصر 2 مصفوفه أيضا , والمصفوفات متشعبه جداً, لكن غالباً تكون مجرد متغير يحمل عناصر بدون تشعب كبير كما في المثال .

المهم , نحن الآن قمنا بطباعه محتوى كثير ويوجد فيه نوع النظام والمتصفح وغيره , أريد الآن أن اقوم بتحديد نوع النظام بدقه, لذا نستخدم الداله strpos التي تقوم بمعرفه "هل يوجد النص المحدد في متغير ما أو لا" ,

شرح الداله ببساطه :

CODE:

```
strpos($var, 'text')
```

حيث ان $var هو المتغير الذي نريد أن نرى هل يحوي بداخله على text او لا ! سوف تقوم بإرجاع قيمه true او false أي نعم أو لا .

مع مثالنا نستخدمها كالتالي:

CODE:

```
<?php

if (strpos($_SERVER['HTTP_USER_AGENT'], 'MSIE') !== FALSE)
{
	echo 'أنت تستخدم متصفح  Internet Explorer.<br />';
}

?>
```

سوف يظهر لو كنت أستخدم متصفح اكسبلور التالي

CODE:

```
أنت تستخدم متصفح Internet Explorer.
```

لو كنت تستخدم متصفح اخر مثل فيرفوكس لن تظهر الرساله لك .. جرب بنفسك ..

إستخدمنا في المثال الأمر if وهو مساوي للكلمة العربيه "لو"
ونستطيع التعبير عن المثال السابق باللفظ كالتالي :

لو أن المتغير $_SERVER['HTTP_USER_AGENT'] يحوي الكلمة MSIE قم بعرض النص أنت تستخدم متصفح Internet Explorer.

لنقم بجعل المثال متطور أكثر .. قم بوضع الكود التالي بدلاً من السابق في ملف hello.php

CODE:

```
<?php

if (strpos($_SERVER['HTTP_USER_AGENT'], 'MSIE') !== FALSE) {
?>
	<h3>strpos() قامت بإرجاع قيمة لاتساوي false</h3>
	<p>أنت تستخدم Internet Explorer</p>
<?php
} else {
?>
	<h3>strpos() قامت بإرجاع قيمة تساوي false</h3>
	<p>أنت لاتستخدم Internet Explorer</p>
<?php
}
?>
```

سوف يظهر لو نفذته :

CODE:

```
strpos() قامت بإرجاع قيمة لاتساوي false
أنت تستخدم Internet Explorer
```

أما لو كنت لم تستعرض من متصفح غير Internet Explorer سوف يظهر

CODE:

```
strpos() قامت بإرجاع قيمة تساوي false
أنت لاتستخدم Internet Explorer
```

قمنا بإستخدام else وتعني "لو لم يكون السابق قم بعمل التالي"

ونستطيع ترجمة المثال السابق لفظياً كالتالي :

لو أن المتغير $_['SERVER['HTTP_USER_AGENT يحوي القيمه MSIE قم بإرجاع ... لو لم تكن فقم بإرجاع ..

ايضا يوجد الأمر else if ونضعه بعد الشرط if لو أردنا ان نقول لو لم يكون الاعلى كذا وكان التالي كذا فإفعل كذا ..

أيضا لو لاحظت المثال لوجدت أننا قفزنا من كود php للهتمل وذلك بإستخدم <?php ... ?> و الهتمل جعلنا العرض له بدون هذه الوسوم ! وهذه ميزه مهمه يجب أن تحاول التمتع بها قليلاً ..

لا حظنا أيضا أن الدوال مهمه في تنفيذ أغلب الأشياء في php والداله عباره عن كلمة او جمله بسيطه مع أقواس , الأقواس تعبأ بالقيم المطلوبه لتنفيذ الداله , كما شرحنا مع strpos ..
وهذه هي الفكره مع أغلب الدوال .

جزء 6 : التعامل مع النماذج والمدخلات

غالباً لا يخلو موقع من النماذج "forms" وهي عباره عن مجموعة حقول مختلفه تطلب من الزائر مثلا إدخال معلومات فيها , هذا هو سر تفاعلية صفحات ال php ..والميزه الأقوى ضمن مميزات php .

لننشأ صفحة بإسم مثلا myform.html ولاحظ هنا أنها مجرد صفحة هتمل عاديه , ولنضع التالي فيها :

CODE:

```
<form action="action.php" method="post">
      <p>إكتب إسمك: <input type="text" name="name" /></p>
      <p>إكتب عمرك: <input type="text" name="age" /></p>
      <p><input type="submit" value="تـنـفـيـذ" /></p>
</form>
```

لو لاحظت الكود السابق لوجدت أنه كود نموذج عادي يبتدأ بـ <form...> ويتنهي بـ </form> كأي وسم html ووسوم الهتمل غالبا تحوي بوسم الفتح معلومات تسمى كل معلومه "خاصيه"

CODE:

```
<tag attribute1="قـيـمـة">...</tag>
```

في مكان tag يكون أي وسم آخر كـ form و ينتهي بنفس الأسم الذي إبتدأ به
وسم الفتح يحوي خواص قد تكون كثير او قليه بحسب ماتريده
والخاصيه بالشرح هي attribute1 ولها يكون **القيمة** التي قد تكون معرفه مسبقا او تكون عامه.

وقد قام الأخ "سردال" مشكوراً بشرحها تفصيلا هنا :
سردال » ما الفرق بين tag والكلمات الأخرى؟
وبعد إنشاء الصفحه myform.html نقوم بحفظها وننشأ بنفس المجلد ملف أخر بأسم مثلاً action.php ونضع التالي فيه :

CODE:

```
مرحباً <?php echo htmlspecialchars($_POST['name']); ?>.

عمرك هو <?php echo intval($_POST['age']); ?> سنه.
```

إستخدمنا الداله htmlspecialchars لتنظيف المدخل الإسم من أكواد الهتمل للحمايه من أكواد الهتمل الخبيثه واستخدمنا داله intval لتنظيف المدخل , لو كان رقما إرجعه لنا, لو كان غير هذا قام بإرجاع قيمة 0 .

الناتج سوف يكون قريب لـ

CODE:

```
مرحباً عبدالرحمن. عمرك هو 20 سنه
```

جرب بنفسك وإكتب مره إسم يحوي مثلا أكواد هتمل وإجعل مره العمر نص وليس رقم للتأكد من عمل الدوال التي شرحناها بالأعلى ...

المتغير $_POST هو عباره عن متغير سوبر , او لنقل مصفوفه يتم تخزين بها المدخلات من النماذج التي يكون فيها "الخاصيه "method="post
كما بالمثال

CODE:

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

لو كان النموذج يملك الخاصيه method مع القيمه get مثل

CODE:

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

إذن نستخدم المتغير السوبر $_GET

CODE:

```
$_POST['إسم الحقل']
```

إسم الحقل هو الاسم الذي قمنا بتعريف كل حقل في النموذج به مثل :

CODE:

```
<input type="text" name="إسم الحقل" />
```

لاحظ

طبعا إسما الحقول يجب أن تكتبها بالانجليزيه , رغم أني لم أجرب لغات أخرى , لكن لتفادي المشاكل ..

جزء 7 : النماذج والمتغيرات الخارجيه

كلمنا في الجزء السابق عن النماذج , و ببساطه عن $_POST و $_GET..

وقلنا أن النماذج يتم إرسال المعلومات بعد إدخالها إلى صفحة php بعدة طرق , مثلاً هذا الكود الهتمل:

CODE:

```
<form action="foo.php" method="post">
   الأسم:  <input type="text" name="username" /><br />
    البريد: <input type="text" name="email" /><br />
    <input type="submit" name="submit" value="إرسل المعلومات!" />
</form>
```

وطبعا كما تعلمنا سابقاً أن صفحة php التي سوف يرسل لها النموذج يوضع في خاصيه action والصفحة في المثال تكون foo.php

وطريقة الإرسال تكون في خاصية method وهنا خاصية الإرسال هي post إذن يتم إستقبال المعلومات عن طريق $_POST .

المعلومات التي يتم إستقبالها هي المتغيرات الخارجيه التي سوف نتكلم عنها ...

المتغيرات المستلمه من النماذج يتم إستلامها بناءاً على نوع النموذج post او get .. لكن هناك طرق أخرى لإستلام المتغيرات هذه بنوعيها .. وهي $_REQUEST وهي تشمل النوعين كلاهما , فلو كنت غير متأكد من نوع النموذج الذي سوف تستقبل منه المتغيرات إستخدم هذه ..

بعض المستضيفين يفعّل الإستقبال بدون الحالجه لهذه الأنواع , فلو كان الحقل بإسم username يتم إستقباله كمتغير عادي $username وتسمى هذه الخاصيه "register globals" أي تسجيل المتغيرات كمتغيرات عامه, ولافرق بين المتغيرات الخارجيه والمحليه .. لكن إنتبه فهذه الخاصيه خطره ولاتستخدمها , ولو كانت مفعله لديك في الاستضافه قم بطلب بتعطيلها.

عرفنا الآن أن استقبال المعلومات من النماذج يتم بناءاً على نوع النموذج وهناك طرق أخرى غير مفضله قلناها , لكن ركز على الأثنين $_POST و $_GET .

المثال السابق , كان عباره عن صفحه هتمل , قم بتسميتها أي أسم وليكن myform.html وضعها على الخادم والآن لنقم بإنشاء ملف الإستقبال foo.php ألذي سيتم إستقبال النموذج فيه . ولتكن محتوياته :

CODE:

```
<?php
	echo $_POST['username'];
	echo $_REQUEST['username'];
?>
```

لاحظ أن الإسم سيطبع مرتين, لان $_REQUEST تجلب كل النوعين .

جميل لحد هنا . توضحت أمور كثيره . لكن لم نعطي $_GET حقه.
$_GET هو النوع الذي يستلم المعلومات من الرابط نفسه .
لاحظ الرابط هنا:

CODE:

```
localhost/foo.php?username=saanina
```

username هو المتغير الذي سوف يحمله $_GET

قم بحذف محتويات ملف foo.php ووضع :

CODE:

```
<?php
	echo $_GET['username'];
	echo $_REQUEST['username'];
?>
```

وقم بطلب العنوان السابق بالمتصفح لديك على إعتبار ان الملف موجود على خادمك المحلي ..
سوف يظهر الأسم saanina مرتين.

لو لاحظت أن $_GET يتم حملها عن طريق الرابط , لذا هي لاتنفع للأمور المهمه مثل كلمات المرور أو مثل الحقول الطويله كحقل نص المقال مثلا ..

جزء 8 : المتغيرات بشكل عام

تكلمنا سابقا ببساطه عن المتغيرات ولم نركز عليها .. لكن الآن لنفهم شيئاً بسيطاً عنها ..

المتغيرات في php تبدأ بعلامة الدولار $ ويتبعها إسم المتغير , وإسم المتغير حساس لحالة الاحرف "case-sensitive" , بعضكم لن يعرف معنى هذه الكلمة , حتى أنا في البدايه لم أتقن الفكره إجمالاً , لكن سوف أخبرك الآن معناها ببساطه , حساس لحالة الأحرف الأنجليزيه أي أن الحرف a ليس هو نفسه A في المتغير فمثلاً :

CODE:

```
$ahmed
```

لا يعني أبداً بأي حال من الأحوال :

CODE:

```
$Ahmed
```

فكلاهما متغير مختلف , فقط لتغير نوع الحرف "كبير كان او صغير" .. هذا الموضوع ببساطه !

اسم المتغير أي مابعد علامة الدولار $ يبتدي بأي حرف او شرطه سفليه "_" يتبع ذلك أي عدد من الحروف او الأرقام.

هناك إسم واحد وهو "this$" لايمكنك إستخدامه .. لأنه محجوز .

لنأتي للفكره عموماً .. لنكتب مثالاً ..

CODE:

```
<?php
	$var = 'Bob';
	$Var = 'Joe';
?>
```

المتغير هو حيّز يحمل قيمة معينه , يمكن تغييرها كل ما أردت ..

في المثال السابق المتغير $var يحمل قيمة هي "Bob" أما المتغير $Var فيحمل القيمة "Joe"

ماذا لو قمنا بطباعه المتغيرات :

CODE:

```
<?php
	echo $var;
	echo $Var;
?>
```

سوف ينتج :

CODE:

```
BobJoe
```

ماذا لو قمنا بتجربة اول مثال مع تغيير بسيط جداً ..

CODE:

```
<?php
	$var = 'Bob';
	$var = 'Joe';
?>
```

ثم نقوم بطباعته :

CODE:

```
<?php
	echo $var;
?>
```

سوف ينتج :

CODE:

```
Joe
```

المتغير , يوصف بأنه أيضا متغير , أي ان الاسم مطابق لوظيفته , فهو يحمل اخر قيمة تم إسنادها إليه (بعكس الثوابت, لاحقا نمر عليها) .. فالمتغير كان يحمل قيمة "Bob" و قمنا بتغييرها لـ "Joe" , إذن القيمة تغيرت واصبحت أخر قيمة تم إسنادها له !

نسيت أن أخبرك أن قيمة المتغير تكون نصوص أو أرقام و أشياء أخرى لاتشغل بالك بها الآن , المهم الآن هو أن المتغير يحمل قيمة وتتغير القيمة بتغييرنا لها كل ما أردنا ..

أيضا تستطيع أن تعطي قيمة متغير لمتغير أخر مثل :

CODE:

```
<?php
      $var1 = $var2;
?>
```

وبالتالي كلاهما يحملان الآن نفس القيمه .

لنتأكد من صحة التالي :

CODE:

```
$4site = 'not yet';
$_4site = 'not yet';
$täyte = 'mansikka';
```

الأول خاطئ لأنه لم يبدأ بحرف أو "_"

الثاني صحيح لأن بدأ بـ"_" ويحوي حروف وأرقام

الثالث أيضا صحيح لأن الحرف "ä" من حروف ASCII وهي حروف مسموحة بالأسماء ..

هناك ميزه في المتغيرات , لو كان لدي متغيران , وكلاهما أريد أن اجعل لهما قيمة واحده , وأن أغير كلاهما في نفس الوقت .. عندما أفعل هذا كل مره مع كل المتغيرين فهذا شي عادي , لكن لو كان هناك شي يوفر لي الوقت فلابأس , هناك ميزه إسمها "assign by reference" أي "وضع قيمة من مرجع" ترجمة ليست حرفيه , لكنها للفهم ... أي ببساطه جعل المتغيرين توأمين بكل شيء

المثال على الميزه :

CODE:

```
$foo = 'Bob';
$bar = &$foo;
$bar = "joe";
echo $bar;
echo $foo;
```

السطر الأول يعني أن $foo يحمل قيمة "Bob"

السطر الثاني قلنا أن المتغير $bar له مرجع من المتغير $foo.

السطر الثالث غيرنا قيمة المتغير $bar وبالتالي سوف تتغير قيمة المتغير $foo

السطر الرابع والخامس قمنا بطباعه المتغيرين وسوف ينتج القيمة "joe" مرتين

أي أن مايحصل لأحد المتغيرين يحصل للآخر ..

لايهم أن كنت لم تفهم اخر ميزه , فهي غير مهمه.

جزء 9 : المتغيرات , أكثر و أكثر

في الجزء السابق تكلمنا عن المتغيرات بشكل مفصل , لكن لم نتطرق لأشياء قد تكون مفيده بنظري ..
المتغيرات لها أنواع ولو سردت لك الانواع وأنهيت الجزء لن تفهمها , لذا لنأتي بها بشكل مبسط !

المتغيرات تكون بأي مكان بالصفحه , تستطيع كتابتها داخل الدوال وأيضاً تستطيع جلبها عن طريق المتغيرات السوبر "superglobal" كـ$_POST وغيرها.

المتغيرات العاديه هي التي تكتب بأي مكان , ولن تتوفر داخل الدوال وستنتهي مهمتها على الصفحه نفسها. أما المتغيرات السوبر فيمكن جلبها داخل الدوال , أما المتغيرات التي داخل الدوال لن تعمل خارج الدوال !

شيء معقد بالبدايه !

فلنفترض أن المثال التالي هو صفحتك ويحوي :

CODE:

```
$ahmed = 'text';

function foo()
{
	$khaled = 'text';
}
```

المتغير $ahmed متغير عام "لكنه ليس متغير سوبر" ولن يعمل داخل الداله foo و $khaled متغير محلي يعمل فقط داخل الداله.

يمكننا تمرير المتغير العام لداخل داله عن طريق global كالتالي :

CODE:

```
$ahmed = 'text';

function foo()
{
	global $ahmed;
	$khaled = 'text';
}
```

المتغير $ahmed الآن يمكن إستخدامه داخل الداله, نسيت أن أخبرك أنه يمكنك تمرير المتغيرات العامه داخل الدوال أيضا
بإستخدام المتغير السوبر $GLOBALS
فـ :

CODE:

```
$GLOBALS['ahmed']
```

هي نفسها :

CODE:

```
$ahmed
```

هنا عرفنا انواع المتغيرات (عامه , سوبر , محليه)
لكن لم نعرف قائمة بالمتغيرات السوبر "Superglobals" ...
هذه هي المتغيرات السوبر مع شرح وظيفة لكل منهما :

المتغير	وظيفته
$_GET	المتغيرات الممره عن طريق رابط
$_POST	المتغيرات الممره عن طريق نموذج يستخدم هذا النوع
$_SERVER	متغيرات تحمل معلومات الخادم وأشياء تتعلق بالصفحه
$_COOKIE	متغيرات الكوكيز
$_FILES	متغيرات تحمل معلومات الملفات المحمله
$_ENV	متغيرات تحمل معلومات تتعلق بـ php
$_REQUEST	المتغيرات التي توجد في $_COOKIE, $_GET , POST
$_SESSION	متغيرات الجلسات

هذا الجدول ليس للحفظ , سوف تفهمه جيداً مع الوقت وستقول يا لتفاهة هذا الجدول :) .

المتغيرات تكون بعدة أنواع ,

إما نصية وتسمى "string"

وإما رقمية وتسمى "integer" أي رقم حتى لو سالب

وإما كسريه وتسمى "float" مثل 2.3

وإما "boolean" وهو إما يحمل قيمة 1 او 0 (true او false)

وبالتالي بما ان المتغيرات ممكن أن تكون أعداداً , منطقياً أننا نستطيع تنفيذ عملية حسابيه بين متغيرات مثل ضرب وجمع وطرح وقسمة ونسبه وغيره

مثل :

CODE:

```
$a = 1;
$b = 2;
$result = $a + $b;
echo $result;
```

كأي رقمين قمنا بعملية حسابيه , لكن يجب أن تنتبه أن بعض العمليات لها رموز أخرى ك

جمع +

طرح -

ضرب *

قسمة /

نستطيع ربط (أي جعل قيمهم بجانب بعض وليس جمعهم كالحساب) بين متغيرين بطرق كالتالي:

CODE

```
$var1 = 'I am ';
$var2 = 'ahmed';
echo $var1 . $var2;
```

لاحظ النقطه بين المتغيرين , هذه ليست فقط خاصه بطباعة العناصر , أيضا في أي مكان يطلب متغير.

أو جعلهم بين " .. "

CODE:

```
echo "$var1$var2";
```

لكن ليس بين ' .. ' لأنها تعطل المتغيرات داخلها , جرّب بنفسك !

اعتقد أننا وصلنا لمرحلة متقدمه بالمتغيرات وسوف ننتقل لجزء أبسط

كن متابع ,,

جزء 10 : الثوابت

كما عرّفنا سابقاً المتغيرات بأنها قيم متغيره ..
سوف نعرّف الآن الثوابت بأنها قيم ثابته ! شي عجيب أليس كذلك .
لاتستغرب إن قلت لك أن هناك كثير من المطورين لايعرفون الفرق بينهما حتى الآن !
هذا لأنهم يترجمون الدروس حرفياً فمثلا كلمة "constant" تظل معهم من أول الدرس حتى أخره ولا يعرف المتلقي معناها ! لندع هذا الأمر جانباً الآن .

قلنا سابقاً أن الشكل هذا :

CODE:

```
<?php
	$var = 'ahmed';
?>
```

يمثل متغير قيمته "ahmed"
ماذا لو قمنا بوضع المثال كالتالي :

CODE:

```
<?php
	$var = 'ahmed';
	$var = 'khaled';
?>
```

سوف تقول لي بأن هذا الكود يمثل متغير تم إعطائه القيمة "ahmed" وتم تغييرها إلى "khaled" ولكن سوف أسألك , سؤال واحد فقط . ماهي قيمة المتغير $var الآن ؟
سوف ترد بعفويه إنها "khaled" , هذا ياعزيزي لأن المتغير يحمل آخر قيمة تم إسنادها له !

لماذا نعيد هذا الكلام ؟ لأن هذا ياعزيزي هو لُب الموضوع والمدخل لفهم الثوابت
الثوابت لايتم إعطائها قيمة بالشكل الذي تنتهجه المتغيرات , أيضا لاتحمل الرمز الدولار $ في بدايتها.

هكذا يتم إنشاء ثابت بقيمة محدده :

CODE:

```
<?php
	define("MOFFED", "website");
?>
```

المثال السابق قمنا بإنشاء "ثابت" بإسم MOFFED وإعطائه القيمة website

أي أن الثوابت لها دالة تعريف أسمها define ولها خواص الأول إسم الثابت والثاني قيمته .

ماذا لو غيّرت رأيي وأعطيت قيمة مختلفه بعد الكود السابق للثابت MOFFED هل تتوقع أن تتغير !

CODE:

```
<?php
	define("MOFFED", "website");
	define("MOFFED", "great website");
	echo MOFFED;
?>
```

الكود السابق قمنا بتغيير قيمة الثابت MOFFED ! لكن جرّب الكود , الثابت لن يتغير , لأنه ببساطه ثابت , يحمل أول قيمة يتم إسنادها له ويتجاهل الباقي !

قد تستفيد لو عرفت أن هناك داله تفحص هل الثابت معرّف أو لا . وهي :

CODE:

```
<?php
	defined('MOFFED')
?>
```

تفحص هل الثابت (وفي المثال MOFFED) موجود أ ولا ,

وترجع قيمة موجبه او سالبه لو كان موجوداً او لا.

تابع الأجزاء .. وركزّ وجرّب !

جزء 11 : الثوابت العجيبه

عرفنا ماهي الثوابت سابقاً .. وهي قيم يتم تعريفها ولايتم تغييرها في اي جزء من السكربت لاحقاً ..
اما في هذا الجزء سوف نعرف ثوابت من نوع اخر .. وهي ثوابت معرفه مسبقاً لك لتستفيد بسرعه منها .. وهذه الثوابت منها ماهو متغير على حسب حالته . مثلا __LINE__ ثابت يعطيك رقم السطر الموجود فيه هذا الثابت . لو غيرت مكانه من سكر لآخر سوف يتغير الرقم الذي يعطيك إياه .
وهذه الثوابت الخاصه غير حساسه لحالة الاحرف أي ان __LINE__ هي نفسها __line__

من هذه الثوابت العجيبه كما احب ان اسميها وكما يسميها البعض الثوابت الخاصه "MAGIC CONSTANTS" :

CODE:

```
<?php
	echo __LINE__;
?>
```

سوف يطبع رقم السطر الموجود فيه هذا الثابت ..

CODE:

```
<?php
	echo __FILE__;
?>
```

سوف يطبع اسم الملف مع المسار كاملاً له ...

CODE:

```
<?php
	echo __FUNCTION__;
?>
```

يعطيك اسم الدالة الموجود فيها هذا الثابت مثل :

CODE:

```
<?php
functon foo()
{
      echo __FUNCTION__;
}
?>
```

لو لاحظت ان الدالة اسمها foo و الثابت داخلها لذا سوف يتم طباعة اسم الدالة .. foo

هناك اكثر من هذه الثوابت العجيبه لكن المهم منك فقط الآن هو انت تعرف ان هناك ثوابت معرفه مسبقاً تستطيع استخدامها في سكربتك وتعطيك معلومات جاهزه .. اما مسألة حفظها فهذا دعه للزمن !

جزء 12 : أنواع البيانات

في كل لغات البرمجه الاخرى هناك انواع للبيانات يتم تحزينها فيها.. كل نوع له ميزه ..
فلنضرب مثلا بعيدا عن البرمجه .. في المطبخ مثلا كل إناء له فائده , مثلاً المقلاة للبيض , الكوب للسوائل كالشاهي ..

هذا هو المعيار . كل شيء له إناء .. كل بيانات لها نوع يتم تخزينها فيه..

الـ PHP لها 8 انواع اساسيه .. حفظ هذه الانواع بدون فهمها . كحفظ ارقام صفحات الكتاب بدون معرفه محتوياته .. فقط افهمها ولا يهم ان لم تحفظها ..

- اربعه انواع منها ذوات قيم :

- boolean
- integer
- float
- string

- نوعين تعتبر مركبات :

- array
- object

- و نوعين خاصين :

- resource
- NULL

سوف نأخذ اول 4 انواع اما الباقي فلاحقاً .. سوف تعرفهم من نفسك ...

معنى ذوات قيم اي تحمل قيمة ..

نوع البيانات يتم تحديده من قبل المطور "المبرمج" ... وكل نوع كما قلنا له ميزه وسوف نشرح كل واحد منهم .

CODE:

```
$bool = TRUE;
```

عندما يكون المتغير عباره عن نعم او لا .. وهي مايعبر عنها بـ true او false على التوالي ..
فائده هذا النوع هو للتأكد , وهذا النوع هو الذي يتم ارساله من اغلب الدوال عندما يكون فيها خطأ مثلا .. وايضا يتم استخدامه في النماذج .. مثل : هل انت متزوج ... وغيرها من الاسئله ذوات الاجابه بنعم او لا .
هذا النوع هو الاول ويسمى "Boolean" .. ولاستخدامه في برنامجك فقط استخدم true او false كالمثال السابق ..

CODE:

```
$a = 1234;
```

النوع هذا يسمى "integer" ويعني ان البيانات تكون ارقام صحيحه ...,-2, -1, 0, 1, 2, ...
والارقام الصحيحه لاتحوي فواصل للارقام العشريه كـ 1.33
وتشمل الارقام السالبه والموجبه

CODE:

```
$a = 1.234;
```

النوع هذا يسمى "float" ويعني ان البيانات تكون ارقام حقيقيه .. اي ارقام تحوي ارقام عشريه اي كل رقم يحوي كسراً وهو مايعبر عن بالفاصله بالكمبيوتر كالمثال السابق "."

CODE:

```
$a = 'moffed.com is a best website';
```

هذا النوع يمسى "string" اي ان البيانات تكون نصوصا أي مجموعه حروف وارقام وحتى روموز .. اي شي يمكن تخزينه في هذا النوع على الاغلب ..

هذه الانواع ببساطه .. لاتحتاج ان تزيد معلوماتك حالياً حول هذه الانواع .. فقط اعرف ماكتب ومالميزه التي تحويه و يكفي .. اما لو كنت طالب نجيب .. وتحب ان تزيد معرفتك .. فقم بالاطلاع على المصادر ..

13

جزء 13 : الشروط

في حياتنا اليوم نتكلم بالشروط , ونعيش بها , ولأنها جزء من الحياة لانعتبرها شي جديد .. هنا في لغات البرمجه الأمر مماثل .. لابد ان نضع شروط لتنفيذ الاولويات والاهداف ..

ماذا نقصد بالشروط ,

ببساطه هي كالجمله التاليه " لو كان هذا كذا , إفعل كذا" ...

IF

احد اهم مميزات ال PHP واللغات البرمجيه الأخرى وتكتب الصيغه كالتالي :

CODE:

```
if (expr)
      statement
```

expr يكتب مكانها الصيغه ..

statement يكتب مكانها الفعل

مثل : لو احمد مريض إذهب للمستشفى ...

فـ الصيغه في المثال هي "احمد مريض" والفعل هو "اذهب للمستشفى"

فلنفرض ان لدينا متغيران ولهم قيم كالتالي:

CODE:

```
<?php
      $a = 1;
      $b = 2;
?>
```

ولنكتب شرط يقول :

لو كان المتغير a اكبر من b اطبع جملة a اكبر من b

CODE:

```
<?php
if ($a > $b)
      echo "a اكبر من b ";
?>
```

وتستطيع ايضا عمل اكثر من فعل داخل الشرط , مثل المثال السابق لكن نقول افعل ايضا " ان a تكون مساويه لـ b "

CODE:

```
<?php
if ($a > $b)
{
      echo "a اكبر من b ";
      $a = $b;
}
?>
```

لاحظ الأقواس المعكوفه {...} تضعها لو كان الفعل اكثر من سطر ..

ELSE

مكملة لـ IF .. وتعني "لو لم يتنفذ الشرط السابق قم بعمل التالي"

ونغير بمثال if قليلاً ..

CODE:

```
<?php
if ($a > $b)
{
      echo "a اكبر من b ";
}
else
{
      echo "a ليس اكبر من b ";
}
?>
```

ELSE IF

تاتي بعد ال IF او بعد ELSE IF وتعني لو لم يكن السابق وكان التالي افعل هذا ..

المثال :

CODE:

```
<?php
if ($a > $b)
{
      echo "a اكبر من b ";
}
else if ($a < $b)
{
      echo "a اقل من b ";
}
?>
```

المطلوب في هذا الجزء فهم الفكره ... وعندما تفهم الفكره سوف ترسخ هذه المعلومه .. حول الشروط.

جزء 14: التعليقات

اعتبر هذا الجزء استراحه , لذا حاول انت تستمتع بقراءته ...

التعليقات غالبا تعني كلاما خارج الموضوع الاصلي .. وهذا معناها ايضا في كل لغات البرمجه
فالتعليقات comments هي السطور التي لا يتم قراءتها من قبل المعالج ويتم قفزها .. وهي عباره عن توضيح لقارئ الكود ..

مثلا , عندما اكتب كود يقوم بطباعه صوره لكني لم احدد الصوره, فقط اكتب تعليقا فوق الكود اوقول "سوف اضع رابط صوره هنا بدل النقاط ..."

CODE:

```
<?php
      //سوف اضع رابط صوره هنا بدل النقاط ...
      echo "...";
?>
```

لاحظ ان التعليق هو اي نص بعد // او # ويسمى للعلم فقط "تعليق السطر الواحد"

تعليق بعد #

CODE:

```
<?php
      #سوف اضع رابط صوره هنا بدل النقاط ...
      echo "...";
?>
```

ويمكن ان يكون التعليق بعدة اشكال ...

CODE:

```
<?php
	/*
	سوف اضع رابط صوره هنا  ...
	 بدل النقاط
	*/
	echo "...";
?>
```

والتعليق السابق كان متعدد الاسطر لذا يسمى "تعليق متعدد الاسطر"

فقط .. هذه التعليقات وهذه وظيفتها ..

- التعليقات هي نصوص لايتم معالجتها من قبل المعالج ويتم قفزها .
- التعليقات وضعت لكي يتم قراءتها من قبل البشر .
- التعليقات اما ذوات سطر واحد وهي مابعد # او // او متعدد الاسطر وهو مابين /*...*/
- التعليقات غالبا تعبر عن شخصية المطور لذا احسن كتابتها واستخدم الفاظا حسنه.

جزء 15: الدوال بشكل عام

الدوال هي دوال :) ...
لا يمكن تعريف الدوال الا بالقول انها اهم ميزه باغلب لغات البرمجه ..
الدوال عموما تقوم بوظائف مختلفه لتبسيط عملية البرمجه و تسريع الوصول لنتيجه
والدوال اما عامه وهي ماتأتي مع اللغه نفسها او مخصصه وهي ماتقوم انت بعملها.

فنلبسط المسأله اكثر ... الدوال هي كلمة بالغالب و يلحقها اقواس.

CODE:

```
<?php
	foo()
?>
```

حيث ان foo هو اسم الداله . والاقواس الملحقه بالداله هي التي تميز الدوال عن غيرها وتجعلنا نقول ان هذه دالة قطعاً ..

الدوال غالبا تعطى مدخلات "parameters" وتوضع هذه المدخلات داخل الاقواس

CODE:

```
<?php
	foo(parameters)
?>
```

المدخلات "parameters" يتم فصلها عن ادخالها للداله بفاصله ,

CODE:

```
<?php
	foo(param1, param2)
?>
```

الدوال غالبا ترجع قيم , اما قيم صح او خطأ true او false
او ترجع القيمة التي ادخلتها بالداله لكن بعد تنفيذ الوظيفه عليها ..

لنأخذ مثال على داله الآن ...

دالة **strlen**
تقوم هذه الدالة بإرجاع عدد حروف المدخل لها ..

مثال :

CODE:

```
<?php
	echo strlen('moffed.com');
?>
```

بعد تنفيذ الكود سوف يطبع "10" وهو عدد حروف moffed.com
اغلب الدوال تقوم بارجاع القيمة ولاتطبعا
اي اننا بالمثال قمنا بالطباعه ولم تقم الداله بذلك ..

لاحظ ان الداله يمكن ان نتلقف القيمة التي ترجعها بواسطة متغير كالتالي :

CODE:

```
<?php
	$var = strlen('moffed.com');
?>
```

الآن المتغير var يملك القيمة 10 ولو قمت بطباعة المتغير لظهرت لك ..

CODE:

```
<?php
	$var = strlen('moffed.com');
	echo $var;
?>
```

بعض الدوال كما قلنا ترسل فقط اما false او true وهي من دوال التحقق غالباً

مثل دالة **empty**

وهي دالة تقوم بالتأكد هل المتغير فارغ او لا.

هذه الدالة تقوم بإرجاع القيمة true لو كان فارغ او false لو كان غير فارغ.

واغلب هذه الدوال (اي التي تنتج قيم صح او خطأ) يتم استخدامها بالشروط ..

مثال :

CODE:

```
<?php
$var = '';
if(empty($var))
{
      echo 'المتغير فارغ';
}
?>
```

وفي المثال قمنا بالتأكد هل المتغير var فارغ او لا ..

لو كان فارغ قم بطباعه "المتغير فارغ"

جزء 16: دوال مخصصه

رغم ان الدوال التي توفرها php لك كثيره وعامه , الا انك تحتاج احيانا ان تصنع داله خاصه بك لتسهيل عملك وبرنامجك ..

كما قلنا بالجزه السابق عن الدوال , الا اننا هنا الآن سوف نتعرف على كيفية تعريف دالتنا الجديده ..

CODE:

```
<?php
function foo()
{

}
?>
```

انشاء الدالة يختلف عن جلبها .. فإنشاء الداله يتطلب وضع كلمة function قبل اسم الداله ومن ثم الاقواس وبعد الاقواس يتم وضع الاقواس المعكوفه {...} وداخل هذه الاقواس المعكوفه يتم وضع شفرتك ..

اسم الداله يبدأ بإما حرف او شرطه سفليه _ يتبع ذلك اي حرف او رقم او شرطه سفليه ...
اسم الداله يجب ان لايكون مستخدم من قبل الدوال العامه , مثلا دالة strlen عبارة عن دالة عامه لذا لايمكنني استخدام هذا الاسم...

للعلم فقط : يمكن تضمين دوال داخل دوال ...

فلنعطي مثال اوضح ..

فلنقم بإنشاء داله تقوم بإرجاع true اذا كان الرقم اقل من 10 وترجع false اذا كان اكبر ..

CODE:

```
<?php
function foo($number)
{
	if($number < 10)
	{
		return true;
	}
	else if ($number > 10)
	{
		return false;
	}
}
?>
```

لاحظ ان الدخل "parameter" هنا هو المتغير number وهو الذي بين الاقواس..

بعدها قمنا بعمل الكود بناءاً على المدخل.

ايضا return يقوم بإنهاء الداله وإرجاع قيمة منها .

الآن لننشأ كود يعتمد على دالتنا المخصصه foo

CODE:

```
<?php
if(foo(4) == true)
{
    echo 'لقد تم ارجاع قيمة true';
}
?>
```

أو هكذا :

CODE:

```
<?php
if(foo(4))
{
	echo 'لقد تم ارجاع قيمة true';
}
?>
```

هلا لاحظت ان عندما تكون القيمة المرجعه true فاننا لايلزم ان نضع " == true"

لكن هذا مهم مع القيمة false

CODE:

```
<?php
if(foo(11) == false)
{
	echo 'لقد تم ارجاع قيمة false';
}
?>
```

يمكنك انشاء دوال عديده مع عدة ادخلات , جرب هذه الدالة المخصصه

CODE:

```
<?php
function plus($a, $b)
{
	return $a + $b;
}
 ?>
```

انتهى هذا الجزء ..

جزء 17: النهايه ...

قمنا بتعلم ال php على مدى 16 جزء .. لم نتعلم كل شيء , حتى الآن كل هذا مقدمه بسيطه وبسيطه جداً , , ,

كل ماقرأته هو مايجب أن تعرفه قبل ان تبدأ في قراءه شفره أي برنامج بسيط ..

الآن يجب ان تقوم بأشياء لترسيخ مبدأ انك مطور , ايضا لتعرف امور اكثر في php
وايضا لتعرف اكثر الدوال العامه ولتعرف كيف يقوم المطورين الآخرين بالعمل و بكتابة التعليقات كما عرفنا ..

قد ازيد الكتاب لاحقاً كيف ننشأ سكربت بسيط . مثلا سجل زوار , سكربت اخبار بسيط , اي شيء عام لتتعلم المبادئ وايضا لتعرف كيف نستخدم قواعد البيانات ايضاً. لكن لاتنتظر قم الان بالبحث عن مالا تعرف لتعرف ! .

ايضا يجب أن تقوم بقراءة سكربتات بسيطة المصدر, ابحث سكربتات php في محرك البحث قوقل.

- لاتنسى زيارة مفيد.كوم كل فتره لترى دروس php و مايتعلق فيه .. وللسؤال
- ايَضاً لاتنسى زيارة مدونتي, اي ديف http://idev.me/

كتب في صيف 2009

تم تنقيحه في ربيع 2010

تم تنقيحه ايضاً في نهاية 2012

ملاحظة :

لم يتم تجاهل المصادر , ولكن لم يتم كتابتها هنا في الكتيب للتخفيف

لكنها موجوده في المواضيع في مفيد.كوم وجميع الحقوق الخاصه بالمؤلفين والناشرين كموقع php.net محفوظة ..

لذا وجب التنبيه