

ESSENTIAL IN ARABIC PHP YASSINE ABOUHRIF

مقدمة:

الحمد لله وحده والصلاة والسلام على من لا نبي بعده، محمد وعلى اله وصحبه اجمعين وبعد:
فهذا كتاب حول لغة البرمجة بلغة php في جزئه الاول، والذي تضمن اساسيات هاته اللغة
باسلوب سلس، وباختصار شديد لاهم المصطلحات والتعاريف،
وهذا العمل المتمثل في مجموعة من الدروس الخاصة بلغة البرمجة بي اتش بي ، موجهة
للمبتدئين الذين لديهم خلفية مع لغة البرمجة html وتقنية css ، وهو مجاني النشر.

فهرس:

١.....	مقدمة
٣.....	الدرس الاول: مدخل الى php
٧.....	الدرس الثاني: تهيئ الكمبيوتر واجاز الصفحة الاولى في بي اتش بي
١٤.....	الدرس الثالث: المتغيرات تعاريف وملاحظات - دورة أكاديمية تعلم
١٧.....	الدرس الرابع: الشروط
٢٢.....	الدرس الخامس: الحلقات
٢٤.....	الدرس السادس: الدوال
٢٨.....	الدرس السابع: المصفوفات
٣٤.....	خاتمة:

مدخل الى لغة PHP

لقد كانت سابقا عملية انشاء موقع تقتصر على تضمينه بالامور الاساسية . يعني مثلا ٤ صور و ٥ نصوص هـووب وهـاهو الموقع جاهز بسهولة تامة.

لكن مع مرور الزمن اختلفت الامور وتطورت ولم يعد مبرمجوا المواقع يكتفون بالامور الاساسية. بقدر ما اصبحوا يفكرون اكثر في خلق (الحياة) في مواقعهم. من هنا ظهرت لغة برمجة المواقع الجديدة php

وقبل ان نتحدث عن لغة php وجب ان نعرف انواع مواقع الانترنت الموجودة. ونعرف ايضا كيف يتم الاتصال بهاته المواقع عبر النت

انواع مواقع الانترنت:

هناك نوعين من المواقع حاليا:

- مواقع ثابتة:

وهي مواقع تعتمد في الغالب في برمجتها على لغة HTML وتقنية CSS وتتميز بكونها سهلة لقصور وظيفته التي تكمن في عرض محتوى ما فقط (صور. نصوص) . وكما ان تغيير محتوى هاته المواقع يتطلب عناءا ومشقة اكبر حيث يطر مبرمجوا هاته المواقع الى اعادة فتح كود صفحات الموقع والتعديل فيه

ثم بعد ذلك رفع ملفات الموقع من جديد بعد الغاء الملفات السابقة طبعا.

- مواقع ديناميكية:

مميزات عديدة وخصائص تحكم اكثر فاعلية وقوة . وتعتمد هاته المواقع اضافة على HTML و CSS على لغات برمجة اخرى مثل PHP و MYSQL مما يجعلها متطورة حيث تسمح بخلق تواصل بين الزائر و الموقع . وايضا تسهيل عملية تغيير او اضافة او حذف محتوى.

الاتصال بموقع عبر الانترنت:

عادة حينما نريد زيارة موقع معين موجودة على الشبكة العنكبوتية . فاننا نفتح المتصفح الموجود لدينا . كان انترنت إكس بلوري . فايرفوكس . سفري . اوبرا او اي متصفح اخر . ثم نقوم بكتابة عنوان الموقع . فيظهر الموقع إن كان موجودا .

لكن لم يسبق لنا ان سألنا انفسنا كيف يظهر لنا هذا الموقع؟؟!!..

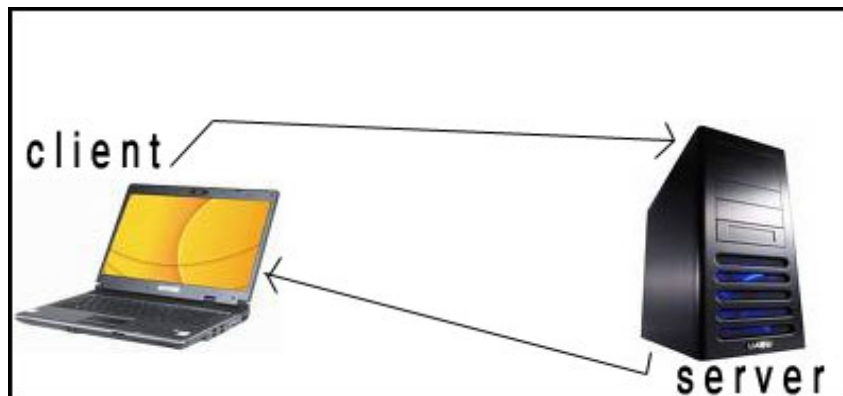
ان عملية الاتصال هاته اعتمدت بالاساس على الانترنت . حيث ان هاته الاخيرة عبارة عن شبكة من اجهزة الكمبيوتر تنقسم الى قسمين:

- اجهزة كمبيوتر خاصة بالزوار . يعني جهازك انت الذي سيمكنك من الاتصال بالموقع الذي تريده .

- واجهزة كمبيوتر اخرى قوية عبارة عن خوادم (سيرفرات) تقوم بعملية تخزين ونشر المواقع .

وعليه فعملية الاتصال بموقع على النت . تقوم على اتصال كمبيوتر خاص بزائر . بكمبيوتر خاص بخادم .

صورة توضيحية بسيطة:



(= السؤال الذي يطرح نفسه الان هو كيف يتم التواصل بين الزائر حين يطلب موقع على النت والخادم؟؟؟..

إن الامر بسيط لكنه يعتمد على نوع الموقع الذي يريد الزائر الاتصال به.
- فإذا كان يريد الاتصال بموقع ثابت: فالعملية سهلة
الزائر يطلب من السيرفر رؤية صفحة موقع, والسيرفر يقوم بالبحث عن صفحة الموقع المطلوبة وتسليمها الى الزائر.

- أما إذا كان الزائر يريد الاتصال بموقع ديناميكي : فالزائر يطلب من السيرفر رؤية صفحة موقع, والسيرفر يقوم بالبحث عن صفحة الموقع المطلوبة. ثم يقوم باعدادها قبل تسليمها للزائر, وعملية الاعداد هاته التي تكون قبل التسليم من طرف الخادم (السيرفر) هي راجعة الى كون المتصفحات التي نستعمل لاتستطيع ترجمة اكواد هاته المواقع . لانها مبرمجة بلغات المواقع الديناميكية مثل php . ومنه فالامر يحتاج الى معرف يعرف للمتصفح اكواد صفحات المواقع الديناميكية => الخادم

- لغة PHP :

الان حديثنا عن لغة PHP سيكون مفيدا.
وهي لغة برمجية مفسرة تستخدم في تطبيقات الويب, وتعمل على جانب الخادم وليس على جانب المستخدم اي هي خاصة بالمواقع الديناميكية.

ونظرا لتميزها بالعديد من الخصائص كسهولة الاستخدام وتدعيم الاتصال بمجموعة كبيرة من قواعد البيانات. وايضا قوة الفهم والتكامل في العديد من الخوادم, فهذا جعل من php تنال شهرة كبيرة في مواقع الويب المبرمجة.

وقد كان اول ظهور للـ php في سنة ١٩٩٤ من قبل الدنماركي راسموس لردورف Rasmus Lerdorf الذي اسماها بـ Personal Home Page الصفحات الشخصية واقتبس معظم اكوادها من لغة C و Java و Perl ومنذ ذلك الوقت و PHP تعرف تطورا وتوسعا اكثر الى يومنا هذا حيث وصل عدد نسخ بي اتش بي الى النسخة ٥ والتي اقل مايقال عنها انها اعطت مفهوما اخر في عالم تطبيقات الويب يتسم بالقوة.



كان هاته مجرد مقدمة للدخول في عالم بي اتش بي . ولو انها تبدو ملة بعض الشيء...
وبين بينكم انا اكره مثل هاته المقدمات ولكن في كل مرة اضطر لكتابتها احاول قدر الامكان الاختزال. فاقصر على الاشياء التي ارها مهمة.

تهيئ الكمبيوتر وانجاز الصفحة الاولى في بي

اتش بي

سبق لنا في درسنا السابق ان اشرنا الا ان بي اتش بي يتعامل مع الخادم (السيرفر) الذي يقوم بترجمة اكواده للمتصفح قبل عرضها. وعليه فإن اجهزتنا تحتاج الى الاتصال بـ سيرفر حتى نستطيع عمل صفحات من بي اتش بي.

إن كيف نحصل على سيرفر؟؟

سؤال وجيه..

طيب ممكن يكون الجواب : ان نبحت عن مصباح علاء الدين. ثم نسمح فوفه فيظهر لنا العفريت .

العفريت: « شوبيك لوبيك انا بين إديك . اطلب ماشئت »
صاحب مصباح علاء الدين: « اريد > سيرفر < حتى استطيع ان اشتغل على لغة بي اتش بي »
العفريت: « مهم تريد > سيرفر < .. شوف يااخي انا اقترح عليك ان تطلب ثلاجة او غسالة احسن بكثير من >السيروف < هذا . قلت لي مااسمه؟؟ »

إن المواقع المبرمجة سواء كانت مواقع ثابتة او مواقع ديناميكية (راجع الدرس السابق). تحتاج الى برامج مختلفة للتعامل معها. فإذا كان الموقع ثابتا فنحن بحاجة الى :

- محرر اكواد : وهو برنامج يمكننا من كتابة الاكواد الخاصة بموقعنا كـ Notepad او Notepad++
- متصفح انترنت: وهو يسمح لنا برؤية صفحات النت. ويمكننا استخدام المتصفح انترنت اكسبلوري او موزيلا او اي متصفح اخر.

وهذا كل مانحتاجه بالنسبة للمواقع الثابتة. يأتي هذا خلفا للمواقع الديناميكية والتي مثلا تعتمد على لغة بي اتش بي في برمجتها فإننا نحتاج إضافة الى البرامج السابقة الى سيرفر . هذا الاخير يكون إما سيرفرا محليا او ما يطلق عليه بـ Localhost : اي اننا نحول اجهزة الكمبيوتر الخاصة بنا الى سيرفر وهنا لانحتاج الى الانترنت للاتصال به.

او يكون عبارة عن سيرفر غير محلي اي مستقل عن اجهزتنا ويتم الاتصال به عن طريق الانترنت.

وفي دروسنا الاولى هنا سنتعامل مع السيرفر المحلي فقط ونقوم بتنصيب البرامج التي يوفرها لنا السيرفر والتي ستقوم بقراءة اكواد بي اتش بي ومن ثم ترجمتها للمتصفح.

تنصيب السيرفر:

هناك العديد من الحزم التي توفر لنا سيرفر محلي . لكن يفضل اختيار برنامج عمل سيرفر محلي يجمع البرامج التالية :

- Apache : وهو اهم برنامج حيث يعتبر هو المسؤول عن إظهار صفحات الانترنت في المتصفح للمشاهد او الزائر

- PHP : بي اتش بي هذا عبارة عن plug-in يحتاجه Apache ليعرف له اكواد لغة بي اتش بيحتى يظهرها

- MySQL : برنامج او نظام إدارة قواعد البيانات و به نستطيع حفظ البيانات التي نريد

وسوف نستعمل برنامج WampServer

لسهولة استخدامها والخواص التي يوفرها لنا زائد الاهم من ذلك انه يجمع البرامج التي تحدثنا عنها. طبعا يمكنكم استخدام اي برنامج اخر لعمل سيرفر محلي كـ appserv وغيره...

نبدأ بتحميل برنامج WAMP من خلال هذا الرابط

http://www.net.com/telecharger/windows/Internet/editeur_de_site/fiches.html.٢٨٧٣٩/٠١

بعد انتهى التحميل نقوم بتنصيب السيرفر

وعملية التنصيب اظن انها واضحة ولاداعي للحديث حولها الا في حالة طبعاً ان واجهتكم مشكلة او كان لديكم استفسار حول امر معين اثناء التنصيب

فيمكنكم طرحه في احدى المواقع المشهورة في تقديم المساعدة، وستجدونه ان شاء الله.

عند الانتهاء من عميلة التنصيب سوف تظهر لنا ايقونة البرنامج الجديد نضغط فوقها فيتم تشغيل السيرفر

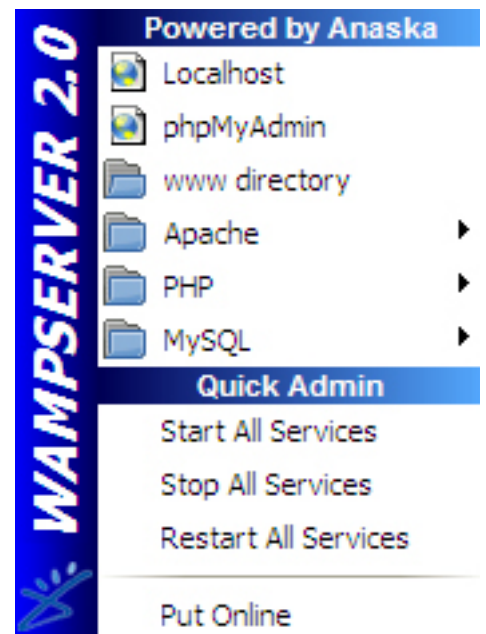
فتظهر الايقونة في اللوحة السفلية من سطح المكتب taskbar



فاذا قمنا بالضغط فوقها بالماوس نكليك يمين فنجد مجموعة من الخيارات من بينها خيار اختيار اللغة حيث يطرح لنا البرنامج مجموعة من اللغات من بينها اللغة العربية

لكن يستحسن اختيار اللغة الانجليزية حتى نعتاد على اللغة، ايضا من الخيارات الموجودة خيار Exit لالغاء عمل السيرفر.

و في حالة قمنا بالضغط على ايقونة WampServer بالماوس نكليك يسار فنجد ايضا مجموعة من الخيارات:



- localhost: وهي صفحة البداية الخاصة بالسيرفر المحلي الذي نستعمله، وتحتوي على معلومات عن السيرفر والتطبيقات المحفوظة فيه واداة الولوج لنظام التحكم.

رابط الصفحة هو :

http://localhost

أو نقوم بادخال عنوان الايبي IP

http://127.0.0.1

صورة توضيحية:

Version 2.0 English Version

Configuration Serveur

معلومات عن اعدادات السيرفر

Version de Apache: 2.2.11
Version de PHP: 5.3.0
Version de MySQL: 5.1.36

Extensions Chargées:

Core	bcmath	calendar	com_dotnet	ctype	date
ereg	filter	ftp	hash	iconv	json
mcrypt	mysqlnd	odbc	pcre	Reflection	session
SPL	standard	tokenizer	zip	zlib	libxml
dom	PDO	Phar	SimpleXML	wddx	xml
xmlreader	xmlwriter	apache2handler	gd	mbstring	mysql
mysqli	pdo_mysql	pdo_sqlite	mhash		

Outils

- phpinfo() معلومات بي أنش بي
- phpmyadmin لوحة التحكم والتي تسمح لنا بانشاء قواعد البيانات

Vos Projets

- wjeez ملفات التطبيقات او المشاريع التي نشغل عليها
- wordpress

Vos Alias

- phpmyadmin

WampServer - Donate - Anaska

- phpMyAdmin: وهي نفس الصفحة التي دخلنا اليها من صفحة البداية وقلنا عنها انها عبارة عن لوحة تحكم وفيها يمكننا ان ننشئ قواعد بيانات



هاته مجرد نظرة عامة وسريعة على الصفحة ومستقبلا سنتعرف ان شاء الله على كيفية انشاء قواعد البيانات.

- الخيار الثالث هو عبارة عن ملف C:\wamp\www

نقوم بوضع تطبيقاتنا فيه حتى نجعلها تنتمي الى السيرفر. يعني اي تطبيق سنقوم به ونخزنه بصيغة بي اتش بي سيتم وضعه داخل هذا الملف.

اظن نكتفي بهاته الخيارات لانها المهمة عندنا حاليا.

[صفحتي الاولى بي اتش بي:](#)

نعلم جميعا ان ملفات html او xhtml عبارة عن وسوم خاصة بهاته اللغة ويعرفها برنامج متصفح الانترنت. ايضا لغة بي اتش بي php لها وسوم خاصة بها. لكن ما سنضيفه هو ان ملفات php يمكن ان تحتوي على وسوم html وهذا مايعطينا امكانية قوية في التحكم والعرض.

نأخذ تطبيق.

تطبيق:

صفحة html

```
<html dir=rtl>
<head>
<meta http-equiv=»Content-Type» content=»text/html; charset=utf8-» />
<title> صفحة اتش إم إل </title>
</head>
<body>

السلام عليكم ورحمة الله وبركاته
<br />
ارجوا ان تكونوا قد استفذتم من دروسنا المتواضعة حول لغة بي اتش بي

</body>
</html>
```

طبعا الكل يعرف ويفهم ماكتب في التطبيق ١
حيث سيتم بعد تسجيل الصفحة بصيغة html
عرض:

السلام عليكم ورحمة الله وبركاته
ارجوا ان تكونوا قد استفذتم من دروسنا المتواضعة حول لغة بي اتش بي.

التطبيق ٢: في هذا التطبيق سوف نتعرف على كيفية كتابة صفحة التطبيق ١ ب بي اتش بي

```
<html dir=rtl>
<head>
<meta http-equiv=»Content-Type» content=»text/html; charset=utf8-» />
<title> صفحتي الاولى ب بي اتش بي </title>
</head>
<body>

<?php
print «<br />. السلام عليكم ورحمة الله وبركاته» ;
print « ارجوا ان تكونوا قد استفذتم من دروسنا المتواضعة حول لغة بي اتش بي » ;
?>

</body>
</html>
```

ثم نسجل التطبيق باسم معين + php.
ولاننسى ان نضعه في مجلد معين نسميه مثلا page
ثم نضع بعد ذلك المجلد في الملف C:\wamp\www
الآن لرؤية التطبيق ٢ نفتح صفحة localhost
وسنجد في المكان المخصص للمشاريع الموجودة اسم المجلد الذي انشئنا فنضغط فوقه ونختار اسم صفحتنا
فتفتح الصفحة ان شاء الله بدون اي مشاكل مظهره لنا:
السلام عليكم ورحمة الله وبركاته
ارجوا ان تكونوا قد استفذتم من دروسنا المتواضعة حول لغة بي اتش بي.

ولاحظ معي ان الامر print هو احد اوامر بي اتش بي وهو المسؤول عن اظهار محتوى معين يمكن تعويضه ب echo لامشكلة.
وعليه فمن التطبيق نستخلص الصيغة العامة لملف php وهي:

```
<?php
```

```
اوامر بي اتش بي التي نريد العمل بها
```

```
?>
```

المتغيرات

مازلنا نذكر اخر شيء في درسا السابق الذي حدثنا فيه عن الصيغة العامة لكتابة لغة بي اتش بي كما استعملنا احد الاوامر التي تمكننا من اظهار كلام او شيء ما: الامر echo الان سنتعرف على اوامر واشياء اخرى سنحاول ترتيبها حتى يسهل استيعابها من طرف الجميع ان شاء الله

التعليقات او مايسمى بـ COMMENTAIRES

حسنا , نفترض لدينا كود برمجي بلغة بي اتش بي او اي لغة برمجة اخرى وارادنا التعديل فيه . في الواقع سيصعب علينا ذلك في حالة عدم وجود تعليقات وارشادات داخل هذا الكود لاننا سنحير ونحن نبحث عن السطر المراد تعديله في هذا الكود. لكن إذا وجدت تعليقات وارشادات بداخل هذا الكود فسيسهل الامر علينا قليلا إذ اننا سنكتفي بقراءة التعليقات المختصرة وهذا افضل من قراءة اسطر طويلة من كود معين . ومن هنا نعرف التعليق في لغتنا المدروسة على انه عبارة عن نص نكتب فيه شيئا له دلالة على اسطر من كود معين حتى يسهل الرجوع اليها. وهذا النص لا يتم ترجمته اي بصيغة اخرى لن يظهر الا عند فتح الكود اي source الصفحة.

ولكتابة تعليق في بي اتش بي نستعمل هذا الرمز
// نكتب التعليق المراد بعد هذا الرمز شريطة الا يتجاوز التعليق سطرا واحدا فقط.
او نستعمل هذا الرمز في حالة كان لدينا تعليق طويل يتكون من عدة اسطر . نشرح فيها امر ما مثلا بالتفصيل:
*/

التعليق المراد

اكتب التعليق الذي تريده بعدد الاسطر التي تريد
ولاتنسى ان تغلق الرمز دلالة على انتهاء التعليق هكذا
/*

مثال بسيط

```
< ?php
// Name of my father
echo «Elhaj Addi <br />» ;
/*
The next line refers to the name of my grandmother.
Of note: My grandmother from my father
*/
echo «elhaja Mahjouba om Elhaj Addi»
?>
```

سنحصل على التالي:

Elhaj Addi
elhaja Mahjouba om Elhaj Addi

نحن عائلة حجاج ما شاء الله ههههه
ماعلينا , كما لاحظتم ان التعليقات لم تظهر معنا وكانت فائدتها كما قلنا هي للدلالة على كود معين. وهي جدا مفيدة فلا يعتبرها احدكم انها غير ضرورية. بالعكس ستجعل صفحة كودك التي تبرمج اكثر تنظيما ويستطيع ان يقرئها زملاؤك بسهولة.

المتغيرات او VARIABLES

تعريف:

الحديث عن المتغيرات يطول جدا ولكن ماينبغي ان نعلمه هو انه وفي اي لغة برمجة قوية لايمكنك ان تقطع اشوطا طويلة بدون مايسمى <المتغيرات>

فهى عنصر اساسى جدا فى اى لغة.
حيث تكمننا المتغيرات من تخزين معلومة او معلومات فى الذاكرة بشكل مؤقت نستطيع التحكم فيها.
ويتكون المتغير فى بي اتش بي من عنصرين اساسيين وهما:
- الاسم : وهو يدل على اسم معين نختاره نحن لـ <متغير> نريد استعماله.
- والقيمة: وهى المعلومة التى نريد ان يحملها <المتغير> والتى تتحول.
ولايجب ان ننسى ان المتغيرات التى ننشئ يجب ان تبتدأ بعلامة الدولار هاته: \$
وكمسلمة : اى اسم يبتدأ بعلامة الدولار \$ هو عبارة عن متغير.

امثلة لمتغيرات:

```
$name=>yassine»;  
$age=۲۱;  
$message=>this forum is the best»
```

أنواع المتغيرات :

هى ببساطة انواع البيانات التى نريد تخزينها ويمكن تصنيفها حسب :
- string : هذا النوع من المتغيرات خاص بالنصوص والقيمة الخاصة بهذا النوع تكون دائمة بين علامتي تنصيص مزدوجة او فردية
- integer: int: اي الأرقام الطبيعية
- float: او - double وهى الأرقام التى تحتوى على الفاصلة
- boolean: bool: هو متغير منطقي لا يَحتمل اكثر من قيمتين: true or false اي اما ان يكون صحيح او يكون خاطئا وهذا المتغير مهم للتأكد من صحة المعلومات.
- Null: قيمة فارغة يعنى عندما ننشئ متغير ولا نعطيه اى قيمة فيكون هذا المتغير فارغا
- array: المصفوفات وسوف يأتي الحديث عنها فى الدروس القادمة ان شاء الله بالتفصيل

مثال بسيط:

```
<?php  
$name= »yalsne <br />»;  
$nickname= »king of the programming»;  
echo $name;  
echo $nickname;  
?>
```

سنحصل على :

```
yalsne  
king of the programming
```

ملاحظات مهمة:

الملاحظات التى سوف ندرج تتعلق بكتابة المتغيرات. حيث ينبغي مراعاة بعض الامور وهى كالتالى:
١- جميع اسماء المتغيرات لايجب ان تبتدأ برقم بعد علامة الدولار مباشرة . مسموح يحتوى الاسم على رقم لكن الغير مسموح به هو ان يكون هذا الرقم هو الاول فى الاسم :

```
۳$amal= «i don>t now.. any work for example: <policeman> lol»
```

خطأ فى اسم المتغير ۳amal لانه يبتدأ برقم

٢- اسماء المتغيرات حساسة لوضعية الحروف :
المتغير \$name ليس هو المتغير \$Name

٣- لاينبغي ان نبدأ بعلامة تنصيص منفردة فى تعريف قيمة متغير حرفي ونغلق عند الانتهاء بعلامة تنصيص مزدوجة او العكس:

```
$name= >Acadamy the php»;
```

خطأ. الصحيح هو ان علامة التنصيص التى بدأت بها يجب ان تنهى بها:

```
$name= »Acadamy the php»;
```

٤- امكانية استخدام العلامة : _ علامة underscore في بداية او داخل اسم المتغير:

```
$_year=٢٠١٠;  
$programming_language=»php»;
```

٥- يمكننا جمع عدة متغيرات في متغير واحد مثال:

```
<?php  
$a="programming language ";  
$b="php";  
$c= ٥;  
$d=$a.$b.$c  
echo $d;  
?>
```

ونلاحظ اننا جمعنا المتغيرات الثلاثة في متغير واحد a , b , c وكنا نضع نقطة بين متغيرين وليس علامة + بحكم اننا نجمع بين متغيرات اعتبرناها كلها نصوص string . فالمرجوا الانتباه لهاته الملاحظة المهمة جدا.

اظن هاته هي الملاحظات الاساسية باقى الملاحظات سوف نتعرف عليها مستقبلا ان شاء الله حين نتعرف ونتعمق اكثر في اللغة.

[نمر الان الى العمليات الحسابية الاساسية في لغة الـ بي اتش بي](#)

الضرب : *

القسمة: /

الطرح: -

الجمع: +

وباقي القسمة: %

نأخذ مثال لكيفية استخدام هاته العمليات

```
<?php  
$Number١ // الرقم الاول ;٧=١  
$Number٢ //الرقم الثاني ;٥=٢  
  
$sum=$Number١ + $Number٢ //متغير عملية الجمع  
$subtraction=$Number١ - $Number٢ //متغير عملية الطرح  
$multiplication=$Number١ * $Number٢ //متغير عملية الجداء  
$division=$Number١ / $Number٢ //متغير عملية القسمة  
$rest_division=$Number١ % $Number٢ //متغير عملية باقي القسمة  
  
echo $sum .»<br />»;  
echo $subtraction .»<br />»;  
echo $multiplication .»<br />»;  
echo $division .»<br />»;  
echo $rest_division .»<br />»;  
?>
```

ماسنحصل عليه سيكون هو:

١٢

٢

٣٥

١,٤

٢

الشروط

في كثير من المرات في صفحات الويب التي ننجز لنجأ الى ربط اشياء متعددة باوامر معينة . يعني نقول للمتصفح: «إذا حدث هذا الشيء قم بالامر التالي (ونع الامر الذي نريده)»
على سبيل المثال نريد في صفحة من صفحات مواقعنا إذا كان اليوم جمعة تظهر جملة: جمعة مباركة ان شاء الله. وإذا كان غير يوم الجمعة لا يظهر شيء.
وهذا هو معنى الشرط في بي اتش بي ببساطة . اي نعطي امرا يكون مربوطا بشرط معين . إذا حقق فالامر يتحقق وإذا لم يتحقق فالامر لا يتحقق.

إذا فما هي الصيغة العامة لكتابة شرط في بي اتش بي؟؟

الصيغة العامة للشرط :

تختلف كتابة الشرط بحسب شكل هذا الشرط.

فممکن ان يكون شكل الشرط مجرد شرط واحد نربطه بامر معين وانتهى . ويمكن ان يكون شرط واحد نربطه بامرين . الاول إذا تحقق. والثاني إذا لم يتحقق. يعني نقول إذا حقق الشرط كذا وكذا ففعل كذا وكذا . وإذا لم يتحقق ففعل كذا..
وهناك اشكال اخرى سوف نتعرض اليها ونعطي صيغة كتابة كل منها فالمرجوا الانتباه في الامر في غاية الهمية.

الصيغة IF

هي الدالة التي تمكننا من وضع الشرط وتعني «إذا» وهي اول شيء نبدأ به في وضع الشرط فنكتب هكذا:

```
if(نكتب الشرط الذي نريده)
{
هنا نكتب الامر الذي نريد تحقيقه في حالة كان الشرط صحيحا
}
```

مثال:

```
<?php
$length=٧٥;
if ($length<=٧٨)
{
echo "proper length";
}
?>
```

كما نلاحظ جميعا بانه وضعنا شرط وقلنا فيه إذا كان المتغير length اصغر من او يساوي ٧٨ فاطبع الجملة : proper length

وبما انا الشرط متحقق فعلا حيث length يساوي ٧٥ و ٧٥ اصغر من ٧٨ التي وضعنا في الشرط فستظهر الجملة المرتبط بهذا الشرط: proper length

وكملاحظة اخرى بالنسبة للرموز التي نضع داخل الشرط وهي:

== : وهذا الرمز ليس كالرمز "=" واحدة . هنا "==" مرتين . ونعني بهذا "إذا وافق الشرط كذا..."
>

الرمز اصغر قطعاً من ..

<

الرمز اكبر قطعاً من ...

>=

اصغر من او يساوي..

<=

اكبر من او يساوي..

!=:

نفي يوافق اي لا يوافق
لدى يرجى الانتباه في وضع الرموز ومعانيها.

الصيغة ELSE ... IF

هنا نأتي لشكل اخر من اشكال كتابة الشرط . ففي الصيغة السابقة ربطنا شرط واحد بامر واحد.

لكن ماذا لو لم يتحقق الشرط في المثال السابق ؟

الذي كان سيقع ببساطة انه لن يظهر اي شيء وفي هاته الحالة لن يفهم اي شيء لانه اصلا لن يعرف الشخص الزائر هل كان هنالك شرط في الاساس ام لا.

وعليه ففي هذا الشكل سنرى كيف نضيف امر ثاني نربطه في حالة لم يتحقق الشرط.

وطريقة كتابة هاته الصيغة هي:

```
if(الشرط)
{
الامر المرتبط بتحقق الشرط
}
else
{
الامر المرتبط بعدم تحقق الشرط
}
```

مثال: ولناخذ نفس المثال السابق فقط سنعدل فيه قليلا

```
<?php

$length=79;
if ($length<=78)
{
echo "proper length";
}

else
{
echo "length is not suitable";
}

?>
```

اكيد نعرف ماذا سيعطي هذا المثال. سنحصل على الجملة : length is not suitable لان الشرط لم يتحقق . وعليه فالامر الذي سيحصل هو الامر المرتبط بعدم تحقق الشرط

تذكير: المرجوا تطبيق هاته الامور واعادة قراءة ماجاء في الدرس مع محاولة تغير الارقام التي جاءت في الامثلة حتى يتسنى لكم اكتشاف امور اخرى .

جيد لقد الان. نبقى مع نفس المثال الاخير ونلاحظ اننا ربطنا شرط واحد بامرين. الاول في حالة تحقق الشرط . والثاني في حالة عدم تحقق الشرط.

لكن ماذا لو اردنا وضع شرط ثاني في حالة عدم تحقق الشرط الاول ونربط بالامر الثاني.

يعني نقول: إذا تحقق الشرط الاول فاننا نأمر بالامر كذا وكذا. وإذا لم يتحقق وكان كذا وكذا.. (لاحظ هنا الشرط الثاني) فافعل الامر كذا وكذا..

اعرف ان الامر قد يبدو مستعصي لدى البعض لذلك سنأخذ مثال وان شاء الله تتوضح الامور للجميع

المثال:

```

<?php
$length=٧٩;
if ($length<=٧٨)
{
echo «proper length»;
}
elseif($length==٧٩)
{
echo «length is not suitable»;
}
else
{
echo «large length»;
}
?>

```

سنحصل على : length is not suitable
وكما نلاحظ فقد استخدمنا الصيغة elseif التي فرضنا فيها شرط ثاني في حالة لم يتحقق الشرط الاول وبما ان الشرط الاول لم يتحقق والشرط الثاني تحقق فان الجملة المرتبطة بالشرط الثاني هي التي ستحصل.

شكل شروط متعددة:

يمكننا ان نضع اكثر من شرط في امر واحد لكن باستعمال رموز جديدة وهي:

AND : وتعني "و" ولها رمز اخر وهو &&

OR : وتعني "او" لها ايضا رمز اخر ||

نأخذ مثال لهذا الشكل حتى نفهم اكثر:

```

<?php
$name="mohamed";
$age= ٩٩ ;
if($name=="mohamed" && $age==٩٩)
{
echo "true";
}
else
{
echo "false"
}
?>

```

وهنا بما ان كلا الشرطين متحققين فان الامر المرتبط بهما سوف يتحقق اي ماسنحصل عليه هنا هو الكلمة : true
مثال ٢:

```

<?php
$name="mohamed";
$age= ٩٨ ;
if($name=="mohamed" || $age==٩٩)
{
echo "true";
}
else
{
echo "false";
}
?>

```

سنحصل على true كذلك لان هنا اشترطنا تحقق احد الشرطين وبما ان الشرط الاول قد تحقق فذلك كافي لظهور الامر المرتبط بالشرط..
اكيد تتعد اشكال كتابة الشروط فيمكننا ان نعقد اكثر من هذا ونضع شرط داخل شرط داخل شروط مربوطة باوامر وما الى ذلك .
لست ادري كيف يفكر كل منا وفي نوع الشروط التي يريد ان يضع ولكن هذا هو الشيء الاساسي في كتابة الشروط وهذا ما يجب ان يفهم.

صيغة اخرى لوضع الشروط : الامر SWITCH

يعد الامر او الدالة switch مثل الامر if و else لكن هذا الامر اكثر مرونة .
وسنأخذ مثال لطريقة استخدامه.

المثال:

```

<?php
$age = ٩٩;
switch ($age) // نحدد المتغير الذي نريد ان نتعامل معه
{
case ١٠ : // هنا نقول في حالة ما إذا كان المتغير المتعامل معه يساوي ١٠
echo "you are a child";
break;
case ٢٥ : // نضع الحالة الثانية اي في حالة ما إذا كان المتغير يساوي ٢٥
echo "you are man";
break;
case ٩٩ : // الحالة الثالثة ولاحظ معي ان هاته الحالة هي الصحيحة إذا الامر المرتبط بها هو الذي سيتحقق
echo "you are old man";
break;
case ١٣٠ : // وهكذا نضع حسب الحالات التي يمكن تواجها
echo "you are so old man";
break;
default:
echo "sorry, i don't have any sentence for your age";
}
?>

```

كما نلاحظ الامر بسيط حيث نقوم بوضع الاحتمالات التي يمكن توقع ونربط كل احتمال بامر معين.

وهنا الاحتمال الصحيح هو case ٩٩:

وعليه فالنتيجة كما هو متوقع ستكون: you are old man

الامر default:

هو الامر الذي يقع في حالة لم يتحقق اي احتمال يعني لو غيرن مثلا قيمة المتغير age ووضعنا قيمة غير موجودة في الاحتمالات مثلا :

;v=age\$

فالنتيجة ستكون :

sorry, i don't have any sentence for your age

اما بالنسبة للامر break;

فهو مهم جدا حيث تكمن وظيفته في الخروج من الدالة switch في حالة تحقق الاحتمال. دون ان يتابع قرأت باقي الاحتمالات .

ولنتحقق من الامر سوف نعدل في المثال الاخير ونضع في الاحتمال ١٣٠ الرقم ٩٩ كذلك

سوف يتحقق الاحتمال ٩٩ الاول ولن يتحقق الثاني وذلك بسبب الامر break الذي خرج من الدالة ولم يتابع القراءة.

لكن ماذا لو حذفنا الامر break المرتبط بالاحتمال ٩٩ الاول ؟؟

سوف يتم متابعة القراءة وسنحصل على الامر المرتبط بالاحتمال الاول ٩٩ وايضا الامر المرتبط بالاحتمال الثاني ٩٩.

الحلقات

LOOPS

تعريف:

لنفترض مثلا اننا نريد عمل صفحة بـ بي اتش بي، تظهر لنا الجملة التالية متكررة ١٠٠ مرة :

-السلام عليكم ايها الطالب
-السلام عليكم ايها الطالب
-السلام عليكم ايها الطالب
-السلام عليكم ايها الطالب
-السلام عليكم ايها الطالب

-السلام عليكم ايها الطالب
الجملة تتكرر ١٠٠ مرة .

طيب إذا اردنا ان نقوم بذلك بالطريقة التقليدية، يعني نطلب إظهار كل جملة على حدى، فسيكون الامر صعبا وشاقا، ان نكرر كتابة الجملة مائة مرة

لكن مع الحلقات او مايسمى بالانجليزية بـ loops نستطيع عمل ذلك بكل بساطة
فما هي الحلقات إذن؟؟

الحلقة او الحلقات ماهي الا صيغة تسمح لنا بعمل تكرار لمجموعة من التعليمات ، حتى نوفر الوقت وايضا نقلص من حجم الكود.
وقد تم تخصيص مجموعة من الدوال الخاصة لذلك ، سنكتفي نحن هنا بمعرفة دالتين هما:

while -

و

for -

ولنبدأ بكيفية عمل حلقة تكرار باستخدام الدالة WHILE السهلة.

جاء في تعريفنا للحلقات بانها صيغة لعمل تكرار لتعليمات معينة نحددها نحن، وهذا يعني ان هذا التكرار سيستمر حتى بعد الوصول الى اخر تعليمة ثم يبدأ من جديد التكرار بدون توقف حتى النهاية ثم يعود للبداية وهكذا .. (التكرار مستمر).

ولهذا فنحن بحاجة ماسة الى شرط يوقف هذا التكرار ويجعلنا نتحكم فيه

هذا الشرط يتم بواسط الدالة while

صيغة كتابتها:

while (الشرط)

```
{
... التعليمات
}
```

وسنطبقها الان على مثالنا السابق حيث ستكون كتابة المثال السابق هكذا:

```
<?php
$n = 1;
while ($n <= 100)
{
echo 'السلام عليكم ايها الطالب <br />';
$n++;
// $n = $n + 1
}
?>
```

إذا حاولنا تطبيق هذا الكود سنحصل على الجملة: "السلام عليكم ايها الطالب". مائة مرة وبدون اي عناء

وكشروح للمثال . فاول شيء بدأنا به هو وضع متغير

n

وإعطائه قيمة ١ وهي قيمة البداية التي نريد ان يبدأ منها التكرار. ثم بعد ذلك استعملنا دالتنا المدروسة وقلنا مادم (شروط) المتغير

n

اصغر من او يساوي ١٠٠

فالمطلوب هو إظهار الجملة: السلام عليكم ايها الطالب

وبما اننا نريد ان تتكرر هاته العملية فقد وضعنا ان المتغير

n

يساوي

نفسه زائد واحد

حتى نضمن ان قيمة المتغير في تصاعد اي تتغير

وهذا سيجعل الكومبيوتر يعود للقرءة من بداية الحلقة ولن يخرج من الحلقة الا بعد إختلال الشرط يعني حين تصبح قيمة المتغير

n

فوق ١٠٠

اظن ان المسألة اصبحت واضحة.

الحلقات باستخدام الدالة FOR

ليست صعبة كذلك وتشبه سابقتها . ولو اني شخصا افضل استخدامها على سابقتها. نظرا لكوني اقرأها بسرعة وارتاح لها. على العموم

كل وكيف يرتاح لمثل هاته الاستخدامات ولا ضير بتاتا في استخدام

while

او

for

مادامت النتيجة واحدة

ولنعرف كيف تكتب هاته الدالة

(for)

سنقوم باعادة كتابة نفس المثال السابق باستخدامها حتى نلاحظ الفرق بينها وبين

while

إن المثال باستخدام الدالة

for

سيكتب:

```
<?php
```

```
for ($n = 1; $n <= 100; $n++)
```

```
{
```

```
echo 'السلام عليكم ايها الطالب ';
```

```
}
```

```
?>
```

اظن وصلنا الى نهاية هذا الدرس . يجدر الاشارة الى انه هنالك دوال اخرى لعمل الحلقات لم نتطرق لها كونها لم تعد تستخدم واغلبية المبرمجين يستخدمون احد هاته الدالتين لاغير. يكفي ان نجيد التحكم في عمليات التكرار بشكل جيد حتى نحصل على النتيجة المرغوب فيها.

الدوال

أولا

ما هي الدوال في بي اتش بي؟

ببساطة الدوال هي مجموعة او سلسلة من التعليمات تطبق اجراءات معينة وترجع قيمة في الاخير. وهي شبيهة الى حد ما بحلقات التكرار حيث ان فائدتها ايضا تكمن في اختصار الكود البرمجي. وتختلف عن هاته الاخيرة بكونها - اي الدوال - تتكيف مع المعلومات التي نرسل لها. ولنفترض مثالا : نريد حساب مساحة ارضية فلاحية , ماسنحتاجه هو عبارة عن دالة بسيطة ترجع لنا قيمة ما نريد , حيث سنقوم باعطائها معلومات عن الارضية الفلاحية وهاته المعلومات هي: طول وعرض الارضية وهي ستقوم بعملية حسابية لتعطينا النتيجة في الاخير. فقط يجدر الاشارة الى ان هناك دوال موجودة سابقا في بي اتش بي تؤدي وظيفة معينة, وهناك دوال ننشئها بحسب الوظيفة التي نريد من دالتنا ان تقوم بها.

- الدوال المعرفة مسبقا في بي اتش بي:

هي مجموعة كبيرة جدا من الدوال ذات وظائف معينة جاهزة للاستخدام, وهي عنصر القوة في بي اتش. شخصا لم اطلع عليها بعد بالكامل لانها كثيرة كما قلت, لكنها تسد جميع حاجياتنا التي نريد ولاغنى لاي ميرمج محترف عنها.. يعني نستطيع ان ننشئ دوال ونعطيهها وظائف معينة لكن لن نصل الى مدى بعيد بدون مكتبة دوال بي اتش بي. وهذا رابط للموقع الرسمي للغة فيه جميع الدوال يمكن الرجوع اليه:

<http://fr.php.net/manual/en/funcref.php>

طبعاً لن يكفينا الوقت للمرور على كل الدوال الموجودة لكن سنكتفي بذكر بعضها ومستقبلاً ان شاء الله ستلاحظون اننا نستعمل دوال اخرى جديدة وهناك سنعرف بها.

مايجب ان نعلمه الان ان الوظائف التي يمكن ان تقوم بها هاته الدوال هي وظائف عديدة من بينها:

- ارسال ملفات الى السيرفر
- انشاء صور
- ارسال رسالة الى البريد الالكتروني
- الرسم والكتابة
- تعويض محتوى بمحتوى وكلمة بكلمة...
- تشفير ارقام سرية
- اظهار الوقت والتاريخ
- اعطاء معلومات عن الزائر
- مسح او الغاء اجراءات
- والعديد من الوظائف الاخرى...

ولعلنا سنتعرف اليوم على دوال جميلة جدا حيث ان وظيفتها تتجلى في التعامل مع النصوص ومع دالتنا الاولى:

STRLEN

بواسطة هاته الدالة نستطيع معرفة طول -واقصد هنا عدد- الحروف او الارقام المكتوبة في جملة او نص ما. مثال لطريقة استخدامها


```
<?php
$sentence= "yalsne is the king of PIA";
$length = strlen($sentence);

echo 'number of characters of the sentence \'' . $sentence . '\'' is: <br />' . $length;
?>
```

وستكون النتيجة هي
:number of characters of the sentence \'' yalsne is the king of PIA'\'' is
٢٥
ونبه الى انه تم حساب المسافة الفاصلة بين الكلمات ايضا

الدالة STR_REPLACE

واضح من اسمها انها تقوم بتعويض كلمة بكلمة او حرف بحرف..
مثال لطريقة عملها:

```
<?php
$sentence = str_replace('evening', 'morning', 'good evening sir');

echo $sentence;
?>
```

والنتيجة ستكون هي:
good morning sir
حيث اننا قمنا بوضع البارمتر الاول في الدالة وهو البارمتر الذي نريد ان نغير في الجملة
ثم البارمتر الثاني وهو البديل الذي نريد ان نعوض به البارمتر الاول. وطبعا البارمتر الثالث هو الجملة التي نريد ان نغير.

الدالة STR_SHUFFLE

دالة طريقة لعمل خلط وتغيير لاماكن الاحرف بشكل عشوائي . حيث كل مرة سنعمل فيها تحديث للصفحة سنحصل على تشكيل جديد
للحروف.
مثال:

```
<?php
$sentence١ = 'thank you, it is very good';
$sentence٢ = str_shuffle($sentence١);

echo $sentence٢;
?>
```

النتيجة ستكون متغير في كل تحديث لان الامر يتم بشكل عشوائي

```
eysoinki hooy avgrd ,ttu
yn urdikgy s otti,voehao
go,haenouitvk ritdos yy
ytan a tku s, ivgdoi hyreo
```

وظيفة هاتين الدالتين تتجلى في التحكم في حالة الحروف ، حيث ان الاولى تجعل الحروف كبيرة والثانية تجعل الحروف صغيرة.
مثال:

```
<?php
$stence1 = 'YALSNE';
$stence2 = strtolower($stence1);
$stence3 = 'morocoo';
$stence4 = strtoupper($stence3);

echo $stence2.'  

```

النتيجة
yalsne
MOROCOO

وكما نلاحظ ان الكلمة الاولى كانت صغيرة فقمنا بجعلها كبيرة والكلمة الثانية العكس.

- دالة التاريخ والوقت

في بي اتش بي هناك دالة تعطينا الوقت والتاريخ الموجود في الجهاز وهي:

date()

وفي داخل علامة القوسين الخاصة بهاته الدالة نكتب العنصر الذي نريد استخراجة. يعني ممكن نريد استخراج السنة. او الساعة. او اليوم او الشهر...

وهاته هي العناصر التي يمكن استخراجها والرموز الخاصة بها:

H: الساعة

i: الدقائق

s: الثواني

d: اليوم

D: اسم اليوم

m: الشهر

M: اسم الشهر

Y: السنة

تطبيق:

```
<?php

$hour= date('H');
$minutes= date('i');
$day= date('d');
$month= date('m');
$year= date('Y');

echo 'The time now is ' . $hour . ':' . $minutes . ' /on ' . $day . '/' . $month . '/' . $year;
?>
```

The time now is ٠٨:٤٩ / on ٢٠١٠ / ٠٥ / ٠٧

اظن سنكتفي بهاته الدوال الجاهزة للوقت الحالي ومستقبلا كما قلنا سنعرف المزيد حولها ان شاء الله.
 الان سنمر للنوع الثاني من الدوال وهي الدوال التي ننشئ نحن. حتى نعرف طريقة ذلك
 يجدر الاشارة الى انه يجب ان تكون الدالة المنشئة منطقية حتى يتم تطبيقها فتظهر.
 اول شيء في انشاء الدالة هو الاسم. حيث يجب ان تحمل هاته الدالة اسما خاصا بها. ولك حرية الاختيار في التسمية. فقط من اجل ترتيب
 الامور يستحسن اختيار اسم مختصر للوظيفة التي ستؤديها تلك الدالة حيث يكون دالا عليها بعد ذلك نعطي الاجراءات التي نريد ان تقوم
 بها الدالة ثم اخيرا نقوم بالنداء على الدالة باسمها حتى تظهر ناآج عملها.
 وهاته هي الصيغة العامة لكتابة الدالة في الغالب:

```
function name_of_function(variables)
{
  الاجراءات او التعليمات
}
name_of_function(values);
```

مثال:

```
<?php
function hello($name)
{
  echo "hi " . $name . " <br />";
}

hello('yalsne');
hello('batol');
hello('khaled');

?>
```

كما نرى اننا اعتمدنا في المثال

اسم الدالة هو : hello

المتغير هو name

وكلما غيرنا المتغير يتم تطبيق الدالة على المتغير الجديد. كما بإمكاننا ان ننشئ اكثر من متغير واحد.

بل اكثر من دالة ونجعل دالة بدخل دالة وكل ذلك بشكل منطقي وسليم.

مثال اخر بسيط على السريع:

```
<?php
function calcul($n1,$n2,$n3)
{
  echo $n1+$n2+$n3;
}
calcul(١٠,٢٠,٣٠);

?>
```

النتيجة ٦٠

المصفوفات

ARRAYS

مقدمة:

لاشك انكم ما زلتتم تتذكرون انه سبق لنا وأن مررنا باسم مصفوفة في دروسنا السابقة وقلنا على ان المصفوفة (array) هي عبارة عن متغير . وفعلا هي كذلك لكن متغير اكثر تشعبية. يحتوي على مجموعة من العناصر لها قيم معينة. كيف؟؟

دعونا نضرب مثلا بسيطا لنفهم معنى المصفوفة. وسنشبه المصفوفة هنا بحافلة بها ٣٠ مقعدا في كل مقعد يجلس شخص معين إذن:

الحافلة هي عبارة عن مصفوفة والثلاثين مقعدا هم عناصر المصفوفة (حيث ان المقاعد تنتمي للحافلة) و الأشخاص الذين يجلسون على المقاعد الخاصة بالحافلة. هم قيم عناصر المصفوفة.

وحتى ندقق اكثر في معنى المصفوفة. فالمصفوفة هي عبارة عن جدول بالضبط ندخل فيه المعلومات او المعطيات التي نريد.

مثال: بطاقة تعريف

الاسم : ياسين

الجنسية: مغربية

المدينة : ميدلت

رقم الهاتف: ٢١٢٠٠٠٠٠٠٠٠٠

وهناك نوعان من الجداول (المصفوفات) :

- نوع يسمى بـ الجداول التجميعية

- ونوع اخر يسمى بـ الجداول المرقمة

وسندرس ان شاء الله فيما يلي. كل نوع على حدا.

-الجداول التجميعية:

مثال: هو نفس المثال السابق

وسنلاحظ ان ذلك الجدول في المثال يعتمد في ترتيبه وتسلسله على كلمات نحن اخترناها بذلك الترتيب:

الاسم - الجنسية - المدينة - رقم الهاتف.

وعليه فحين يعتمد الجدول في ترتيب قيم عناصره على كلمات (حروف,اسماء او افعالا)

فإن ذلك الجدول يسمى بجدول تجميعي

(Associative arrays)

إذن

اسم المصفوفة هنا سيكون هو : بطاقة تعريف

عناصر المصفوفة: الاسم - الجنسية - المدينة - رقم الهاتف

قيم عناصر المصفوفات : ياسين - مغربية - ميدلت - ٢١٢٠٠٠٠٠٠٠٠٠

ترجمة ذلك في لغة php

لقد عرفنا بشكل نظري وعام كيف هي المصفوفة التجميعية والان سنتطرق الى طريقة كتابة تلك المصفوفة في php

فقط سنعتمد على اللغة الانجليزية في كتابة المثال السابق

```
<?php
// Identification_card سنسمي المصفوفة في المثال السابق بـ
```

```
$Identification_card = array (
    'name'=> 'Yassine',
    'nationality' => 'Moroccan',
    'city' => 'Midelt',
    'phone_number' => 'Marseille');
?>
```

كما نلاحظ ان الامر بسيط فقط يحتاج الى شيء من الممارسة والتعود . فقط يجدر الإشارة الى انه ليست هاته فقط هي الطريقة الوحيدة في كتابة مصفوفة. بل هناك طريقة اخرى ايضا لو اني لاحب ان ارهقكم بكم الطرق الموجودة في دروسنا لان الاساس هو معرفة الطريقة السهلة والتي نختصر فيها وقتنا وسطورا في الكتابة. لكن لابس سنعرض الطريقة الثانية فرما قد تشاهدونها او تمرن عليها مستقبلا وحتى لا تستغربوا من ذلك.

```
<?php
$Identification_card["name"]="Yassine";
$Identification_card["nationality"]="Moroccan";
$Identification_card["city"]="Midelt";
$Identification_card["phone_number"]="٢١٢٠٠٠٠٠٠٠٠٠";
?>
```

وهاته هي الطريقة الاخرى حيث تلاحظون انه لم يرد في الكود كلمة array وتم تعويض هاته الكلمة بالمعقوفتين [و] ايضا تم تعريف كل عنصر وقيمه في سطر معين عكس الطريقة السابقة التي جمعت كل شيء في سطر واحد. وعليه ف:

```
$variable=array(elements) == $variable[ ]=elements;
```

سؤال : عرفنا كيفية انشاء مصفوفة في بي اتش بي. فكيف إذن سنظهر قيمة عنصر من المصفوفة؟؟

الامر كذلك بسيط. يكفي ان نضيف الكود الذي يطبع لنا 'المتغير' مع تحديد العنصر المراد طباعة قيمته. (أوليس المصفوفة بمتغير؟؟) إذن الكود الذي سيتم اضافته هو:

```
<?php
echo $Identification["nationality"];
?>
```

الناج سيكون هو:

Moroccan

ولقراءة كل المصفوفة نستخدم حلقة التكرار فتكون الاضافة على المثال السابق

هي كالتالي:

كود

```
for($i=٠;$i<٤;$i++)
{
    echo $Identification_card[$i]. "<br />";
}
```

</ br> :كانت من اجل الرجوع للسطر وطباعة كل قيمة عنصر في سطر معين

المصفوفات المرقمة:

تشبه المصفوفات التجميعية فقط يتم تعويض الكلمات التي نعطيها للعناصر حتى نميز ترتيبها بالأرقام.
مثال: ايام الاسبوع

الرقم او العنصر	القيمة
٠	الاحد
١	الاثنين
٢	الثلاثاء
٣	الاربعاء
٤	الخميس
٥	الجمعة
٦	السبت

طريقة الكتابة في php:

```
<?php
// سنسمي المصفوفة هنا باسم days
$days=array('Sunday','Monday','Tuesday','Wednesday','Thursday','friday','Sutrday');
for($i=٠;$i<=٦;$i++)
{
echo $days[$i]."<br />";
}

?>
```

النتيجة:

Sunday
Monday
Tuesday
Wednesday
Thursday
friday
Sutrday

ملحوظة : ترقيم وترتيب قيم عناصر مصفوفة يبدأ من ٠ دائما
طبعا كما توجد طريقة ثانية للكتابة عند المصفوفات التجميعية يوجد ايضا طريقة ثانية عند المصفوفات المرقمة.

```
<?php
$days[٠]='sunday';
$days[١]='Monday';
$days[٢]='Tuesday';
$days[٣]='Wednesday';
$days[٤]='Thursday';
$days[٥]='friday';
$days[٦]='Sutrday';
// سنطلب اظهار على سبيل المثال يوم الجمعة بحكم انه يوم عيد بالنسبة لنا
echo $days[٥];
// كتابنا رقم خمسة لانه هو رقم العنصر الذي يحتوي على يوم الجمعة الذي نريد طباعته
?>
```

وفي حالة اردنا اظهار محتوى المصفوفة كلها من عناصر و قيم سنقوم بتعويض الامر
echo \$days[0];
بالامر
print_r(\$days);

ووظيفة الامر print_r تكمن في الحقيقة في استعراض المعلومات التي يتوفر عليها المتغير. ودائما يرجى الانتباه ان المتغير الذي يستعمل معاهاته الدالة يوضع بين قوسين.

وهكذا نكون قد تعرفنا على نوعي المصفوفات اللذان يبدوان متشابهين الى حد ما في طريقة الكتابة لكن الفرق الوحيد يكمن في الاستعمال الذي نريد .
فينصح باستخدام المصفوفة المرقمة في حالة كان لدينا عناصر كلها من نوع واحد مثلا اسماء الايام (كلها عبارة عن ايام) اما في حالة كنا نريد تخزين بيانات متخلفة يعني مثلا الاسم, السن , البلد, المهنة... فمن الاحسن استخدام المصفوفة التجميعية.

دوال متعلقة بالمصفوفات

تحدثنا في الدرس السابق عن الدوال وقلنا ان الدوال نوعان اما دوال جاهزة اي خاصة ب php او دوال نقوم نحن بإنشائها حسب ماتطلبه الوظيفة التي نريد ان نحصل عليها.
وهنا سنلمس نوعا من الدوال الجاهزة وهي دوال تستخدم بكثرة مع المصفوفات, واول دالة سنتعرف عليها هي:

الدالة KEY:

من الدوال البسيطة التي تستعمل لمعرفة العنصر الاول ويجب التفريق بين العنصر وقيمة العنصر حيث ان هاته الدالة تعطينا رقم او اسم العنصر الاول فقط دون ان تعطينا القيمة التي يحمل.
وطريقة كتابته في مثال:

```
<?php  
$a=array("name"=>"yassine","city"=>"midelt","phone"=>"٢١٢٠٠٠٠٠٠٠٠٠");  
$b=key($a);  
echo $b;  
  
?>
```

الناج سيكون هو :
name

الدالة

ARRAY_KEY_EXISTS

وهي دالة نستعملها للتأكد من وجود عنصر معين في مصفوفة
مثال:

```
<?php  
$a=array("name"=>"yassine","city"=>"midelt","phone"=>"٢١٢٠٠٠٠٠٠٠٠٠");  
if (array_key_exists('city', $a))  
{  
    echo 'yes';  
}  
else  
{  
    echo "no";  
}  
  
?>
```

وطبعا سيكون الناتج هو :

yes

كون العنصر city ينتمي الى المصفوفة a

الدالة

[IN_ARRAY](#)

للتحقق من وجود قيمة معينة داخل مصفوفة

مثال لطريقة استعمال الدالة:

```
<?php
$a=array("name"=>"yassine","city"=>"midelt","phone"=>"٢١٢٠٠٠٠٠٠٠٠");
if (in_array('yassine', $a))
{
    echo 'yes';
}
else
{
    echo "no";
}
?>
```

الناتج

yes

الدالة

[ARRAY_PUSH](#)

هي دالة تسمح لنا باضافة قيم جديدة لمصفوفة معينة

مثال:

```
<?php
$names=array("yassine ","sofia ","ahmed ");
array_push($names,"btisam ","ali");
for($i=0;$i<4;$i++)
{
    echo $names[$i];
}
?>
```

الدالة

[ARRAY_POP](#)

تقوم بحذف اخر عنصر/قيمة من مصفوفة

مثال:

```
<?php
$a=array("yasso"," sofo", " hamo", " hado");
$b=array_pop($a);
for($i=0;$i<4;$i++)
{
    echo $a[$i];
}
?>
```


سيتم حذف الاسم :

hado

الدالة

[ARRAY_SHIFT](#)

تقوم بحذف اول قيمة من مصفوفة

```
<?php
$a=array("yasso"," sofo", " hamo", " hado");
$b=array_shift($a);
for($i=0;$i<4;$i++)
{
echo $a[$i];
}
?>
```

سوف يتم حذف :

yasso

وسنحصل فقط على :

sofo hamo hado

الدالة

[ARRAY_UNSHIFT](#)

تنجلى وظيفة هاته الدالة في اضافة قيمة او مجموعة من القيم عند بداية مصفوفة.
وهذا مثل لطريقة كتابتها وعملها:

```
<?php
$a = array("yassine ", "sofia ");
array_unshift($a, "ahmed ", "ali ");
for($i=0;$i<4;$i++)
{
echo $a[$i];
}
?>
```

الدالة

[ARRAY_FLIP](#)

تقوم هاته الدالة العجيبة بتعويض العناصر بقيمها. والقيم بعناصرها
مثال :

```
<?php
$a = array("name" => 1, "age" => 2);
$a = array_flip($a);
print_r($a);
?>
```

الناج سيكون هو

```
Array ( [1] => name [2] => age )
```

بحيث تم تحويل العناصر بقيمها والقيم بعناصرها
فلو الغينا السطر

```
$a = array_flip($a);
```

فلن يتم ذلك وسنحصل على الشكل العادي والمتوقع ظهور به المصفوفة

```
Array ( [name] => 1 [age] => 2 )
```

خاتمة

انتهت دروس الجزء الاول من اساسيات لغة php . والتي حاولت فيها تقليص حجم المفاهيم واعطاء تعاريف مبسطة وسهلة حتى تسهل على كل مبتدئ . ارجوا ان تحاولوا قدر الامكان اعادة قراءة الدروس ومحاولة استيعاب كل سطر والاستفادة ايضا مما تطرقنا اليه . ولا تنسوا التطبيق ثم التطبيق فالتطبيق .
والله أسأل ان يوفقني على اتمام باقي الاجزاء من دروس لغة php . ويوفق المستفيدين منها لما يحبه ويرضاه . إنه سميع مجيب الدعاء .

هاته الدروس من اعداد اخوكم في الله
ياسين ابوالشريف
صاحب عضوية codm في الانترنت

ملحوظة:
يمكنكم مراسلتي او الاتصال بي
عبر الايميل التالي:
yalsne@hotmail.com
او عبر رقم الهاتف
٢١٢٦٧٦٥٣٩٢٠٥
المملكة المغربية