

به نام خدا

آموزش زبان PHP



سایت: www.RightClick.ir

فهرست :

صفحه	عنوان
	آموزش جلسه ای PHP
۳	جلسه اول
۷	جلسه دوم (متغیرها و شرط ها)
۱۰	جلسه سوم (حلقه ها)
۱۳	جلسه چهارم (فرم ها)
۱۶	جلسه پنجم (توابع)
۱۹	جلسه ششم (ارتباط با MySQL)
۲۴	جلسه هفتم (SQL)
۲۹	جلسه هشتم (ساخت قسمت جستجو)
۳۴	جلسه نهم (ساخت Login)
۳۷	جلسه دهم (Session)
۴۲	جلسه یازدهم (رشته ها)
۴۷	جلسه دوازدهم (File Database)
۵۰	جلسه سیزدهم (تابع Date)
۶۲	کد امنیتی
۶۴	برای سایت خود یک سیستم رای گیری بسازیم ... آموزش ساخت خبرنامه
۶۸	ساخت خبرنامه (قسمت صفر)
۷۱	ساخت خبرنامه (قسمت اول)
۷۶	ساخت خبرنامه (قسمت دوم)
۸۰	ساخت خبرنامه (قسمت سوم)
۸۴	ساخت خبرنامه (قسمت چهارم)
۸۸	استفاده از تاریخ و ساعت گرینویچ در PHP
۹۰	آموزش کلاس Template
۱۰۳	SQL Injection و مقابله با آن ...
۱۰۸	کد افراد آنلاین
۱۱۱	برای سایت خود یک قسمت "ارتباط با ما" بسازیم
۱۱۵	آموزش برنامه آپلود عکس (خیلی ساده)

سؤال های کاربران و جواب ها

- ۱۱۷ GD چیست ؟
- ۱۱۷ تابع رندوم (شانسی)
- ۱۱۷ نوشتن جاوا اسکریپت در PHP
- ۱۱۸ فرستادن محتوای یک صفحه ی HTML ؛ بصورت ایمیل
- ۱۱۸ چه طوری میشه از تابع SMTP استفاده کرد ؟؟؟
- ۱۲۲ چه طوری با کوکی ها در پی اچ پی میشه کار کرد ؟
- ۱۲۳ چه طوری میشه آدرس یک RSS را داخل پی اچ پی وارد کنیم
- ۱۲۵ چه جوری میشه نام یک فیلد و تعداد فیلدها را در جدولی پیدا کرد ؟
- ۱۲۶ کد برای پخش تصادفی فلش در PHP و JavaScript
- ۱۲۷ چگونه برای سایت خود آرشیو درست کنیم ؟
- ۱۲۸ چگونه برای نوشته ها ادامه مطلب بگذاریم ؟
- ۱۲۸ دستورات کش کردن و جلوگیری از کش چیست ؟
- ۱۲۹ علامت @ در پشت توابع چه کاربردی دارد ؟
- ۱۲۹ چه جوری میشه بفهمیم که کاربر از چه آدرسی وارد سایت شده و یک سری از امکاناتش مثل مرورگر سیستم عامل ip و.. رو بدست آورد ؟
- ۱۳۰ الگوریتم رمز کردن، قابل بازگشت و غیر قابل بازگشت
- ۱۳۰ توضیح در مورد توابع **imagestring** و **imagejpeg** و **imagedestroy**
- ۱۳۱ دانلود متن و دانلود فایل
- ۱۳۱ ساختن جستجو و نمایش چند صفحه ای اطلاعات

جلسه اول :

امروز اولین بخش **php** را شروع می کنیم .
ببینید **php** از نظر گرامری خیلی شبیه به زبان **C** و جاوا اسکریپت ... است .
اگر شما از قبل با زبان **C** و یا بطور کلی زبان های برنامه نویسی کار کرده باشید خیلی سریعتر از چیزی که فکرش رو می کنید اون و یاد می گیرید ...

خوب برای شروع کار اول باید بدونیم که **php** چی هست و کجا اجرا می شه و چه چیزایی لازم داره ...

php یک زبان برنامه نویسی هست که در وب (به طور کلی) به کار می رود .

ببینید زبان های برنامه نویسی وب را به دو دسته می توانیم تقسیم کنیم ...

- 1- ClientSide
- 2- ServerSide

Client Side به برنامه هایی گفته می شود که بر روی کامپیوتر کاربر (کلاینت) اجرا می شوند مثل: جاوا اسکریپت ...

ServerSide برنامه هایی است که بر روی سرور (سرویس دهنده) اجرا می شوند مثل: **PHP** ، **ASP** ، **CGI** و ...

مثلا شما می خواهید وقتی کاربر بر روی قسمتی از صفحه وبتون کلیک کند به پیغام به اون نشون داده بشه ... این برنامه روی کامپیوتر کاربر اجرا می شود ... این رو می تونیم با جاوا اسکریپت بنویسیم ...

اما به موقع هست که می خواهید مثلا کاربر بتونه به چیزی رو توی محتوای سایت شما جستجو کنه که این جستجو باید بر روی سرور انجام بشه چرا ؟ خوب معلومه چون تمامی سایت ما روی سرور است ...

خوب حالا که به مقدمه ای گفته شد بیایید **php** رو شروع کنیم .

مواد لازم برای **php** :

یک ویرایشگر مثل **DreamWeaver** البته با نوت پد هم می شه ولی با این راحت ترید ...
یک محیط که برنامه های **php** بتونن اونجا اجرا شوند مثل یک سرور که **php** رو هم ساپورت می کنه ...
البته برنامه هایی هم وجود دارند که اون سرور رو روی کامپیوتر ما شبیه سازی می کنند... مثل **Easy PHP** و **Wamp** و ...

آیا پی.اچ.پی محیط خاصی برای برنامه نویسی داره یا نه ؟

در واقع **php** محیط خاصی نداره اونو می تونین تو نوت پد هم بنویسید ولی محیط اجرایی اون فرق میکنه یعنی باید روی یک سرور اجرا بشه ... که البته **Easy Php** و برنامه هایی مثل این اون سرور رو بر روی کامپیوتر ما شبیه سازی می کنند .

نصب Easy PHP هم ساده است و اون رو می تونید از آدرس زیر دانلود کنید ...

http://easyphp.abbal.com/depot/easyphp1-6_setup.exe

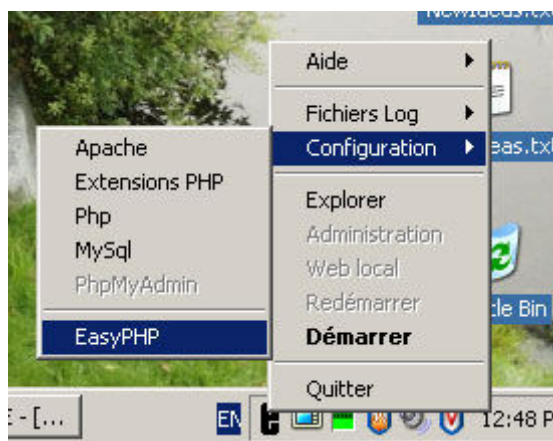
خوب حالا که نصبش کردید، اون رو اجرا کنید؛ اگه فایروال داشته باشید فایروالتون یه هشدار میده... شما بهش اجازه بدید ...

بعد از اجرا یه آیکون در کنار ساعت به نمایش در می آید .

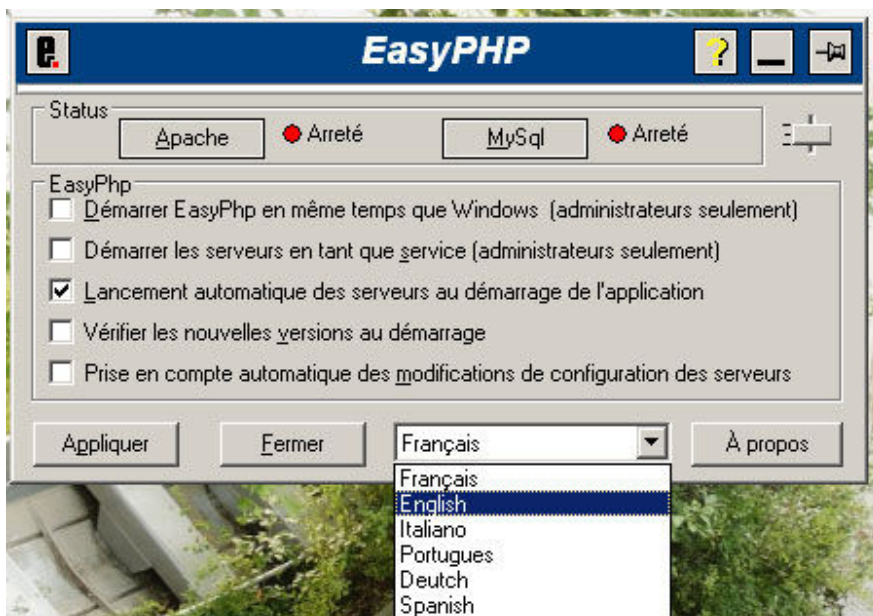


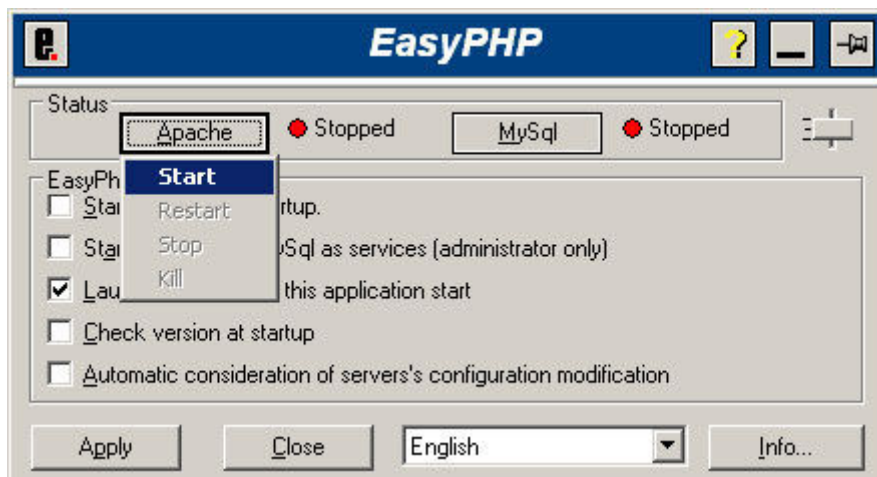
خوب برای این که قابل استفاده باشد مراحل زیر را طی می کنیم:

ابتدا بر روی آیکون این برنامه در کنار ساعت راست کلیک می کنیم و از منوی باز شده **Configuration** و از زیر منوی باز شده **EasyPHP** را مطابق شکل باز می کنیم ...



بعد در پنجره باز شده زبان برنامه را انگلیسی انتخاب کنید و بر روی دکمه **Apply** کلیک کنید ، خوب حالا سرور آپاچی رو راه می اندازیم ، به همین صورت سرور **MySQL** رو هم راه میاندازیم ...





خوب حالا باید آیکون برنامه به شکل چشمک زن در بیاد .



خوب حالا فایل های روی سرور را باید کجا قرار دهیم ؟

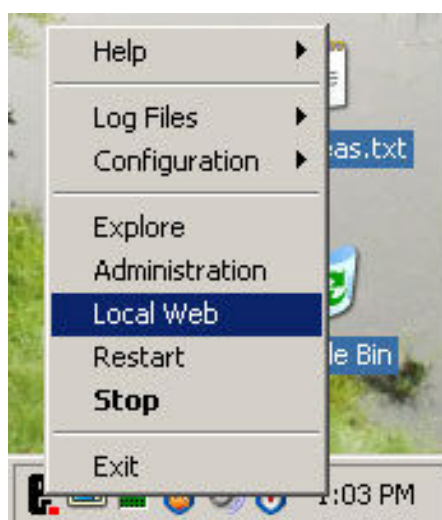
به فرض این که برنامه رو توی درایو C و پوشه **Program Files** نصب کرده باشید ... بایستی به این مسیر بروید :

کد :

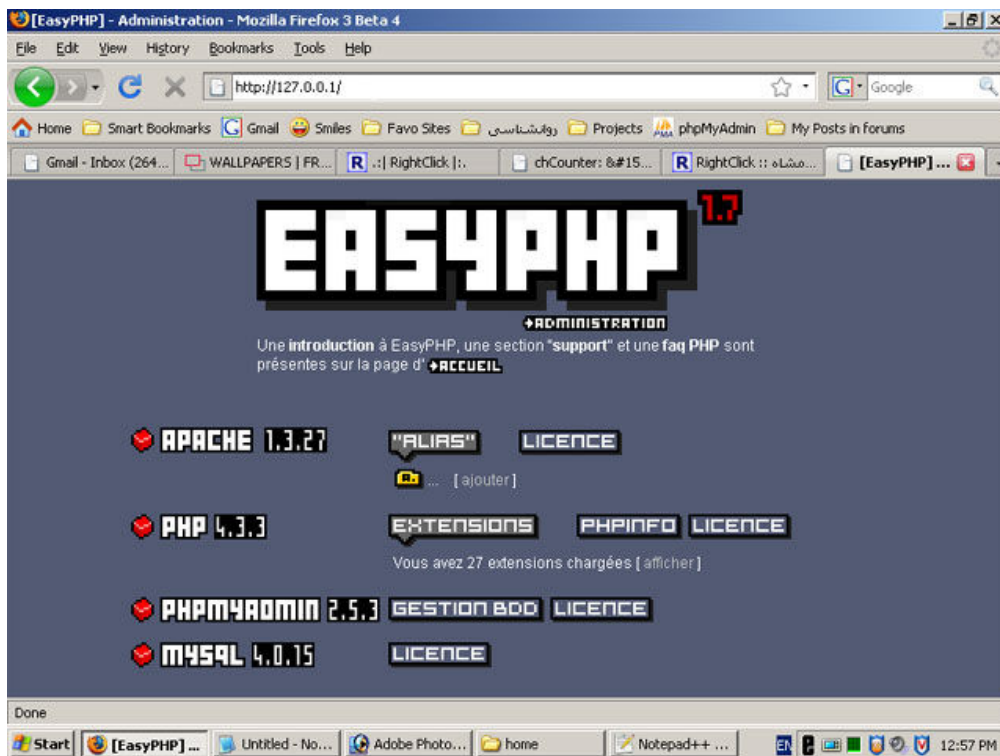
C:\Program Files\EasyPHP1-7\www

و صفحات وب خودتون رو اونجا کپی کنید ...

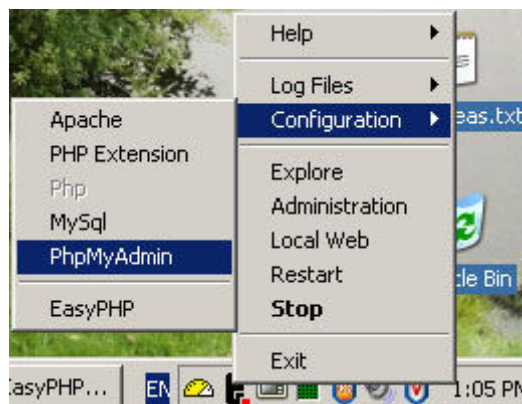
برای اجرای فایل هاتون یعنی یه چیزی شبیه به اینترنت، بر روی **Local Web** کلیک کنید .



تا در مرورگر پیش فرض صفحه وب اصلی را باز کند .



برای مدیریت پایگاه داده برنامه معروف و قوی **Php My Admin** نیز همراه **Easy PHP** وجود دارد که برای دسترسی به آن، بر روی **PhpMyAdmin** از منوی **Configuration** کلیک می کنیم .



خوب حالا که همه چیز برای شروع برنامه نویسی **php** آماده شد برای امتحان به کد کوچولو می نویسیم ...

```

کد :
<?php
print("WelCome To PHP World!");
?>

```

این کد را در نوت پد وارد کنید و در فایل به نام **index.php** در مسیر اصلی لوکال هاست **www** در **Easy php** ذخیره کنید . حال مرور گر خود را باز کنید و به این آدرس بروید **http://127.0.0.1** یا **http://localhost** ، تا صفحه خود را ببینید .

از جلسه بعد کد نویسی **php** رو شروع می کنیم .

جلسه دوم (متغیرها و شرط ها) :

در این قسمت به مفهوم متغیرها و شرط ها می پردازیم .

متغیر چیه ؟

فرض کنید شما میخواهید یک مقدار عددی یا یک رشته مثلا اسم یک نفر رو نگه دارید و بعد از یک سری کارها دوباره ازش استفاده کنید . برای این کار از چیزی به نام متغیر استفاده می کنیم . در واقع وقتی یک متغیر را تعریف می کنیم یک قسمت از حافظه **Ram** سیستم را رزرو می کنیم و هر چه را که می خواهیم آنجا می نویسیم . در برنامه های دیگر مثل **C , Pascal , ...** ما هنگام تعریف متغیر بایستی نوع آن را هم تعریف کنیم ... (نوع چیه ؟) اما توی **php** نیازی به تعریف نوع نمی باشد خود **php** نوع آن را تشخیص می دهد . به مثال ساده :

کد :

```
$a = 25;
```

دقت کنید که در این مثال **a** متغیر است و عدد ۲۵ مقداری است که ما در آن متغیر قرار داده ایم .

همان طور که قبلا گفته شد بعد از هر دستور باید سیمیکلن ; قرار می دهیم . علامت \$ چیه ؟ توی **php** قبل از نام هر متغیر بایستی حتما این علامت رو قرار دهیم . نامگذاری متغیرها یک سری قوانین داره که مثلا اولین کاراکتر اون نمیتونه _ باشه یا با عدد شروع بشه و همچنین بایستی از حروف الفبا استفاده کنیم مثلا همه نمونه های زیر نادرستند :

کد :

```
$_salam = 12;  
$65a = 10;
```

حال که با نام گذاری آشنا شدیم ببینیم چه مقادیری توی متغیرها می توانیم قرار دهیم یا بهتر بگم چه نوع هایی داریم :

کد :

```
$a = 12;
```

این نوع یک عدد صحیح است .

کد :

```
$a = 12.5;
```

این نوع یک عدد ممیز شناور (اعشاری) است .

کد :

```
$a = "Ali";
```


این یک رشته است دقت کنید که برای وارد کردن رشته بایستی از دو علامت " یا ' ' استفاده کنیم یعنی :

کد :

```
$a = 123;
```

با :

کد :

```
$a = '123';
```

خیلی فرق داره اولی عدد ۱۲۳ است و دومی کاراکتر ۱ و ۲ و ۳ است که به دنبال هم یک رشته رو تشکیل می دن .

حالا به مثال :

کد :

```
$num1 = 10;  
$num2 = 5;  
$total = $num1 + $num2;  
print($total);
```

اگر دقت کنید می بینید که در خط اول و دوم دو متغیر به نام های `num1` و `num2` رو تعریف کردیم و مقدار دادیم و در خط بعد مجموع این دو را در متغیر `total` قرار دادیم و درخط بعد نتایج را چاپ می کنیم (به مرور گر ارسال می کنیم) .

توجه: سعی کنید که همیشه نام های با مسما به متغیر هاتون بدین برای این که بعد به فهم برنامه کمک می کنه و دلیل دیگر اون اینه که اگه برنامهتون طولانی بشه دیگر گیج نمی شوید .

خوب حالا که با متغیر ها آشنا شدیم بریم سراغ **شرط ها** .

فرض کنید توی برنامهتون فراره که کاربر نام کاربری و کلمه عبور خود را وارد نماید و در صورت درست بودن هر دو بتواند وارد صفحه دیگری شود برای این کار (یعنی چک کردن `username` و `password` از شرط استفاده می کنیم .

مثلا :

کد :

```
if(شرط)  
{  
----  
----  
----  
}
```

شکل کلی شرط به این صورته . داخل پرانتز به شرط رو می نویسیم و بین { } دستوراتی که اگر شرط داخل پرانتز برقرار بود باید اجرا شوند رو قرار می دهیم .
مثلا :

```
کد :  
if($user == "ali")  
{  
    print("Welcome Ali!");  
}
```

در صورتی که متغیر **user** برابر با رشته **ali** بود دستور **print** داخل شرط اجرا می شه .
آقا این == چیه ؟
خوب حالا بریم سراغ جواب این سوال :

برای بررسی دو چیز با هم از یک سری علامت ها استفاده می کنیم که معنی اونا رو زیر نوشتم :

== بررسی تساوی دو چیز

> کوچکتر

< بزرگتر

=> کوچکتر یا مساوی

=< بزرگتر یا مساوی

!= مخالف (نامساوی)

نکته خیلی مهم: دقت کنید که برای بررسی تساوی از = استفاده نکنید، زیرا این عمل انتساب است و باعث میشه شرط شما همیشه برقرار باشد ...

شرط ها مباحث زیادی دارند که در قسمت های بعدی به بررسی آنها می پردازیم .

جلسه سوم (حلقه ها) :

برای این قسمت حلقه ها رو در نظر گرفتیم ...

حلقه ها در هر زبانی برای انجام کارهای تکراری به کار می رود مثلا برای نمایش لیستی از نتایج جستجو و یا هر چیز دیگه ای ... در این جا من سه نوع از حلقه هایی که در **php** به کار می رود را توضیح میدم ... با یه مثال ساده شروع می کنم ...

من میخوام یه کلمه ۱۰ بار چاپ بشود ...

کد :

```
for($i = 0 ; $i<10 ;$i++)
{
print("Salam<br />");
}
```

این تکه کد کلمه **Salam** رو ۱۰ بار زیر هم توی مرورگر نمایش میده ...

خوب حالا به توضیح می پردازیم :

این حلقه که به کار بردیم به حلقه **for** معروف است و شکل کلی اون این طوری است ...

کد :

```
for(1 ; 2 ; 3)
{
بدنه حلقه
}
```

توضیح :

- ۱- در این قسمت مقدار اولیه رو می نویسم . یعنی شمارنده حلقه (؟) از کجا شروع بشه ...
- ۲- در این قسمت یه عبارت شرطی رو می نویسیم و به محض این که این شرط نادرست شود یعنی **false** بشه از حلقه خارج می شویم ... دقت کنید باید شرطی بنویسیم که بالاخره یه موقعی **false** بشه و از حلقه خارج بشیم چون اگه خارج نشیم همین طور توی حلقه می چرخیم و تا ابد بیرون نمیایم .
- ۳- در این قسمت که به گام حلقه موسوم است بایستی عبارتی رو بنویسیم که مثلا به مقدار اولیه یه چیزی اضافه کنه یا ازش کم کنه ... بستگی به شرایط داره ...

بدنه حلقه ؟

بدنه حلقه دستوراتی است که ما می خواهیم آنها را تکرار کنیم ... در مثال بالا دستور **print** بود بدنه حلقه میتونه هر چند تا دستور که بخواهیم باشه حتی می تونه چند تا حلقه دیگه هم توش باشه ...

نکات مهم :

* دقت کنید که ما به متغیر به عنوان مقدار اولیه در نظر گرفتیم ... و این مقدار می تونه هر چیزی باشه یعنی می تونه عدد یا کاراکتر باشه ...

* شرط حلقه می تونه هیچ ربطی به متغیری که ما توی مقدار اولیه تعریف کردیم نداشته باشه ...

* گام حلقه می تونه کم بشه یا زیاد بشه همچنین میتونه یکی یکی کم و زیاد بشه و یا حتی هر بار ۰،۱ تغییر کنه ...

* $i++$ چیه ؟

این دقیقا معادل اینه :

کد :

```
 $i = i + 1 ;$ 
```

یعنی یکی به متغیر i اضافه می کنه، به چیز دیگه توی همین مایه ها داریم که به این شکل می نویسیم $i--$ یعنی به دونه از i کم می کنه ... به جور ساده نویسیه ...

* ما میتونیم هر یک از قسمت های سه گانه حلقه (یعنی مقدار اولیه و شرط و گام حلقه) را ننویسیم البته باید ؛ را بگذاریم . مثلا :

کد :

```
for(;;)  
{  
    چند تا دستور  
}
```

یه سوال ؟ آقا مگه نگفتی که باید به جوری از حلقه خارج بشیم - حالا که شرط رو نمی نویسیم چطوری از حلقه خارج بشیم ... ؟

یه دستور داریم به نام **break** که هر جا برنامه به این دستور رسید از حلقه خارج می شود ... ما می تونیم بر حسب نیازمون توی حلقه هر جا که خواستیم این دستور رو اجرا کنیم .

حلقه بعدی که بررسی می کنیم حلقه **while** است ... مثال بالا را با همین حلقه مینویسیم :

کد :

```
 $i = 0;$   
while( $i < 10$  )  
{  
  
    print("Salam<br>");  
     $i++$ ;  
}
```

ساختار کلی این حلقه به این شکل است :

```
کد :  
while(شرط)  
{  
بدنه حلقه  
}
```

این شرط به محض این که نقض (**false**) شود از حلقه خارج می شویم ...

حلقه بعدی **do while** است :

```
کد :  
$i = 0;  
do  
{  
$i++;  
print("Salam<br>");  
}while($i<10);
```

خوب این چه فرقی با **while** داشت ؟

این حلقه دقیقا مثل **while** است با این تفاوت که شرط حلقه در پایان حلقه بررسی میشه این حلقه در هر صورت حداقل یک بار اجرا می شود ولی حلقه هایی مثل **for** , **while** در صورتی که قبل از حلقه شرط برقرار نباشد اصلا وارد حلقه نمیشود اما این نوع حلقه حتما یک بار وارد می شود ...

و این بود حلقه ...

در جلسات بعدی در مورد فرم ها و گرفتن اطلاعات از کاربر بحث می کنیم .

جلسه چهارم (فرم ها) :

چون برنامه هایی که ما می نویسیم روی سرور هستند و به درخواست کاربر اجرا می شوند در این قسمت به فرم ها (که یک نوع درخواست هستند) می پردازیم .

فرم چیه ؟؟ به صفحه که یک (چند) تا دکمه و یا جایی برای ورود اطلاعات داره مثل جایی که نام کاربری و کلمه عبور رو وارد می کنیم و یا مثل صفحه اول گوگل که یه دکمه و یه محل ورود داره ...

با فرم ها به دو صورت اطلاعات رو به سمت سرور می فرستیم (یا اصطلاحا درخواست میدهیم) :

1- Get

2- Post

ساخت فرم :

برای ساختن فرم بایستی از کد **Html** استفاده کنیم ربطی به **php** نداره
یک مثال از یک فرم :

کد :

```
<form method="POST" action="Page.php" name="MyForm" target="_self">  
<input type="text" name="T1" size="20">  
<input type="submit" value="Submit" name="B1">  
<input type="reset" value="Reset" name="B2">  
</form>
```

همون طور که می بینید چند تا تگ ساده هستند که یک تگ **<form>** **</form>** هست و چند تا تگ دیگه وسط اونا ...

اون وسطی ها مربوط به اشیای فرم هستند .

بررسی تگ **<form>** :

این تگ هم مثل بقیه تگ های **html** یک سری خاصیت داره و از یه جا شروع می شه و به یه جا ختم می شه .

method : نوع درخواست را مشخص می کند که می تواند **POST** و یا **GET** باشد در صورتی که اصلا این را ننویسیم به صورت پیشفرض **Get** در نظر گرفته می شود .

Action : صفحه مقصد را مشخص می کند یعنی این که پس از پر کردن فرم برای ارسال اون به سمت سرور به کدوم صفحه ارسال بشه ... در ادامه می بینیم که کد های مربوط به فرم رو باید توی صفحه مقصد بنویسیم. در صورتی که این را ننویسیم صفحه جاری صفحه مقصد در نظر گرفته می شود .

Name: نام فرم است که برای **php** نیازی به اون نداریم ولی اگه بخواهیم از جاوا اسکریپت استفاده کنیم نیاز می شه ... مثل چک کردن صحت ایمیل قبل از ارسال اون تا به وقت یکی شیطونی نکنه.

Target: نوع باز شدن صفحه مقصد که شامل صفحه جاری ، صفحه جدید ، صفحه پدر ، یک فریم خاص و ... است . ممکنه تا به حال به صفحاتی بر خورده باشید که پس از فشردن دکمه مربوط به ارسال فرم نتیجه رو توی یه صفحه جدید مشاهده کردید .

خوب تا اینجاش **html** بود که خودتون با **FrontPage** یا برنامه های مربوطه می تونید بسازید. و مربوط به ارسال اطلاعات بود .

خوب برای این که با دریافت هم آشنا بشویم یه مثال کوچولو می زنیم :
فرض کنیم می خواهیم برنامه ای بنویسیم که یک رشته را دریافت کرده و اونو ۵ بار چاپ کنه؟
اول فرم اون رو می سازیم (که **html** هست) :

```
کد :  
<form method="POST" action="print.php">  
  <p><input type="text" name="T1" size="20"></p>  
  <p><input type="submit" value="Submit" name="B1"></p>  
</form>
```

این فرم یه فرم خیلی ساده است که صفحه مقصد اون **print.php** هست و چون آدرسی داده نشده پس صفحه ای که فرم توی اون و صفحه ای که کد های **php** توی اون نوشته شده (**print.php**) بایستی توی یه فولدر باشند .

این فرم یه فیلد (جایی که اطلاعات رو توی اون می نویسیم) و یه دکمه داره .

خوب حالا کد **php** اون (صفحه **print.php**) :

```
کد :  
<?  
$my_str = $_POST['T1'];  
for($i = 0 ; $i < 5 ; $i++)  
{  
  print($my_str);  
}  
?>
```

در خط اول به یه متغیر به نام **my_str** مقدار

```
کد :  
$_POST['T1']
```

رو انتصاب دادیم . این مقدار برای دسترسی به اشیای فرم هست و **T1** نام اون فیلد ما است که محتوای اون رو توی یه متغیر ریختیم و توی حلقه اون رو ۵ بار چاپ کردیم .

برای دسترسی به اشیای فرم در صورتی که با **Get** ارسال شده باشند از

کد :

```
$_GET['FieldName']
```

استفاده می کنیم .

جلسه پنجم (توابع) :

تابع چیست...؟

فرض کنید که ما به کارخونه خودرو سازی داریم که توی اون به دستگاه هست که از به طرف مواد اولیه رو بهش میدهیم و از طرف دیگر خدا بیامرز پیکان میآد بیرون ... حالا اومدیم و به نقص کوچولو پیدا کرد مثلا چراغ های جلو روشن نمی شوند ... اینجاست که دردرس ها شروع میشه یعنی ما باید اون دستگاه بسیار بزرگ که پیکان رو میسازه رو از اول این که مواد اولیه که بهش میدیم تا زمانی که پیکان ساخته شده را به ما تحویل میده زیر نظر بگیریم تا ببینیم کجای کار می لنگه... واقعا کار سخت و زمان گیری است .

حال فرض کنید کارخونه ما به جای به دستگاه بزرگ چند تا دستگاه کوچولو داره که هر دستگاه قطعه تولید شده خودش رو به به دستگاه دیگه میده تا بالاخره خودروی ما سرهم بشه . اگه مشکلی مثل مشکل بالا صورت گرفت ما فقط به دستگاه کوچولو رو می خواهیم رفع عیب کنیم ، و این خیلی راحتتره ...

توی برنامه هامون هم ما بهتره که برنامه رو بر اساس کارهای مختلف تکه تکه کنیم که اگه به قسمت برنامه ایراد به هم زد فقط اون قسمت رو بخوایم ارور گیری کنیم . به هر کدام از این تکه برنامه ها به فانکشن (تابع) میگی که این طوری نوشته می شه .

کد :

```
function salam()  
{  
    print("Salam");  
}
```

این به فانکشن ساده است که کلمه سلام رو چاپ میکنه ...
ما قبلا هم از فانکشن ها استفاده میکردیم بدون این که اونو بلد باشیم !!!

مثلا فانکشن

کد :

```
print();
```

که با اون آشنا هستید و از این قبیل . که البته این یک فانکشن داخلی خود php هست .

اصطلاحات مربوط به توابع :

فراخوانی : یعنی این که تابع را صدا بزنیم توی مثال بالا برای صدا زدن تابعی که نوشتیم فقط کافیست که هر جا که لازم داشتیم نام تابع رو بنویسیم :

کد :

```
salam();
```

یعنی کامپایلر **php** وقتی به این عبارت برسد روند اجرای برنامه رو متوقف میکنه و به سراغ تابع میرود و خط به خط تابع را اجرا میکند تا تابع تمام شود ، بعد دوباره بر میگردد و ادامه میدهد .

ورودی تابع : تابع ما میتونه یک یا چند تا ورودی هم داشته باشه مثلا یه تابع که نام کاربری و کلمه عبور رو بگیره و ببینه که کاربر معتبر هست یا نه و ... برای نمونه به تابع زیر نگاه کنید که یه ورودی میگیره و اون رو ۵ بار زیر هم چاپ میکنه :

```
کد :  
function my_prn($mystr)  
{  
    for($i = 0 ; $i < 5 ;$i++)  
        print($mystr."<br>");  
}
```

و اونو این طوری صدا میزنیم :

```
کد :  
my_prn("My Name");
```

البته به جای "My Name" میتونستیم یه متغیر قرار بدیم :

```
کد :  
$my_name = "Mojtaba";  
my_prn($my_name);
```

خروجی تابع : بعضی از توابع هستند که چیزی را چاپ نمیکنند بلکه یک مقدار را برای ما بر میگردانند مثلا تابعی که کلمه عبور یک کاربر را بگیرد و آن را به رمز خاصی تبدیل کند و بعد از آن ما این رمز را با رمزی که خودمان داریم مقایسه کنیم (یعنی این که نمیخواهیم چیزی چاپ شود) یک مثال ساده :

```
کد :  
function sum($a,$b)  
{  
    $c = $a +$b ;  
    return $c ;  
}
```

تابع فوق دو عدد را می گیرد و مجموع آنها را بر می گرداند ، با استفاده از کلمه کلیدی **return** چیزی را بر می گردانیم .

چگونه به مقدار برگشتی دستیابی داشته باشیم ؟

برای تابع فوق می توانیم این چنین بنویسیم :

کد :

```
$s = sum(2,5);
```

کد بالا مقدار برگشتی تابع `sum` را به ازای ورودی های ۲ و ۵ که میشود ۷ را در متغیر `S` می ریزد. چون `S` به متغیر مثل بقیه است پس ما می توانیم خود

کد :

```
sum(2,5)
```

را در شرط ها ، حلقه ها ، دستور چاپ و ... به کار ببریم .

مثلا :

کد :

```
if(sum(2,5) > 7)
{
    print ("Grater");
}
else
{
    print ("Less or Equal");
}
```

نکات مهم :

- ۱- کلمه کلیدی `function` , `return` حتما بایستی کوچک نوشته شوند .
- ۲- قوانین نام گذاری توابع مثل نام گذاری متغیر ها است .
- ۳- متغیر هایی که درون بدنه تابع به کار می روند هیچ ربطی به متغیر های برنامه ندارند (یعنی اختلالی ایجاد نمی کنند) .
- ۴- در تابع هر جا که به دستور `return` برسیم تابع تمام شده است و دستورات بعدی که در بدنه تابع نوشته شده اند اجرا نمی شوند .

جلسه ششم (ارتباط با MySQL) :

امروز آموزش ارتباط با دیتابیس MySQL رو آموزش میدم .

اصلا دیتابیس چیه ؟

دیتابیس یه جایی هست که داده هامون رو اونجا ذخیره می کنیم داده هایی مثل نام کاربران ، کلمه عبور آنها ، ایمیل آنها و ... مثل همین سایت **RightClick.ir** که تمامی تایپک ها و پست ها رو توی دیتابیس نگهداری می کنه .

برای این که داده ها سازماندهی بهتری داشته باشند خود دیتابیس هم به قسمت هایی تقسیم می شه که به اونا جدول (Table) میگیم . برای درک بهتر می تونید دیتا بیس رو یه فولدر در نظر بگیرید و **Table** ها رو فایل های درون آن البته خود **Table** تقسیم بندی های خاص خودش رو داره .

هر **Table** از یک سری فیلد (ستون) و رکورد (سطر) تشکیل شده است مثلا **Table** دانشجویان از فیلد های شماره دانشجویی ، نام ، نام خانوادگی ، سال ورود ، رشته ، مقطع و ... تشکیل شده که به هر کدام از این مشخصات یه فیلد می گیم .

خوب حالا فرض کنید یه سری داده توی این جدول وارد کردیم مثلا مشخصات دانشجویانی با مشخصات زیر :

834421 ، مجتبی ، اسکندری ، ۸۳ ، کامپیوتر ، کارشناسی

842135 ، حسن ، محمدی ، ۸۴ ، زراعت ، کارشناسی

820012 ، حسین ، مرادی ، ۸۲ ، فیزیک ، کارشناسی

.
. .

خوب به هر کدام از سری این مشخصات (هر سطر) یه رکورد می گوئیم. در ادامه مبحث بیشتر آشنا می شوید .

برای کار با دیتابیس MySQL باید از یه سری دستورات استفاده کنیم که به **Query** معروفند و با زبان **SQL** نوشته می شوند یاد گیری **SQL** خیلی راحت و بسیار سریع اون رو یاد میگیرید .

اما یه برنامه بسیار قدرتمند (تحت وب) برای مدیریت MySQL نوشته شده که به **PhpMyAdmin** معروفه .

خوب برنامه **PhpMyAdmin** رو باز کنید . (راهنمایی : اگه از **Easy PHP** استفاده می کنید به آدرس زیر توی مرورگرتون بروید :)

کد :

<http://127.0.0.1/mysql/>

خوب در سمت چپ یه لیست کشویی هست که دیتابیس های شما رو نشون میده . و توی قسمت سمت راست یه قسمت هست با عنوان **Create new database** که میتونید دیتابیس جدید ایجاد کنید .

مثلا نام **MyDB** : رو وارد کنید و روی **Create** کلیک کنید . خوب وارد یه صفحه دیگه می شویم که یه پیغام میده که دیتابیس شما ساخته شد . و در قسمت زیرین اون یه قسمت هست که نوشته :

کد :

```
SQL query:  
CREATE DATABASE `MyDB` ;
```

اینجا برنامه **PhpMyAdmin** اون **Query** که اجرا کرده تا عملیات درخواستی ما (در اینجا ساختن دیتابیس) رو انجام بده رو مینویسه و این به یاد گیری ما کمک میکنه .

خوب در قسمت **Create new table on database MyDB** نام **Table** ی که میخوایم بسازیم رو وارد میکنیم مثلا **users** :

و در قسمت **Number of fields** اون تعداد فیلد هایی که برای جدول نیاز هست رو مینویسیم مثلا ۲ (منظور **username** و **password**) هست .

در صفحه بعد بسته به اون عددی که توی مرحله قبل وارد کرده اید از شما نام فیلد و نوع آن و دیگر مشخصات را میخواهد . حالا یکی یکی توضیح میدم .

Field : در این قسمت اسم فیلد را می نویسید . مثلا : **username**

Type : در این قسمت نوع فیلد را مشخص میکنید مثلا کاراکتری ، رشته ، عدد صحیح ، عدد اعشاری و ... برای نام کاربری شما از همون (**VARCHAR** رشته حداکثر ۲۵۵ کاراکتری) استفاده کنید .

Length/Values : در این قسمت برای نوع هایی که نیاز به طول دارند طول فیلد را میدهیم مثلا **VARCHAR** نیاز دارد که حتما طول آن را بنویسیم شما باید با توجه به مورد استفاده خود طولی رو در نظر بگیرید که نه کم بیاد و نه زیاد باشه . مثلا برای نام کاربری (در این مثال) ۲۰ عدد خوبیه .

Attributes : یه سری خاصیت های مربوط به فیلد است که اکثر موارد استفاده نمی شود . شما هم آن را تغییر ندین .

Null : مشخص می کند که این فیلد میتونه خالی باشه یا نه (توی برنامه نویسی توضیح بیشتر میدم).

Default : مقدار پیش فرض فیلد شما است .

Extra : در این قسمت یه سری امکانات اضافی برای فیلد وجود داره که من تا حالا بیشتر از یکی ندیدم و همون هم توضیح می دم . اسم اون هست **auto_increment** اگر این امکان رو به فیلدی بدهید (باید نوع آن **INT** باشه) اون فیلد با اضافه کردن هر رکورد این فیلد رو یکی اضافه میکنه (از 1 شروع می کنه).

Primary : این قسمت نوشته نشده فقط یه عکس کشیده که با رفتن ماوس روی اون این متن رو نشون میده . اگه این قسمت رو برای یه فیلد علامت بزیند اون فیلد به اصلاح کلید اصلی جدول میشه و معمولا برای فیلد هایی که نباید تکراری باشه استفاده می شه مثل فیلد شماره دانشجویی که نباید دو دانشجو یه شماره داشته باشند .

برای قسمت های دیگه مثل **Index , Unique** بعد توضیح میدم .

خوب فیلد بعدی که **password** هست رو هم طبق راهنمایی های بالا درست میکنیم .بعد بر روی دکمه **Save** کلیک میکنیم .

جدول ما ساخته میشه باید **Query** شما به چیزی توی این مایه ها باشه .

کد :

```
SQL query:
CREATE TABLE `users` (
  `username` VARCHAR( 20 ) NOT NULL ,
  `password` VARCHAR( 32 ) NOT NULL ,
  PRIMARY KEY ( `username` )
) TYPE = MYISAM ;
```

خوب حالا اون بالا بر روی **Insert** کلیک کنید و به **username , password** بنویسید و بر روی **Go** کلیک کنید .

PhpMyAdmin امکانات زیادی داره که با کارکردن همه اونا رو یاد می گیرید . اما ما نمی خواهیم این طوری با دیتا بیس کار کنیم ، ما می خواهیم این کارها رو با برنامه نویسی انجام بدیم . مثلا به فرم قشنگ داشته باشیم و اطلاعات رو از کاربر بگیریم و اونا رو توی جدول وارد کنیم یا به جستجو توی جدول انجام بدیم .

برای این که توی **php** با **MySQL** کار کنیم به سری مراحل داره که باید اونا رو به ترتیب انجام بدیم .

اتصال به هاست : منظور از هاست اون کامپیوتری هست که **MySQL** روی اون نصبه و همه داده ها اونجا ذخیره میشن . توی برنامه هایی که ما می نویسیم معمولا همون **localhost** است . برای اتصال به هاست نیاز به **username** , **password** داریم . با استفاده از دستور زیر به هاست متصل می شویم .

کد :

```
mysql_connect( HostName , HostUserName , HostPassword);
```

اگه روی کامپیوتر خودتون دارید کار می کنید برای مثال از این دستور این طوری استفاده کنید :

کد :

```
$myconn = mysql_connect( "localhost" , "root" , "");
if($myconn)
  print ("Connection Successful!");
else
  print("Connection Failed!");
```

یا این طوری :

کد :

```
$myconn = mysql_connect( "۱۲۷.۰.۰.۱" , "root" , "");
if(! $myconn)
```

```
print("Connection Failed!");  
else  
print ("Connection Successful!");
```

البته **myconn** به متغیره که من دوست داشتم اسمش این باشه شما هر چی دوست داشتید بگذارید .

خوب حالا که به هاست کانکت شدیم باید دیتا بیسمون رو انتخاب کنیم . با دستور زیر

```
کد :  
mysql_select_db(MySqlDataBaseName,ConnectionID);
```

برای مثال خودمون این طوری می شه :

```
کد :  
mysql_select_db("MyDB",$myconn);
```

خوب حالا دیتا بیس خودمون رو هم انتخاب کردیم حال هر کاری که خواستیم انجام بدهیم باید براش **Query** بنویسیم همون چیزایی که توی **PhpMyAdmin** هم بهمون نشون می داد .

اجرای یک Query :

دستور اجرای یک **Query** اینه :

```
کد :  
mysql_query(QueryString , ConnectionID);
```

مثال :

```
کد :  
mysql_query("SELECT * FROM users" , $myconn);
```

برای این که به نتیجه از کار امروز رو دیده باشید . کد زیر رو اجرا کنید :

```
کد :  
<?  
$myconn = mysql_connect( "localhost" , "root" , "");  
if($myconn)  
print ("Connection Successful!<br><br>");  
else  
die("<font color=#ff0000> Connection Failed !!! </font><br><br>");  
  
mysql_select_db("MyDB",$myconn);  
$result = mysql_query("SELECT * FROM users" , $myconn);
```

```
for($i = 0 ; $i != mysql_num_rows($result) ; $i++)  
{  
    print(@mysql_result($result,$i,0)." " .@mysql_result($result,$i,1)."<br><br>");  
}  
?>
```

توی جلسه بعد کلا در باره SQL نویسی بحث خواهیم کرد .

جلسه هفتم (SQL) :

امروز در مورد SQL می‌گیم .

SQL چیه ؟ مخفف Structured Query Language و به معنی زبان پرس و جوی ساخت یافته است .

پرس جو ؟

اصطلاح پرس و جو یا Query به این معنی است که ما اطلاعاتی که داریم رو میخوایم باهاش چکار کنیم . مثلاً اون رو بخونیم ، قسمتی از اون رو بخونیم ، تغییرش بدیم ، بهش یه چیزایی اضافه کنیم و ...

مثال :

۱- لیست تمام کاربرانی که از تاریخ ۱۲ فروردین ۱۳۸۵ به بعد ثبت نام کرده اند را می‌خواهم .

۲- حساب کاربری فلانی را قفل کن .

۳- پسورد کاربری که user اون برابر با ali هست رو به ۱۲۳ تغییر بده .

حال ما می‌خواهیم بدونیم که چطوری این‌ها رو به کامپیوتر بفهمونیم .

اکثر Query هایی که ما استفاده می‌کنیم از ۴ حالت زیر خارج نیستند :

۱- Select خواندن ، جستجو ...

۲- Insert درج اطلاعات جدید .

۳- Update به هنگام سازی .

۴- Delete حذف اطلاعات .

SELECT:

اول به مثال می‌زنیم تا گیج نشوید :

کد :

```
SELECT user FROM userTable
```

این Query از تمامی رکورد های جدول userTable فیلد user را پیدا می‌کند .

یکی دیگه :

کد :

```
SELECT user, name, lastName, Id FROM userTable
```

این یکی تمامی فیلد های ذکر شده جلوی دستور SELECT را از جدول userTable پیدا می‌کند .

اگه همه فیلد ها رو بخوایم نشون بده چی ؟ باید همه فیلد ها رو جلوی SELECT بنویسیم؟

برای این کار به جای این که نام همه فیلد ها رو بنویسید می‌تونید از * استفاده کنید یعنی :

کد :

```
SELECT * FROM userTable
```

اگه یه شرط داشتیم چی؟ مثلا "کاربری رو پیدا کن که نام کاربری اون **ali** و پسورد اون ۱۲۳ است..."
برای شرط در **query** ها از **WHERE** استفاده میکنیم. اینطوری :

کد :

```
SELECT * FROM userTable WHERE userName = 'ali'
```

معنی این : تمام رکورد های جدول **userTable** رو پیدا کن که شرط

کد :

```
userName = 'ali'
```

رو داشته باشند .

اگه شرط هامون چند حالت داشته باشند ؟ مثلا این :

کد :

```
SELECT * FROM userTable WHERE userName = 'ali' AND userPassword='123'
```

یعنی : تمام رکورد های جدول **userTable** را پیدا کن که فیلد **userName** اون برابر با **ali** و فیلد **userPassword** اون برابر با ۱۲۳ باشه .

به همین ترتیب میتونه به جای **AND** کلمه **OR** را قرار بدین همچنین میتونین هر شرطی خواستید اضافه کنید .

مرتب سازی داده های خروجی :

برای مرتب کردن خروجی براساس یک فیلد از این دستور توی **Query** استفاده میکنیم .

کد :

```
ORDER BY fieldName ASC
```

که **fieldName** نام فیلدی هست که مرتب سازی بر اساس اون صورت می گیره و اون **ASC** میگه که مرتب سازی صعودی باشه ؛ اگه بخواییم مرتب سازی نزولی باشه به جای **ASC** از **DESC** استفاده می کنیم .

مثال :

کد :

```
SELECT * FROM postTable WHERE postId>50 ORDER BY postTitle ASC
```

یعنی چه ..؟

تمام رکورد های جدول **postTable** رو پیدا کن که مقدار فیلد **postId** اون بزرگتر از ۵۰ باشه و اون رو بر اساس فیلد **postTitle** به صورت صعودی مرتب کن .

اگه نخواهیم داده های تکراری نشون داده بشن چی ؟
از

```
      کد :  
GROUP BY FieldName
```

استفاده میکنیم .

مثلا :

```
      کد :  
SELECT * FROM postTable WHERE senderId=103 GROUP BY posted
```

یعنی : تمام رکورد ها را از جدول **postTable** نمایش بده که فیلد **postId** (کد ارسال کننده) اون برابر ۱۰۳ است .
و پست های تکراری رو حذف کن . (اون رکورد هایی که **postId** مشابهی دارند فقط یکی شون رو نشون بده) .
شاید مثال خوبی نباشه ...

جستجو :

برای جستجو در داده های یه جدول به صورتی که مثلا اگه **ali** رو جستجو کردیم تمامی نتایجی که شامل عبارت **ali** میشود را نشان دهیم .

```
      کد :  
SELECT * FROM postTable WHERE postBody LIKE '%ali%'
```

تمام چیزهایی که توش **ali** باشه رو پیدا میکنه ، اگه بخوایم فقط اونایی که اولشون با **ali** شروع می شه به جای **'%ali%'** فقط آخرش **'%'** می گذاریم یعنی :

```
      کد :  
SELECT * FROM postTable WHERE postBody LIKE 'ali%'
```

برای اونایی که به **ali** ختم می شن وارونه فقط اولش **%** می گذاریم .

داشت یادم می رفت برای شمردن نتیجه یک **SELECT** که فقط تعداد برامون مهم باشه از تابع **COUNT** استفاده می کنیم :

```
      کد :  
SELECT COUNT(*) FROM userTable WHERE registerDate < '85/10/15'
```

البته توابع دیگری ای هم هست که فعلا نمی گم چون ممکنه سر در گم بشید .

INSERT:

این **query** برای درج رکورد جدید توی جدول استفاده میشه .
این طوری :

کد :

```
INSERT INTO myTable (userName, userPassword, userReallyName)
VALUES ('ali', '123', 'Ali Agho')
```

تو چند خط نوشتم که راحتتر خونده بشه البته توصیه می شه توی برنامه هاتون هم اینطوری بنویسید .

خوب این **query** به رکورد جدید رو توی جدول **myTable** درج میکنه که مقدار فیلد **userName** اون **ali** و مقدار فیلد **userPassword** اون **123** و مقدار فیلد **userReallyName** اون **Ali Agho** هست .

تو پرانتز اولی اگه فیلدی رو ننویسید اشکالی نداره (البته اگه فیلد اون **Not NULL** نباشه) مقدار **Default** توش قرار میگیره که اگه موقع ساختن جدول تعیین نشده باشه **Null** خواهد شد . در هر صورت بایستی تعداد فیلدی که توی پرانتز اولی می نویسید با تعداد فیلد پرانتز دوم برابر باشه .

در ضمن میتونید کلا پرانتز اول رو نگذارید اونوقت توی پرانتز دوم برای تمام فیلد ها بایستی مقدار تعیین کنید .

یه نکته :

توی **MySQL** میتونید **n** تا رکورد رو با یه **query** درج کنید . مثال :

کد :

```
INSERT INTO myTable
VALUES
('ali', '123', 'Ali Agho'),
('mojataba', '1100012', 'Agh Mojtaba'),
('reza', '1545587ax1', 'Agha Reza')
```

به همین سادگی .

UPDATE:

برای ویرایش یه چیز استفاده میشه. مثلا تغییر پسورد ، ویرایش یه مطلب و ...

مثال :

کد :

```
UPDATE myTable SET userName='ali'
```

یعنی : برای تمام رکورد های جدول **myTable** مقدار فیلد **userName** رو به **ali** تغییر بده ...

تمام رکورد ها، چرا؟

برای این که شرطی تعیین نکردیم. آگه این طوری بنویسیم اون وقت محدود به رکورد (های) خاصی می شه :

کد :

```
UPDATE myTable SET userName='ali' WHERE userID=53
```

فقط رکورد هایی که مقدار فیلد **userID** اونا برابر با ۵۳ باشه تغییر میکنند .
ما میتونیم همزمان چند تا فیلد رو با هم تغییر بدیم :

کد :

```
UPDATE myTable  
SET userName='ali', userPassword='123', userReallyName = 'Ali Aghoye Gol'  
WHERE userID=53
```

WHERE اینجا مثل توی **SELECT** هست .

DELETE:

برای حذف رکورد از جدول استفاده میشه. مثلا :

کد :

```
DELETE FROM myTable
```

تمامی رکورد های جدول **myTable** رو حذف میکنه .

توجه :

- ۱- اینجا بعد از **DELETE** و قبل از **FROM** چیزی نیست با **SELECT** اشتباه نکنید .
- ۲- باید خیلی مواظب این جور **query** ها باشید . چون بعضی وقت ها به خاطر یه اشتباه کوچولو خیلی چیزا از دست میروند .

اگه بخواهیم رکورد(های) خاصی رو حذف کنیم بایستی از شرط استفاده کنیم مثل **update** :

کد :

```
DELETE FROM myTable WHERE userID=662
```

فقط رکورد هایی رو حذف می کنه که مقدار فیلد **userID** اونا برابر ۶۶۲ باشه .

خوب و این بود **query** .

من سعی کردم خیلی ساده بگم و چیزهایی که فعلا لازم نبود نگفتم ، به موقع که بهش رسیدیم چیزهایی که لازمه رو میگم .

جلسه هشتم (ساخت قسمت جستجو) :

امروز آموزش ساخت یک جستجو در بانک اطلاعاتی رو آموزش میدیم .
توی این قسمت با **Request** ها و بازیابی اطلاعات از دیتا بیس آشنا می شویم .

خوب اول به فرم نیاز داریم که فرم جستجوی ما می شود :

کد :

```
<form method="get" action="search.php">
<p>
<input type="text" name="q" size="20">
<input type="submit" value="Search" name="searchBtn">
</p>
</form>
```

کد بالا رو توی یه فایل با پسوند **html** ذخیره کنید .

حالا برنامه **phpMyAdmin** رو باز می کنیم و یه دیتابیس جدید و توی اون یه جدول جدید می سازیم (البته آگه جدولی که می خواهیم توش بگردیم وجود نداشته باشه) .

برای ساخت جدول از **SQL** زیر استفاده کنید :

کد :

```
CREATE TABLE `text` (
`id` INT NOT NULL AUTO_INCREMENT ,
`title` VARCHAR( 255 ) NOT NULL ,
`body` TEXT NOT NULL ,
PRIMARY KEY ( `id` )
) TYPE = MYISAM ;
```

(توی **phpMyAdmin** پس از انتخاب دیتا بیس مورد نظر بر روی **SQL** کلیک کنید و کد بالا رو اونجا کپی کنید بعد بر روی **Go** کلیک کنید تا جدول مورد نظر ساخته بشه) .

بعد از این که جدول **text** رو ساختید روی **Insert** کلیک کنید و چند تا رکورد توی جدول وارد کنید . (فعلا سعی کنید لاتین وارد کنید تا بعد ...) .

خوب حالا ما به فرم جستجو داریم و یه جدول که یه سری اطلاعات توی اون و میخوایم توی اون جستجو کنیم ...

* خوب بریم سراغ ساختن فایل **search.php** که مهمترین قسمت کاره:

یه ویرایشگر مثل **notepad** یا **Dream Weaver** باز کنید .

موقعی که کاربر بر روی دکمه **search** توی فایل **html** اولی کلیک میکنه اون فرم و تمامی اشیایی که روی اونیه با مقادیرشان میان تو سرور و فایل **search.php** که ما براش تعیین کردیم . به آرایه هست که مال خود **php** هست و تمامی مقادیر فرم ارسال شده توی اون قرار میگیره مثلا مقدار **Edit Box** ی که نام اون رو **q** گذاشتیم میره توی این :

کد :

```
$_GET["q"]
```

این **GET** برای موقعی به کار میره که فرم ما با **get - method** ارسال شده اگه با **post** ارسال شده بود از **POST** استفاده میگردیم (دقت کنید که باید با حروف بزرگ نوشته بشن) .

خوب حالا که میدونیم که کاربر چی وارد کرده پس میریم سراغ قسمت بعد . راستی تا یادم نرفته بگم اگه به کاربر شیطون اومد از راه دیگه ای فایل **search.php** رو باز کرد مثلا آدرس اون رو توی **Address Bar** نوشت ... اونوقت چیزی به نام

کد :

```
$_GET["q"]
```

وجود نداره و برنامه به ما ارور میده ، همچنین توی جاهایی که چیزی رو قراره ثبت کنیم (**Insert**) نباید رکورد خالی ثبت بشه ...

برای این کار بایستی از تابعی به نام **isset** استفاده کنیم ، اینطوری :

کد :

```
if(isset($_GET['q']))
{
...
}
```

این تابع در صورتی که همچین متغیری وجود داشته باشه مقدار **true** بر میگرددونه و در غیر این صورت **false** بر می گردونه . پس ما توی خطوط اولی فایلمون اون رو چک میکنیم .

کد :

```
if(!isset($_GET['q']))
die("The Search key word is not set !!!");
```

در صورتی که کاربر از طریق فرم به این صفحه منتقل نشده باشه به پیغام خطا چاپ می کنیم .

خوب حالا باید چک کنیم که کاربر مقدار مورد جستجو رو خالی نگذاره با این :

کد :

```
$key = $_GET['q'];
if($key == "")
die("The Search key word must be entered !!!");
```

وقتی به اینجا میرسیم دیگه مطمئنیم که کاربر از طریق فرم به اینجا اومده چون اگه اینطور نبود در خطوط بالایی با دستور **die** برنامه به پایان می رسید و به اینجا نمی رسیدیم .

خوب حالا کارهای پایگاه داده (جستجو) :

اول از همه کد های ارتباط با دیتابیس رو که قبلا توضیح داده شده رو مینویسیم :

کد :

```
$DbConn = mysql_connect("localhost", "root", "" ) or die(mysql_error());  
mysql_select_db("test",$DbConn) or die("Can Not Select DataBase Because: "  
mysql_error());
```

خوب حالا که اتصال با موفقیت انجام شد ، بریم سراغ **SQL** :

کد :

```
$sql = "SELECT * FROM `text`  
WHERE `title` LIKE '%" . $key . "%' ";
```

دقت کنید که ما جستجو را بر اساس فیلد **title** از جدول **text** انجام میدیم و هر رکوردی که فیلد **title** اون خود اون یا قسمتی از اون برابر چیزی باشه که کاربر وارد می کنه را برامون پیدا میکنه .

خوب حالا **query** رو اجرا میکنیم :

کد :

```
$SearchResult = mysql_query($sql) or die(mysql_error()). "<br />SQL: ". $sql);
```

اگه تا حالا همه چیز با موفقیت انجام شده باشه با اجرای خط بالا نتایج جستجو را با استفاده از متغیر **SearchResult** به دست می آوریم . اول از همه بیایید ببینیم جستجوی ما چند تا نتیجه داشته :

کد :

```
$TotalResults = mysql_num_rows($SearchResult);
```

خط بالا تعداد نتایج را در متغیر **TotalResults** می ریزه .

خوب حالا تعداد نتایج به دست آمده رو چک میکنیم که اگه نتیجه ای در بر بوده اون(ها) رو چاپ کنیم و اگه نبود پیغام مناسب چاپ کنیم :

کد :

```
if($TotalResults <= 0)  
die("Not found any record for your key word !!!");
```

خوب حالا نتایج رو چاپ میکنیم :

اول باید بدونیم که برای دستیابی به نتایج یک **query** باید از تابع **mysql_result** استفاده میکنیم . با مثال بهتر متوجه می شوید :

```
کد :
for($i = 0; $i != $TotalResults; $i++)
{
    //<!-- Start of Print Results...

    $TextTitle = mysql_result($SearchResult, $i, 1);
    $TextBody = mysql_result($SearchResult, $i, 2);

    print($TextTitle ." ==> ". $TextBody ."<br /><br />");

    //End of Print Results. -->
}
```

توی حلقه بالا تابع **mysql_result** رو میبینیم که با سه ورودی فراخونی شده ورودی اول متغیری هست که نتیجه اجرای **Query** رو توی اون ریختیم ، ورودی دوم شماره رکودی است که توی جستجوی ما پیدا شده و از صفر شروع میشه (چون ممکنه که جستجوی ما چند رکورد را پیدا کند بایستی توی حلقه اونا رو یکی یکی چاپ کنیم ، ورودی سوم اندیس فیلد (ستون) جدول ما است که اون هم از صفر شروع میشه .

و این هم سورس کامل امروز :

```
کد :
<?php
if(!isset($_GET['q']))
    die("The Search key word is not set!");

$key = $_GET['q'];
if($key == "")
    die("The Search key word must be entered!");

$dbConn = mysql_connect("localhost", "root", "" ) or die(mysql_error());
mysql_select_db("test",$dbConn) or die("Can Not Select DataBase Because: ".
mysql_error());

$sql = "SELECT * FROM `text`
WHERE `title` LIKE '%" . $key . "%' ";

$searchResult = mysql_query($sql) or die(mysql_error(). "<br />SQL: ". $sql);

$totalResults = mysql_num_rows($searchResult);

if($totalResults <= 0)
    die("Not found any record for your key word!");

for($i = 0; $i != $totalResults; $i++)
{
```

```
//<!-- Start of Print Results...

$TextTitle = mysql_result($SearchResult, $i, 1);
$TextBody = mysql_result($SearchResult, $i, 2);

print($TextTitle ." ==> " . $TextBody . "<br /><br />");

//End of Print Results. -->
}
?>
```

نکته: فایل **html** ی که اول ساختیم و فایل **search.php** رو توی یه فولدر مشترک باشند .

توجه: اگه خواستید از این کد توی وبسایتتون استفاده عملی بکنید حتما با من مشورت کنید ، به این دلیل که چون این کد جنبه آموزشی داره مسائل امنیتی توی اون رعایت نشده که خیلی پیچیده نشه .

در ضمن این خیلی ساده است فقط برای یه مورد جستجو انجام میده با تلاش بیشتر میتونید **query** های پیچیده بنویسید و خیلی کارای دیگه بکنید مثلا مدت زمان جستجو رو به دست بیارید یا کلمه های کلیدی که مورد جستجو قرار گرفته رو توی نتایج رنگی نشون بدیم و ...

جلسه نهم (ساخت Login) :

از این به بعد چون روش کار کردن با دیتابیس رو یاد گرفتیم می تونیم برنامه های جالب ، به درد بخور و کاربردی بنویسیم .

امروز ساخت یک برنامه **Login** را آموزش میدم به بهانه یاد گیری این طرز استفاده از **Session** ها رو هم یاد میگیرید .

اول از همه نیاز به یک **Table** داریم که نام کاربری و رمز عبور کاربران ما داخل اونیه :

با اجرای **Query** زیر در **PhpMyAdmin** جدول مورد نظر رو بسازید :

کد :

```
CREATE TABLE `user_table` (  
  `user` VARCHAR( 32 ) NOT NULL ,  
  `pass` VARCHAR( 32 ) NOT NULL  
) TYPE = MYISAM ;
```

یه چند تا **user** هم توی اون **Insert** کنید .

بعد یه فرم **Login** درست میکنیم :

کد :

```
<form method="POST" >  
  <p>User name: <input type="text" name="user_name" size="20"></p>  
  <p>Password: <input type="password" name="password" size="20"></p>  
  <p><input type="submit" value="Login" name="login_btn"></p>  
</form>
```

خوب حالا توی فایل **login.php** که قراره با هم بسازیم اول میگیریم که اگه هیچ درخواستی (**Request**) ارسال نشده فرم رو نشون بده . پس در شرط زیر بررسی میکنیم اگر بر روی دکمه ورود کلیک شده است کارهای مربوط به **Login** رو انجام بده و اگر نه توی خطوط بعدی فرم رو نشون می دیم .

کد :

```
<?php  
if(isset($_POST['login_btn']))  
{  
  // ---  
  // ---  
  // ---  
} //End of if(isset($_POST['login_btn']));  
?>
```

خوب حالا فرض کنیم درخواستی داریم یعنی فرم ارسال شده است ، پس باید چک کنیم ببینیم که نام کاربری و پسورد وارد شده درسته یا نه ؟

خوب اول مقادیر دو **Text Box** نام کاربری و کلمه عبور را در دو متغیر میریزیم :

کد :

```
$user_name = $_POST['user_name'];  
$password = $_POST['password'];
```

خوب حالا برای این که ببینیم درست هستند یا نه باید از یه **Query** استفاده کنیم :

کد :

```
$sql = "SELECT `pass` FROM `user_table` WHERE `user` = '$user_name'";
```

و حالا اجرا کردن **Query** :

کد :

```
mysql_connect('localhost','root','');  
mysql_select_db('test');  
$sql = "SELECT `pass` FROM `user_table` WHERE `user` = '$user_name'";  
$result = mysql_query($sql) or die(mysql_error());
```

خوب حالا در صورت پیدا شدن نتیجه ای کلمه عبور دریافتی رو هم چک می کنیم . اول ببینیم که تعداد نتایج **SELECT** چند تا است ، اگه یک نتیجه داریم بریم سراغ بررسی پسورد :

کد :

```
if(mysql_num_rows($result) == 1)  
{  
    //Check Password ...  
    if($password == mysql_result($result,0))  
    {  
        //Logined Ok!  
        print("Ok");  
        exit();  
    }  
}  
print("The User name And/Or password is invalid!");
```

خوب حالا کد کامل صفحه رو می نویسم :

کد :

```
<html>  
  
<head>  
<title>Login Page...</title>
```

```

</head>

<body style="font-family:arial;font-size:10pt">
<?php
if(isset($_POST['login_btn']))
{
$user_name = $_POST['user_name'];
$password = $_POST['password'];

mysql_connect('localhost','root','');
mysql_select_db('test');
$sql = "SELECT `pass` FROM `user_table` WHERE `user` = '$user_name'";
$result = mysql_query($sql) or die(mysql_error());

if(mysql_num_rows($result) == 1)
{
//Check Password ...
if($password == mysql_result($result,0))
{
//Logged Ok!
print("Ok");
exit();
}
}
print("The User name And/Or password is invalid!");
} //End of if(isset($_POST['login_btn']));

?>

<form method="POST">
<p>User name: <input type="text" name="user_name" size="20"></p>
<p>Password: <input type="password" name="password" size="20"></p>
<p><input type="submit" value="Login" name="login_btn"></p>
</form>
</body>

</html>

```

: Session

خوب ما هویت کاربر رو توی این صفحه بررسی کردیم ... اما اگه به یه صفحه دیگه رفتیم چی؟ فرض کنید ما یه صفحه برای کاربر داریم که مثلا توی اون صفحه کاربر میتونه یه چیزی رو ارسال کنه و به یه سری اطلاعات دسترسی داشته باشه ... نمی تونیم توی هر صفحه از کاربر نام کاربری و کلمه عبور رو پرسیم که ...؟ برای این کار باید متغیر هایی داشته باشیم که توی تمامی صفحات سایت قابل دستیابی باشه به این نوع متغیر (Session) می گیم .

جلسه دهم (Session) :

توی جلسه قبل یک برنامه **Login** را مورد بررسی قرار دادیم و با هم به برنامه **Login** نوشتیم ...

خوب فرض کنید کاربر **Login** کرد و **UserName** و **Password** اون درست بود و خواست به صفحات دیگه سایت بره و در صورتی که **Login** کرده باشه به اون صفحات دسترسی داشته باشه یا مثلا امکانات بیشتری بهش بدیم ... خوب وقتی توی اون صفحه اومدیم از کجا بفهمیم که کاربر **Login** کرده ... یکی از راههاش اینه که مثلا به متغیر بگیریم و اون رو **true** کنیم بعد توی هر صفحه اونو چک کنیم ... نه نمی شه چون که وقتی که اجرای یک صفحه به پایان می رسه تمام منابعی که از سیستم گرفته آزاد میشه یکی از این منابع حافظه اصلی (RAM) که متغیرهای برنامه توی اون ذخیره می شن پس این روش به درد نمی خوره ...

خوب پس چکار کنیم ... به روش دیگه هست و اون اینه که توی کامپیوتر کاربر متغیرمون رو ذخیره کنیم ... مگه میشه ؟ بله توی کوکی های مرورگر ... ولی این روش توصیه نمی شه چون که از امنیت بسیار پایینی برخورداره ...

خوب به روش معمول دیگه می مونه که به (Session جلسه) معروفه و روش کار اون از این قراره که به جور متغیر خاص که ما بهش میگیریم رو روی سرور ذخیره میکنه که از توی تمام صفحات سایت بهش دسترسی داریم و برای هر کاربر منحصر به فرد و این دقیقا چیزیه که برای یک سیستم **Login** کاربردی بهش نیاز داریم ... خوب چطوری با هاش کار کنیم ... ؟

به مثال :

کد :

```
$_SESSION['LoggedIn'] = true;
```

خوب توی این مثال **LoggedIn** نام متغیر **session** ما هست و **true** مقداری هست که توی اون قرار دادیم .

نکات مهم :

- ۱- کلمه کلیدی **SESSION** بایستی حتما با حروف بزرگ نوشته شود و به همین شکلی که نوشتیم به کار برده شود .
- ۲- به جای 'میتونید از " هم استفاده کنید .
- ۳- تا وقتی که با دستور صریح مقدار اون رو عوض نکردید خود به خود تغییر نمی کنه یا از بین نمی ره .
- ۴- محدود به استفاده از فقط یک متغیر **Session** نمی باشید. مثلا میتونید هر چند تا که بخواید با نام های مختلف از اونا داشته باشید .
- ۵- در صورتی که کاربر ارتباطی با سرور نداشته باشه بعد از یک زمان مشخص **Session** اون از بین میره. (ارتباط با سرور یعنی این که مثلا یک صفحه از سایت ما را باز کند ...) این مدت زمان مشخص توسط مدیر سرور قابل تنظیم هست .

خوب ببینید به برنامه **Login** کوچولو درست کنیم که اگه کاربر **Login** کرده باشه به یه صفحه مورد نظر دسترسی داشته باشه ...

نکته مهم :

* قبل از این که در یک صفحه **php** از **Session** ها استفاده کنیم بایستی حتما **Session** رو **Start** کنیم .
با نوشتن دستور

```
کد :  
session_start();
```

در اول فایلمون میتونیم از **Session** در برنامهمون استفاده کنیم .

**** نکته مهم :** قبل از این که از دستور

```
کد :  
session_start();
```

استفاده کنیم نباید هیچ خروجی داشته باشیم یعنی از دستور **print** و امثال این نباید استفاده کنیم همچنین **HTML** عادی هم نباید داشته باشیم .

خوب اول از همه صفحه **Login** رو درست میکنیم که یک صفحه **HTML** هست :

```
کد :  
  
<html>  
  
<head>  
<title>Login User...</title>  
</head>  
  
<body>  
  
<form method="POST" action="login.php">  
<p>UserName: <input type="text" name="username" size="20"></p>  
<p>Password:&nbsp;   <input type="password" name="password" size="20"></p>  
<p><input type="submit" value="Login" name="LoginBtn"></p>  
</form>  
  
</body>  
  
</html>
```

اسم این صفحه ای که ساختیم رو میگذاریم **index.html** .

خوب بریم سراغ صفحه **login.php** که توی اون صفحه **user, Password** بررسی میشه
اول از همه دستور

```
کد :  
session_start();
```

بعد ریختن دو فیلد **username, password** توی دوتا متغیر و بررسی اونا (در این قسمت برای این که زیاد پیچیده نشه **user,pass** رو با دیتا بیس چک نمی کنیم ، توی قسمت قبل توضیح داده شد).

کد :

```
<?php
session_start();

$user = strtolower($_POST['username']);
$pass = $_POST['password'];

if($user == 'ali' && $pass == '123')
{
    $_SESSION['LoggedIn'] = true;
    print('<h1>Welcome</h1><br /><br />Click on the below link :<br /><br />');
    print('<a href="user.php">Go to User Page</a>');
}
else{
    print('<font color="#FF0000">The User Name And/Or Password is incorrect!
Please try again...<br /></font>');
    print('<a href="index.html">Back</a>');
}
?>
```

خوب حالا اگه **userName** را **ali** وارد کنیم و کلمه عبور را **۱۲۳** بزنیم وارد میشود و به لینک به ما نشون میده که ما رو به صفحه **user.php** هدایت می کنه .

نکته : اونجایی که نام کاربری رو از متغیر های فرم خوندم اون رو با تابع **strtolower** به حروف کوچک تبدیل کردم تا اگه کاربر نام کاربری خود را بزرگ وارد کرد باز هم بتونه وارد سیستم بشه

کد :

```
$user = strtolower($_POST['username']);
```

خوب حالا بریم سراغ صفحه **user.php** ...

چون می خواهیم از **session** استفاده کنیم اون رو توی خط اول **start** میکنیم. بعد چک میکنیم که اصلا **session** ی که توی صفحه **login.php** ساخته شد وجود داره یا نه ...

کد :

```
<?php
session_start();

if(isset($_SESSION['LoggedIn']))
{
```



```

if($_SESSION['LoggedIn'] == true)
{
    print('<center><h3>Welcome</h3><center>');

    ?>
    <form method="POST" action="user.php">
    <input type="submit" value="Logout" name="LoutBtn">
    </form>
    <?

    exit();
}
}

print('<font color="#FF0000">Access Denied</font>');

?>

```

خوبی حالا بریم سراغ **Logout** کاربر به دکمه توی همین صفحه گذاشتیم اما فعلا کار نمی کنه ...

برای **Logout** کافیه که **Session** رو نابود کنیم ... با دستور :

کد :

```
session_destroy();
```

این کار رو انجام میدیم .

خوب حالا کامل شده **user.php** :

کد :

```

<?php
session_start();

if(isset($_POST['LoutBtn']))
{
    session_destroy();
}

if(isset($_SESSION['LoggedIn']))
{
    if($_SESSION['LoggedIn'] == true)
    {
        print('<center><h3>Welcome</h3><center>');

        ?>
        <form method="POST" action="user.php">
        <input type="submit" value="Logout" name="LoutBtn">
        </form>

```

```
<?
    exit();
}
}
print('<font color="#FF0000">Access Denided</font>');
?>
```

خوب ...

و این بود **Session** به زبان ساده .

در قسمت های بعد توی به برنامه کوچولو که با هم می سازیم تجربیات بیشتری کسب می کنید.

جلسه یازدهم (رشته ها) :

امروز در مورد رشته ها در php صحبت می کنیم .

ali is a good student, یک رشته هست. همین طور که می بینید رشته چیز عجیب و غریبی نیست به مشت کاراکتر که پشت سر هم میان .

تعریف رشته ها در php :

تعریف رشته در php مثل تعریف بقیه متغیر ها است :

کد :

```
$MyStr = "salam";
```

رشته **salam** را در متغیر **MyStr** ریختیم .

دقت کنید که رشته در " " قرار میگیره البته میتونه تک quote هم باشه یعنی این طوری ' ' ، چه فرقی با هم دارند ؟

رشته ها رو میتونیم به راحتی تغییر بدیم یا چاپ کنیم و یا توی دیتا بیس بریزیم ... رشته ها می تونن کد های **HTML** و یا **JavaScript** باشد ... خلاصه هر چیز متنی ...

مثلا به کد زیر دقت کنید :

کد :

```
$MyStr = "salam";  
$MyBrackLine = '<br>';  
  
for($i = 0; $i < 5; $i++)  
{  
    print($i);  
    print(' - ');  
    print($MyStr);  
    print($MyBrackLine);  
}
```

5 تا **salam** زیر هم چاپ میکنه و شماره خط هم میزنه از 0 تا 4 .

دقت کنید که تک **
** که یک تک **HTML** هست هم چاپ می شه اما چون یک تک **HTML** هست ما توی صفحه مرورگر نتیجه اون رو میبینیم (که رفتن به خط بعده). برای دیدن دقیق رشته بایستی **source** صفحه را ببینید (در مرورگر **IE** باید به این مسیر بروید **View -> Source**).

(Literals) کاراکتر های با معنی :

آقا اگه خواستیم یه سری کاراکتر خاص توی رشتهمون باشه باید چکار کنیم مثلا این که بخواهیم یه علامت " وسط رشتهمون باشه ،

ببینید اگه ما رشتهمون رو وسط " " بگذاریم وسط اون نمی تونیم " بگذاریم چون فکر میکنه که اینجا رشته تموم شده و از بقیه اون خطا میگیره ... برای حل این مشکل یه کاراکتر قبل از " میگذاریم و با این کار بهش میگی که به این کاری نداشته باش این مال منه ...

کاراکتر های با معنی :

n خط جدید

r سر سطر

t یک Tab به جلو

\$ علامت دلار

" دابل کوتیشن

هر کدوم از این کاراکتر ها که در رشته ما به صورتی بیابند که قبل از اون باشه توی خروجی معنی خاصی می دهد (همون هایی که جلوش نوشته) .

توجه : توقع نداشته باشید که اگه توی یک رشته n یا r یا t گذاشتید توی مرورگرتون نتیجه رو ببینید چون روی source تاثیر میگذارد نه روی خروجی ... HTML حتما می پرسید پس به چه درد می خوره ...؟

نکته : توی php یه رشته لزومی نداره که توی یه خط باشه مثلا کد زیر کاملا درست است :

کد :

```
$MyString = 'Salam bacheha ...  
hame khubin ...  
Felan Bye';
```

فرق " با ' چیه ؟

حالا که لیترال ها را گفتیم میتونیم فرق این ها رو هم بگیم ...

اگه از کاراکتر های با معنی استفاده کنیم بایستی حتما از " " استفاده کنیم اگه از ' ' استفاده کنیم کاراکتر های با معنی به جای اعمال شدن چاپ می شوند مثلا :

کد :

```
print('Salamnn bacheha');
```

دقیقا رشته داخل ' ' رو چاپ میکنه یعنی خروجی اینه :

کد :

```
Salamnn bacheha
```

و این :

کد :

```
print("Salamnn bacheha");
```

خروجی :

کد :

```
Salam
```

```
bacheha
```

* باز هم دقت کنید که این خروجی در صفحه مرورگر این طوری نیست فقط در **Source** صفحه به این شکل است
اگر بخواهیم توی مرورگر این طوری باشه باید این طوری بنویسیم :

کد :

```
print("Salam<br /><br /> bacheha");
```

عملیات روی رشته ها :

عزیزان این قسمت می خواهیم روی رشته ها عملیات هایی رو انجام بدیم مثل جستجو ، تکه تکه کردن رشته ، به هم چسبوندن رشته ها و ...

- به هم چسبوندن رشته ها :

کد :

```
$Str1 = 'Salam';  
$str2 = "khubi?";  
$str3 = $Str1 . $str2;  
print($str3);
```

در این کد **str1** , **str2** رو به هم می چسبونیم و در **Str3** ریختیم . میبینید که به همین راحتی با به نقطه به هم می چسبند .

یه نکته :

کد :

```
$Str1 = 'Salam';  
$str2 = "khubi?";  
$Str1 .= $str2;
```

این کد هم **Str2** را به انتهای **Str1** می چسبونه .

- بریدن رشته ها :

با استفاده از تابع **substr** میتونیم قسمتی از یک رشته را ببریم

کد :

```
$Str1 = 'Salam bacheha khubin';  
$str2 = substr($Str1, 15);
```

این کد از کاراکتر ۱۵ به بعد از رشته **Str1** را می برد و در رشته **Str2** می ریزد .

کد :

```
$Str1 = 'Salam bacheha khubin';  
$str2 = substr($Str1, 5,10);
```

این کد از کاراکتر ۵ تا ۱۰ را از رشته **Str1** را می برد و در رشته **Str2** می ریزد .

- جستجو در رشته ها :

تابع **strpos** این کار را برای ما انجام میدهد ...

کد :

```
$Str1 = 'Salam bacheha khubin';  
print(strpos($Str1, 'bacheha'));
```

این کد عدد ۶ را چاپ می کند چون کلمه **bacheha** کاراکتر ششم است (شمارش از صفر شروع میشه). اگه پیدا نشد هیچی چاپ نمیکنه .

ما میتونیم بهش بگیم که از کاراکتر چندم به بعد رو بگرده :

کد :

```
$Str1 = 'Salam bacheha khubin';  
print(strpos($Str1, 'bacheha',10));
```

این کد از کاراکتر ۱۰ به بعد به دنبال **bacheha** می گرده .

- حذف Space های اضافی :

با استفاده از **trim** فضاهای خالی یک رشته رو حذف می کنیم :

کد :

```
$Str1 = ' Salam ';  
print(trim($Str1));
```

خروجی این کد فقط کلمه **Salam** است .

- جایگزینی یک رشته با رشته دیگر :

کد :

```
$Str1 = 'Salam bacheha khufin ... !!?';  
$Str2 = str_replace("salam", "Hello", $Str1);
```

این کد توی رشته **Str1** به دنبال کلمه **salam** میگرده و هر جا اونو دید کلمه **Hello** رو می گذاره جای اون بعد خروجی رو توی **Str2** میریزه .

جلسه دوازدهم (File Database):

آموزش php قسمت سیزدهم (دیتابیس فایلی)

اگر شما دیتابیس ندارید و نمی توانید به آن دست رسی پیدا کنید، میتوانید از DBM خود پی اچ پی استفاده کنید . همه کاری هم برایتون انجام میده ، بدون اجرا دستورات اس کیو ال فقط چند تا دستور برای خودش دارد . با استفاده از دستور dbmopen() میتونید دیتابیس را اگر نبود بسازید و اگر بود باز کنید . این دستور دو تا ورودی میگیره ...

اولی میسر فایل و دیگری برای انجام عملیات است :

برای بازیابی اطلاعات r

برای بازیابی اطلاعات و اضافه کردن w

برای ساختن دیتابیس و اگر وجود دارد بازیابی اطلاعات و اضافه کردن c

ساختن یک دیتابیس جدید n

پس از کار با دیتابیس آن را با دستور dbmclose() میندیم تا برنامه به مشکل بر نخورد .

برای درج کردن اطلاعات به این صورت عمل میکنیم :

کد :


```
<?php
$db = dbmopen("./data","c") or die("Could not open DBM");
dbminsert($db, "Name", "rightclick");
dbmclose($db);
?>
```

اول اومدم و dbmopen() رو تو یک متغیر ریختم این تو اس کیو ال میشه مثل Table ، و با استفاده از دستور dbminsert() که سه تا ورودی که اولی اطلاعات اون فایلی که ساختیم قرار میگیره ، دومی میشه نام فیلد مون تو اس کیو ال و سومی هم مقدار توش است .

برای تغییر اطلاعات یا همون بروز رسانی به این صورت عمل میکنیم :

کد :

```
<?php
$db = dbmopen("./data","c") or die("Could not open DBM");
dbmreplace($db,"Name","Mahdi");
dbmclose($db);
?>
```

خط اول و سوم را می دونیم چه هست؟! 

خط دوم اولیش ورودی اطلاعات فایلمون هست ، دومی هم نام فیلمون و سومی هم مقدار آن است که من فیلد name را مقدارش را از **rightclick** به **mahdi** تغییر دادم .

برای خواندن اطلاعات به صورت زیر عمل میکنیم :

```
کد :
<?php
$db = dbmopen("./data","c") or die("Could not open DBM");
dbmfetch($db, "Name");
dbmclose($db);
?>
```

خط اول و سوم را می دونیم چه هست؟! 😊

دستور **dbmfetch()** دو تا ورودی میگیره که اولی اطلاعات مربوط به فایل و دومی نام فیلمون است .

برای پاک کردن یک فیلد هم از این کد استفاده می کنیم :

```
کد :
<?php
$db = dbmopen("./data","c") or die("Could not open DBM");
dbmdelete($db, "Name");
dbmclose($db);
?>
```

توضیح خط دوم : دستور **dbmdelete()** دو تا ورودی میگیره که اولی اطلاعات مربوط با فایل است و دومی نام فیلد برای پاک شدن .

با دستور زیر هم میشه فهمید توی فیلد چیزی ثبت شده یا نه ، اگر ثبت شده باشه مقدارش برابر **true** و اگر ثبت نشده باشه مقدارش برابر **false** می شود .

توضیحات : اولی اطلاعات مربوط به فایل و دومی نام فیلد است .

```
کد :
dbmexists($db, "Name");
```

برای نمایش اولین فیلد با استفاده از دستور زیر پیدا می کنند :

```
کد :
dbmfirstkey($db);
```

فقط یک ورودی میگیره اون هم اطلاعات فایل است .

برای نمایش فیلد های بعدی از دستور زیر استفاده کنید :

کد :

```
dbmnextkey($db, $key);
```

اولین ورودیش مربوط به اطلاعات فایل است و دومین ورودیش مربوط به فیلد قبلی فیلد مورد نظر شما .

جلسه سیزدهم (تابع Date):

این جلسه، آموزش تابع تاریخ را میزوارم که هر کی تو جایی از آن مشکل داره مشکلتش رفع بشه .
اول اینکه تابع `date()` دو تا ورودی میگیره که اولیش مربوط به قالب نمایش است و دومیش اگر ندهیم اتفاق خاصی نمی افتد اگر بهش چیزی دادیم باید بهش تاریخ مورد نظر را بدهیم تا با آن فرمتی که بهش دادیم نمایش بده .

جدول نمایش قالب این تابع؛ این شکلی است :

A قبل از ظهر و یا بعد از ظهر را نشان می دهد / قبل از ظهر - بعد از ظهر

a ق.ظ و یا ب.ظ را نشان می دهد / ق.ظ - ب.ظ

d چندمین روز از ماه را به صورت عدد دورقمی / ۰۱ - ۰۲ - ۰۳ - ... - ۳۱

D نام روز هفته را با یک کلمه / ی - د - ... - ج

F نام کامل ماه / فروردین - اردیبهشت - ... - اسفند

g نمایش ساعت به صورت ۱۲ ساعته / ۱ - ۲ - ۳ - ... - ۱۲

G نمایش ساعت به صورت ۲۴ ساعته / ۰ - ۱ - ۲ - ... - ۲۳

h نمایش ساعت به صورت ۱۲ ساعته دورقمی / ۰۱ - ۰۲ - ... - ۱۲

H نمایش ساعت به صورت ۲۴ ساعته دو رقمی / ۰۰ - ۰۱ - ۰۲ - ... - ۲۳

i نمایش دقیقه / ۰۰ - ۰۱ - ۰۲ - ... - ۵۹

I (lower L) نمایش روزهای ماه به صورت عدد / ۱ - ۲ - ... - ۱۲

m نمایش ماه به صورت عدد دورقمی / ۰۱ - ۰۲ - ... - ۱۲

n نمایش ماه به صورت عدد / ۰ - ۱ - ۲ - ... - ۱۲

s نمایش ثانیه یک دقیقه / ۰۱ - ۰۲ - ... - ۵۹

t تعداد روزها مشخص شده ماه (خورشیدی) / ۲۹

w چندمین روز هفته به صورت یک عدد تک رقمی / (یکشنبه) ۱ - ۲ - ... - ۶

y نمایش سال به صورت عدد دورقمی / ۸۵ - ۸۶ ...

Y نمایش سال به صورت عدد چهار رقمی / ۱۳۸۵ - ۱۳۸۶ ...

حالا برای قرار دادن تاریخ در پارامتر دوم از این دستور استفاده کنید (`strtotime()` یا اعدادی که با این دستور تبدیل شده اند استفاده کنید چون در این صورت تاریخ اشتباهی برایتان بوجود می آورد .

برای بدست آوردن وقت **گیرنویج** هم میتونید از این تابع استفاده کنید :

کد :

```
gmdate()
```

این تابع هم مثل تابع بالایی است .

حالا اگر بخواهید زمان حال را با این تابع حساب کنید، این اعمال را انجام می دهید :

کد :

```
gmdate("Y-m-d H:i:s", strtotime(time() + (3.5 * 3600)));
```

برای گذاشتن تاریخ از این کد استفاده کن :

کد :

```
date("l j F Y g:i A");
```

که **l j F Y g:i A** قالب نمایش ساعت و تاریخ است (طبق اون قاعده ای که بالا گفتم).
حالا اگر خواستی تاریخی را که می خواهی نمایش بده به جای ۱۹۹۱-۱۲-۰۱ ۱۲:۳۰:۰۰ در کد زیر تاریخ را به میلادی وارد کن .

کد :

```
date("l j F Y g:i A", strtotime("1991-12-01 12:30:00"));
```

برای گذاشتن تاریخ فارسی هم از این کد استفاده کنید :

کد :

```
<?php
/*
Jalali Date function by Milad Rastian (miladmovie AT yahoo DOT com)
```

```
//The main function which convert Gregorian to Jalali calendars is:
// Copyright (C) 2000 Roozbeh Pournader and Mohammad Toossi
//you can see complete note of those function in down of the page
```

```
AND JALAI DATE FUNCTION
this function is simillar than date function in PHP
you can find more about it in http://jdf.farsiprojects.com
Copyright (C)2003 FARSI PROJECTS GROUP
```

```
    // /\      // /\      //      //
    // // \    // // \    //      //
    // // \    // // \    //      //
\ // // // \ // // // \ // //
\ \ // // \ \ // // \ \ // // //
```

```
//////// // \ // // // //
// // // \ // //
// // // \ // //
// // // \ // //
//////// // \ // //
```

```
*/
```

```
function jdate($type,$smaket="now")
{
//set 1 if you want translate number to farsi or if you don't like set 0
```

```

$transnumber=0;
///chosse your timezone
$TZhours=0;
$TZminute=0;
$need="";
$result1="";
$result="";
if($maket=="now"){
    $year=date("Y");
    $month=date("m");
    $day=date("d");
    list( $jyear, $jmonth, $jday ) = gregorian_to_jalali($year, $month, $day);

    $maket=mktime(date("H")+ $TZhours,date("i")+ $TZminute,date("s"),date("m"),date("d"),date("Y"));
} else{
    // $maket=0;
    $maket+= $TZhours*3600+ $TZminute*60;
    $date=date("Y-m-d", $maket);
    list( $year, $month, $day ) = preg_split ( '/-/', $date );

    list( $jyear, $jmonth, $jday ) = gregorian_to_jalali($year, $month, $day);
}

$need= $maket;
$year=date("Y", $need);
$month=date("m", $need);
$day=date("d", $need);
$i=0;
$subtype="";
$subtypetemp="";
list( $jyear, $jmonth, $jday ) = gregorian_to_jalali($year, $month, $day);
while($i<strlen($type))
{
    $subtype=substr($type,$i,1);
    if($subtypetemp=="\")
    {
        $result.= $subtype;
        $i++;
        continue;
    }

    switch ($subtype)
    {

        case "A":
            $result1=date("a", $need);
            if($result1=="pm") $result.= "بعداز ظهر";
            else $result.= "قبل از ظهر";
            break;

        case "a":
            $result1=date("a", $need);
            if($result1=="pm") $result.= "ب.ظ";

```

```

else $result.="ق.ظ";
break;
case "d":
if($jday<10)$result1="0".$jday;
else $result1=$jday;
if($transnumber==1) $result.=Convertnumber2farsi($result1);
else $result.=$result1;
break;
case "D":
$result1=date("D",$need);
if($result1=="Thu") $result1="پ";
else if($result1=="Sat") $result1="ش";
else if($result1=="Sun") $result1="ی";
else if($result1=="Mon") $result1="د";
else if($result1=="Tue") $result1="س";
else if($result1=="Wed") $result1="چ";
else if($result1=="Thu") $result1="پ";
else if($result1=="Fri") $result1="ج";
$result.=$result1;
break;
case "F":
$result.=monthname($jmonth);
break;
case "g":
$result1=date("g",$need);
if($transnumber==1) $result.=Convertnumber2farsi($result1);
else $result.=$result1;
break;
case "G":
$result1=date("G",$need);
if($transnumber==1) $result.=Convertnumber2farsi($result1);
else $result.=$result1;
break;
case "h":
$result1=date("h",$need);
if($transnumber==1) $result.=Convertnumber2farsi($result1);
else $result.=$result1;
break;
case "H":
$result1=date("H",$need);
if($transnumber==1) $result.=Convertnumber2farsi($result1);
else $result.=$result1;
break;
case "i":
$result1=date("i",$need);
if($transnumber==1) $result.=Convertnumber2farsi($result1);
else $result.=$result1;
break;
case "j":
$result1=$jday;
if($transnumber==1) $result.=Convertnumber2farsi($result1);
else $result.=$result1;
break;
case "l":

```

```

$result1=date("l",$need);
if($result1=="Saturday") $result1="شنبه";
else if($result1=="Sunday") $result1="یکشنبه";
else if($result1=="Monday") $result1="دوشنبه";
else if($result1=="Tuesday") $result1="سه شنبه";
else if($result1=="Wednesday") $result1="چهارشنبه";
else if($result1=="Thursday") $result1="پنجشنبه";
else if($result1=="Friday") $result1="جمعه";
$result.= $result1;
break;
case "m":
if($jmonth<10) $result1="0".$jmonth;
else $result1=$jmonth;
if($transnumber==1) $result.=Convertnumber2farsi($result1);
else $result.= $result1;
break;
case "M":
$result.=short_monthname($jmonth);
break;
case "n":
$result1=$jmonth;
if($transnumber==1) $result.=Convertnumber2farsi($result1);
else $result.= $result1;
break;
case "s":
$result1=date("s",$need);
if($transnumber==1) $result.=Convertnumber2farsi($result1);
else $result.= $result1;
break;
case "S":
$result.="م";
break;
case "t":
$result.=lastday ($month,$day,$year);
break;
case "w":
$result1=date("w",$need);
if($transnumber==1) $result.=Convertnumber2farsi($result1);
else $result.= $result1;
break;
case "y":
$result1=substr($jyear,2,4);
if($transnumber==1) $result.=Convertnumber2farsi($result1);
else $result.= $result1;
break;
case "Y":
$result1=$jyear;
if($transnumber==1) $result.=Convertnumber2farsi($result1);
else $result.= $result1;
break;
case "U" :
$result.=mktime();
break;
case "Z" :

```

```

    $result.=days_of_year($jmonth,$jday,$jyear);
    break;
case "L" :
    list( $tmp_year, $tmp_month, $tmp_day ) = jalali_to_gregorian(1384, 12, 1);
    echo $tmp_day;
    /*if(lastday($tmp_month,$tmp_day,$tmp_year)==="31")
        $result.="1";
    else
        $result.="0";
    */
    break;
default:
    $result.=$subtype;
}
$subtypetemp=substr($type,$i,1);
$i++;
}
return $result;
}

function jmaketime($hour="", $minute="", $second="", $jmonth="", $jday="", $jyear="")
{
    if(!$hour && !$minute && !$second && !$jmonth && !$jmonth && !$jday && !$jyear)
        return mktime();
    list( $year, $month, $day ) = jalali_to_gregorian($jyear, $jmonth, $jday);
    $i=mktime($hour,$minute,$second,$month,$day,$year);
    return $i;
}

//Find num of Day Beginning Of Month ( 0 for Sat & 6 for Sun)
function mstart($month,$day,$year)
{
    list( $jyear, $jmonth, $jday ) = gregorian_to_jalali($year, $month, $day);
    list( $year, $month, $day ) = jalali_to_gregorian($jyear, $jmonth, "1");
    $timestamp=mktime(0,0,0,$month,$day,$year);
    return date("w",$timestamp);
}

//Find Number Of Days In This Month
function lastday ($month,$day,$year)
{
    $jday2="";
    $jdate2="";
    $lastdayen=date("d",mktime(0,0,0,$month+1,0,$year));
    list( $jyear, $jmonth, $jday ) = gregorian_to_jalali($year, $month, $day);
    $lastdatep=$jday;
    $jday=$jday2;
    while($jday2!="1")
    {
        if($day<$lastdayen)
        {

```



```

$day++;
list( $jyear, $jmonth, $jday2 ) = gregorian_to_jalali($year, $month, $day);
if($jdate2=="1") break;
if($jdate2!="1") $lastdatep++;
}
else
{
$day=0;
$month++;
if($month==13)
{
$month="1";
$year++;
}
}
}
return $lastdatep-1;
}

//Find days in this year untile now
function days_of_year($jmonth,$jday,$jyear)
{
$year="";
$month="";
$year="";
$result="";
if($jmonth=="01")
return $jday;
for ($i=1;$i<$jmonth || $i==12;$i++)
{
list( $year, $month, $day ) = jalali_to_gregorian($jyear, $i, "1");
$result+=lastday($month,$day,$year);
}
return $result+$jday;
}

//translate number of month to name of month
function monthname($month)
{
if($month=="01") return "فروردین";
if($month=="02") return "اردیبهشت";
if($month=="03") return "خرداد";
if($month=="04") return "تیر";
if($month=="05") return "مرداد";
if($month=="06") return "شهریور";
if($month=="07") return "مهر";

```

```

if($month=="08") return "آبان";
if($month=="09") return "آذر";
if($month=="10") return "دي";
if($month=="11") return "بهمن";
if($month=="12") return "اسفند";
}

function short_monthname($month)
{
if($month=="01") return "فرو";
if($month=="02") return "ارد";
if($month=="03") return "خرد";
if($month=="04") return "تير";
if($month=="05") return "مرد";
if($month=="06") return "شهر";
if($month=="07") return "مهر";
if($month=="08") return "آبا";
if($month=="09") return "آذر";
if($month=="10") return "دي";
if($month=="11") return "بهم";
if($month=="12") return "اسف";
}

////here convert to number in persian
function Convertnumber2farsi($string)
{
$num0="۰";
$num1="۱";
$num2="۲";
$num3="۳";
$num4="۴";
$num5="۵";
$num6="۶";
$num7="۷";
$num8="۸";
$num9="۹";

```

```

$stringtemp="";
$len=strlen($srting);
for($sub=0;$sub<$len;$sub++)
{
if(substr($srting,$sub,1)=="0")$stringtemp.=$num0;
elseif(substr($srting,$sub,1)=="1")$stringtemp.=$num1;
elseif(substr($srting,$sub,1)=="2")$stringtemp.=$num2;
elseif(substr($srting,$sub,1)=="3")$stringtemp.=$num3;
elseif(substr($srting,$sub,1)=="4")$stringtemp.=$num4;
elseif(substr($srting,$sub,1)=="5")$stringtemp.=$num5;
elseif(substr($srting,$sub,1)=="6")$stringtemp.=$num6;
elseif(substr($srting,$sub,1)=="7")$stringtemp.=$num7;
elseif(substr($srting,$sub,1)=="8")$stringtemp.=$num8;
elseif(substr($srting,$sub,1)=="9")$stringtemp.=$num9;
else $stringtemp.=substr($srting,$sub,1);

}
return $stringtemp;

}///end conver to number in persian

function is_kabise($year)
{
if($year%4==0 && $year%100!=0)
return true;
return false;
}

function jcheckdate($month,$day,$year)
{
$j_days_in_month = array(31, 31, 31, 31, 31, 31, 30, 30, 30, 30, 30, 29);
if($month<=12 && $month>0)
{
if($j_days_in_month[$month-1]>=$day && $day>0)
return 1;
if(is_kabise($year))
echo "Asdsd";
if(is_kabise($year) && $j_days_in_month[$month-1]==31)
return 1;
}

return 0;

}

function jtime()
{
return mktime() ;
}

function jgetdate($timestamp="")
{
if($timestamp=="")

```

```

$timestamp=mktime();

return array(
    0=>$timestamp,
    "seconds"=>jdate("s",$timestamp),
    "minutes"=>jdate("i",$timestamp),
    "hours"=>jdate("G",$timestamp),
    "mday"=>jdate("j",$timestamp),
    "wday"=>jdate("w",$timestamp),
    "mon"=>jdate("n",$timestamp),
    "year"=>jdate("Y",$timestamp),

    "yday"=>days_of_year(jdate("m",$timestamp),jdate("d",$timestamp),jdate("Y",$timestamp))
,
    "weekday"=>jdate("l",$timestamp),
    "month"=>jdate("F",$timestamp),
);
}

// "jalali.php" is convertor to and from Gregorian and Jalali calendars.
// Copyright (C) 2000 Roozbeh Pournader and Mohammad Toossi
//
// This program is free software; you can redistribute it and/or
// modify it under the terms of the GNU General Public License
// as published by the Free Software Foundation; either version 2
// of the License, or (at your option) any later version.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// A copy of the GNU General Public License is available from:
//
// <a href="http://www.gnu.org/copyleft/gpl.html"
target="_blank">http://www.gnu.org/copyleft/gpl.html</a>
//

function div($a,$b) {
    return (int) ($a / $b);
}

function gregorian_to_jalali ($g_y, $g_m, $g_d)
{
    $g_days_in_month = array(31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
    $j_days_in_month = array(31, 31, 31, 31, 31, 31, 30, 30, 30, 30, 30, 29);

```

```
$gy = $g_y-1600;
$gm = $g_m-1;
$gd = $g_d-1;

$g_day_no = 365*$gy+div($gy+3,4)-div($gy+99,100)+div($gy+399,400);

for ($i=0; $i < $gm; ++$i)
    $g_day_no += $g_days_in_month[$i];
if ($gm>1 && (($gy%4==0 && $gy%100!=0) || ($gy%400==0)))
    /* leap and after Feb */
    $g_day_no++;
$g_day_no += $gd;

$j_day_no = $g_day_no-79;

$j_np = div($j_day_no, 12053); /* 12053 = 365*33 + 32/4 */
$j_day_no = $j_day_no % 12053;

$jy = 979+33*$j_np+4*div($j_day_no,1461); /* 1461 = 365*4 + 4/4 */

$j_day_no %= 1461;

if ($j_day_no >= 366) {
    $jy += div($j_day_no-1, 365);
    $j_day_no = ($j_day_no-1)%365;
}

for ($i = 0; $i < 11 && $j_day_no >= $j_days_in_month[$i]; ++$i)
    $j_day_no -= $j_days_in_month[$i];
$j_m = $i+1;
$j_d = $j_day_no+1;

return array($jy, $j_m, $j_d);
}

function jalali_to_gregorian($j_y, $j_m, $j_d)
{
    $g_days_in_month = array(31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
    $j_days_in_month = array(31, 31, 31, 31, 31, 31, 30, 30, 30, 30, 30, 29);

    $jy = $j_y-979;
    $jm = $j_m-1;
    $jd = $j_d-1;

    $j_day_no = 365*$jy + div($jy, 33)*8 + div($jy%33+3, 4);
    for ($i=0; $i < $jm; ++$i)
        $j_day_no += $j_days_in_month[$i];

    $j_day_no += $jd;

    $g_day_no = $j_day_no+79;
```

```

$gy = 1600 + 400*div($g_day_no, 146097); /* 146097 = 365*400 + 400/4 - 400/100 +
400/400 */
$g_day_no = $g_day_no % 146097;

$leap = true;
if ($g_day_no >= 36525) /* 36525 = 365*100 + 100/4 */
{
    $g_day_no--;
    $gy += 100*div($g_day_no, 36524); /* 36524 = 365*100 + 100/4 - 100/100 */
    $g_day_no = $g_day_no % 36524;

    if ($g_day_no >= 365)
        $g_day_no++;
    else
        $leap = false;
}

$gy += 4*div($g_day_no, 1461); /* 1461 = 365*4 + 4/4 */
$g_day_no %= 1461;

if ($g_day_no >= 366) {
    $leap = false;

    $g_day_no--;
    $gy += div($g_day_no, 365);
    $g_day_no = $g_day_no % 365;
}

for ($i = 0; $g_day_no >= $g_days_in_month[$i] + ($i == 1 && $leap); $i++)
    $g_day_no -= $g_days_in_month[$i] + ($i == 1 && $leap);
$gm = $i+1;
$gd = $g_day_no+1;

return array($gy, $gm, $gd);
}
?>

```

فقط به جای `date()` از `jdate()` استفاده کنید.

کد امنیتی :

سلام دوستان امروز آموزش ساخت کد امنیتی رو برای شما گذاشتم ...



قبل از همه باید بدونیم که اصلا چه نیازی هست به این کار ...

ببینید شما یه فرم توی سایتتون دارید مثلا ارتباط با ما حالا یه نفر یه برنامه کوچولو مینویسه و هی فرم شما رو با یه سری مقادیر **submit** میکنه و یه دفعه میبینید که ۹۹۹۹۹۹۹۹۹۹۹۹۹۹ میل از فرمتون اومده توی **Inbox** تون ... 😊

حالا میخوایم یه کاری بکنیم که فقط آدمیزاد بتونه برامون چیز بفرسته ...

برای این که کد امنیتی داشته باشیم پیشنهاد من اینه که یه فایل داشته باشیم برای ساخت تصویر و هر جا که خواستیم ازش استفاده کنیم .

فایل تولید تصویر :

```
کد :
<?
//SecImage.php
session_start();

if( !isset( $_SESSION['SecImageStr']))
exit();

$Str = $_SESSION['SecImageStr'];

$SecImage = imagecreate( 50, 20);

$BgColor = imagecolorallocate( $SecImage, 255, 150, 220);//Background color
$FrColor = imagecolorallocate( $SecImage, 0, 0, 255);//ForeGround color
$LineColor = imagecolorallocate( $SecImage, 255, 0, 150);//Line color

for($Index = 0; $Index != 3; $Index++)
{
    $LineDegree = rand(15, 50);
    imageline ( $SecImage, $Index, $LineDegree, ($Index+1) * 20, $Index,
$LineColor );
}

imagestring ( $SecImage, 4, 5, 1, $Str, $FrColor );
imagejpeg ( $SecImage , "", 100 ) ;

header("Content-type: image/jpg");

imagedestroy ( $SecImage ) ;

?>
```

خوب حالا جایی که میخوایم استفاده کنیم:

کد :

```
<?
//test.php
session_start();
// Create Security Code ...

$SecCode = md5(rand(0,9999));
$SecCode = strtolower($SecCode);
$_SESSION['SecImageStr'] = strtoupper(substr($SecCode,0,4));

?>

```

خوب بعد از این که کد بالا رو توی یه فرم گذاشتیم برای این که چک کنیم ببینیم درسته یا نه بریم توی فایل که تو **Action** فرم مینویسیم:

کد :

```
<?
//check.php
session_start();

$UserSecCode = strtolower($_POST['SecCode']);
$SysSecCopde = strtolower($_SESSION['SecImageStr']);

if($UserSecCode != $SysSecCopde)
{
    die("The Sec Code is invalid!");
}

//... ادامه برنامه ...
?>
```

به همین سادگی ... فقط دقت کنید که این مقاله رو به این خاطر نوشتم که در این زمینه اکثر جاهایی که دیدم امنیتش پایینه و فقط یه کد تولید میکنه و اونو به **client** ارسال میکنه ولی اینجا ما از **session** ها استفاده کردیم .

نکته : برای ایجاد تصویر بایستی کتابخانه **GD** روی سرور شما نصب و فعال باشه .

برای سایت خود یک سیستم رای گیری بسازیم ...

در این مقاله می خواهیم یک اسکریپت **php** رو بهتون نشون بدم که باهاش می شه یه قسمت رای گیری برای سایتمون درست کنیم (مثل همون چیزی که توی صفحه اول www.RightClick.ir هست).

برای این کار ما فقط از یک جدول استفاده می کنیم ، جدول را مطابق این دستورات می سازیم :

کد :

```
CREATE TABLE `poll` (  
  `id` int(11) NOT NULL auto_increment,  
  `topic` text NOT NULL,  
  `type` char(1) NOT NULL default "",  
  `count` int(11) NOT NULL default '0',  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM COMMENT='My Web Poll' AUTO_INCREMENT=14 ;
```

خوب حالا یه توضیح کوچولو در مورد جدول می دم :

این جدول 4 تا فیلد داره که **id** فیلد **Auto increment** است و **topic** سوال رای گیری و گزینه های رای گیری رای گیری که فقط یکی است در این فیلد **q** قرار می دهیم و برای گزینه ها **s** قرار می دهیم .
فیلد **count** هم که تعداد رای برای هر گزینه را در خود نگه می دارد است همچنین در فیلد **type** مشخص میکنیم که این رکورد سوال ما است یا یکی از گزینه های ما است ... برای سوال .

نکته مهم : چرا از دو جدول استفاده نکردیم ؟ **?** برای این که سرعت دستیابی به یک جدول خیلی بیشتر از دو جدول است .

خوب بریم سراغ کد :

کد :

```
<?php  
$host = "localhost";  
$db_name = "temp";  
$db_user = "root";  
$db_pass = "MyPassword";  
  
$link = mysql_connect($host,$db_user,$db_pass) or die(mysql_error());  
mysql_select_db($db_name,$link) or die("Can not select Data base!");  
  
//Check Action Set ...  
if(isset($_GET['action']))  
{  
  if($_GET['action'] == 'vote')  
  {  
    $poll_id = (isset($_POST['webpoll'])) ? $_POST['webpoll'] : "";  
    if($poll_id != "")
```

```

{
    $sql = "UPDATE `poll` SET `count`=count+1 where `id`='". $poll_id.'" AND
`type`='s'";
    $result = mysql_query($sql) or die(mysql_error());
}
}
$count = 0;
$sql = "SELECT * FROM `poll` WHERE `type`='s'";
$result = mysql_query($sql) or die(mysql_error());
$n = mysql_num_rows($result);
for ($i=0;$i<$n;$i++)
{
    $count += mysql_result($result,$i,3);
};
if($count == 0){ $count = 1;};
$output = 'The Result of Vote<br><br>';
for ($i=0;$i<$n;$i++)
{
    $num = mysql_result($result,$i,3) * 100;
    $num /= $count;
    $num = (int) $num;
    $output .= mysql_result($result,$i,1).' '.$num.'%<table border="0"
width="'. $num.'%" height="20" bgcolor="#0000FF"><tr><td></td></tr></table>';
};

    $output .= '<br>The webPoll Result From '.$count.' Votes<br><a target="_self"
href="'. $ _SERVER['PHP_SELF'].'" >[Back]</a>';
}
else

{
//Web Poll...

$sql = "SELECT * FROM `poll` WHERE `type`='s'";
$poll = mysql_query($sql) or die(mysql_error());

$topic_sql = "SELECT `topic` FROM `poll` WHERE `type`='q'";
$poll_result = mysql_query($topic_sql) or die(mysql_error());
$poll_question = mysql_result($poll_result,0,0);

$poll_total = mysql_num_rows($poll);

$output = '<p align="center" dir="ltr">'.$poll_question.'</p><form method="POST"
action="?action=vote">';

$vote_count = 0;

for ($i = 0; $i < $poll_total; $i++)
{
    $vote_count += mysql_result($poll,$i,3);

    $poll_value = mysql_result($poll,$i,0);
    $poll_name = mysql_result($poll,$i,1);

```

```

$output .= '<input type="radio" value="'. $poll_value.'" name="webpoll"
style="CURSOR:Hand;" >'. $poll_name.'<br>';

}

$output .= '<input type="submit" value="vote" name="submit_poll" > <br><br><a
title="View Poll Result" target="_self" href="?action=show_result" >Total Votes :
'. $vote_count.'</a>';

} //End of Action Set

//-----
//Print the output

print($output);

?>

```

راهنمایی : این کد را در فایل‌ها با هر نامی که خواستید و با پسوند **php** ذخیره کنید و اجرا نمایید .

در

```

کد :
$db_name = "temp";

```

به جای **temp** ؛ نام دیتابیس خود را بنویسید .

در

```

کد :
$db_user = "root";

```

به جای **root** ؛ نام کاربری دیتابیس خود را بنویسید .

در

```

کد :
$db_pass = "MyPassword";

```

به جای **My Password** ؛ کلمه عبور دیتابیس خود را بنویسید .

و اما بررسی این کد :

اساس کار این کد به این صورت است که تعداد آرای هر گزینه را بر تعداد کل آرا تقسیم می کند و در صد آن را به دست می آورد و بر اساس آن با استفاده از تگ **<table>** برای هر گزینه یک جدول ترسیم می کند .

در خط ۷ و ۸ اتصال به هاست و انتخاب دیتابیس را انجام می دهیم .

از خط ۱۰ تا ۴۲ مربوط به ثبت آرا و نمایش نتایج می باشد .
بقیه آنها هم برای ترسیم **Radio Button** ها و دکمه رای و ... است

بررسی ثبت رای ...

در خط ۱۱ چک می کنیم که **action** ست شده است یا نه اگر ست شده باشد وارد شرط می شویم در خط ۱۳ در صورتی شرط برقرار می شود که کاربر دکمه ارسال رای را کلیک کرده باشد . پس از ورود به این شرط در خط ۱۵ **poll_id** که کد منحصر به فرد هر گزینه است را به دست می آوریم در صورتی که **poll_id** مقداری نداشته باشد به این معنی است که کاربر گزینه ای را انتخاب نکرده است پس ما فقط نتایج آرا را به او نمایش می دهیم حالا فرض بر این که کاربر گزینه ای را انتخاب کرده باشد ... در خط ۱۸ و ۱۹ یک **query** را نوشته و اجرا می کنیم ؛ این **query** یک واحد به رای انتخاب شده اضافه می کند .

خط ۲۲ تا ۴۱ مربوط به نمایش نتایج آرا است در این قسمت همان طور که گفته شد برای نمایش درصد هر گزینه تعداد آرای مربوط به آن را بر کل تعداد آرا تقسیم می کنیم و در ۱۰۰ ضرب می کنیم و آنها را زیر هم چاپ می کنیم .

چرا فقط از یک دستور **print** استفاده کردیم ؟

برای این که در برنامه هر چه کمتر از این دستور استفاده کنیم سرعت آن بالاتر است .

امیدوارم که این آموزش به درد شما عزیزان بخورد .

 موفق باشید .

ساخت خبرنامه (قسمت صفر) :

اگر آموزش های php این سایت رو دنبال کرده باشید الان به جایی رسیدیم که میتونیم با هم یه پروژه کوچولو رو شروع کنیم ...

من برای نمونه ساخت یه برنامه خبرنامه رو پیشنهاد می کنم و چند تا دلیل هم دارم ...

با ساختن یه اسکریپت خبرنامه که توی هر جلسه یه قسمت ازش رو میسازیم یه چیزایی یاد میگیریم که در آخر انتظار میره که چیزهای زیر رو یاد بگیرید :

- ۱- با اصول کلی ساخت یک پروژه آشنا می شوید .
- ۲- با Session ها بتونید به راحتی و به صورت کاربردی کار کنید .
- ۳- با توابع php که توی این پروژه به کار می بریم آشنا می شوید .
- ۴- طریقه ارسال ایمیل رو یاد می گیرید و این که چطور تعداد زیادی ایمیل ارسال کنیم .
- ۵- با انواع Query ها آشنا می شوید .
- ۶- با راههایی که امنیت صفحه وب را بالا میبرند آشنا می شوید .

خلاصه برای این که دستتون راه بیفته خیلی خوبه .

شروع پروژه :

اول از همه باید ببینیم که نیاز های ما از برنامه چیه ؟ من یه سری رو که به ذهنم می رسید اینجا لیست کردم:

- ۱- هر کاربر بتونه خودش ثبت نام کنه .
- ۲- نام و ایمیل و تاریخ پیوستن او به خبر نامه ثبت بشه. (البته نام اختیاری باشه)
- ۳- کاربر از طریق ایمیلی که موقع ثبت نام براش ارسال میشه بتونه عضویت خودش رو فعال کنه .
- ۴- کاربر خودش بتونه ایمیلش رو حذف کنه (البته باز هم با ایمیلی که براش ارسال میشه)
- ۵- متن هایی که مینویسیم رو بتونیم ذخیره کنیم و هر وقت که دوست داشتیم اون رو برای همه یا تعدادی بفرستیم .
- ۶- مشخصات کاربران توسط مدیر قابل ویرایش باشه .
- ۷- خبر ها همه آرشیو بشن و در دسترس ملت قرار بگیرن. (البته با اجازه مدیر)
- ۸- دیگه چیزی به ذهنم نمیرسه ...

طراحی دیتابیس ...

خوب برای این کار هایی که گفتم سه جدول در نظر گرفتیم :

- ۱- جدول اعضا :

کد :

```
CREATE TABLE `users` (  
  `uid` INT NOT NULL AUTO_INCREMENT,  
  `email` VARCHAR(255) NOT NULL,  
  `name` VARCHAR(150) NOT NULL,
```

```

`active` VARCHAR(1) NOT NULL,
`join_date` VARCHAR(10) NOT NULL,
`delete_code` VARCHAR(32) NOT NULL,
`sec_code` VARCHAR(32) NOT NULL,
PRIMARY KEY (`uid`)
)
TYPE = myisam;

```

۲- جدول متن ها (نامه ها) :

کد :

```

CREATE TABLE `letters` (
`lid` INT NOT NULL AUTO_INCREMENT,
`title` VARCHAR(255) NOT NULL,
`sent` VARCHAR(1) NOT NULL,
`date` VARCHAR(10) NOT NULL,
`show` VARCHAR(1) NOT NULL,
`body` TEXT NOT NULL,
PRIMARY KEY (`lid`)
)
TYPE = myisam;

```

۳- جدول مدیریت :

کد :

```

CREATE TABLE `admin` (
`user` VARCHAR(32) NOT NULL,
`pass` VARCHAR(32) NOT NULL,
`last_log_date` VARCHAR(10) NOT NULL,
`last_log_ip` VARCHAR(16) NOT NULL
)
TYPE = myisam;

```

ظاهر برنامه :

خوب ما باید به فکر ظاهر برنامه هم باشیم ... چون قصد ما آموزش و فعلا اول راه هستیم استفاده از قالب رو آموزش نمیدم تا به روز مفصل در موردش توضیح بدم .

۱- به صفحه ثبت نام می خواهیم که خیلی سادس :
(برای این که پیچیده نشه به ظاهرش نرسیدم)

کد :

```

<html>

<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Join to Us...</title>

```

```

</head>

<body>

<form method="POST" action="Register.php">
  <p>Name :<input type="text" name="T1" size="20"></p>
  <p>Email : <input type="text" name="T2" size="20"></p>
  <p><input type="submit" value="Join" name="register"></p>
</form>

</body>

</html>

```

۲- به صفحه برای ورود مدیر می‌خواهیم :

```

: کد

<html>

<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Newsletter Login...</title>
</head>

<body>

<form method="POST" action="Login.php">
  <p>User name: <input type="text" name="user" size="20"></p>
  <p>Password : <input type="password" name="pass" size="20"></p>
  <p><input type="submit" value="Login" name="LoginIt"></p>
</form>

</body>

</html>

```

صفحات مدیریت رو بعد بررسی می‌کنیم چون قراره به صورت داینامیک ساخته بشه .
 فعلا تا همین جا خوبه تا جلسه بعد کد نویسی ثبت نام رو شروع کنیم .

ساخت خبرنامه (قسمت اول) :

امروز قسمت اول از ساخت خبرنامه رو با هم پیگیری میکنیم توجه کنید که به قسمت هم قبل از این داشتیم که عنوان اون آموزش **php** (ساخت خبر نامه قسمت صفر) است .

خوب امروز قسمت ثبت نام رو می سازیم ...

اول از همه فرم ثبت نام رو توی یه صفحه به نام **Register.htm** میسازیم :

```
کد :
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Join to Us...</title>
</head>
<body>
<form method="POST" action="Register.php">
  <p>Name : <input type="text" name="Name" size="20"></p>
  <p>Email : <input type="text" name="Email" size="20"></p>
  <p><input type="submit" value="Join" name="register"></p>
</form>
</body>
</html>
```

خوب این صفحه بعد از ارسال فرم به یه صفحه دیگه میره به نام **Register.php** (چرا؟ خوب معلومه چون توی قسمت **Action** فرم آدرس این صفحه رو نوشتیم)

حالا بریم سراغ صفحه Register.php :

* قبل از این که شروع کنیم باید بگم که چون ما خیلی جاهای برنامهمون نیاز به کار با دیتا بیس داریم من یه تابع می نویسم و اون رو توی یه فایل به نام **config.php** قرار می دم و هر جایی که نیاز به کار با دیتا بیس داشتیم اون فایل رو به برنامه ام الحاق می کنم و ازش استفاده میکنم ... اگه متوجه نشدید وقتی شروع کردم کاملا متوجه می شوید .

فایل **config.php** :

```
کد :
<?php
function connect()
{
  mysql_connect("localhost", "root", "") or die(mysql_error());
```



```
mysql_select_db("newsletter");
}
?>
```

توی این فایل به تابع تعریف کردیم به نام **connect** و هر جا که خواستیم به دیتا بیس متصل بشیم فقط کافیست که این تابع رو فراخونی کنیم ...

نکته ۱: دقت کنید که ما میتونستیم هر نامی که بخواییم برای تابعمون انتخاب کنیم فقط برای این که ارتباطی با کارش داشته باشه من این نام رو انتخاب کردم .

نکته ۲: توابع **mysql_connect** و **mysql_select_db** همین طور که قبلا هم گفتیم توابع خود **php** هستند که برای اتصال به دیتا بیس **MySQL** استفاده می شه .

نکته ۳: **newsletter** نام دیتا بیس ما است و **root** نام کاربری دیتابیس و بدون پسورد .

خوب بریم سراغ فایل **Register.php** که عملیات ثبت نام توی اون صورت میگیره ...

۱- اولین کاری که باید بکنیم اینه که فایل **config.php** رو به این فایل الحاق کنیم چون با دیتابیس کار داریم :

کد :

```
include("config.php");
```

۲- چک کنیم ببینیم که کاربر از طریق فرم وارد این صفحه شده یا نه همین طوری آدرس زده و اومده تو ...

کد :

```
//Check The Form is Submitted!?
if(!isset($_POST['Name']))
exit();
```

۳- خوب حالا اطلاعات فرم ارسال شده را توی دو تا متغیر می ریزیم که ازش استفاده کنیم

کد :

```
//Read Data From Form...
$Name = $_POST['Name'];
$email = $_POST['Email'];
```

۴- برای این که فضاهای خالی (**Space**) که کاربر در ابتدا و انتها وارد کرده از بین برود از تابع **trim** استفاده می کنیم .

کد :

```
//Trim the Strings...
$Name = trim($Name);
$email = trim($Email);
```

۵- بررسی می کنیم که کاربر حتما ایمیل را وارد کرده باشد (چون در این برنامه نام مهم نیست اون رو چک نمی کنیم)

```
کد :  
//Check Empty Email Addr...  
if(empty($Email))  
die("Please Enter your Email Address!");
```

۶- جایگزینی کاراکتر های خاص HTML با مقدار نمایشی آن (این کار برای اینه که کاربر نتونه کد HTML و یا JavaScript به برنامه ما تزریق کنه)

```
کد :  
//Set The special HTML Characters 2 HTML Codes  
$Name = htmlspecialchars($Name);
```

۷- از داخل نام کاربر کاراکتر ' را حذف میکنیم که توی کار با دیتابیس دچار مشکل نشویم ...

```
کد :  
//Remove Slashes...  
$Name = str_replace("'", "", $Name);
```

۸- حروف آدرس ایمیل را به حروف کوچک تبدیل میکنیم تا همه جا یکدست باشه ...

```
کد :  
//Convert Email Address to Lower Case  
$Email = strtolower($Email);
```

۹- بررسی میکنیم که آدرس ایمیل کاربر از نظر عبارتی درست هست یا نه ... (بعدا به جلسه مفصل در موردش توضیح می دم)

```
کد :  
//Check For Valid Email Address Expression  
if(!preg_match("/^[A-z0-9_-]+@[A-z0-9_-]+(\.[A-z0-9_-]+)+[A-z]{2,4}$/",  
$Email))  
die("Invalid Email Address!");
```

۱۰- بررسی می کنیم که ایمیلی که کاربر وارد کرده قبلا در دیتابیس ذخیره شده یا نه ...

```
کد :  
//Check Exist...  
$Sql = "SELECT COUNT(*) FROM `users`  
WHERE `email` = '$Email'";  
  
//Connect to database...
```

```
connect();

$Result = mysql_query($Sql) or die(mysql_error() . "<br>SQL: " . $Sql);
if(mysql_result($Result, 0) > 0 )
    die("This Email Address already Exist!");
```

۱۱- به کد امنیتی تولید می کنیم تا کاربر بوسیله اون بتونه عضویت خودش رو فعال کنه

کد :

```
//Create A Random Security Code...
$RandomNum = rand(0, 9999);
$SecCode = md5($RandomNum);
```

۱۲- تاریخ امروز

کد :

```
//Calculate The Today Date...
$Today = date("Ymd");
```

۱۳- ثبت کاربر جدید :

کد :

```
//Insert New User...
$Sql = "INSERT INTO `users` (`email`, `name`, `active`, `join_date`, `sec_code`)
VALUES
($Email, '$Name', '0', '$Today', '$SecCode)";

$Result = mysql_query($Sql) or die(mysql_error() . "<br>SQL: " . $Sql);

print("Registration Complete Successfully!");
```

کد کامل صفحه Register.php :

کد :

```
<?
include("config.php");

//Check The Form is Submitted!?
if(!isset($_POST['Name']))
    exit();

//Read Data From Form...
$Name = $_POST['Name'];
$Email = $_POST['Email'];

//Trim the Strings...
$Name = trim($Name);
$Email = trim($Email);
```

```

//Check Empty Email Addr...
if(empty($Email))
    die("Please Enter your Email Address!");

//Set The special HTML Characters 2 HTML Codes
$Name = htmlspecialchars($Name);

//Remove Slashes...
$Name = str_replace("'", "", $Name);

//Convert Email Address to Lower Case
$Email = strtolower($Email);

//Check For Valid Email Address Expression
if(!preg_match("/^[.A-z0-9_-]+@[.A-z0-9_-]+(\.[.A-z0-9_-]+)+[A-z]{2,4}$/",
$Email))
    die("Invalid Email Address!");

//Check Exist...
$sql = "SELECT COUNT(*) FROM `users`
WHERE `email` = '$Email'";

//Connect to database...
connect();

$result = mysql_query($sql) or die(mysql_error() . "<br>SQL: " . $sql);
if(mysql_result($result, 0) > 0 )
    die("This Email Address already Exist!");

//Create A Random Security Code...
$RandomNum = rand(0, 9999);
$SecCode = md5($RandomNum);

//Calculate The Today Date...
$Today = date("Ymd");

//Insert New User...
$sql = "INSERT INTO `users` (`email`, `name`, `active`, `join_date`, `sec_code`)
VALUES
('$Email', '$Name', '0', '$Today', '$SecCode')";

$result = mysql_query($sql) or die(mysql_error() . "<br>SQL: " . $sql);

print("Registration Complete Successfully!");

?>

```

دوستان من خیلی سعی میکنم که ساده توضیح بدم ... با این وجود آگه باز هم جایی از اون نیاز به توضیح بیشتر داره بگید تا توضیح بدم ...

در قسمت بعد در مورد شیوه ارسال ایمیل به کاربر در قالب یک صفحه HTML زیبا و فعال کردن عضویت از طریق ایمیل بحث خواهیم کرد.

ساخت خبرنامه (قسمت دوم) :

امروز بخش سوم (قسمت دوم) از آموزش ساخت خبر نامه رو دنبال می کنیم .
اگر بار اول است که این سری از آموزش ها را می بینید سری های قبلی موجود در همین سایت را مطالعه نمایید .

امروز فعال سازی حساب کاربری کاربر از طریق ایمیل را بررسی میکنیم .
در جلسه قبل یک کاربر جدید ساختیم و اون رو در **Table** ، **Insert** کردیم. حال میخواهیم به محض ثبت نام کاربر ایمیلی را دریافت کند و از طریق آن ایمیل حساب خود را فعال نماید .

ابتدا ببینیم که توی **php** چطور **mail** ارسال کنیم ...

برای ارسال ایمیل توی php دو راه پیش روی ما است :

- ۱- استفاده از تابع داخلی **mail** که خیلی راحت است .
- ۲- استفاده از **SMTP** که به خورده مشکل تر از تابع فوق است .

(برای این آموزش از تابع داخلی **mail** استفاده می کنیم بعد که خوب راه افتادیم یه کلاس مرتب برای ارسال ایمیل با هم می نویسیم.)

طرز استفاده از تابع mail :

کد :

```
mail("MailTo@domain.com" , "Mail Subject", "Salam khubi?");
```

که در کد فوق **MailTo@domain.com** ایمیل شخص دریافت کننده است .
Mail Subject ؛ عنوان ایمیل ارسالی .
Salam khubi? ؛ هم متن ایمیل ما است .

این تابع در صورتی که بتواند میل ارسال کند مقدار **True** بر می گرداند و در غیر این صورت **False** .

نکته: با این تابع نمی توانیم تشخیص دهیم که ایمیل به دست گیرنده رسیده است یا نه .

خوب حالا یه خط کد به انتهای فایل **Register.php** می افزاییم که ایمیل تایید عضویت رو برای کاربر ارسال نماید .

کد :

```
//<!-- Start of Mail Body
```

```
$MailBody = '
```

```
Salam dooste aziz,
```

```
Az in ke dar khabar nameye ma ozv shodehid kheyli mamnoonim.
```

```
lotfan baraye faal shodan acount khod bar rooye link zir click nemayeed:
```

```
http://www.MySite.com/Newsletter/Active.php?Email='. $Email
.'&SecCode='. $SecCode .'
```

```
Ba tashakor,  
Movafagh bashid.');
```

```
// End of Mail Body-->
```

```
mail($Email , "Active your account", $MailBody);
```

خوب حالا بریم سراغ فایل **Active.php** که صحبتش اون بالا شد .

خوب چون با **DataBase** کار داریم :

کد :

```
include("config.php");
```

چون توی لینک ما چیز هایی نوشتیم که مثل در خواست **GET** عمل می کنه پس اول چک میکنیم ببینیم که لینک درسته یا نه (یعنی همه چیزایی رو که میخوایم رو داره یا نه)

کد :

```
//Check The Link is Correct  
if(!isset($_GET['Email']) || !isset($_GET['SecCode']))  
exit();
```

خوب حالا مقادیری که از ورودی گرفتیم رو توی متغیر هایی میریزیم که راحت تر بشه باهاش کار کرد :

کد :

```
//Set The variables...  
$Email = $_GET['Email'];  
$SecCode = $_GET['SecCode'];
```

حالا **SQL** مربوط به فعال کردن **Account** رو می نویسیم :

کد :

```
//Update The Account ...  
$Sql = "UPDATE `users`  
SET  
`active` = '1'  
WHERE  
`email` = '$Email'  
AND  
`sec_code` = '$SecCode'  
";
```

ارتباط با دیتا بیس (از طریق تابعی که قبلا خودمون نوشتیم):

کد :

```
//Connect to database...
connect();
```

اجرای SQL :

کد :

```
//Execute The SQL...
$result = mysql_query($sql) or die(mysql_error() . "<br>SQL: " . $sql);
```

معرفی تابع جدید :

`mysql_affected_rows` این تابع در مواردی که ما عملیات هایی که در داده هامون تغییر ایجاد می کنند مثل (Insert, Update, Delete) تعداد رکورد هایی که تغییر کرده اند رو بر میگرددونه. که اینجا ما ازش استفاده کردیم :

کد :

```
//Check Affected Rows...
if(mysql_affected_rows() == 1)
    die("Your account successfully activated!");
else
    die("No accounts activated!");
```

خوب و این کد کامل امروز :

کد :

```
<?
//Active.php

include("config.php");

//Check The Link is Correct
if(!isset($_GET['Email']) || !isset($_GET['SecCode']))
    exit();

//Set The variables...
$email = $_GET['Email'];
$secCode = $_GET['SecCode'];

//Update The Account ...
$sql = "UPDATE `users`
SET
    `active` = '1'
WHERE
    `email` = '$email'
AND
    `sec_code` = '$secCode'
```

```
";  
  
//Connect to database...  
connect();  
  
//Execute The SQL...  
$Result = mysql_query($Sql) or die(mysql_error() . "<br>SQL: " . $Sql);  
  
//Check Affected Rows...  
if(mysql_affected_rows() == 1)  
    die("Your account successfully activated!");  
else  
    die("No accounts activated!");  
  
?>
```

موفق و سربلند باشید.

ساخت خبرنامه (قسمت سوم) :

- سلام دوستان، امروز قسمت سوم آموزش ساخت خبرنامه رو پیگیری میکنیم .
- امروز قسمت Login مدیر به بخش مدیریت رو انجام میدیم .

ابتدا به مرور روی جدول admin میکنیم ...

کد :

```
CREATE TABLE `admin` (  
  `user` varchar(32) NOT NULL default "",  
  `pass` varchar(32) NOT NULL default "",  
  PRIMARY KEY (`user`)  
) TYPE=MyISAM;
```

خوب این جدول مربوط به کاربر (های) ما است که با اون وارد سیستم میشیم و خبر نامه ارسال میکنیم و اعضا رو مدیریت میکنیم .

برای تست فعلا از توی phpMyAdmin به دونه کاربر توش insert میکنیم بعد برنامهش رو می نویسیم .

کد :

```
INSERT INTO `admin` VALUES ('a', '202cb962ac59075b964b07152d234b70');
```

خوب حالا بریم سراغ ساخت صفحه Login :

من پیشنهاد میکنم به فولدر جداگانه برای admin درست کنیم و فایل های مخصوص مدیر رو توی اون بریزیم .

فایل login.html :

این فایل فرم ورود رو به ما نشون میده و محتوای اون اینه :

کد :

```
<form method="POST" action="login.php">  
<p>User name:  
<input type="text" name="UserName" >  
<p>  
Password: <input type="password" name="Password" >  
<p>  
<input type="submit" name="BtnSubmit" value="Login">  
</form>
```

فایل login.php :

این فایل کار تعیین اعتبار کاربر رو انجام میده .

اول از همه چون از جلسات میخواهیم استفاده کنیم پس تابع مربوط به راه اندازی session رو فراخونی میکنیم .

کد :

```
session_start();
```

بعد نام کاربری و پسورد را از ورودی میگیریم و توی دوتا متغیر میریزیم :

کد :

```
$User = $_POST['UserName'];  
$Pass = $_POST['Password'];
```

چون ما کلمه عبور رو به صورت رمزی برگشت ناپذیر توی **DataBase** ذخیره میکنیم پس چیزی که از کاربر میگیریم رو هم به همون صورت به رمز در میاریم تا بتونیم مطابقت اون رو بررسی کنیم .

کد :

```
$Pass = md5($Pass);
```

البته الگوریتم های دیگری هم هستند ولی این بیشتر مرسومه ... البته توی برنامه های بزرگ که بعد خواهید نوشت از این الگوریتم به صورتی که بالا گفته شد استفاده نمیکنید بلکه با رشته ها بازی میکنیم و یه الگوریتم من در آوردی با استفاده از **md5** و ... می سازیم تا امنیت بیشتری داشته باشه .

حالا **Query** مربوط به پیدا کردن کاربر در جدول را مینویسیم تا ببینیم که اصلا همچین کاربری وجود داره یا نه ...

کد :

```
$sql = "SELECT `pass`  
FROM `admin`  
WHERE `user` = '$User'";
```

حالا با دیتابیس ارتباط برقرار میکنیم :

کد :

```
include('../config.php');  
connect();
```

اون فایل **config.php** همون فایلی هست که توی مراحل قبل یه تابع توش نوشتیم برای ارتباط با **DataBase** .

حالا اجرای **query** :

کد :

```
$Result = mysql_query($sql);
```

حالا شرط بررسی این که پسورد درسته یا نه و انجام کارهای مربوط به لاگین :

کد :

```
if(mysql_num_rows($Result) == 1 && mysql_result($Result, 0) == $Pass)  
{  
    $_SESSION['UserName'] = $User;  
  
    $ClientString = $_SERVER['HTTP_USER_AGENT'] .
```

```

$_SERVER['REMOTE_ADDR'];
$_SESSION['ClientStr'] = md5($ClientString);

header("Location: admin.php");
}
else
{
die("The specified username and/or password is incorrect!");
}

```

توضیح: توی عبارت شرطی اول گفتیم که بررسی شود که اصلا **query** ما یک نتیجه داشته یا نه اگه داشت حالا قسمت دوم شرط که میگیریم پسوردی که از فرم گرفتیم با پسورد موجود در دیتابیس یکی هست یا نه .

حالا اگه شرط درست بود وارد بدنه شرط می شویم :
توی خط اولش نام کاربری رو توی **session** ریختیم چون توی صفحات دیگه به دردمون میخوره .

در خط دوم و سوم ما مشخصات مرورگر، سیستم عامل و ... رو به همراه **IP** کاربر رو توی **session** ریختیم و برای این که کمتر جا بگیره و امنیت بیشتر بشه اون رو با **md5** کد کردیم. که توی صفحات مدیریت اون رو چک میکنیم تا کسی نتونه با **session** تقلبی **login** کنه .

در خط بعد هم که یه هدر میفرستیم و کاربر رو به صفحه مدیریت (**admin.php**) که در جلسات بعد خواهیم ساخت هدایت میکنیم .

در زیر کد کامل صفحه **login.php** را می بینید :

```

کد :
<?php
//Starting the session...
session_start();

//Get The data from form.
$user = $_POST['UserName'];
$pass = $_POST['Password'];

//Hashing the password...
$pass = md5($pass);

//Checking the user and pass...
$sql = "SELECT `pass`
FROM `admin`
WHERE `user` = '$user'";

//Connecting to db...
include('../config.php');
connect();

$result = mysql_query($sql);

```

```

if(mysql_num_rows($Result) == 1 && mysql_result($Result, 0) == $Pass)
{
    $_SESSION['UserName'] = $User;

    $ClientString = $_SERVER['HTTP_USER_AGENT'] .
$_SERVER['REMOTE_ADDR'];
    $_SESSION['ClientStr'] = md5($ClientString);

    header("Location: admin.php");
}
else
{
    die("The specified username and/or password is incorrect!");
}
?>

```

این قسمت چه کاربردی دارد؟

```

$ClientString = $_SERVER['HTTP_USER_AGENT'] .
$_SERVER['REMOTE_ADDR'];
$_SESSION['ClientStr'] = md5($ClientString);

```

کد :

برای اینته که از دزدیدن **Session** یا همون **session** تقلبی جلوگیری کنیم؛ توی جلسه بعد مفصل توضیح میدم.

ساخت خیرنامه (قسمت چهارم) :

خوب امروز میریم سراغ قسمت مدیریت (ثبت نامه ها...)

خوب فایل **admin.php** رو با هم می سازیم ...

توی این صفحه لینک های مختلفی به قسمت های مختلف مدیریت میگذاریم :

خوب چون این صفحه صفحه ای هست که کاربر باید حتما **Login** کرده باشه تا بتونه اون رو ببینه پس اول از همه اعتبار سنجی کاربر رو انجام میدیم . برای این کار ابتدا تابع **session_start** رو صدا میزنیم، سپس با یه شرط بررسی میکنیم که کاربر **Login** کرده یا نه. اگر نه که کاربر رو به صفحه **login.html** هدایت میکنیم و اگر **Login** کرده که کاری بهش نداریم ...

```
کد :
<?
    session_start();
    $ClientString = $_SERVER['HTTP_USER_AGENT'] .
    $_SERVER['REMOTE_ADDR'];

    if( !isset( $_SESSION['UserName'] ) || $_SESSION['ClientStr'] != md5(
    $ClientString))
    {
        header( "location: login.html" );
        exit();
    }
?>

<a href="InsertLetter.php">Insert New Letter</a>
```

اون آخر هم که می بینید یه لینک به یه صفحه دیگه گذاشته شده که توی اون صفحه نامه هایی رو که قراره ارسال کنیم رو تو دیتابیس ثبت میکنیم .

خوب بریم سراغ صفحه **InsertLetter.php** و ببینیم که چطور اخبار رو ثبت می کنیم .

چون توی این صفحه هم نیاز هست که کاربر معتبر باشه پس همون اعتبار سنجی بالا رو برای این هم تکرار میکنیم ...

```
کد :
session_start();
$ClientString = $_SERVER['HTTP_USER_AGENT'] .
$_SERVER['REMOTE_ADDR'];

if( !isset( $_SESSION['UserName'] ) || $_SESSION['ClientStr'] != md5(
$ClientString))
{
    header( "location: login.html" );
    exit();
}
```

خوب حالا به روش جدید یاد می گیریم و اون اینه که اگه بخوایم توی یه صفحه فرمی داشته باشیم که هر وقت دلمون خواست باشه و هر وقت خواستیم اون رو نشون ندیم و فرمون به همین صفحه جاری ارسال بشه ...

خوب اول چک میکنیم تا ببینیم که کاربر فرم رو ارسال کرده یا نه که اگه چیزی ارسال نشده فرم رو نشون بدیم :

```
کد :
if( !isset( $_POST[ 'Title' ]))
{
?>
<form method="POST">
<p>Letter Title: <input type="text" name="Title" >
<p>Letter Body: <textarea name="Body"></textarea>
<p><input type="submit" name="BtnSubmit" value="Insert">
</form>
?<
} //End of If(isset)
```

توی چند خط بالا چند تا نکته هست که باید بگم:

اول این که من توی شرط چرا در تابع **isset** اسم دکمه را ننوشتم و اسم یکی از **EditBox** ها رو نوشتم... دلیلش اینه که وقتی که فرمی به سمت سرور ارسال میشه بعضی از چیزهای اون **set** نمیشه (توی آرایه **POST** ریخته نمی شه ...). مثلا وقتی دکمه **submit** رو کلیک نکنیم یعنی با کلید **Enter** وقتی که توی یکی از **EditBox** ها هستیم فرم رو ارسال کنیم اون دکمه **Set** همیشه همچنین در مورد **CheckBox** ها هم همین طور هست. اگه **check** نخورده باشند ارسال نمیشن در مورد بقیه اجزای فرم هم اگه **disabled** باشند باز هم ارسال نمیشن .

یه نکته دیگه که توی تگ فرم مشاهده میشه اینه که من هیچ **Action** ی برای فرم خودم مشخص نکردم وقتی این کار رو میکنم خود صفحه جاری به عنوان **action** فرم در نظر گرفته میشه ... اگه **method** هم ننویسم **get** در نظر میگیره ...

خوب حالا بریم سراغ بقیه کد ها ...

توی **else** مربوط به همین **if** ی که بالا نوشتیم کد مربوط به **insert** رو مینویسیم ...

اولین کاری که میکنیم اینه که چیزهایی که از فرم برای ما ارسال شده رو توی چند تا متغیر میریزیم :

```
کد :
$title = $_POST[ 'Title' ];
$body = $_POST[ 'Body' ];
$today = date("Ymd");
```

حالا به دیتابیس متصل می شیم :

```
کد :
//Connecting to db...
include('../config.php');
connect();
```

یه Query برای Insert می نویسیم و اون رو اجرا می کنیم :

کد :

```
$Sql = "INSERT INTO `letters`  
  (`title`, `sent`, `date`, `show`, `body`)  
  VALUES ( '$Title', '0', '$Today', '0', '$Body')";  
  
$Result = mysql_query($Sql) or die(mysql_error() . "<br>SQL: " . $Sql);
```

حالا یه نکته جدید :

کد :

```
if( mysql_affected_rows()  
{  
  print("The Letter Inserted Sucessfully!");  
}  
else  
{  
  print("Error: The Letter Inserting Stopped!");  
}
```

بله تابع `mysql_affected_rows` رو معرفی میکنم که برای SQL هایی که مربوط به تغییرات در اطلاعات میشن تعداد رکورد هایی رو که تغییر کرده رو بر میگرددونه مثلا وقتی که `insert` ، `Update` یا `Delete` که انجام میدیم میتونه خیلی کمکمون کنه ... که به نمونش رو بالا ببینید .

کد کامل صفحه `insert` رو زیر براتون گذاشتم :

کد :

```
<?  
  session_start();  
  $ClientString = $_SERVER['HTTP_USER_AGENT'] .  
  $_SERVER['REMOTE_ADDR'];  
  
  if( !isset( $_SESSION['UserName'] ) || $_SESSION['ClientStr'] != md5(  
  $ClientString)  
  {  
    header( "location: login.html" );  
    exit();  
  }  
  
  if( !isset( $_POST[ 'Title' ] ) )  
  {  
    ?>  
    <form method="POST">  
    <p>Letter Title:  
    <input type="text" name="Title" >  
    <p>  
    Letter Body: <textarea name="Body"></textarea>  
    <p>
```

```

<input type="submit" name="BtnSubmit" value="Insert">
</form>
<?
} //End of If(isset)
else{

    //Insert The Letter...

    $Title = $_POST[ 'Title' ];
    $Body = $_POST[ 'Body' ];
    $Today = date("Ymd");

    //Connecting to db...
    include('../config.php');
    connect();

    $Sql = "INSERT INTO `letters`
        ( `title`, `sent`, `date`, `show`, `body` )
        VALUES ( '$Title', '0', '$Today', '0', '$Body' )";

    $Result = mysql_query($Sql) or die(mysql_error() . "<br>SQL: " . $Sql);

    if( mysql_affected_rows() )
    {
        print("The Letter Inserted Successfully!");
    } else {
        print("Error: The Letter Inserting Stopped!");
    }

} //End of Else If...
?>

```

در جلسات بعد در مورد ارسال ایمیل دسته جمعی بحث میکنیم و این که کاربر چطور بتونه خودش ایمیلش رو از لیست ما حذف کنه .

استفاده از تاریخ و ساعت گرینویچ در PHP

سلام دوستان 😊،

توی این مبحث می خوام در مورد استفاده از تاریخ و ساعت گرینویچ (Greenwich Time) توی پروژه های PHP صحبت کنم .

کاربرد :

۱- شما می توانید به کاربران سایت خود این امکان رو بدهید که هر جای دنیا که باشند تمامی تاریخ و ساعت ها را به همون جا داشته باشند .

۲- معمولا ساعت Server ها با ساعت ما set نیستند .

خوب برای این کار فیلد Date و Time رو در DataBase تون از نوع Int بگیرید .

تابع time در php لحظه جاری رو به صورت یک عدد صحیح (Integer) به گرینویچ بر می گردونه... که این عدد شامل ثانیه، دقیقه، ساعت، روز، ماه و سال هست .

طرز استفاده :

```
کد :  
  
<?php  
  
$Now = time();  
print ( $Now);  
  
?>
```

توضیحات بیشتر در مورد این تابع :

<http://ir.php.net/manual/en/function.time.php>

خروجی :

```
کد :  
  
1205692465
```

خوب حالا برای ذخیره تاریخ و ساعت در DataBase همین عدد رو ذخیره می کنیم .

* حالا خوندنش

برای خوندنش چون به گرینویچ هست اول باید به میزان اختلاف ساعت جایی که می خوایم بهش اضافه کنیم یا ازش کم کنیم. و این اختلاف رو باید به ثانیه تبدیل کنیم. مثلا برای ایران که اختلاف ۳،۵ ساعت داره، این طوره :

```
کد :  
  
<?php
```

```
$TheTime = time();  
  
$TimeZone = 3.5;  
  
$TheTime += $TimeZone * 3600;  
print ( $TheTime);  
  
?>
```

حالا که تاریخ و ساعت رو به صورت یک عدد صحیح به دست آوردیم اونو به شکلی که می خواهیم نمایش می دیم. برای این کار از تابع `gmdate` استفاده می کنیم :

```
کد :  
  
<?php  
  
$TheTime = time();  
  
$TimeZone = 3.5;  
  
$TheTime += $TimeZone * 3600;  
  
$StrTime = gmdate( "Y/m/d - g:i A", $TheTime);  
print( $StrTime);  
  
?>
```

توضیحات بیشتر در مورد این تابع :

<http://ir.php.net/manual/en/function.gmdate.php>

پارامتر اول این تابع مثل همون چیزیه که به ورودی تابع `date` می دیم. دومیش هم که `time` هست .

امیدوارم این آموزش کوچولو به دردتون بخوره ... 😊👍

آموزش کلاس Template

سلام دوستان، چند تا از دوستان خواسته بودن در مورد طرز کار و استفاده از کلاس **Template** توضیحاتی بدم. که البته نوبت این پست فقط در مورد طرز استفادش توضیح می دم .

اول کد کلاس رو ببینید :

این کلاس رو از یه **Counter** آلمانی کپی زدم .

کد :

```
<?php
class Template
{
    var $tags = array();
    var $required_tags = array();
    var $blocks = array();
    var $tpl = "";
    var $parsed_tpl = "";

    function Template( $templatefile = " )
    {
        if( !empty( $templatefile ) )
        {
            if( !file_exists( $templatefile ) )
            {
                print 'Could not find the template file <i>'. $templatefile . "</i>!\n";
                return FALSE;
            }
            $this->load_file( $templatefile );
        }
        return TRUE;
    }

    /*-----*/

    function load_template( $template )
    {
        $this->tpl .= $template;
    }

    /*-----*/

    function load_file( $file )
    {
        if( empty( $file ) )
        {
            return FALSE;
        }

        $template = @implode( " ", @file( $file ) );
        if( $template == FALSE )
        {
            print 'Could not read the template file!';
            return FALSE;
        }
    }
}
```

```

    }
    $this->tpl .= $template;
}

/*-----*/

function assign( $input, $value = "", $required = FALSE )
{
    if( is_array( $input ) )
    {
        foreach( $input as $tag => $value )
        {
            if(empty( $tag ) )
            {
                print "Tag name ist empty!";
                return FALSE;
            }
            if( $required == TRUE )
            {
                $this->required_tags[$tag] = $value;
            }
            else
            {
                $this->tags[$tag] = $value;
            }
        }
    }
    elseif( is_string( $input ) )
    {
        if( empty( $input ) )
        {
            print "Tag name ist empty!";
            return FALSE;
        }
        else
        {
            if( $required == TRUE )
            {
                $this->required_tags[$input] = $value;
            }
            else
            {
                $this->tags[$input] = $value;
            }
        }
    }
    else
    {
        return FALSE;
    }
    return TRUE;
}

/*-----*/

```

```

function add_block($block_name, $block_array)
{
    if( !is_string($block_name) || empty($block_name))
    {
        print 'Block name is not a string or is empty!';
        return FALSE;
    }
    if( !is_array($block_array))
    {
        print 'Block array is not an array!';
        return FALSE;
    }
    $this->blocks[$block_name][] = $block_array;
}

/*-----*/

function parse()
{
    if( empty( $this->tpl ) )
    {
        return;
    }

    # blocks
    $tmp_blocknames = array();
    foreach( $this->blocks as $block_name => $block_arrays )
    {
        if( $anzahl = preg_match_all( '/<!-- BEGIN BLOCK ' . preg_quote(
$block_name, '/' ) . '-->(.*?)<!-- END BLOCK ' . preg_quote( $block_name, '/' ) . '--
>/sU', $this->tpl, $matches ) )
        {
            for( $i = 0; $i < $anzahl; $i++ )
            {
                $block_plus_definition = $matches[0][$i];
                $block = $matches[1][$i];

                if( is_int( strpos( $block, '<!-- IF' ) ) )
                {
                    $parse_control_structures = TRUE;
                }

                $parsed_block = "";
                foreach( $block_arrays as $block_array )
                {
                    $tmp = $block;
                    if( isset( $parse_control_structures ) )
                    {
                        $tmp = $this->parse_control_structures( $tmp, array_merge(
$block_array, $this->tags, $this->required_tags ) );
                    }
                    foreach( $block_array as $tag_name => $tag_value )
                    {

```

```

        $tmp = str_replace( '{'.$tag_name.'}', $tag_value, $tmp );
    }
    $parsed_block .= $tmp;
}
$this->tpl = str_replace( $block_plus_definition, $parsed_block, $this->tpl
);
    $tmp_blocknames[] = $block_name;
    unset( $parse_control_structures );
}
}
}
if( count( $this->blocks ) > 0 )
{
    $this->tpl = preg_replace( "/<!-- (BEGIN|END) BLOCK (. implode( '|',
$tmp_blocknames ).") -->/", "", $this->tpl );
}

# unbenutzen bl.cke entfernen
$this->tpl = preg_replace( "/<!-- BEGIN BLOCK ([a-zA-Z0-9_-]+) -->.*<!--
END BLOCK \\1 -->( \\r\\n)?/msU", "", $this->tpl );

# single tags
foreach( $this->required_tags as $tag_name => $tag_value )
{
    if( !is_int( strpos( $this->tpl, $tag_name ) ) )
    {
        print 'Could not find tag <i>'.$tag_name.</i> in the template file!';
        return FALSE;
    }
    else
    {
        $this->tpl = str_replace( '{'.$tag_name.'}', $tag_value, $this->tpl );
    }
}
foreach( $this->tags as $tag_name => $tag_value )
{
    $this->tpl = str_replace( '{'.$tag_name.'}', $tag_value, $this->tpl );
}

# if & else
$this->tpl = $this->_parse_control_structures(
    $this->tpl,
    array_merge( $this->tags, $this->required_tags ),
    $this->blocks
);

$this->parsed_tpl = $this->tpl;
$this->tpl = "";
}

/*-----*/

function print_template()

```

```

{
    if( !empty( $this->tpl ) )
    {
        $this->parse();
    }
    print $this->parsed_tpl;
}

/*-----*/

function get_tpl_as_var()
{
    if( !empty( $this->tpl ) )
    {
        $this->parse();
    }
    return $this->parsed_tpl;
}

/*-----*/

function free()
{
    $this->tpl = "";
    $this->parsed_tpl = "";
    $this->tags = array();
    $this->required_tags = array();
    $this->blocks = array();
}

/*-----*/

function _parse_control_structures( $tpl, $vars, $blocks = array() )
{
    if ( $matchnumber = preg_match_all( '/<!-- IF (!?)(BLOCK)?([_a-zA-Z0-9\-\
]+) -->(.*)(<!-- ELSEIF !(\1\2\4) -->(.*))?<!-- ENDIF \1\2\4 -->/msU', $tpl,
    $matches ) )
    {
        for( $i = 0; $i < $matchnumber; $i++ )
        {
            //print( $matches[8 ][$i] . '<br />');
            if( !empty( $matches[2][$i] ) )
            {
                $code = 'if( ' . $matches[1][$i] . 'isset($blocks["' . $matches[4][$i] . '\'])' . "\n";
            }
            else
            {
                $code = 'if( ' . $matches[1][$i] . '( isset($vars["' . $matches[4][$i] . '\']) )' . "\n";
            }
            $code .= '{ $tpl = str_replace( $matches[0][$i], $this-
>_parse_control_structures( $matches[5][$i], $vars, $blocks ), $tpl ); }' . "\n";
            $code .= ' else ' . "\n";
            $code .= '{ $tpl = str_replace( $matches[0][$i], !empty($matches[7][$i]) ?
$this->_parse_control_structures( $matches[8][$i], $vars, $blocks ) : \\' , $tpl ); }';

```

```

        eval( $code );
    }
}
return $tpl;
}
}
?>

```

خوب اول طرز استفادش :

برای این کار به مثال میزنم

فرض کنید کد بالا رو توی یه فایل به نام **Template.php** ذخیره کردیم و یه فایل به نام **index.php** داریم که برنامه اصلیمون هست و یه **Folder** به نام **Tpls** داریم که تمامی قالب هامون توی اون، فعلا یه قالب به نام **body.html** داریم .

گام اول - محتویات فایل index.php :

کد :

```

<?php

include('Template.php');

$MyTpl = new Template();
$MyTpl -> load_file( 'Tpls/body.html' );

$MyTpl -> print_template();

?>

```

و محتوای body.html :

کد :

```

<html>
<head></head>
<body>

<p align="left">
Name:
<font color="#0000FF">
<b>
Ahmad
</b>
</font>
</p>

</body>
</html>

```


توضیح گام اول :

در کل به فایل **template** رو باز می کنیم و اونو نمایش می دیم .
body.html که به صفحه ساده **HTML** هست .
توی **index.php** خط اول که فایل حاوی کلاس **Template** رو **include** کردیم .

خط بعد به **Object** جدید از این کلاس رو تعریف کردیم با نام **MyTpl** :

کد :

```
$MyTpl = new Template();
```

خط بعد فایل قالب رو **Load** کردیم :

کد :

```
$MyTpl -> load_file( 'Tpls/body.html' );
```

دقت کنید که نام و پسوند فایل قالب اصلا مهم نیست چی باشه. من برای راحتی پسوندش رو **HTML** گرفتم .

خط بعد کل قالب رو چاپ کردیم :

کد :

```
$MyTpl -> print_template();
```

خوب این از گام اول بریم سراغ گام بعد ...

ما قالب ها رو می سازیم که کد **HTML** رو از کد **PHP** جدا کنیم و کارمون تمیزتر باشه. ولی باید بتونیم توی قالب تغییرات ایجاد کنیم .

توی این کلاس چیزهایی که توی کد **HTML** مربوط به قالب با داده های تولید شده در کد **PHP** جایگزین می شوند بین **{}** قرار می گیرند . مثل قالب وبلاگ ها که چیزایی مثل این دارند **<*>** ، **<##>** ، ... ، **[]** ،

مثال... محتویات **body.html** :

کد :

```
<html>
<head></head>
<body>
  <p align="left">
    Name:
    <font color="#0000FF">
      <b>
        {TheVarName}
      </b>
    </font>
  </p>
</body>
</html>
```

در اینجا **TheVarName** که در میان {} قرار گرفته است در کد **php** با داده تولید شده جایگزین می شه. توجه کنید که این کلاس به حروف بزرگ و کوچک حساس است .

کد : **index.php**

```
کد :
<?php
include('Template.php');

$MyTpl = new Template();
$MyTpl -> load_file( 'Tpls/body.html' );

if( 1 > 2 )
{
    $Name = 'Ahamad';
}
else
{
    $Name = 'Mohsen';
}

$MyTpl -> assign( 'TheVarName' , $Name );

$MyTpl -> print_template();

?>
```

خوب همون طور که می بینید به خط جدید داریم :

```
کد :
$MyTpl -> assign( 'TheVarName' , $Name );
```

با صدا زدن این **Method** توی قالب **Load** شده هر جا که **{TheVarName}** دید اون رو با محتویات متغیر **Name** جایگزین می کنه .

اگه چند تا متغیر داشتیم می تونیم این **Method** رو چند بار **Call** کنیم ولی به راه ساده تر هم هست و اون اینه که ورودی آرایه بهش بدیم این طوری :

محتویات **body.html** :

```
کد :
<html>
<head></head>
<body>

<p align="left">
    First name:
```

```

<font color="#0000FF">
  <b>
    {FrstName}
  </b>
</font>
</p>
<p align="left">
  Last name:
  <font color="#0000FF">
    <b>
      {LstName}
    </b>
  </font>
</p>

</body>
</html>

```

محتویات **index.php** :

```

: کد
<?php

include('Template.php');

$MyTpl = new Template();
$MyTpl -> load_file( 'Tpls/body.html' );

$FName = 'Maryam';
$LName = 'Hamed';

$NamesArray = array( 'FrstName' => $FName, 'LstName' => $LName);

$MyTpl -> assign( $NamesArray );

$MyTpl -> print_template();

?>

```

معمولا برای راحتی آرایه رو مستقیما توی ورودی **Method** تعریف می کنیم و به یه شکل تمیز تر می نویسیم. برای این که توی متغیر های زیاد قاطی نکنیم یعنی این طوری :

```

: کد
$FName = 'Maryam';
$LName = 'Hamed';

$MyTpl -> assign( array(

  'FrstName' => $FName,
  'LstName' => $LName

```

```
));  
//End of assign template vars -->
```

گام بعد IF - در قالب ...

بعضی وقتا می خواییم با توجه به شرایطی قسمت هایی از قالبتون نمایش داده نشه. مثال :

کد قالب :

کد :

```
<html>  
<head></head>  
<body>  
  
<p align="left">  
  First name:  
  <font color="#0000FF">  
    <b>  
      {FrstName}  
    </b>  
  </font>  
</p>  
  
<!-- IF LstNameCond -->  
  
<p align="left">  
  Last name:  
  <font color="#0000FF">  
    <b>  
      {LstName}  
    </b>  
  </font>  
</p>  
  
<!-- ENDIF LstNameCond -->  
  
</body>  
</html>
```

توی کد قالب به `<!-- IF LstNameCond -->` و `<!-- ENDIF LstNameCond -->` دقت کنید که یک شرط رو بیان می کنند `LstNameCond` عنوان دلخواهی برای شرط هست. خوب کد `php` :

کد :

```
$MyTpl -> assign( array(  
  
  'FrstName' => $FName,  
  'LstName' => $LName,  
  'LstNameCond' => 1  
  
) );  
//End of assign template vars -->
```

طبق شرط بالا محتویات میان **IF** نمایش داده می شوند. توجه کنید که به جای عدد ۱ هرچیز دیگری می تواند باشد حتی ۰ یا **false**، اما برای عدم نمایش کافیه که توی لیست **assign** نباشه. مثل این :

```
کد :
$MyTpl -> assign( array(
    'FrstName' => $FName,
    'LstName' => $LName,
    '//LstNameCond' => 1
)
);
//End of assign template vars -->
```

گام بعد- تکرار کد HTML ...

فک کنم مهمترین قسمت اینجا باشه که ما می خواهیم به تعداد داده رو با قالب یکسان زیر هم چاپ کنیم. برای این کار به کد زیر توجه کنید :

کد قالب :

```
کد :
<html>
<head></head>
<body>

<p align="left">
#-
Name
</p>

<!-- BEGIN BLOCK MyBlock -->

<p align="left">
{Row}-
<b>
{Name}
</b>
</p>

<!-- END BLOCK MyBlock -->

</body>
</html>
```

توی این کد هم **<!-- BEGIN BLOCK MyBlock -->** و **<!-- END BLOCK MyBlock -->** به چشم می خوره ... هر چی بین این دو **Tag** باشه تکرار می شه مثل وبلاگ ها که برای نمایش پست ها معمولا اسم خود سرویس دهنده وبلاگ رو می ذارن **MyBlock**. نام این قسمت هست که دلخواه .

کد php :

```
کد :
<?php
include('Template.php');

$MyTpl = new Template();
$MyTpl -> load_file( 'Tpls/body.html' );

$Names = array( 'Ali', 'Morteza', 'Javad', 'Maryam', 'Ehsan', 'Fatemeh', 'Ahmad');
$TotalRecords = count( $Names);

for( $Index = 0 ; $Index != $TotalRecords ; $Index++ )
{
    $MyTpl -> add_block('MyBlock', array(
        'Row' => $Index + 1,
        'Name' => $Names[ $Index ],
    )
);
//End of assign template vars -->
}

$MyTpl -> print_template();

?>
```

برای این که پیچیده نشه پای **DataBase** رو وسط نکشیدم. فرض کنید داده های ما توی آرایه **Names** هست و تعداد **Record** هامون هم توی **TotalRecords** هست. به ازای هر تکرار داده باید متد **add_block** رو صدا بزنینم و همین طور که می بینید داده ها رو بهش اختصاص بدیم .

نکات قابل توجه :

۱- ما میتونیم مستقیما کدهای **HTML** تولید شده رو چاپ نکنیم ... مثلا شاید بخواییم برای خبرنگارمون یه قالب داشته باشیم و این رو باید **Mail** کنیم. برای این کار یه **Method** دیگه وجود داره که محتوای تولید شده رو بر می گردونه :

```
کد :
$htmlCode = $MyTpl -> get_tpl_as_var();
```

۲- ما به هر تعداد که بخواییم می تونیم قالب **Load** کنیم. یعنی متد **load_file** رو می تونیم به هر تعداد برای قالب های مختلف **Call** کنیم، تمام قالب ها مثل رشته ای پشت سر هم قرار می گیرند .

کاربرد: برای جاهایی استفاده می شه که مثلا شما یه **Header** و **Footer** ثابت دارید و فقط قسمت وسط صفحه عوض می شه، مثل همین صفحه انجمن. مثال :

کد :

```
<?php

include("Template.php");

$MyTpl = new Template();
$MyTpl -> load_file( 'Tpls/Header.html' );

...

switch ( $Action)
{

case 'Login': $Page = 'Login.html'; break;

case 'Logout': $Page = 'Logout.html'; break;

case 'Contacts': $Page = 'Contacts.html'; break;

case 'ContactsSent': $Page = 'ContactsSent.html'; break;

default $Page = 'Index.html';

}

$MyTpl -> load_file( 'Tpls/' . $Page ); //Load the Body of page

...

$MyTpl -> load_file( 'Tpls/Footer.html' );
$MyTpl -> print_template();

?>
```

فقط توی این قسمت باید دقت کنید که چون این صفحات پشت سر هم می آیند توی کد های HTML فاطمی نشه که از تگ های HTML, Head , Body و ... هر کدام ۳ تا وجود داشته باشه .



ان شاء ... از این به بعد بتونم بیشتر فعال باشم . با آرزوی موفقیت و سربلندی برای تمامی عزیزان .

نکات اضافه شده:

مهدی: فکر کنم مجتبی جان یادش رفته راجع به تابع free در این TemplateEngine توضیح بدهند، که من توضیحش را می دهم که آموزش کامل بشود ...

تابع free را وقتی از داخل Class صدا بزنید، شی شما را reset می کند (همه ی تنظیمات و بلاک های اضافه شده و ... از بین می رود) و باید دوباره همه چیز را مقدار دهی کنید ... از تعیین قالب گرفته تا...

SQL Injection و مقابله با آن ...

سلام دوستان،

همون طور که قول داده بودم می خوام در زمینه (SQL Injection تزریق SQL یا Query) صحبت کنم ...

همون طور که می دونید Query یک واسطه بین برنامه و DataBase هست ...
و همچنین می دونید که برای داشتن برنامه های Dynamic باید Query ها را با توجه به ورودی (های) کاربر تغییر بدیم ...

فرض کنید ما به فرم Login داریم و برای ورود کاربر از قطعه کد زیر استفاده می کنیم :

```
کد :
<?php
$username = $_POST['Username'];
$password = $_POST['Password'];

$sql = "SELECT * FROM `users` WHERE `u_name` = '$username' AND
`u_pass` = '$password'";
$result = mysql_query($sql);

if( mysql_num_rows($result) > 0)
{
    print('welcome to your profile...');
} else {

    die('The specified username and/or password is invalid!');
}
?>
```

خوب مشاهده می کنید که به ظاهر همه چیز مرتبه و کاربر با وارد کردن user و pass خودش می تونه Login کنه و اگه اشتباه بود، نمی تونه ...

خوب به کاربر شیطان اول می آد به username که Single Quotation توش هست مثل jo'hn رو امتحان می کنه تا ببینه که میتونه از این روش استفاده بیره یا نه ... خوب بعد از فهمیدن ورودی ها رو به این شکل وارد می کنه :

```
کد :
username: a
password: ' or '1'='1
```

خوب ببینیم SQL ما به چه شکل در میاد :

کد :

```
SELECT * FROM `users` WHERE `u_name` = 'a' AND `u_pass` = " or '1'='1'
```

خوب نتیجه Query رو خودتون می دونید دیگه آگه حداقل یک کاربر عضو سایت باشه به راحتی می تونیم Login کنیم .

حالا به فرض این که هکر username رو می دونه ...

کد :

```
username: admin'--  
password: 123
```

خوب SQL :

کد :

```
SELECT * FROM `users` WHERE `u_name` = 'admin'--' AND `u_pass` = '123'
```

همون طور که می دونید در MySQL ؛ -- علامت Comment هست ... البته علامت های دیگری مثل /* ... */ و # هم در Mysql هستند ... یعنی در واقع Query ما به این شکل در می آد :

کد :

```
SELECT * FROM `users` WHERE `u_name` = 'admin'
```

خوب این هم که بدیهیه و Login با موفقیت انجام می شه .

البته همه چیز به یک Login ختم نمی شه بلکه هکر می تونه کل اطلاعات مربوط به ساختار و داده های DataBase ما رو در بیاره ... به این شکل :

نکته : البته به این شکلی که نوشته شده MySQL خطا نمی ده ولی SQL server و ... خطا می دن ...

کد :

```
username: ' having 1=1--  
password: 123
```

کد SQL :

کد :

```
SELECT * FROM `users` WHERE `u_name` = " having 1=1--' AND `u_pass` = '123'
```

پیغام خطایی که SQL server میده، اینه :

کد :

```
[Microsoft][ODBC SQL Server Driver][SQL Server]Column 'user.u_name' is  
invalid in the select list because it is not contained in an aggregate function and  
there is no GROUP BY clause.
```

که از اون می شه اسم **Table** و اولین فیلد جدول رو به دست آورد ...
همین طور که میبینید هکر از **Error** ها خیلی بهره می بره ...
به همین ترتیب می شه کار های دیگه ای هم کرد مثلا :

کد :

```
username: '; DROP TABLE `users` --  
password: 123
```

```
SELECT * FROM `users` WHERE `u_name` = "; DROP TABLE `users` -- AND  
`u_pass` = '123'
```

خوب می بینید که این اصلا خوب نیست ...

استفاده از **UNION** :

کد :

```
username: ' UNION select max(`u_name`) from `users` --  
password: 123
```

```
SELECT * FROM `users` WHERE `u_name` = " UNION select max(`u_name`)  
from `users` -- AND `u_pass` = '123'
```

خطایی که سرور به ما میدهد اینته :

کد :

```
#1222 - The used SELECT statements have a different number of columns
```

خوب ما تعداد فیلد های **SQL** برنامه رو نمی دونیم به همین دلیل کار خودمون رو با افزودن یک فیلد دیگر در **UNION** ادامه میدیم :

کد :

```
username: ' UNION select max(`u_name`),0 from `users` --  
password: 123
```

```
SELECT * FROM `users` WHERE `u_name` = " UNION select max(`u_name`),0  
from `users` -- AND `u_pass` = '123'
```

```
#1222 - The used SELECT statements have a different number of columns
```

باز هم همون خطای قبلی رو دیده ... این کار رو این قدر ادامه میدیم تا تعداد فیلد ها رو به دست بیاریم .
همچنین با استفاده از این باگ میتونیم اعمالی چون **Insert, Delete** رو هم انجام بدیم ...
در ضمن فقط فرم **Login** که نیست هر جایی که با دیتابیس کار کنه میتونه مورد حمله واقع بشه اما **Login** حساسیت
بیشتری داره ...

خوب حالا چاره چیه ...

اول بریم سراغ **Login** ، کد مربوط به **Login** رو به این شکل تغییر می دیم :

کد :

```
<?php
$username = $_POST['Username'];
$password = md5($_POST['Password']);

$username = strtolower($username);
$username = preg_replace("[^a-z0-9_]*", "", $username);

$sql = "SELECT `u_pass` FROM `users` WHERE `u_name` = '$username'
LIMIT 0,1";
$result = mysql_query($sql);

if(mysql_num_rows($result) == 1 && $password == mysql_result($result, 0))
{
    print('welcome to your profile...');
} else {
    die("The specified username and/or password is invalid!");
}
?>
```

اولا که **Password** رو حتما **Hash** کنید که الگوریتم های مختلفی در این زمینه وجود داره ... البته باز هم محدود به رد کردن یک یا چند بار از **md5** نشید بلکه با چیزهای دیگری مثل تاریخ عضویت یا کد منحصر به فردی مخلوطش کنید تا حسابی قرو قاطی بشه ... مثلا این طوری :

کد :

```
$hashedPass = md5(strev(md5($password) . $registerDate));
```

خلاصه به روال من در آوردی ... که حتی تو هر پروژه ای که می نویسد هم عوض بشه .

نکته بعد که خیلی مهمه اینه که نام کاربری فقط همونایی باشه که می خواهیم و دیگه کاراکتر های اضافی توش نباشه :

کد :

```
$username = preg_replace("[^a-z0-9_]*", "", $username);
```

که البته موقع ثبت نام کاربر هم **Username** رو از این فیلتر رد میکنیم و بهش میگیم که از چه چیزهایی می تونه استفاده کنه .

نکته بعد طرز نوشتن **SQL** مون هست :

کد :

```
SELECT `u_pass` FROM `users` WHERE `u_name` = '$username' LIMIT 0,1
```

همون طور که می بینید فقط **Password** کاربر از دیتابیس بیرون کشیده شده (البته توی برنامه های واقعی چیزای دیگه ای هم هست که می خونیم) ...

بعد هم به شرط SQL دقت کنید که فقط **username** رو مورد بررسی قرار می ده همچنین به **LIMIT 0,1** هم دقت کنید که فقط به رکورد رو بیرون می کشیم .

نکته خیلی مهم اینجاست که **Password** رو خود ما چک می کنیم و اونو به دست SQL نمی سپاریم :

کد :

```
if( mysql_num_rows( $Rslt) == 1 && $Password == mysql_result( $Rslt , 0))
```

یعنی اگه به هر شکل هکر بتواند SQL رو هم دور بزنه، این جا رو نمی تونه دور بزنه ...

نکات دیگه ای هم هست که برای جاهای دیگه استفاده می شه و اون اینه که کاراکترهای اضافی رو از طریق استفاده از تابع زیر از ورودی هامون حذف کنیم :

کد :

```
function EscapeString( $str , $Conn = 0)
{
    if(function_exists('mysql_real_escape_string') && $Conn) {
        return mysql_real_escape_string( $str, $Conn );
    } else {
        return mysql_escape_string( $str );
    }
}
```

نکته بسیار مهم دیگه اینه که برای دیتابیسمون دوتا **user** بسازیم، یکی فقط می تونه **SELECT** رو انجام بده و دیگری فقط **SELECT, UPDATE, INSERT, LOCK TABLE** رو بتونه انجام بده، و موقعی که می خواهیم **Query** فقط خواندن رو اجرا کنیم با **user** اولی به دیتابیس **Connect** می شیم، که اگه هکر بتونه چیزی رو هم تزریق کنه که توی اطلاعات ما تغییر ایجاد کنه، خود **DBMS** اجازه همچنین کاری رو نده ... در ضمن به **user** های دیتابیس دسترسی های خطرناکی چون **DROP TABLE, ALTER, ...** رو ندیم .

یک نکته مهم دیگه اینه که مثلا در فیلد هایی که **int** هستند و بر اساس این فیلد جستجو انجام میدین ورودی **GET** یا **POST** رو چک کنید که حتما همون چیزی باشه که باید باشه مثلا اگه قراره **int** باشه اون رو از تابع **intval** رد کنید

کد :

```
$NewsID = intval( $_GET[ 'ID' ] );
$SQL = "SELECT * FROM `news` WHERE `ID` = $NewsID LIMIT 0,1";
```

و نکته بسیار مهم آخر اینه که نگذارید کسی خطاهای برنامه شما را ببیند (حتی اطلاعات سرور و زبان برنامه نویسی رو هم مخفی کنید خیلی خوبه، چون همین خطاها هستند که به هکرها اطلاعات میدن ... برای این کار بعد از این که کل پروژه ساخته شد و خواستید اون رو **publish** کنید بالای یکی از فایل هایی که توی همه فایل های برنامه **include** می شه تابع زیر رو به این شکل صدا بزنید :

کد :

```
error_reporting( 0);
```



با آرزوی موفقیت و سربلندی برای همه شما عزیزان .

کد افراد آنلاین



عباس: من به کد ساختم که تعداد افراد آنلاین رو نشون میده .

جدولش در دیتابیس به شکل زیره :

کد :

```
CREATE TABLE online(  
session char(100) NOT NULL,  
time int(11) NOT NULL,  
UNIQUE(session),  
)
```

خود کد هم به شکل زیر است :

کد :

```
<html dir=rtl>  
<body style="font-family:tahoma;font-size:8pt">  
<?  
session_start();  
$session=session_id();  
$time=time();  
$time_c=$time-600;  
$tbl_name="online";  
mysql_connect("localhost","root","") or die("اتصال به سرور مشکل در");  
mysql_select_db("new");  
  
$q="INSERT INTO $tbl_name(session,time) VALUES('$session','$time)";  
mysql_query($q);  
  
$q1=mysql_query("SELECT * FROM $tbl_name");  
$num=mysql_num_rows($q1);  
echo "نفر آنلاین : $num";  
  
mysql_query("DELETE FROM $tbl_name WHERE time<$time_c");  
?>
```

طرز کارش هم اینه که session های تمامی افراد رو در جدول دیتابیس ثبت میکنه و بعد هم تعداد session های ثبت شده رو میشماره .

معرفی سایت به دوستان

یکی از راههای به درد بخور در افزایش بازدید کننده قسمت معرفی به دوستان است که در زیر کد **php** اون رو گذاشتم :

```
کد :
<?
$body = 'Catalog of Our Site';

function send_mail_to_friend($mail_to,$mail_from) //Send Catalog of Site...
{
    global $body;
    $mail_subject = "My Site Catalog Mail Subject";
    if(mail($mail_to, $mail_subject, $body, "From: <$mail_from>n"."MIME-Version:
1.0n"."Content-type: text/html; charset=utf-8")) {
        return true;
    }
    return false;
} //End of send_mail_to_friend Function

if(isset($_POST['send_mail']))
{
    $mail_to = isset($_POST['to']) ? htmlspecialchars($_POST['to']) : "";
    $mail_from = isset($_POST['from']) ? htmlspecialchars($_POST['from']) : "";

    if(send_mail_to_friend($mail_to,$mail_from))
    {
        print("Your Mail is Send!");
    } else {
        print("Error While Sendin Mail!");
    }
    exit();
}
//This Code Downloaded From www.RightClick.ir
?>

<head>
<meta http-equiv="Content-Language" content="fa">
</head>
<body>

<form method="POST">

    <p><span lang="en-us">Your Email : </span><input type="text" name="from"
size="20"></p>
    <p><span lang="en-us">Your Friend Email :</span><input type="text"
name="to" size="20"></p>
    <p><input type="submit" value="Submit" name="send_mail"></p>
</form>

</body>
</html>
```

توضیح :

این کد در یک صفحه با هر نامی که خواستید و با پسوند **php** ذخیره کنید .
حال ببینیم چیزهایی که می خواهید برای شخص دریافت کننده ارسال شود کجا می نویسیم .

خوب **Front Page** رو باز کنید و یک صفحه قشنگ توی اون درست کنید و توی اون صفحه سایت خود و امکانات اون رو معرفی کنید. راستی اگه فارسی نوشتید به صورت یونیکد باشه .
حالا در خط دوم

کد :

```
$body = 'Catalog of Our Site';
```

به جای عبارت **Catalog of Our Site** سورس صفحه ای که ساختید رو قرار بدین و دیگه تموم شد.

در ضمن عبارت های زیر رو هم می تونید تغییر بدین تا پیغام مناسب چاپ بشه :

My Site Catalog
Mail Subject
Your Mail is Send!
Error While Sending Mail!

برای سایت خود یک قسمت "ارتباط با ما" بسازیم

سلام دوستان 😊

آیا فکر نمی کنید که برای ارتباط بازدید کنندگان سایت شما با شما نیاز به یک قسمت است که آنها بتوانند به راحتی با شما ارتباط داشته باشند...؟ 😊

خوب با هم این قسمت را درست می کنیم :

اول از همه چون این کدی که می‌گم با **php** هستش و نیاز به ارسال ایمیل داره باید مطمئن شویم که سرور ما از **php** پشتیبانی میکند و ما می توانیم از **mail Server** استفاده کنیم .

کد :

```
<?php
$admin_email = "YourEmail";

if(isset($_POST['submit_msg']))
{
    if (empty($_HTTP_POST_VARS['name_msg'])){
        die('<font face="Tahoma" color="#003399" style="font-size: 9pt"> لطفا نام خود را
        نمایش دهید.</font>');
    };
    if (empty($_HTTP_POST_VARS['email_msg'])){
        die('<font face="Tahoma" color="#003399" style="font-size: 9pt"> لطفا ایمیل خود
        نمایش دهید.</font>');
    };
    if (empty($_HTTP_POST_VARS['msg_body'])){
        die('<font face="Tahoma" color="#003399" style="font-size: 9pt"> لطفا پیام خود را
        نمایش دهید.</font>');
    };

    $name_msg = htmlspecialchars($_HTTP_POST_VARS['name_msg']);
    $name_msg = str_replace("&","&",$name_msg);
    $name_msg = str_replace("ي","ي",$name_msg);

    $msg_body = htmlspecialchars($_HTTP_POST_VARS['msg_body']);
    $msg_body = str_replace("&","&",$msg_body);
    $msg_body = str_replace("ي","ي",$msg_body);
    $msg_body = nl2br($msg_body);

    // check empty for url...
    if (!empty($_HTTP_POST_VARS['url_msg']))
    {
        $url_msg = str_replace("http://","$_POST['url_msg']");
        $name_msg = "<a target=_blank
href=http://".$_htmlspecialchars($url_msg).">".$name_msg."</a>";
    }

    if(!strpos($_POST['email_msg'], "@"))
    {
        print("<font color=#ff0000 face=tahoma > ورودی نا معتبر >");
        die("<br><br>Example : <u>YourEmail@SiteName.Com</u></font>"\);
    }
}
```



```

};

$from = htmlspecialchars($HTTP_POST_VARS['email_msg']);
// start of Email Body...
$body = '<html><head><title>.:| Contact |:.</title><base
target="_blank"></head>
<body text="#000000">
<p align="right" dir="rtl"><font face="Tahoma" size="2">سلام
به دوست عزیزم
است.یک نفر برای شما پیامی ارسال نموده
</font></p>
<p align="right" dir="rtl"><font face="Tahoma" size="2">نام :
'. $name_msg.</font></p>
<p align="right" dir="rtl"><font face="Tahoma" size="2">پیام :
</font></p>
<p align="right" dir="rtl"><font face="Tahoma" size="2"
color="#003399">'. $msg_body.</font></p>
<p align="center" dir="rtl">&nbsp;</p>
<p align="center" dir="rtl"><font face="Tahoma" style="font-size: 9pt">آرزوی با
شما موفقیت برای شما
</font></p>
<p align="center" dir="rtl">
<font face="Tahoma" color="#003399" style="font-size: 9pt">
<a href="http://www.rightclick.ir/">کلیک رایت</a></font></p>
</body>
</html>';

//End of Email Body ...
if(!mail($admin_email,'New Contact',$body,"From: <$from>\n"."MIME-Version:
1.0\n"."Content-type: text/html; charset=utf-8"))
{
print("The Mail Could not be send.");
}else
{
print('<font face="Tahoma" color="#003399" style="font-size: 9pt">پیام شما با
شد موفقیت برای مدیر سایت ارسال
</font>');
}

} //End of set Submit...
else
{
?>
<html>
<head>
<title>Contact Us ...</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta http-equiv="Content-Language" content="fa">
<style>
.TEXT{
font-family:Tahoma; font-size:9pt;
}
</style>
</head>
<body>
<form name="msg_form" method="POST" >
<div align="center">

```

```

<table border="0" width="359" cellpadding="2">
  <tr>
    <td width="242" dir="rtl"><font face="Tahoma"><span style="font-size:
9pt">
<input name="name_msg" size="22" class="TEXT" dir="rtl" tabindex="1"
maxlength="35" lang="fa" style="float: right"></span></font></td>
    <td dir="rtl">
<p dir="rtl" align="center"><font face="Tahoma"><span style="font-size: 9pt">نام
شما<font color="#FF0000">
<span lang="en-us">*</span></font></span></font></p></td>
  </tr>
  <tr>
    <td width="242" dir="rtl">
    <p align="right"><font face="Tahoma"><span style="font-size: 9pt">
    <input type="text" name="url_msg" size="22" dir="ltr" tabindex="2"
maxlength="30" class="TEXT"></span></font></td>
    <td dir="rtl">
    <p dir="ltr" align="center"><span lang="en-us">
    <font face="Tahoma" style="font-size: 9pt">Web</font></span></td>
  </tr>
  <tr>
    <td width="242" dir="rtl"><font face="Tahoma"><span style="font-size:
9pt">
    <input name="email_msg" size="22" dir="ltr" tabindex="3" maxlength="40"
class="TEXT" style="float: right"></span></font></td>
    <td dir="rtl"><p dir="ltr" align="center">
    <font face="Tahoma"><span style="font-size: 9pt"><font color="#FF0000">
<span lang="en-us">*</span> </font></span></font>
    <font face="Tahoma" style="font-size: 9pt"><span lang="en-
us">Email</span></font></p>
    </td>
  </tr>
  <tr>
    <td width="242" dir="rtl">
    <p align="right"><font face="Tahoma"><span style="font-size: 9pt">
    &nbsp;<textarea rows="5" name="msg_body" cols="33" dir="rtl"
class="TEXT" tabindex="4" lang="fa"></textare></span></font></td>
    <td dir="rtl">
    <p align="center"><font face="Tahoma"><span style="font-size: 9pt">
    شما پیام<font color="#FF0000"><span lang="en-
us">*</span></font></span></font></td>
  </tr>
</table>
</div>
<p dir="rtl" align="center">
  <input type="reset" value="reset" name="reset_btn" ><input type="submit"
value="Submit" name="submit_msg" tabindex="5" ></p></form>

<?php
} //End of Else...
?>
</body>
</html>

```

برای این کار از تابع **mail** استفاده می کنیم که یکی از ساده ترین راههای ارسال ایمیل در **php** است .
طرز کار این کد به این شکل است که اطلاعات را در قالب یک فرم از کاربر دریافت نموده و برای مدیر سایت ارسال می کند .

توجه کنید که در این کد در خط دوم به جای عبارت **YourEmail** بایستی ایمیل مدیر سایت نوشته شود و فایل را با هر نامی که خواستید ، با پسوند **php** و در هر جایی که خواستید ذخیره کنید . 😊

آموزش برنامه آپلود عکس (خیلی ساده)



یه آموزش خیلی ساده برای افراد مبتدی. 😊
میخواهیم یه برنامه بنویسیم که به وسیله اون عکسامون رو آپلود کنیم. (البته یه برنامه خیلی خیلی ساده)
برای این کار نیاز به دو فایل داریم یکی با اسم **upload.php** و دیگری به اسم **upload-a.php** :
فایل **upload.php** به شکل زیره :

کد :

```
<html dir=rtl>
<form action=upload.php method="post" enctype="multipart/form-data">
<input type=file name=up><br>
<input type=submit value=UPLOAD>
</form>
</html>
```

نکته: به این قسمت از کد دقت کنید، حتما باید باشه !

کد :

```
enctype="multipart/form-data"
```

فایل دوم :

کد :

```
<html dir=rtl>
<?
$path=$HTTP_POST_FILES['up']['name'];
if($path!='none'){
if(copy($HTTP_POST_FILES['up']['tmp_name'],$path)){
echo "شد فایل شما آپلود"<br>;
echo "اسم:". $HTTP_POST_FILES['up']['name']. "<br>";
echo "نوع:". $HTTP_POST_FILES['up']['type']. "<br>";
echo "اندازه:". $HTTP_POST_FILES['up']['size']. "<br>";
echo "<img src='$path' width=150 height=150>";
}
}else{
die("انتخاب کن فایل رو");
}
?>
```

توضیحات :

کد :

```
$path=$HTTP_POST_FILES['up']['name'];
```

متغیر از پیش تعریف شده خود **php** هست که به وسیله این کد ما نام فایلی که میخوایم آپلود کنیم رو بدست میاریم .
در مرحله بعد از یه شرط ساده استفاده شده که میگه اگه مسیر (متغیر **path**) خالی نبود فایل رو آپلود کن .
به وسیله تابع **copy()** که دو قسمت داره فایل رو در مسیری که میخوایم با همون نام که روی کامپیوتر هست ذخیره کنیم .

نکته: متغیر **path** رو به شکل زیر هم میشه گذاشت .

کد :

```
$path="upload/".$HTTP_POST_FILES['up']['name'];
```

با این فرق که عکسامون در فولدر **upload** ذخیره میشه .

نکته :

کد :

```
$HTTP_POST_FILES['up']['name']  
$HTTP_POST_FILES['up']['type']  
$HTTP_POST_FILES['up']['size']
```

این متغیرها هم شکل کلیشون همینه.(متغیرهای از پیش تعریف شده در **php** هستن)

شما کارهای دیگه هم میتونید بکنید مثلا با یه شرط ساده برای اندازه فایل محدودیت بزارید . 😊

یه نکته رو هم من اضافه کنم و اون اینکه که از **php 4** به بعد می تونیم به جای

کد :

```
$HTTP_POST_FILES
```

از

کد :

```
$_FILES
```

هم استفاده کنیم.

سؤال های کاربران و جواب ها:

GD چیست ؟

یکی از کتابخانه های **php** هست که به ما اجازه میده روی عکس و فلش کار کنیم ... کتابخونه چیه ...؟ مجموعه ای از توابع از پیش نوشته شده در مورد خاص مثل کامپوننت توی زیون دلفی ...
برای نمونه این هم میتونی توی فرم ارتباط با ما همین سایت اون کد امنیتی رو بررسی کنی میبینی که **URL** کد امنیتی یه فایل **php** هست که اونجا تولید میشه .

آموزش تابع رندوم (شانسی):

طرز استفاده از تابع رندوم :

کد :

```
$a = rand(0 , 50);
```

توی متغیر **a** یک مقدار تصادفی از ۰ تا ۵۰ قرار میگیره .

توجه : ۰ و ۵۰ حتما باید **int** باشند. خروجی تابع هم **int** هست.

می خواهم بدانم که کدی وجود دارد که با آن بشود وسط پی اچ پی جاوا اسکریپت نوشت؟
و کدی وجود دارد که آلرت جاوا اسکریپت را بدهد؟ منظورم همان پنجره ی کوچکی که باز می شود و پیامی می دهد.
بین اگه منظورت اینه که کد **php** تا جایی اجرا بشه و بعد **Alert** بدهد و بعد با کلیک کردن روی **Ok** بقیه کد اجرا بشه نه ... ولی میتونی با دستور **print** ساده و یا این که تگ **php** رو ببندی بعد تگ **java Script** رو باز کنی :

: ۱

کد :

```
print('<script language="javascript">
alert("Salam");
</script>');
```

: ۲

کد :

```
...
?>
<script language="javascript">
alert("Salam");
</script>
<?php
...
```

البته اگر کد جاوا اسکریپت وسط **php** استفاده کنی وقتی کد جاوا اسکریپت اجرا میشه که اجرای صفحه **php** تموم شده باشه منظورم اینه که کد های زیر دستور **print** و اون تگه هم اجرا میشود مگر این که خودت اون رو **exit** کنی.

کدی وجود دارد که بتوان با آن محتوای یک صفحه ی اچ تی ام ال را بصورت ایمیل فرستاد ؟ آخه من یک با کمک سیستم آموزشی شما و یک کد سیستمی برای فرستادن ایمیل ساختم اما مشکل در این هست که کد های اچ تی ام ال را بصورت مستقیم در نامه نشان می دهد و اونو بصورت یک صفحه نمی فرسته چکار باید کنم ؟

برای این کد ها رو توی ایمیل نشون می ده که فکر میکنه اونا **text** هستند ما باید موقع ارسال بهش بگیم که اینا **html** هستند . با فرستادن هدر مربوط به اون این کار رو انجام میدیم :

توی تابع **mail** یه آرگومان چهارم به این شکل اضافه کنید :

کد :

```
mail($mail_to, $mail_subject, $mail_body, "From: <$mail_from>n"."MIME-Version: 1.0n"."Content-type: text/html; charset=utf-8")
```

یعنی کد زیر که در قسمت بالا اومده بهش می فهمونه که محتوای صفحه **HTML** هست و انکود اون **UTF-8** هست :

کد :

```
"MIME-Version: 1.0n"."Content-type: text/html; charset=utf-8"
```

اگر انکود صفحه شما چیز دیگه ای هست به جای **utf-8** انکود اون رو می نویسد ولی اگر می خواهید ایمیل فارسی بفرستید حتما **utf-8** باشه ... البته **Arabic (windows-1256)** هم میتونه باشه ولی وب میل هایی مثل یاهو درست نمی تونن نشونش بدن یعنی پیشفرضشون همون **utf-8** هست.

چه طوری میشه از تابع **SMTP** استفاده کرد ؟؟؟

SMTP تابع خاصی نیست اگر خودت بخوایی بنویسی خیلی در دسر داره و خیلی مشکله ... اما کلاس ها و توابعی در این زمینه نوشته شده که میتونیم از اینه استفاده کنیم ... یه منبع خوب که خودم تستش کردم رو بهت معرفی میکنم :

برنامه **Contact us** که یک برنامه مجانی و **Open Source** است رو دانلود کن و در فولدر **includes** یه فایل هست به نام **smtp.php** که از برنامه **phpbb** گرفتم و اون جوری که می خواستمش ویرایشش کردم و کار باهاش هم راحتیه ...

توی همون برنامه میتونی ببینی که چکار کردم.

کد :

```
<?php
/*****
*
* RightClick CMS var 2.0
* -----
* begin : Monday, July 29, 2006
```

```

* copyright      : (C) 2006 The RightClick $Team = "www.rightclick.ir";
* email         : info@rightclick.ir
*
* $Id: RightClick CMS var 2.0 2006/07/29 13:20:00 $
*
*****/

//Configuration...

$Config = array(
    'Smtphost' => 'localhost',
    'Smtusername' => 'user',
    'Smtpassword' => 'pass'
);

function server_parse($socket, $response, $line = __LINE__)
{
    while (substr($server_response, 3, 1) != ' ')
    {
        if (!$server_response = fgets($socket, 256))
        {
            die ("Couldn't get mail server response codes". $line. __FILE__);
        }
    }

    if (!(substr($server_response, 0, 3) == $response))
    {
        die("Ran into problems sending Mail. Response: $server_response". $line. __FILE__);
    }
}

// Replacement or substitute for PHP's mail command
function smtpmail($mail_to, $SenderMail, $subject, $message, $headers = ")
{
    global $Config;

    // Fix any bare linefeeds in the message to make it RFC821 Compliant.

    $message = preg_replace("#(?<!r)n#si", " ", $message);

    if ($headers != ")
    {
        if (is_array($headers))
        {
            if (sizeof($headers) > 1)
            {
                $headers = join("\n", $headers);
            }
            else
            {
                $headers = $headers[0];
            }
        }
    }
}

```



```

$headers = chop($headers);

// Make sure there are no bare linefeeds in the headers
$headers = preg_replace('#(?<!r)n#si', " ", $headers);

// Ok this is rather confusing all things considered,
// but we have to grab bcc and cc headers and treat them differently
// Something we really didn't take into consideration originally
$header_array = explode(" ", $headers);
@reset($header_array);

$headers = "";
while(list(, $header) = each($header_array))
{
    if (preg_match('#^cc:#si', $header))
    {
        $cc = preg_replace('#^cc:(.*)#si', '1', $header);
    }
    else if (preg_match('#^bcc:#si', $header))
    {
        $bcc = preg_replace('#^bcc:(.*)#si', '1', $header);
        $header = "";
    }
    $headers .= ($header != "") ? $header . " " : "";
}

$headers = chop($headers);
$cc = ";//explode(',', $cc);
$bcc = ";//explode(',', $bcc);
}

if (trim($subject) == "")
{
    die ("No email Subject specified". __LINE__ . __FILE__);
}

if (trim($message) == "")
{
    die ("Email message was blank". __LINE__ . __FILE__);
}

// Ok we have error checked as much as we can to this point let's get on
// it already.
if( !$socket = @fsockopen($Config['SmtpHost'], 25, $errno, $errstr, 20) )
{
    die ("Could not connect to smtp host : $errno : $errstr". __LINE__ . __FILE__);
}

// Wait for reply
server_parse($socket, "220", __LINE__);

// Do we want to use AUTH?, send RFC2554 EHLO, else send RFC821 HELO
// This improved as provided by SirSir to accomodate
if( !empty( $Config['SmtpUsername']) && !empty($Config['SmtpPassword']) )

```

```

{
    fputs($socket, "EHLO " . $Config['SmtpHost'] . " ");
    server_parse($socket, "250", __LINE__);

    fputs($socket, "AUTH LOGIN ");
    server_parse($socket, "334", __LINE__);

    fputs($socket, base64_encode($Config['SmtpUsername']) . " ");
    server_parse($socket, "334", __LINE__);

    fputs($socket, base64_encode($Config['SmtpPassword']) . " ");
    server_parse($socket, "235", __LINE__);
}
else
{
    fputs($socket, "HELO " . $Config['SmtpHost'] . " ");
    server_parse($socket, "250", __LINE__);
}

// From this point onward most server response codes should be 250
// Specify who the mail is from....
fputs($socket, "MAIL FROM: <" . $SenderMail . "> ");
server_parse($socket, "250", __LINE__);

// Specify each user to send to and build to header.
$to_header = "";

// Add an additional bit of error checking to the To field.
$mail_to = (trim($mail_to) == "") ? 'Undisclosed-recipients;' : trim($mail_to);
if (preg_match('#[^\ ]+@[^\ ]+#[', $mail_to))
{
    fputs($socket, "RCPT TO: <$mail_to> ");
    server_parse($socket, "250", __LINE__);
}

// Ok now we tell the server we are ready to start sending data
fputs($socket, "DATA ");

// This is the last response code we look for until the end of the message.
server_parse($socket, "354", __LINE__);

// Send the Subject Line...
fputs($socket, "Subject: $subject ");

// Now the To Header.
fputs($socket, "To: $mail_to ");

// Now any custom headers....
fputs($socket, "$headers ");

// Ok now we are ready for the message...
fputs($socket, "$message ");

// Ok the all the ingredients are mixed in let's cook this puppy...

```

```

fputs($socket, ". ");
server_parse($socket, "250", __LINE__);

// Now tell the server we are done and close the socket...
fputs($socket, "QUIT ");
fclose($socket);

return true;
}

?>

```

بین توی کد بالا که در ورژن جدید همین سایت استفاده خواهد شد (البته برای شما به کوچولو تغییرش دادم که به تنهایی قابل استفاده باشه) به تابع هست به نام **smtpmail** که کار ارسال رو انجام میده فقط کافیه بعد از این که مقادیر **Set** و **Config** رو کردید این تابع رو با ورودی هاش فراخونی کنید.

چه طوری با کوکی ها در پی اچ پی میشه کار کرد؟

برای ایجاد کوکی ها از تابع **set_cookie** استفاده میشه :

```

: کد
setcookie ("name", "value");

//Ex:
setcookie ("Color", "#FF0000");

```

این تابع ورودی های دیگه ای هم داره که امکانات بیشتری رو در استفاده از کوکی ها در اختیار ما میگذاره مثلا تاریخ انقضای کوکی، دومین مربوط به کوکی، مسیر ذخیره سازی و ...

```

: کد
setcookie ("name", "value", "expiration", "path", "domain", "secure");

//Ex:
setcookie ("Color", "#FF0000", time() + 3600);
setcookie ("Color", "#FF0000", time() + 3600, "/user/");
setcookie ("Color", "#FF0000", time() + 3600, "", "", "1");
setcookie ("Color", "#FF0000", time() + 10000000, "", "", "");

```

برای خواندنش :

```

: کد
//Ex:
print($_COOKIE['Color']);

```

و برای حذف :

```

: کد
//Delete:
setcookie ("UserName", "");

```

```
setcookie("UserName","",time()-60);
```

چه طوری همیشه آدرس یک RSS را داخل پی اچ پی وارد کنیم و محتویات آن را پی اچ پی به روی دیتابیس انتقال بده ؟

فرض کن جدول ما اینه :

کد :

```
CREATE TABLE `bairro` (  
  `hash` int(11) NOT NULL default '0',  
  `descr` varchar(15) NOT NULL default '',  
  PRIMARY KEY (`hash`)  
) TYPE=MyISAM;
```

به فایل Xml هم داریم این طوری :

کد :

```
<?xml version="1.0" encoding="iso-8859-1"?>  
<data>  
<table name="bairro" primary_key="hash">  
  <row>  
    <field name="hash">1</field>  
    <field name="descr">Centro</field>  
  </row>  
</table>  
</data>
```

یه کلاس پیدا کردم که به راحتی این کار رو انجام میده ... زیر اون کلاس کد نمونه گذاشته شده :

کد :

```
<?php  
class xmlToDB {  
  var $parser;  
  var $cn;  
  var $table;  
  var $recordset = array();  
  var $content;  
  var $value;  
  var $field;  
  
  function xmlToDB() {  
    $this->parser = xml_parser_create();  
    xml_set_object($this->parser,$this);  
    xml_set_element_handler($this->parser,"tag_open","tag_close");  
    xml_set_character_data_handler($this->parser,"cdata");  
  
    $this->cn =  
    @mysql_pconnect(XOOPS_DB_HOST,XOOPS_DB_USER,XOOPS_DB_PASS)  
    or trigger_error('Erro ao conectar ao servidor de dados',E_USER_ERROR);  
    @mysql_select_db(XOOPS_DB_NAME,$this->cn) or trigger_error('Erro ao  
    selecionar o banco de dados',E_USER_ERROR);
```

```

}
function parse($data) {
    @xml_parse($this->parser, $data) or
    die(sprintf("Erro de XML: %s na linha %d",
    xml_error_string(xml_get_error_code($this->parser)),
    xml_get_current_line_number($this->parser)));
}
function tag_open($parser, $tag, $attributes) {
    $parser;
    switch ($tag) {
        case 'DATA': break;
        case 'TABLE': {
            $this->table = $attributes['NAME'];
            mysql_unbuffered_query("TRUNCATE $this->table");
            $this->recordset = array();
            //[todo]primary key n?o usada
        } break;
        case 'ROW': break;
        case 'FIELD': {
            $this->content = (isset($attributes['CONTENT'])) ? $attributes['CONTENT'] : "";
            $this->field = $attributes['NAME'];
        }
    }
}
function cdata($parser, $cdata) {
    $parser;
    switch ($this->content) {
        case 'base64': {
            $this->value = base64_decode($cdata);
        } break;
        default: {
            $this->value = $cdata;
        }
    }
}
function tag_close($parser, $tag) {
    $parser;
    switch ($tag) {
        case 'DATA': break;
        case 'TABLE': break;
        case 'ROW': {
            $sql = "INSERT INTO $this->table";
            $iC=0;
            foreach ($this->recordset as $caption => $tvalue) {
                if ($iC==0) $sql .= "nSET "; else $sql .= ", ";
                $sql .= "`$caption` = ";
                $sql .= mysql_real_escape_string($tvalue,$this->cn);
                $sql .= """";
                $iC++;
            }
            mysql_unbuffered_query($sql,$this->cn);
        } break;
        case 'FIELD': {
            $this->recordset[$this->field] = $this->value;
        }
    }
}

```

```

    }
  }
}
}

//Example for use class

$file = 'MySample.xml';//Xml File

//require('mainfile.php');

//require('config.php');
define('XOOPS_DB_HOST', 'localhost');
define('XOOPS_DB_USER', 'root');
define('XOOPS_DB_PASS', '');
define('XOOPS_DB_NAME', 'testDb');

$xmlparse = new xmlToDB();

$num = @fopen($file,'r') or exit();//trigger_error('Erro ao abrir o
XML',E_USER_ERROR);

while (($data = fread($num,4194304))) {
  $xmlparse->parse($data);
}

fclose($num);

?>

```

آقا فقط یک مشکلی داره فارسی را امپورت نمیکنه !!!
 یه بررسی کن ببین اگه انکود فایل XML یا فایل php ی که توی DB ثبت میکنه رو تغییر بدی شاید کارکنه ...

کد :

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

مهدی: چه جوری میشه نام فیلدی در جدولی پیدا کرد؟

حدس میزنیم :

کد :

```
mysql_result(mysql_query("select * from `".$table_prefix."themes`"),1,3);
```

نام فیلد شماره ی ۳ را چه جوری میشه فهمید؟ و تعداد فیلدهای یک جدول را چه جوری میشه پیدا کرد؟ نه ردیف ها !!!

کد :

```
$ResultRs = mysql_query("SELECT * FROM `".$table_prefix."themes`");
```

```
$FieldName = mysql_field_name($ResultRs, 2);//Name sevomin field.
```

```
$TotalFields = mysql_num_fields($ResultRs);//Teedade Field ha.
```

کد برای پخش تصادفی فلش در PHP و JavaScript :

```
کد :  
<?  
$my_array[] = "swf/swf%20(0).swf";  
$my_array[] = "swf/swf%20(1).swf";  
$my_array[] = "swf/swf%20(2).swf";  
$my_array[] = "swf/swf%20(3).swf";  
$my_array[] = "swf/swf%20(4).swf";  
$my_array[] = "swf/swf%20(5).swf";  
$my_array[] = "swf/swf%20(6).swf";  
$my_array[] = "swf/swf%20(7).swf";  
$my_array[] = "swf/swf%20(8).swf";  
$my_array[] = "SWF Address ... ";  
  
$swf_addr = $my_array[rand(1,count($my_array))-1];  
  
?>  
  
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"  
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/  
swflash.cab#version=6,0,29,0" width="441" height="332">  
<param name="movie" value="<? print $swf_addr; ?>">  
<param name="quality" value="high">  
<embed src="<? print $swf_addr; ?>" quality="high"  
pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-  
shockwave-flash" width="441" height="332"></embed>  
</object>
```

راهنمایی : در کد بالا به هر تعداد که می خواهید می توانید از

```
کد :  
$my_array[] = "";
```

استفاده کنید . در کوتیشن " " آدرس فایل فلش را مینویسیم . هر کدام از این آدرس ها در یک سلول آرایه ریخته می شود سپس با تولید یک عدد تصادفی یکی از این آدرس ها رو انتخاب می کنیم و به نمایش در می آوریم .

و این هم کد جاوا اسکریپت اون :

```
کد :  
<script language='JavaScript' type='text/javascript'>  
  
var num_swf = 5  
day = new Date()  
seed = day.getTime()
```

```

ran = parseInt(((seed - (parseInt(seed/1000,10) * 1000))/10)/100*num_swf + 1,10)

switch (ran)//***** 1 *****
{
case 1 : swf_addr = "swf/swf%20(0).swf";break;
case 2 : swf_addr = "swf/swf%20(2).swf";break;
case 3 : swf_addr = "swf/swf%20(3).swf";break;
case 4 : swf_addr = "swf/swf%20(4).swf";break;
case 5 : swf_addr = "swf/swf%20(5).swf";break;
}

document.write('<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version
=6,0,29,0" width="441" height="332">');
document.write('<param name="movie" value="'+ swf_addr +'><param name="quality"
value="high">');
document.write('<embed src="'+ swf_addr +" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-
shockwave-flash" width="441" height="332"></embed></object>');

</script>

```

برای تغییر این کد به تعداد بیشتر یا کمتر بایستی ابتدا در خط دوم مقدار `var num_swf = 5` رو به تعدادی که می خواهید عوض کنید .

سپس در قسمتی که با

```

//***** 1 *****

```

مشخص شده یک خط به این صورت اضافه می کنید :

کد :

```

case 5 : swf_addr = "SWF ADDRESS";break;

```

که در این خط `SWF ADDRESS` آدرس فایل فلش ما است و عدد 5 نشان دهنده این است که این آدرس مربوط به پنجمین فلش ما است.

چگونه برای سایت خود آرشیو درست کنیم ؟

فرض کن یه جدول داریم به نام `Article` و چند تا فیلد داره که فیلد اول اون `id` هست که نوع اون `int` و `AutoIncrement` هست... همچنین یه فیلد برای تاریخ ثبت به نام `SendDate` داریم و یه فیلد دیگه به نام `ArchiveDate` داریم که به طور پیشفرض خالیه .

کد :

```

SELECT * FROM `Article` WHERE `ArchiveDate` = "

```

خوب سر هر ماه هم `Query` زیر رو اجرا میکنی :

کد :

```
UPDATE `Article` SET `ArchiveDate` = '$ThisMonth' WHERE `ArchiveDate` != "  
AND `SendDate` < '$Today'
```

ThisMonth متغیری هست که تاریخ ماه جاری رو داره و **Today** تاریخ امروز رو داره .

برای خواندن آرشیو :

کد :

```
SELECT `ArchiveDate` FROM `Article` WHERE `ArchiveDate` != " GROUP BY  
`ArchiveDate` ORDER BY `ArchiveDate` DESC
```

ساده ترین راهی که به ذهن من می رسید. 😊

چگونه برای نوشته ها ادامه مطلب بگذاریم ؟

کد :

```
<?  
$Str = 'Salam halet khoobeh?';  
  
$OutPutStr = substr($Str, 0, 5);  
  
$OutPutStr .= '...';  
  
$MoreLink = '<a href="?Action=ShowNews&NewsID=5">More...</a>';  
  
print($OutPutStr . '<br />' . $MoreLink);  
?>
```

اون لینکی هم که می سازه خودت باید براش برنامه بنویسی و بگی که مثلا **NewsID** که ۵ هست همه اون رو نشون بده و...

دستورات کش کردن و جلوگیری از کش چیست ؟

برای کش نکردن از این دستور استفاده کن ...

کد :

```
header("Pragma: no-cache");  
header("Cache-Control:no-cache");
```

و برای کش کردن از این دستور ...

کد :

```
header("Pragma: cache");
```

علامت @ در پشت توابع چه کاربردی دارد؟

@ وقتی قبل از نام به تابع می‌آید در صورتی که تابع با شکست روبرو بشه و خواست خطای خودش رو چاپ کنه نمی‌ذاره. یعنی اگه ارور هم داد هیچی.

کد :

```
@mysql_connect("localhost");
```

کد :

```
mysql_connect("localhost");
```

چه جوری میشه بفهمیم که کاربر از چه آدرسی وارد سایت شده و یک سری از امکاناتش مثل مرورگر سیستم عامل ip و.. رو بدست آورد؟
برای آدرس ورودی :

کد :

```
$_SERVER['HTTP_REFERER']
```

ip :

کد :

```
$_SERVER['REMOTE_ADDR']
```

سیستم عامل و مرورگر و ...

کد :

```
$_SERVER['HTTP_USER_AGENT']
```

برای دیدن بقیه این تکه کد رو اجرا کن :

کد :

```
foreach( $_SERVER as $i => $v)
{
    print( "$i => $v <br />");
}
```

بعد از اجرای این تکه کد، بسته به سیستم کاربر، به خروجی تقریباً این شکلی مشاهده می‌کنید:

کد :

```
COMSPEC => C:\WINDOWS\system32\cmd.exe
DOCUMENT_ROOT => c:/wamp/www
HTTP_ACCEPT => image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, application/vnd.ms-excel, application/
vnd.ms-powerpoint, application/msword, */*
HTTP_ACCEPT_ENCODING => gzip, deflate
```

```

HTTP_ACCEPT_LANGUAGE => fa
HTTP_CONNECTION => Keep-Alive
HTTP_COOKIE => phpbb2mysql_data=a%3A2%3A%7Bs%3A11%3A%22
autologinid%22%3Bs%3A0%3A%22%22%3Bs%3A6%3A%22
userid%22%3Bs%3A1%3A%22%22%3B%7D; bblastvisit=1206256268;
bblastactivity=0; PHPSESSID=5e17b7bde6a9d7782ff8781f2fdd173d
HTTP_HOST => 127.0.0.1
HTTP_UA_CPU => x86
HTTP_USER_AGENT => Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1;
.NET CLR 1.1.4322)
PATH => C:\Program Files\PC Connectivity
Solution\C:\Inprise\vbroker\bin;C:\Program Files\Borland\Delphi7\Bin;C:\Program
Files\Borland\Delphi7\Projects\Bpl\C:\WINDOWS\system32;C:\WINDOWS;
C:\WINDOWS\System32\Wbem
REMOTE_ADDR => 127.0.0.1
REMOTE_PORT => 2790
SCRIPT_FILENAME => c:/wamp/www/pathinfo1.php
SERVER_ADDR => 127.0.0.1
SERVER_ADMIN => webmaster@localhost
SERVER_NAME => 127.0.0.1
SERVER_PORT => 80
SERVER_SIGNATURE =>
Apache/1.3.33 Server at 127.0.0.1 Port 80
SERVER_SOFTWARE => Apache/1.3.33 (Win32) PHP/5.1.1
SystemRoot => C:\WINDOWS
WINDIR => C:\WINDOWS
GATEWAY_INTERFACE => CGI/1.1
SERVER_PROTOCOL => HTTP/1.1
REQUEST_METHOD => GET
QUERY_STRING =>
REQUEST_URI => /pathinfo1.php
SCRIPT_NAME => /pathinfo1.php
PATH_TRANSLATED => c:/wamp/www/pathinfo1.php
PHP_SELF => /pathinfo1.php
REQUEST_TIME => 1206631035
argv => Array
argc => 0

```

الگوریتم رمز کردن، قابل بازگشت:

```

base64_decode()
base64_encode()

```

الگوریتم رمز کردن، غیر قابل بازگشت:

```
md5
```

توضیح در مورد این توابع:

imagejpeg و **imagestring** و **imagedestroy**

imagestring مال نوشتن متن بر روی عکس است که ۶ آرگومان می‌گیرد که به صورت زیر است :

```

: کد
imagestring ( resource $image, int $font, int $x, int $y, string $string, int $color )

```

imagejpeg مال خروجی تصویر بر روی مرورگر یا پوشه است و سه تا آرگمان می‌گیرد به شکل زیر است :

کد :

```
imagejpeg ( resource $image, string $filename , int $quality )
```

اگر خاصی آن را بر روی پوشه‌ای ذخیره کنی آدرس آن را به **\$filename** وارد کن و **quality** که مال کیفیت تصویر است .

imagedestroy مال نابود ساختن متغیر ساخت تصویر است که این امر موجب خالی شدن رم می‌شود ...

کد :

```
imagedestroy ( resource $image )
```

دانلود متن و دانلود فایل:

هدری که برای دانلود متن استفاده میکنم اینه :

کد :

```
$FileName="myfile.txt";  
header('Content-Type: text/x-delimtext; name="'. $FileName .'"');  
header('Content-disposition: attachment; filename="'. $FileName .'"');  
  
print('Salam');
```

و هدری که برای دانلود فایل استفاده میکنم اینه :

کد :

```
$size = filesize($FilePath . $FileName) ;  
header("Content-Type: application/force-download; name=\"". $FileName ."\");  
header("Content-Transfer-Encoding: binary");  
header("Content-Length: ". $size .");  
header("Content-Disposition: attachment; filename=\"". $FileName ."\");  
header("Expires: 0");  
header("Cache-Control: no-cache, must-revalidate");  
header("Pragma: no-cache");  
  
echo (readfile($FilePath . $FileName));
```

ساختن جستجو و نمایش چند صفحه ای اطلاعات :

کد :

```
<?php  
  
if(isset($_GET['Page']))  
    $Start = $_GET['Page'];  
else
```

```

$Start = 0;

ShowData($Start);

function ShowData($Base = 0)
{
    $DbConn = mysql_connect("localhost", "root", "" ) or die(mysql_error());
    mysql_select_db("test",$DbConn) or die("Can Not Select DataBase Because: ".
mysql_error());

    $sql = "SELECT * FROM `salam` LIMIT ". intval($Base) .", 10";

    //Connect To DB...

    $ResultRs = mysql_query($sql) or die(mysql_error(). "<br />SQL: ". $sql);

    $TotalResults = mysql_num_rows($ResultRs);

    if($TotalResults <= 0)
        die("â ãÑ íÇÝÊ ä");

    $OutPut = "";

    for($i = 0; $i != $TotalResults; $i++)
    {
        //<!-- Start of Print Results...

        $Textname = mysql_result($ResultRs, $i, 0);
        $Textjob = mysql_result($ResultRs, $i, 1);
        $Textadres= mysql_result($ResultRs, $i, 2);
        $Texttelphon= mysql_result($ResultRs, $i, 3);

        $OutPut .=( $Textname . " - " . $Textjob . " - " . $Textadres . " - " . $Texttelphon . "<br
/><br /><br /><br />");
    }

    print($OutPut . "<p style='text-align:center'>" . ShowPageLinks($Base));
}

function ShowPageLinks($Base)
{
    $DbConn = mysql_connect("localhost", "root", "" ) or die(mysql_error());
    mysql_select_db("test",$DbConn) or die("Can Not Select DataBase Because: ".
mysql_error());
    $SQL = "SELECT COUNT(*) FROM `salam`";

    //Connect To DB...
    $ResultRs = mysql_query ($SQL) or die (mysql_error() . "<br />SQL: ". $SQL);
    $TotalResults = mysql_result($ResultRs,0);

    $TotalLinks = $TotalResults / 10;
    $TotalLinks = $TotalResults % 10 ? intval($TotalLinks) + 1 :
intval($TotalLinks);
}

```

```

$ThisPage = intval($Base / 10);
$ThisPage++;

$page_nav_inlk = "";
if($TotalLinks > 1)
{
    $page_nav_inlk .= 'Page Num: ';
    for($Index = 1 ; $Index <= $TotalLinks ; $Index++)
    {
        if($Index == $ThisPage)
        {
            $page_nav_inlk .= '<b>'. $Index . '</b> ';
            continue;
        }
        $BaseStart = ($Index - 1) * 10;
        $page_nav_inlk .= '<a href="?Page='. $BaseStart .'">'. $Index . '</a> ';
    }
}
return $page_nav_inlk;
}
?>

```

این هم نمونه کد دیگری برای چند صفحه ای کردن اطلاعات:

```

: کد
<?
$res = mysql_query("SELECT * from tbl_name WHERE id='fild name' ");
$numrows=mysql_num_rows($res);
$page=@$_GET['page'];
if(!$page)
$q = "SELECT * from tbl_name WHERE id='fild name' order by lid DESC limit
0,10";
else {
$from=($page-1)*10;
$q = "SELECT * from tbl_name WHERE id='fild name' order by lid DESC limit
$from,10 "; }
$result=mysql_query($q);
$num=mysql_num_rows($result);
?>
<? صفحه
$i=1;
$str=$num;
$tp=$str/10;
$tp=$tp+1;
while ($i<$tp)
{
    print " [<a href=\"page.php?page=$i\"> $i </a>] ";
    $i++;
}
?>

```

توجه : تمامی این مطالب و آموزش ها از انجمن های سایت www.RightClick.ir گرفته شده است ... (بعضی از این مطالب علاوه بر مدیر سایت، توسط اعضای سایت رایت کلیک نوشته شده است؛ که از تمامی عزیزان تشکر می نمایم)

نویسنده : مجتبی اسکندری

ایمیل : Mojtaba.Esk@Gmail.Com

جمع آوری اطلاعات : قاسم آقابابایی نبی

ایمیل : Ghasem.Babaie@Yahoo.Com

سایت : www.RightClick.Ir

فروردین سال ۱۳۸۷

پایان