

# آموزش php به زبان ساده

نویسنده: سید محمدرضا حسینیان راوندی

[www.barnamenevisi.blogspot.com](http://www.barnamenevisi.blogspot.com)

[ravand.city@gmail.com](mailto:ravand.city@gmail.com)

[ravand@iran.ir](mailto:ravand@iran.ir)

تاریخ : ۱۳۹۱/۱/۶

## مقدمه

با سلام و درود به روح پاک شهدای صدر اسلام بخصوص شهدای انقلاب اسلامی و شهدای هشت سال دفاع مقدس، در آغاز از همه ی دوستانی که در سایت [barnamenevis.org](http://barnamenevis.org) بخصوص آقای محمد مصطفی شهرکی و آقای مهدی یقویی که در یادگیری php و حتی نوشتن این کتاب کمک کردن تشکر می کنم.

## شروع

php یک زبان open source (منبع باز) هست که برای برنامه نویسی شما میتونید حتی توی notepad ویندوز هم کدهاش رو بنویسید فقط برای اجرای کدها، شما باید از برنامه ی خاصی استفاده کنید. و برای ثبت مطالب هم می تونید از mysql استفاده کنید. ویرایشگر هایی که برای php استفاده میشن زیاده ولی من از برنامه ی ++notepad استفاده میکنم. و برای اجرا کدهای php و ثبت مطالب در mysql از برنامه ی wamp استفاده میکنم. برای یادگیری این زبان باید حتماً html رو هم بلد باشید. چون html زبان پایه ی برنامه نویسی صفحات وب هست. از مزیت های زبان php سرعت و امنیت ، open source و موجود بودن کدهای این زبان در اینترنت می باشد. وقتی توی برنامه ی ++notepad مشغول کدنویسی هستید دکمه های ctrl و space رو با هم بگیرید با دستورات مختلف php برخورد میکنید. به این شکل کدنویسی برای شما راحت تر می شود. ولی قبل از این کار حتما در منوی language زبان php را انتخاب کنید.

کدهای php با پسوند php. ذخیره میشن. برای اجرای یک فایل php در برنامه ی wamp باید فایل مورد نظر رو داخل پوشه ی www ذخیره کنید. مثلاً نام فایل را test.php می گذارید. برای مشاهده ی این فایل به آدرس <http://localhost/test.php> بروید. تگ های شروع و پایان php به صورت زیر است:

1)

```
<?php
```

```
?>
```

2)

```
<?
```

```
?>
```

۳)

```
<script language="php">
```

```
</script>
```

برای اینکه به روش ۲، تگ رو استفاده کنیم باید در فایل php.ini مقدار short\_open\_tag رو on کنیم. روش ۳ هم به طور پیش فرض فعال است. برای نوشتن یک کد php که یک مقداری رو چاپ کند به روش زیر عمل کنید:

```
<?php
```

```
echo "salam php";
```

```
?>
```

این کد مقدار salam php رو چاپ میکند. این مقدار که جلوی دستور echo هست رو ما داخل دابل کوتیشن " قرار دادیم چون یک رشته (string) است.

می تونید این رشته را به یک متغیر هم نسبت بدید:

```
<?php
```

```
$ravandi="salam php";
```

```
echo $ravandi;
```

```
?>
```

```
out :salam php
```

مقدار رشته ی salam php در حافظه ی متغیر \$ravandi قرار میگیره. و با دستور echo چاپ میشه.

اگر عدد (integer) و در واقع عدد صحیح باشد نیازی به دابل کوتیشن نیست.

```
<?php
```

```
echo 3;
```

```
?>
```

```
<?php
```

```
$ravandi=3;
```

```
echo $ravandi;
```

```
?>
```

```
out: 3
```

در خروجی مقدار ۳ را چاپ میکند.

سمی کالن (یک ویرگول و نقطه روش) در پایان هر خطی که تمام شده باشه نوشته میشه. echo: این دستور برای چاپ یک مقدار بکار میره.

متغیر در php با \$ مشخص میشه.

**توضیحات:**

برخی مواقع پیش میاد که شما میخوای توضیحی رو در مورد کدی که نوشتی بدی برای این کار از روش های زیر استفاده می کنی.  
// برای دادن توضیحات یک خطی در کدهای php به کار میره.

```
/*توضیح  
چند خطی  
*/
```

روش دیگه :

```
# توضیح
```

روشی دیگر برای چاپ مقدار :

```
<?php  
print 3;  
?>
```

## فرق echo و print :

echo یک دستور است که از تایع print سریع تر است.  
echo فقط خروجی رو نمایش می ده و هیچ مقدار بازگشتی نداره ولی print در صورت استفاده یک مقدار بازگشتی داره و موقعیت خودش در اجرا و نمایش رو اعلام میکنه.  
برای متصل کردن دو رشته به هم ، هم میشه از . نقطه و هم ویرگول , استفاده کرد ولی print فقط از نقطه استفاده میکند.

```
<?php  
$ravandi="ravandi";  
$ravand="ravand";  
echo $ravandi.$ravand;  
echo "<br>";  
echo $ravandi,$ravand;  
?>  
ravandiravand  
ravandiravand
```

```
<?php  
$ravandi="ravandi";  
$ravand="ravand";  
print $ravandi.$ravand;  
?>  
ravandiravand
```

## فرق کوتیشن ' و دابل کوتیشن "

به خروجی این دو اسکرپت توجه کنید:

```
<?php  
$ravandi="mohammad reza";  
echo "$ravandi";  
?>  
out: mohammad reza
```

```
<?php  
$ravandi="mohammad reza";
```

```
echo '$ravandi';
?>
out : $ravandi
```

## استفاده از کدهای php در html :

برخی مواقع پیش میاد که شما نیاز دارید از کدهای html داخل کدهای php استفاده کنید. مثلاً برای رفتن يك متن به خط بعدي یا ایجاد جدول: رفتن متن به خط بعدي:

```
echo "<br>";
<br>
<?php
$site="barnamenevisi.blogspot.com";
$ravandi="mohammad reza";
echo $ravandi;
echo "<br>";
echo $site;
?>
out:
mohammad reza
barnamenevisi.blogspot.com
```

## ساخت جدول :

```
<?php
echo "<table border=2>";
echo "<tr>";
echo "<td>";
echo "matn";
echo "</td>";
echo "</tr>";
echo "</table>";
?>
```

یا حتی می تونید به جای دابل کوتیشن " از کوتیشن ' استفاده کنید. یا به شکل زیر می تونید بنویسید:

```
<?php
echo "<table border=2>
<tr>
<td>
matn
</td>
</tr>
</table>";
?>
```

یا حتی پیش میاد که نیاز باشه که کدهای html رو بدون نیاز به کارهای بالا بنویسید. می تونید وقتی به کدهای html رسید تگ php رو ببندید و بعد از کدهای html ، برای بقیه ی کدهای php تگ php رو باز کنید و در آخر ببندید.

## دستور \n

```
<?php
echo "World\n";
echo "k";
?>
World k خروجي :
```

```
World
k
```

وقتي سورسش رو ميگيريم اينطوري ميشه :

حالا بدون \n

```
<?php
echo "World";
echo "k";
?>
out : Worldk
```

```
Worldk
```

وقتي سورسش رو ميگيريم :

## دستور \r

```
<?php
echo "World\r";
echo "k";
?>
out : World k
```

```
Worldk
```

وقتي سورسش رو ميگيريم :

\n\r : هر دو با هم

```
<?php
echo "World\n\r";
echo "k";
?>
out : World k
```

```
source :
World
K
```

روش بعدي :

```
<?php
echo nl2br("matn1\r matn2");
?>
Out:
matn1
matn2
```

**ثابت ها :** بعد از بحث متغير ها مي پردازيم به بحث ثابت ها :

ثابت ها مفاديري هستند که در طول برنامه تبديلي در آن ها ايجاد نميشه و ثابت هستند.  
براي تعريف ثابت از define استفاده ميکنيم:

```
<?php
define('NAME', '3');
```

```
NAME = 60 + 20;
// Error
echo NAME;
?>
```

NAME که بیشتر با حروف بزرگ می نویسند نام ثابت هست و ۳ مقدار ثابت.  
الان اگه در کد بالا دقت کنید می بینید که به خط سوم اخطاز میده :  
Parse error: syntax error, unexpected '=' in F:\wamp2\wamp\www\test.php on line 3  
چون میخواد بگه که به یک ثابت همیشه مقداری رو داد. در صورتی که متغیر این مشکل رو نداره و همیشه بهش مقداری رو داد.

## تابع : strpos

```
<?php
echo strpos("Hello world!","world");
?>
```

عدد ۶ را بر می گرداند یعنی کلمه ی world از بعد از کاراکتر ششم متن Hello world! شروع می شود. فاصله هم یک کاراکتر است.

## تابع strlen : تعداد کاراکترها رو می شمارد:

```
<?php
echo strlen("Hello world!");
?>
12
```

با ایجاد یک فاصله دو متن کنار هم قرار میگیرن:

```
<?php
$txt1="Hello World!";
$txt2="What a nice day!";
echo $txt1 . " " . $txt2;
?>
Hello World! What a nice day!
```

حالا اگه فاصله ی مابین دو " رو حذف کنید فاصله در خروجی هم حذف می شود.

## عملگرهای ریاضی

عملگر	توضیحات	مثال	نتیجه
+	اضافه x=2	x+2	4
-	تفریق x=2	5-x	3

*	ضرب	$x=4$	$x*5$	20				
/	تقسیم	15/5	5/2	3	2.5			
%	ضرب (باقیمانده تقسیم)	5%2	10%8	10%2	1	2	0	
++	افزایش	$x=5$	$x++$	$x=6$				
--	کاهش	$x=5$						
x--		$x=4$						

۵ تقسیم بر دو همیشه دو و باقیمانده اش یک هست.  $5\%2=1 \Rightarrow$

## عملگرهای انتساب

عملگر	مثال	مانند
=	$x=y$	$x=y$
+=	$x+=y$	$x=x+y$
-=	$x-=y$	$x=x-y$
*=	$x*=y$	$x=x*y$
/=	$x/=y$	$x=x/y$
.=	$x.=y$	$x=x.y$
%=	$x%=y$	$x=x\%y$

## عملگرهای مقایسه

عملگر	توضیحات	مثال
==	برابر است با	غلط را بر میگرداند $5==8$
!=	برابر نیست با	درست را بر می گرداند $5!=8$
<>	برابر نیست با	درست را بر می گرداند. $5<>8$
>	بزرگتر از	غلط را بر میگرداند $5>8$
<	کوچکتر از	درست را بر می گرداند. $5<8$
>=	بزرگتر یا مساوی با	غلط را بر میگرداند $5>=8$
<=	کمتر از یا مساوی با	درست را بر می گرداند. $5<=8$

## عملگرهای منطقی

عملگر && : و

```
<?php
$x=14;
if($x > 10 && $x < 20){
echo "صحیح";
}else{
echo "نادرست";
}
?>
```

خروجی : صحیح  
توضیح : اگر متغیر \$x بزرگتر از ۱۰ و کوچکتر از ۲۰ باشد. چاپ کن صحیح وگرنه نادرست.

عملگر || : یا

```
<?php
$x=20;
if($x==10 || $x==20){
echo "صحیح";
}else{
echo "نادرست";
}
?>
```

خروجی : صحیح  
توضیح : اگر متغیر \$x مساوی ۱۰ باشد یا مساوی ۲۰ باشد صحیح را چاپ میکند.  
نکته: علامت == برای مقایسه بکار میره .

3+5

سه و ۵ عملوند و + عملگر است.

**تاریخ :**

```
<?php
echo date("Y-m-d");
?>
2012-02-14
```

```
<?php
echo date('h:i:s');
?>
01:26:55
```

```
<?php
echo time();
?>
```

این تابع زمان جاری را که بر اساس تعداد ثانیه‌ها از مبدأ تاریخ Unix که از تاریخ ۱ ژانویه ۱۹۷۰ ۰۰:۰۰:۰۰ GMT تا الان هست را چاپ میکند.  
به مثال زیر دقت کنید:

```
<?php
$t=time();
echo($t . "<br>");
echo(date("D F d Y",$t));
?>
out:
1329227242
2012-02-14
```

کد بالا در خروجی تاریخ امروز رو به ثانیه و تاریخ به صورت سال و ماه و روز در خروجی برای شما چاپ میکند.

شرط :

```
<html>
<body>
<?php
$d=date("D");
if ($d=="Fri")
echo "آخر هفته خوبی داشته باشید";
```



else

```
echo "یک روز خوبی داشته باشید";
?>
</body>
</html>
```

اول با تابع date و D که داخل آن هست تاریخ امروز رو دریافت میکند و در متغیر \$d قرار میدهد. و سپس در خط بعدی شرطی رو مطرح میکنه که آگه تاریخ امروز که داخل حافظه ی متغیر قرار گرفته مساوی جمعه باشه پیام "آخر هفته خوبی داشته باشید!" را چاپ کند **وگرنه** پیغام "یک روز خوبی داشته باشید!" را با echo چاپ کند.

```
<html>
<body>
<?php
$d=date("D");
if ($d=="Fri")
    echo "آخر هفته خوبی داشته باشید";
elseif ($d=="Sun")
    //وگرنه متغیر د ی مساوی یکشنبه باشد/
    echo "یک یکشنبه ی خوبی داشته باشید";
else
    echo "یک روز خوبی داشته باشید.";
?>
</body>
</html>
```

## نوع دیگری از شرطی:

switch

```
<html>
<body>
<?php
switch ($x)
{
case 1:
    echo "Number 1";
    break;
case 2:
    echo "Number 2";
    break;
case 3:
    echo "Number 3";
    break;
default:
    echo "هیچ کدام از اعداد بین یک تا سه نیست";
}
?>
</body>
</html>
```

به هر کدام از case ها يك مقدار داده میشه كه این مقادیر با مقدار داخل پرانتز یعنی \$x بررسی میشه كه آگه متغیر \$x با یکی از این مقادیر case برابر باشه پیغام مختص هر کدام با echo چاپ می شود. و آگه x\$ با هر کدام برابر نباشه با دستور break شکستی ایجاد میشه و به خط بعدی میره. وقتی مقادیر case

تمام همیشه با دستور default مقدار پیش فرض با دستور echo چاپ میشه . چون دیگه مقداری نیست که مقایسه بشه. برای مثال اگه به متغیر \$x مقدار ۲ بدهید پیغام Number 2 را چاپ میکند.

```
<html>
<body>
<?php
$x=2;
switch ($x)
{
case 1:
    echo "Number 1";
    break;
case 2:
    echo "Number 2";
    break;
case 3:
    echo "Number 3";
    break;
default:
    echo "هیچ کدام از اعداد بین يك تا سه نیست";
}
?>
</body>
</html>
```

Number 2

## آرایه :

روش اول :

```
$cars=array("tiba","peykan","BMW","Toyota");
```

مقادیری که داخل array هست داده هستن .

```
$ages = array("ravandi"=>27, "shahraki"=>30, "mehdi"=>16);
```

به هر داده کلیدی اختصاص داده شده.

روش دوم:

```
<?php
$cars[0]="tiba";
$cars[1]="Volvo";
$cars[2]="BMW";
$cars[3]="Toyota";
echo "$cars[0]";
?>
tiba
```

```
<?php
```

```
$days = array (1 => 'Sun', 'Mon', 'Tue','man');
echo $days[4];
?>
man
```

**حلقه** : مثل حلقه های زنجیر که به همدیگه متصل هستند.  
**حلقه ی while** :

```
<?php
$i=1;
while($i<=5)
{
    echo "The number is " . $i . "<br />";
    $i++;
}
?>
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
```

حلقه تا عدد پنج ادامه پیدا میکنه زمانی تمام میشه که شرط حلقه به اتمام برسه. اول متغیر \$i مساوی یک است بعد در خط بعدی در حالی که توی پرانتز متغیر \$i کوچکتر مساوی ۵ باشد حلقه ادامه پیدا میکند. تا ۵ رسید حلقه خاتمه میابد. \$i++ هم برای اضافه شدن عدد هست. حالا اگه \$i-- باشد کاهش میابد.

## حلقه ی do...while

در این نوع شرط ، اول کار انجام میشه بعد شرط بررسی میشه:

```
<?php
$i=1;
do
{
    $i++;
    echo "The number is " . $i . "<br />";
}
while ($i<=5);
?>
```

## حلقه ی for

```
<?php
for ($i=1; $i<=5; $i++)
{
    echo "The number is " . $i . "<br />";
}
```

```
}  
?>  
The number is 1  
The number is 2  
The number is 3  
The number is 4  
The number is 5
```

توضیحات : برای متغیر \$i مساوی ۱ و متغیر \$i کوچکتر و مساوی ۵ و متغیر \$i تا مقدار ۵ ادامه پیدا میکند.

## حلقه ی foreach :

```
<?php  
$x=array("one","two","three");  
foreach ($x as $value)  
{  
    echo $value . "<br />";  
}  
?>
```

Out :  
one  
two  
three

متغیر \$x مساوی داده های آرایه هست . با foreach متغیر \$x که شامل داده ها هست هر کدام یکی از داده ها را به عنوان مقدار جدا کرده و بعد با echo چاپ می شود.

می تونه به هر مقدار يك کلید هم بده :

```
<?php  
$arr = array("one", "two", "three");  
foreach ($arr as $key => $value) {  
    echo "Key: $key; Value: $value<br />\n";  
}  
?>
```

Out :  
Key: 0; Value: one  
Key: 1; Value: two  
Key: 2; Value: three

## ایجاد تابع : Function

تابع در زبان ساده کسی هست که تابع فرماتی هست. مثلا در برنامه نویسی تابع mail پیامی رو دریافت میکنه و وظیفه اش اینه که این پیام رو ارسال کنه.

تابع نوعی کتابخانه از عملیات خاص و تعریف شده میباشند که برای کاهش زمان کد نویسی و حجم کد برنامه و همچنین تسهیل در امر تحلیل کد برنامه مورد استفاده قرار میگیرند.  
دو نوع تابع داریم (۱) تابعی از قبل تعریف شده (۲) تابعی که ما میسازیم .

```
<?php  
function writeName()  
{
```

```
echo "ravandi";
}
echo "My name is ";
writeName();
?>
My name is ravandi
```

حالا این خط رو حذف کنید تا نتیجه رو ببینید: writeName();

```
<?php
function writeName()
{
echo "ravandi";
}
echo "My name is ";

?>
My name is
```

ممکن است در هر نقطه از صفحه بخواهید یک کار رو انجام بدید. شما می تونید در یک صفحه مقدراری رو چند بار با تابع فراخوانی کنید.

```
<?php
function writeName()
{
echo "محمدرضا";
}
echo " نام من ";
writeName();
echo "<br>";
echo " نام دوست من ";
writeName();
?>
نام من محمدرضا
نام دوست من محمدرضا
```

آرگومنت :

```
<?php
function writeName($fname)
{
echo $fname . " است.<br />";
}
echo " نام من ";
writeName("محمدرضا");
echo " نام دوستم ";
writeName("احمد");
echo " نام برادرم ";
writeName("حسین");
?>
نام من محمدرضا است.
نام دوستم احمد است.
نام برادرم حسین است.
```

توضیحات بیشتر در مورد تابع از سایت [ncis.ir](http://ncis.ir):

توی برنامه نویسی، گاهی اوقات شرایط طوری پیش میاد که میخوایم یکسری کارها رو چندبار انجام بدیم. حالا یا این کارها دقیقاً پشت سر هم انجام میشن (که میتونیم از حلقه ها برای تکرار دستورات استفاده کنیم) و یا اینکه در بخشهای مختلفی از برنامه که الزاماً پشت سر هم نیستن و یا حتی ممکنه در فایل‌های مختلفی از برنامه باشن، قرار دارن که در این حالت، میتونیم دستورات تکرارشونده رو درون یک تابع قرار بدیم و هر زمان بهش نیاز داشتیم، با کمک اسمش، اون رو فراخوانی کنیم تا دستوراتش اجرا بشن. توابع میتونن دارای مقدار بازگشتی باشن یا نباشن، یعنی اینکه بعد از اجرا، یک مقدار رو بعنوان نتیجه اجرای خودشون به بخشی از برنامه که اون تابع رو صدا زده، برگردونن یا نه. همچنین میشه موقع فراخوانی توابع، یکسری مقادیر رو هم براش ارسال کنیم که بهشون میگی وودیهای تابع. اینجا دو تعریف پیش میاد: آرگومان و پارامتر. آرگومان، مقداری هست که موقع صدا زدن تابع، براش ارسال میکنیم و پارامتر، اون چیزی هست که توی تعریف تابع، درکنار اسمش توی پرانتز نوشتیم و توی بدنه تابع از اون استفاده میکنیم

```
function test($x) {  
    $x *= 2;  
    echo $x;  
}
```

```
$y = 5;  
test($y);  
//Output: 10
```

توی کد فوق، \$x پارامتر و \$y آرگومان هست. بنابراین میبینید که آرگومان و پارامتر با هم متفاوت هستند. در حالت عادی، موقع ارسال آرگومان برای تابع، یک کپی از مقدار اون در پارامتر قرار میگیره و توی تابع استفاده میشه و طبیعتاً تغییرات پارامتر در درون تابع، ربطی به آرگومان که خارج از تابع هست، نداره. به این حالت میگوین استفاده مقداری یا ByVal که حالت رایج هست. اما در حالت ارجاعی یا ByRef، بجای اینکه یک کپی از آرگومان داخل پارامتر قرار بگیره، آدرس آرگومان در حافظه داخل پارامتر ذخیره میشه و اینطوری، آرگومان پارامتر هر دو به یک محل از حافظه اشاره میکنن و طبیعتاً تغییرات روی پارامتر در درون تابع، مقدار آرگومان رو در خارج از تابع هم تغییر خواهد داد. مثال:

```
function byval_func($x) {  
    $x *= 2;  
    echo $x;  
}
```

```
function byref_func(&$x) {  
    $x *= 2;  
    echo $x;  
}
```

```
$y = 5;  
  
byval_func($y);  
echo $y;  
// Output: 10 5
```

```
byref_func($y);  
echo $y;  
// Output: 10 10
```

نکته آخر اینکه میتونیم برای استفاده از توابع در فایل‌های مختلف، اونها رو داخل یک فایل قرار بدیم و هر موقع لازم داشتیم، اون فایل رو ضمیمه کنیم و از توابع داخلش استفاده کنیم. به چنین فایل‌هایی که چند تابع داخلشون هست، کتابخانه توابع میگوین و به توابع داخلشون، توابع کتابخانه ای گفته میشه.

<http://www.ncis.ir>

## ساخت يك فرم:

توي سايت ها با فرم هاي ثبت نام ، نظر سنجي ، نظر دهيد و غيره برخورد كرديد .

```
<html>
<body>
<form action="get.php" method="post">
Name: <input type="text" name="fname" />
<input type="submit" />
</form>
</body>
</html>
```

```
get.php
<html>
<body>
<?php echo $_POST["fname"]; ?>
</body>
</html>
```

در قسمت html فرم ، متني كه در تكست نوشته ميشه با استفاده از مقداري كه به name يعني fname داده شده به فايل php آن مقدار ارسال ميشه . روش دريافت هم با توجه به فايل html دستور post هست . كه ميشه به جاي اون از دستور GET هم استفاده كرد به شكل زير:

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

و در قسمت فايل php به اين صورت :

```
<?php echo $_GET["fname"]; ?>
```

روش get باعث ميشه بعد از ارسال مقدار در صفحه ي html در قسمت آدرس بار متن نوشته شده در تكست به نمايش در بيايد كه وقتي شما ميخواهيد براي وارد كردن پسورد استفاده كنيد از لحاظ امنيتي مشكل ساز بوده و فردي كه مانيتور شما رو نگاه ميكند پسورد شما رو هم ميتواند مشاهده كند. به آدرس زير دقت كنيد:

<http://localhost/8.php?fname=4>

از روش get استفاده شده است.

روش ديگه استفاده از دستور \$\_REQUEST هست :

با اين متغير از قبل تعيين شده ميشه هم به روش post و هم روش get مطالب رو ارسال كرد.

```
<?php echo $_REQUEST["fname"]; ?>
```

**include** : با اين دستور مي توانيم مطالب صفحه اي ديگر را در صفحه ي جاري اجرا كنيم.

يه وقت هايي **مثلاً** نياز ميشه كه مشخصاتي چون هاست و يوذر و پسورد و نام ديتابيس را در چند صفحه فرار بديم ولي با اين روش مي تونيم همه ي اين اطلاعات رو داخل يك فايل قرار داده و با دستور زير در هر فايلي اجرا كنيم. يا روش هاي ديگه .

```
<?php include("ravandi.php"); ?>
```

روش هاي ديگه براي اين كار :

```
<?php require("ravandi.php"); ?>
```

```
<?php include-once("ravandi.php"); ?>
```

include-once : اين روش يك بار فقط اجرا ميشه

## جابجايي فايل :

```
<?php
$file=fopen("test.txt","r");
echo fgets($file);
fclose($file);
?>
```

در خط اول فايل test.txt باز ميشه و با r مقاديروي كه داخل اين فايل هست خوانده ميشه و با دستور fgets مقادير داخل فايل كه در حافظه ي متغير \$file قرار گرفته ، گرفته ميشه و با كمك echo چاپ ميشه . در خط بعدي دستور fclose فايل رو مي بنده. خروجي ما مقاديروي است كه داخل فايل test.txt ذخيره شده است.

```
<?php
$file=fopen("test.txt","r");
echo fgetc($file);
fclose($file);
?>
```

fgetc براي خواندن اولين كاراكثر يك فايل به كار ميرهد. دقت كنيد كه فايل test.txt را با گزينه ي Encod in UTF-8 without BOM در ويرايشگر خود (notepad++) ذخيره كنيد. و همچين خود همين كدها رو.

r فقط خواندني است. يعني فقط مطالب داخل فايل را مي خواند و كار ديگه اي نمي كند.  
r+ خواندني و نوشتني است

```
<?php
$str = "Hello";
$file = fopen("test.txt","r+");
echo fprintf($file,$str);
?>
```

w فقط خواندني است و مطالب يك فايل را پاك ميكند و اگه فايل وجود نداشته باشد يك فايل جديد را ايجاد ميكند.

```
<?php
$str = "Hello";
$file = fopen("test.txt","w");
echo fprintf($file,$str);
?>
```

w+ خواندني و نوشتني است و مطالب يك فايل را پاك ميكند و اگه فايل وجود نداشته باشد يك فايل جديد را ايجاد ميكند

a اگر نوشته اي در فايل باشد نوشته ي جديد را به آخر نوشته ي قبلي اضافه ميكند و اگه فايل وجود نداشته باشد يك فايل جديد را ايجاد ميكند

a+ خواندني.. اگر نوشته اي در فايل باشد نوشته ي جديد را به آخر نوشته ي قبلي اضافه ميكند

x اگر فايل وجود داشته باشد اخطار ميدهد. ---ايجاد ميكند يك فايل جديد.---فقط نوشتني

x+

file\_exists

اين دستور بررسي ميكند بين فايل وجود داره يا نه؟

```
<?php
```



```

if(!file_exists("test.txt"))
{
echo("فایل موجود نیست");
}else{
echo "فایل موجود است";
}
?>

```

## ساخت يك فرم آپلود فایل:

```

<html>
<body>
<form action="8.php" method="post"
enctype="multipart/form-data">
<label for="file">Filename:</label>
<input type="file" name="file" id="file" />
<br />
<input type="submit" name="submit" value="Submit" />
</form>
</body>
</html>

```

```

<?php
//شرطي ميذاريم كه نوع فایل موارد مشخص شده هست
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/pjpeg")))
//حجم فایل دو مگابایت باشد
&& ($_FILES["file"]["size"] < 20000))
{
//کد خطا ناشی از آپلود فایل
if ($_FILES["file"]["error"] > 0)
{
//پیغام بده
echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
}
//وگرنه
else
{
//این چهار مورد رو چاپ کن
//نام فایل آپلود شده را چاپ کن
echo "Upload: " . $_FILES["file"]["name"] . "<br />";
//نوع فایل آپلود شده رو چاپ کن
echo "Type: " . $_FILES["file"]["type"] . "<br />";
//اندازه ي فایل آپلود شده رو تقسیم بر ۱۰۲۴ که همیشه دو مگابایت رو چاپ کن
echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
//محل موقت فایل آپلود شده رو چاپ کن
echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";
//اگر نام فایل وجود داشته باشد یعنی تکراری باشد

```

```

if (file_exists("upload/" . $_FILES["file"]["name"]))
{
    // در این صورت این پیغام رو چاپ کن
    echo $_FILES["file"]["name"] . " already exists. ";
}
//وگرنه
else
{
    //حرکت بده فایل رو از پوشه ی موقت
    move_uploaded_file($_FILES["file"]["tmp_name"],
    //به پوشه ی آپلود
    "upload/" . $_FILES["file"]["name"]);
    //وسپس آدرس فایل آپلود شده رو با پیغامی چاپ کن
    echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
}
}
//وگرنه
else
{
    //پیغام بده فایل نامعتبراست.
    echo "Invalid file";
}
?>

```

۲۰۰۰۰ = بیست هزار کیلو بایت معدل ۲ مگابایت.  
 Upload : پوشه ی آپلود فایل است. و باید پوشه ای در هاست با این نام بسازیم.

**Cookie :** اطلاعاتی است که یک وب سایت روی کامپیوتر شما قرار می دهد تا بتواند در زمان های بعدی اطلاعاتی در مورد شما را به خاطر بیاورد.

```

<?php
setcookie("user", "meghdar", time()+3600);
?>

```

ما در کد بالا يك كوكي به نام USER و مقدار meghdar مشخص کردیم که طبق زمان ۳۶۰۰ ثانیه (معادل ۶۰ دقیقه) از بین می رود.

برای دیدن کوكي ایجاد شده در ویندوز ایکس پی و Internet Explorer به آدرس زیر بروید:  
 C:\Documents and Settings\khaki\Local Settings\Temporary Internet Files

نام فایل :

user@localhost[1]

user همان نام یوزر سیستم شماست.  
 وقتی فایل رو باز میکنید به همچین مطلبي نوشته شده:

```

user
meghdar
localhost/
۱2۳۶
۲۳۹۰۶۶3۳۸2
۳۰۲۰3۳2۷
۷۵۳۷3۶۶3۶
۳۰۲۹۵9۳۹
*
```

اگر در این آدرس: C:\Documents and Settings\khaki\Local Settings\Temporary Internet Files: دقت کنید می بینید تاریخ آخرین ویرایش یا آخرین بازدید رو مثلاً ۴:۴۹ دقیقه و expires یا تاریخ انقضا رو ۵:۴۹ دقیقه نوشته هست طبق time()+3600 که تاریخ انقضاء رو يك ساعت بعد تعیین کرده است.

```
<?php
$expire=time()+60*60*24*30;
setcookie("user", "meghdar", $expire);
?>
```

تاریخ انقضای کوکی بالا : سي روز و ۲۴ ساعت و ۶۰ دقیقه و ۶۰ ثانیه.

```
<?php
if (isset($_COOKIE["user"]))
    echo "Welcome " . $_COOKIE["user"] . "!<br />";
else
    echo "Welcome guest!<br />";
?>
```

## طریقه ي پاك کردن کوکی:

ما آمديم با این مثال يك کوکی ایجاد کردیم :

```
<?php
setcookie("user", "meghdar", time()+3600);
?>
```

حالا همین کوکی رو به روش زیر پاك میکنیم:

```
<?php
setcookie("user", "", time()-3600);
?>
```

اگر مرورگر شما از کوکی پشتیبانی نکند می تونید برای انتقال اطلاعات از فرم استفاده کنید.

## توضیحات بیشتر در مورد کوکی:

بطور کلی وظیفه کوکی ذخیره یکسری اطلاعات روی کامپیوتر کلاینت برای استفاده های بعدی هست. مثلاً آگه دقت کرده باشین، اکثر سایتها (مثل همین سایت برنامه نویس) موقع لاگین گزینه ای بنام Remember me یا «مرا بخاطر بسپار» و... دارن که آگه این گزینه رو انتخاب کنید، برحسب تنظیمات اون سایت، تا زمانی که مشخص شده، به محض وارد کردن آدرس سایت، وارد حساب کاربری خودتون میشین و فرم لاگین ظاهر نمیشه. در اینجور موارد، از کوکی استفاده شده و یک فایل حاوی نام کاربری و رمز عبورتون (حالا یا بصورت متن خام و یا بصورت کدشده) روی سیستم شما ذخیره میشه و در مراجعات بعدی، اول چک میکنه بینه اون فایل هست یا نه و آگه بود، اطلاعات ورود به سایت رو از توی اون میخونه و وارد میشه و دیگه از شما نام کاربری و... رو نمیخواد. وقتی هم که عمر اون کوکی به سر رسید، خودبخود حذف میشه (یا اینکه خودتون میتونید حذفش کنید) و اونوقت، چون فایل موجود نیست، دوباره فرم لاگین ظاهر میشه. در واقع برای استفاده از کوکی باید برنامه نویسی کنید .

سایت: <http://www.ncis.ir>

این مثال عملی رو دقت کنید:

```
<form action="8.php" method="post">
<input type="text" name="user"><br>
<input type="password" name="pass"><br>
<input name="remember" type="checkbox" id="remember" value="1"><br>
<input type="submit" value="submit">
</form>
```

```
<?php
// اگر یوزر و پسورد صحیح باشد
```

```

if(strtolower($_POST['user']) == 'ali' && $_POST['pass'] == '1234') {
$username=$_POST['user'];
$password=$_POST['pass'];
// اگر چک باکس انتخاب شود.
if(isset($_POST['remember']) == 'true') {
// کوکی ساخته می شود.
setcookie("user", "$username", time()+120);
setcookie("pass", "$password", time()+120);
}
// و به کاربر پیام خوش آمدگویی داده میشود.
echo "welcom $username";
// وگرنه
}else{
echo "نام کاربری و پسورد اشتباه است";
}
?>

```

الان توی فرم بعد از نوشتن صحیح یوزر و پسورد تیک چک باکس Remember me رو نمی زنی ، بعد دکمه ی submit رو می زنی بعد میرم توی این آدرس :  
C:\Documents and Settings\khaki\Local Settings\Temporary Internet Files  
می بینم هیچ کوکی ایجاد نشده .  
برای بار بعد این چک باکس رو میزنی و میرم می بینم که کوکی ایجاد شده.

## Session

شما در يك صفحه که کار میکنید و بعد به صفحه ی دیگر می روید سیشن شما رو تشخیص میدهد .  
سیشن اطلاعات کاربر رو وقتی که در چند صفحه هست در خود میتواند نگه داری کند.

سایت : <http://www.iran-eng.com>

**به صورت خلاصه اگر بخوام اشاره کنم همان کوکی ها هستند اما ذخیره شده در سرور**  
نحوه کار هم به این شکل هست که وقتی فردی برای اولین بار از یک سایت دیدن میکنه PHP یک جلسه با یک شناسه منحصر به فرد به صورت هیش شده در سرور ایجاد میکنه.  
و اطلاعات مورد نیاز برای مراحل بعدی رو در داخل همین جلسه نگه داری میکنه  
این شناسه ایجاد شده توسط یک کوکی به نام PHPSESSID به مرورگر فرستاده میشه و در سیستم مشتری ذخیره میشه و در بار دوم که فرد از سایت دیدن میکنه کوکی به سرور ارسال میشه و اطلاعات در جلسه خوانده میشه  
کاربرد هم خیلی زیاد هست  
یک نمونه بارز اینجاست که وقتی شما در صفحه Login وارد میشد تا آخرین لحظه که مرورگر باز هست شما رو در تمامی صفحات می شناسه و در هر Page نیاز به ورود مجدد نیست.  
منبع :

<http://www.iran-eng.com/archive/index.php/t-237033.html?s=1de17755372d6eb337da83e642b4300f>

شما وقتی لوگین می کنی اسم شما می ره تو یه متغیر تو سیشن ، خوب حالا شما هر صفحه که از صفحه فعلی (پدر) باز کنی (open a new window) متغیر سیشن در اونا هم (یعنی فرزندان) قابل دسترسی و اسم شما در بالای صفحه نشان داده می شه.

منبع :

<http://www.astovan.com/>

وقتی شما با یک برنامه کاربردی کار می کنید، آنرا را باز می کنید، یکسری تغییرات انجام داده و سپس آنرا می بندید. به این فرآیند یک session یا جلسه کاری می گویند. کامپیوتر می داند که شما کی هستید. همچنین می داند که چه وقت برنامه را شروع کرده و چه وقت پایان داده اید. ولی در اینترنت یک مشکل

وجود دارد: وب سرور نمی داند شما کی هستید و چه می خواهید انجام دهید. چون آدرس های HTTP حال را نگه نمی دارند ، در واقع پروتکل HTTP بصورت state-less عمل می کند. session ها در PHP مشکل را حل می کنند و به شما اجازه می دهند تا اطلاعاتی را برای استفاده های بعدی ذخیره کنید (اطلاعاتی مثل نام کاربری، اقلام خریداری شده و غیره) . با این حال اطلاعات session موقتی هستند و با خارج شدن شما از وب سایت از بین خواهند رفت. اگر می خواهید اطلاعات به صورت دائمی باقی بمانند باید آنها را در بانک اطلاعاتی ذخیره کنید. session ها با ایجاد یک شناسه یکتا (UID) برای هر مراجعه کننده و ذخیره متغیرها بر اساس UID کار می کند. UID خودش در یک کوکی ذخیره شده و یا با URL جایجا می شود.  
منبع : [http://www.astovan.com/public/dcontent.php?content\\_id=65&cat=9](http://www.astovan.com/public/dcontent.php?content_id=65&cat=9)

در مثال زیر یک شمارنده تعداد مراجعات به صفحه ساده ایجاد کرده ایم. تابع isset () بررسی می کند که متغیر "views" از قبل وجود دارد و انتخاب شده است. اگر "views" از قبل وجود داشته باشد، آنرا یک واحد اضافه کرده و گرنه یک متغیر "views" ایجاد کرده و آنرا مقدار ۱ می دهد.

```
<?php
session_start();
$_SESSION['views']=1;
echo "Pageviews=". $_SESSION['views'];
?>
```

```
<?php
session_start();
if(isset($_SESSION[views]))
$_SESSION[views]=$_SESSION[views]+1;
else
$_SESSION[views]=1;
echo "Views=". $_SESSION[views];
?>
```

منبع اسکریپت : <http://www.w3schools.com>

مثال عملی برای سیژن :

```
index.php
<form action="8.php" method="post" >
<input type="text" name="user"><br>
<input type="password" name="pass"><br>
<input type="submit" value="submit">
</form>
```

```
8.php
<?php
session_start();
$username=$_POST['user'];
$password=$_POST['pass'];
```

```

$_SESSION['user']="$username";
$_SESSION['pass']="$password";
if((($_SESSION['user']) == 'ali' && $_SESSION['pass'] == '1234') {
echo "welcom $username";
}else{
echo "نام کاربري و پسورد اشتباه است";
}
?>

```

این هم صفحه ای دیگر در صورتی که کاربر قبلاً لوگین کرده باشد بهش پیغام خوش آمدگویی میدهد ولی در صورتی که لوگین نکرده باشه و وارد این صفحه بشه نمیتونه وارد بشه و از صفحه خارج میشه.

```

<?php
session_start();
$user=$_SESSION['user'];
if((($_SESSION['user']) == 'ali' && $_SESSION['pass'] == '1234')
{
echo "welcom $user";
}else{
header("location: index.php");
exit;
}
?>

```

یک اشکال کد بالا این بود که پسورد در سیژن ذخیره شده است. و باید از دیتابیس پسورد رو بگیرد آن هم به صورت رمزگذاری شده که در میحث md5 به آن می پردازیم. این دو روش برای غیرفعال کردن سیژن که همیشه برای دکمه ی خروج ارزش استفاده کرد:

```
unset($_SESSION['user']);
```

```

<?php
include_once("8.php");
session_destroy();
header("location: index.php");
?>

```

## ارسال ایمیل :

```

<?php
//ایمیل فرد دریافت کننده ی ایمیل
$to = "name@name.com";
$subject = "موضوع ایمیل";
$message = "متن پیام ایمیل";
//ایمیل فرد ارسال کننده
$from = "name2@name.com";
mail($to,$subject,$message,$from);
echo "ایمیل ارسال شد.";
?>

```

نکته : این موضوع رو به خاطر داشته باشید که توی لوکال هاست برای ارسال ایمیل شما باید با برنامه ی خاصی نصب داشته باشید یا از تنظیمات جیمیل استفاده کنید وگرنه برای ارسال ایمیل باید از یک هاستی استفاده کنید که این امکان رو داشته باشه.

send.php

```

<form method="post" action="8.php">
from: <input name="from" type="text" /><br/>
Email: <input name="to" type="text" /><br/>
Subject: <input name="subject" type="text" /><br/>
Message: <br/><textarea name="message" rows="5" cols="25">
</textarea><br />
<input type="submit" value="ارسال" />
</form>

```

```

8.php
<?php
//ایمیل فرد ارسال کننده
$from = $_POST['from'];
//ایمیل فرد دریافت کننده ی ایمیل
$to = $_POST['to'];
// موضوع
$subject = $_POST['subject'];
// متن پیام
$message = $_POST['message'];
mail($from,$to,$subject,$message);
echo "ایمیل ارسال شد.";
?>

```

می تونید برای کدهای بالا اعتبارسنجی هم بذارید که مثلاً اگر نکست ایمیل یا متن پیام به هر کدوم خالی باشه اخطار بده. یا اگه ایمیل رو درست وارد نکرده باشید پیام بده.

## :MySQL

یک بانک اطلاعاتی هست که ما برای ذخیره ی اطلاعاتمون ازش استفاده میکنیم. در بحث های قبلی اشاره ای به این موضوع کردیم. برای ذخیره ی اطلاعاتمون در اول این کتاب از فایل استفاده میکردیم به طوری که مطالبی رو به یک فایل با پسوند txt ارسال می کردیم و در آن ذخیره می کردیم. ولی الان میخوایم اطلاعاتمون رو داخل بانک اطلاعاتی mysql ذخیره کنیم. به غیر از متن، عکس را هم میشه در mysql ذخیره کرد ولی بهتره عکس رو داخل پوشه ذخیره کنیم و آدرس عکس را داخل بانک اطلاعاتی mysql. چون اگه عکس در دیتابیس ذخیره بشه هم حجم اضافی و سنگینی دیتابیس است و هم بار اضافی روی سرور. من در سیستمم برای اجرای فایل های php و استفاده از بانک اطلاعاتی mysql از برنامه ی wamp استفاده میکنم که برنامه ی خیلی خوبی هست. بعد از نصب این برنامه شما به قسمت taskbar رفته و روی آیکن آن که به شکل یک گوشی تلفن هست کلیک کنید. با انتخاب گزینه ی www directory می تونید به پوشه ی www که برای قرار دادن فایل php، برای اجرا هست بروید. و با کلیک روی گزینه ی phpMyAdmin می تونید به بانک اطلاعاتی خود دسترسی داشته باشید تا به ویرایش و ایجاد بانک اطلاعاتی و جدول و فیلد بپردازید. یا به آدرس <http://localhost/phpmyadmin/> بروید.

تعریفی ساده از فیلد و جدول و دیتابیس (بانک اطلاعاتی) فیلد هر مقداری مثل نام و نام خانوادگی، ایمیل یا هر مقدار دلخواه که به بانک اطلاعاتی ارسال میشود را فیلد میگن و جدول هم حاوی چند تا فیلد هست. و دیتابیس هم حاوی چند جدول هست. رکورد هم مطالبی هست که داخل فیلد ثبت می شود.

نمونه ي رکورد و فيلد:

نام شهر	شغل	نام خانوادگي	نام	فيلدها رکوردها
راوند	طراح سايت	حسينيان راوندي	سيد محمدرضا	۱
کاشان	الکترونيک	مشرقي	مهدي	۲

براي ساخت ديتابيس و جدول و فيلد دو روش رو اينجا بررسي ميکنيم .  
(۱) از طريق کدنويسي:

```
<?php
// اين خط اتصالي رو به لوکال هاست و يوزر و پسورد ايجاد ميکند و داخل اين متغير قرار ميدهد.
$con = mysql_connect("localhost","root","");
// يک شرط مي داره بينه مقادير داده شده به متغير موجود هست يا نه؟
if (!$con)
{
// در صورت موجود نبودن يا در واقع اشتباه بود اخطار ميده
die("Could not connect: ' . mysql_error());
}
// پيامي ارسال ميشه که ديتابيس ايجاد شده يا نه؟
if (mysql_query("CREATE DATABASE testi",$con))
{
echo " ديتابيس ايجاد شد.";
}
else
{
echo " خطايي رخ داده " . mysql_error();
}
// ديتابيس انتخاب ميشه
mysql_select_db("testi", $con);
// ساخت جدول
$sql = "CREATE TABLE Persons
(
// ساخت فيلدها
FirstName varchar(15),
LastName varchar(15),
Age int
)";
// پيام براي بررسي ايجاد جدول و فيلدها ارسال ميشه
mysql_query($sql,$con);
// براي بستن ديتابيس
mysql_close($con);
?>
```

mysql\_error() : براي بررسي خطا مي تونيد از اين کد استفاده کنيد.

```
mysql_connect("localhost","root","");
```



این خط اتصالی رو به لوکال هاست (هاست محلی برقرار میکنه) شما اگه روی سایت بخواهید این کد رو امتحان کنید باید از هاست و نام کاربری و رمز مختص سایت خودتون استفاده کنید. ما در اینجا root رو به عنوان نام کاربری پیش فرض هاست محلی ازش استفاده میکنیم. و پسورد رو چون به عنوان پیش فرض برایش در نظر نگرفتیم نمیداریم شما اگه خواستید در سایتتون از این کدها استفاده کنید باید نام کاربری و پسورد مختص خود سایت رو بکار ببرید. varchar هم نوع فیلد و عدد داخل پرانتز طول آن هست. برای سازگاری با utf-8 و زبان فارسی به روش زیر عمل کنید:

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
// Create database
if (mysql_query("CREATE DATABASE `testi` DEFAULT CHARACTER SET utf8 COLLATE
utf8_persian_ci",$con))
{
    echo "Database created";
}
else
{
    echo "Error creating database: " . mysql_error();
}
// Create table
mysql_select_db("testi", $con);
$sql = "CREATE TABLE Persons
(
`FirstName` VARCHAR( 21 ) CHARACTER SET utf8 COLLATE utf8_persian_ci NOT NULL ,
`LastName` VARCHAR( 21 ) CHARACTER SET utf8 COLLATE utf8_persian_ci NOT NULL
)ENGINE = INNODB CHARACTER SET utf8 COLLATE utf8_persian_ci";
mysql_query($sql,$con);
mysql_close($con);
?>
```

روش دیگر برای ساخت دیتابیس و جدول و فیلد :  
به این آدرس بروید <http://localhost/phpmyadmin>

در قسمت MySQL connection collation گزینه ی utf8\_persian\_ci را انتخاب کنید.

و در قسمت Create new database نام بانک اطلاعاتی (database) را نوشته . و جلوی آن گزینه ی utf8\_persian\_ci را انتخاب کنید این گزینه همانطور که گفتیم برای سازگاری با زبان فارسی است. حال دکمه ی Create رو کلیک کنید. مثلاً نام دیتابیس را testi می گذاریم. حالا در صفحه ی بعد در قسمت Create new table on database testi و در قسمت Name نام جدول را مثلاً Persons انتخاب کرده و برای اینکه میخواهیم سه فیلد بسازیم در قسمت Number of fields: عدد ۳ را وارد کنید. دکمه ی Go رو کلیک کنید. در اولین تکست میخواهیم یک فیلد بسازیم به نام id ، پس id را در قسمت Field وارد کرده و نوع آن یا Type آن را INT است . را انتخاب کنید. این فیلد رو روی گزینه ی primary و Extra را روی گزینه ی auto\_increment قرار دهید . تا فیلد اصلی ما باشد و با ثبت هر رکورد در فیلد های جدول یک شماره رو ثبت کند. در قسمت Length/Values یعنی طول و مقدار فیلد نیازی نیست چیزی بنویسید. می تونید Collation این فیلد و بقیه ی فیلد ها رو خالی بذارید و Collation اصلی که در پایین همه ی فیلدهاست را روی utf8\_persian\_ci قرار بدید. اینطوری به طور خودکار همه ی فیلد ها با utf8\_persian\_ci سازگار میشه. و به طور خودکار انتخاب میشه. و در قسمت Null هم برای همه ی فیلد ها گزینه ی not null انتخاب شود. دو فیلد بعدی را هم بسازید با نام های FirstName و LastName هر دو رو از نوع varchar و Length/Values1 مثلاً ۲۱ .

نکته : فقط ۲۱ کاراکتر توی این فیلد همیشه قرار داد.  
و سپس دکمه ی save را کلیک کنید . جدول ساخته میشه.  
نکته : در قسمت Field در نوشتن حروف بزرگ و کوچک دقت کنید.  
با این پیغام روبرو می شوید:

SQL query:

```
CREATE TABLE `testi`.`Persons` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `FirstName` VARCHAR( 21 ) NOT NULL ,  
  `LastName` VARCHAR( 21 ) NOT NULL ,  
  PRIMARY KEY ( `id` ) ) ENGINE = InnoDB CHARACTER SET utf8 COLLATE utf8_persian_ci
```

نوع varchar را برای اینکه خواستیم متن وارد فیلد شود انتخاب کردیم و int را برای عدد انتخاب کردیم.  
اگر متنی را به فیلد های بخواهید وارد کنید می تونید روی گزینه ی درج یا insert کلیک کنید برای مشاهده  
ی مطلب روی گزینه ی Browse کلیک کنید. و برای خروجی گرفتن و در واقع پشتیبان گرفتن از دیتابیس و  
جدول و فیلد ها گزینه ی Export را بزنید و برای وارد کردن پشتیبان از گزینه ی import استفاده کنید. و  
برای پاک کردن دیتابیس یا جدول گزینه ی Drop و برای پاک کردن خود فیلد ، آن را انتخاب و روی گزینه ی  
ضربدر کلیک کنید.  
گزینه ی sql برای این است که وقتی کدی رو مثل کد بالا آماده داشتید به راحتی می تونید در این قسمت  
وارد کنید . و دیگه نیازی به این همه کار که برای ساخت دیتابیس و جدول و فیلد گفتم ندارید.  
روش دیگه هم استفاده از mysql تحت داس یا همون command prompt جدید خودمون هست.  
کافیه بر روی آیکن که شبیه تلفن هست در قسمت تاسک بار کلیک کنید و گزینه ی MySQL رو انتخاب کرد  
و سپس گزینه ی mysql consol رو کلیک کنید اگه در هنگام نصب برنامه پسورد نداده باشید Enter کنید  
این قسمت دیگه خیلی کم ارزش استفاده میشه .  
نکته : برای فیلدتان مقداری را وارد کنید که اندازه ی کاراکترهایی باشد که در آینده قرار است در آن وارد  
کنید. اگر تعداد کاراکتر های وارد شده به فیلد زیادتر از مقدار در نظر گرفته شده برای فیلد باشد مقادیر  
اضافه دور ریخته می شود.  
حالا میخوایم یک فرم بسازیم که اطلاعاتی چون نام و نام خانوادگی رو به دیتابیس ارسال کند:

```
<form action="8.php" method="POST">  
نام<input type="text" name="FirstName"><br>  
نام خانوادگی<input type="text" name="LastName"><br>  
<input type="submit" value="ارسال">  
</form>
```

```
<?php
```

```
/* با این خط از دستور به سرور محلی متصل می شوید*/
```

```
کارهای انجام شده در حافظه ی متغیر قرار میگیرد
```

```
*/
```

```
$handle=mysql_connect('localhost','root','');
```

```
/* با این خط شرطی برقرار میشه که انجام مراحل بالا درست است یا غلط و در واقع ارتباط برقرار شده یا نه؟
```

```
*/
```

```

if($handle==false){
//اگر نادرست باشد پیغام خطا میدهد
die("\r\n");
}
//بانك اطلاعاتي انتخاب میگردد
$db=mysql_select_db('testi');
//شرطي میذاریم که اگر به دیتابیس متصل نشد اخطار بده
if($db==false){
die("\r\n");
}
/*
مقداري که از فرم ارسال توسط کاربر ارسال شده در حافظه ي متغییر قرار میگیرد
اینجا مقدار نام
*/
$FirstName=$_POST['FirstName'];
/*
مقداري که از فرم ارسال توسط کاربر ارسال شده در حافظه ي متغییر قرار میگیرد.
اینجا مقدار نام خانوادگی
*/
$LastName=$_POST['LastName'];
//برای سازگاری به زبان فارسی
mysql_set_charset('utf8',$handle);
//پیام یا يك کوئري برای وارد کردن مقادیر دریافت شده از فرم به جدول ارسال می شود.
mysql_query("INSERT INTO Persons (`FirstName`,`LastName`) VALUES
('$FirstName','$LastName')")or die(mysql_error());
$num=mysql_affected_rows($handle);
if($num==1){
echo "ارسال شد.";
}
else
if($num == 0){
echo "ارسال نشد.";
}
// برای بستن اتصال به mysql
mysql_close($handle);
?>

```

این خط در واقع به شما میگوید که وارد کن به جدول Persons مقادیر FirstName و LastName را با توجه به مقادیر داده شده به دو متغییر داخل پرانتز یعنی \$FirstName و \$LastName و آنگاه مشکلی پیش آمد اخطار بده.

دستور `mysql_affected_rows` تعداد سطرهایی را که توسط آخرین دستور `insert` تحت تاثیر قرار گرفته را بر می گرداند.

حالا میخوایم مقادیری که به دیتابیس ارسال کردیم چاپ کنیم.

```
<?php
$connection=mysql_connect("localhost","root","") or die (mysql_error());
$db=mysql_select_db("testi",$connection) or die (mysql_error());
//انتخاب همه ي فيلد هاي جدول/
$sql="SELECT * FROM Persons";
mysql_query("SET CHARACTER SET utf8",$connection);
//ارسال پیام برای جدول و دیتابیس برای دریافت مطالب/
$result=mysql_query($sql,$connection) or die (mysql_error());
while ($row=mysql_fetch_array($result)){
//شماره ي هر رکورد ثبت شده رو چاپ مي کند/
echo $row['id'];
//فيلد بعدي رو براي چاپ به خط بعدي مي برد/
echo "<br>";
echo $row['FirstName'];
echo "<br>";
echo $row['LastName'];
echo "<br>";
}
mysql_close($connection);
?>
```

`mysql_fetch_array`: یک مجموعه جواب بازگشتی را در آرایه ای انجمنی یا اندیسی یا هر دو می ریزد.

**در این خط اگر \* ستاره باشد همه ي فيلد هاي جدول انتخاب مي شود.**

```
SELECT id,FirstName,LastName FROM Persons
```

خلاصه ي دستور بالا:

```
SELECT * FROM Persons
```

برای ایجاد محدودیت در چاپ فیلدها:

مثلا ما می خواهیم در خروجی از ۲۰ رکورد ثبت شده ۵ تا رکورد چاپ بشه از این روش استفاده میکنیم:

```
SELECT * FROM jadval limit 5
```

برای اینکه هر مطلبی بر اساس id چاپ بشه:

```
SELECT * FROM Persons ORDER BY id
```

بر اساس id يك رکورد رو حذف میکند:  
`DELETE FROM Persons WHERE id='1'`

```
<?php
$handle = mysql_connect("localhost","root","")or die (mysql_error());
mysql_select_db("testi", $handle)or die (mysql_error());
```

```

$result=mysql_query("DELETE FROM Persons WHERE id='1'");
if($result==1){
echo " پاك شد ";
}else{
echo" پاك نشد";
}
mysql_close($handle);
?>

```

با رفرش صفحه ي بالا رکورد يك حذف مي شود.

```

if($result==1)

```

در خط بالا از دو مساوي استفاده شده که در واقع يك نوع مقايسه ايجاد ميشه . اگر پاك کردن به درستي انجام بشه در متغير \$result مقدار يك يا true که از مقادير boolean هست ذخيره ميشه . و با خط بالا مقدار يك که داخل متغير ذخيره شده با مقدار يك جلوي دو مساوي مقايسه ميشه . در صورت يك بودن يا true بودن پيغام " تغيير يافت" چاپ ميشه .

نکته : boolean فقط دارای دو مقدار true يا false هست. در بولين مقدار غير صفر به عنوان درست و مقدار صفر به عنوان نادرست استفاده می شود.

حالا اگر به صورت زیر متغير \$result را جاي پيغام قرار بديد عدد يك چاپ ميکنه:

```

if($result==1){
echo $result;
}else{
echo $result;
}

```

اگر SET CHARACTER SET utf8 در کدهاي ما نباشد چون متوني که شايد ارسال ميکنم فارسي هست يا مطالب ما به صورت علامت سوال چاپ ميشه يا بهم ريخته. به مثال هاي زیر توجه کنيد:

## مشکل علامت سوال در قسمت چاپ :

```

$servername="localhost";
$dbname="cms2";
$user="root";
$pass="";
$dblink=mysql_connect($servername,$user,$pass);
mysql_select_db($dbname,$dblink);
mysql_query("SET CHARACTER SET utf8",$dblink);
$dbresult=mysql_query("SELECT * FROM jadval",$dblink);

```

بايد هميشه اين خط آبي قبل از ارسال کوئري باشه اگه بعد از **قسمت کوئري** نوشته بشه يا اصلا نباشه علامت سوال در صفحه چاپ ميشه. روش هاي ديگر:

```

$link= mysql_connect($Host,$User,$Password);
// وارد کردن مقادير به جدول
$query="INSERT into $Tablename values ('$_POST[sname]','$_POST[sfamily]')";
mysql_query('SET NAMES \utf8\');
if (mysql_db_query($Dbname,$Query,$Link)) {

```

```

mysql_set_charset('utf8',$Link);

```

## مشکل نوشته های بهم ریخته:

اگر این **خط رو** قبل از ارسال کوئری و insert در قسمت ارسال اطلاعات به mysql قرار ندیم نوشته ها به صورت بهم ریخته نشون داده میشه.

```
$handle=mysql_connect('localhost','root','');
mysql_query("SET CHARACTER SET utf8",$handle);
$query = "INSERT INTO jadval SET msg = '$msg',title='$title',date='$date'";
$result=mysql_query($query,$handle);
```

سعی کنید حتما در قسمت فرم html از کد زیر استفاده کنید . چون امکانش هست که وقتی شما مطلبي رو ارسال میکنید به جای نوشته های فارسی نوشته های بهم ریخته برای دیتابیس ارسال میشه.  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

## :update

حالا میخوام به مثال ساده بزنم برای به روز کردن مطلب:

send.htm

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<HTML xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>آپدیت</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body>
<form action="8.php" method="POST">
<input type="text" name="FirstName" size="20"><br>
<input type="text" name="LastName" size="20"><br>
<input type="submit" value="ارسال">
</form>
</body>
</html>
```

8.php

```
<?php
$handle=mysql_connect('localhost','root','');
if($handle==false){
die("can not connect mysql.\r\n");
}
$db=mysql_select_db('testi');
if($db==false){
die("can not connect db.\r\n");
}
```

```

$FirstName= $_POST['FirstName'];
if($FirstName=="")
    die('فيلد خالي است');
$LastName= $_POST['LastName'];
if($LastName=="")
    die('فيلد خالي است');
    mysql_set_charset('utf8',$handle);
$query = "UPDATE persons SET LastName = '$LastName',FirstName='$FirstName'";
$result=mysql_query($query,$handle)or die (mysql_error());
// اگر مقدار يك ارسال بشه
if($result==1){
echo " تغيير يافت ";
}else{
echo"تغيير نيافت";
}
mysql_close($handle);
?>

```

این فقط يك مثال ساده بود شما مي تونيد این مثال رو به شکل بهتري هم بنويسيد.

## اعتبارسنجي فرم :

اصلاً شاید شما که در زمینه ي php تازه کار هستيد به خودتون بگيد اعتبارسنجي چي هست؟ اعتبارسنجي مثلاً يك فرم ثبت نام مي سازيد که مي خواهيد فردي که فرم شما رو پر مي کند برخي گزينه ها رو مثل نام و نام خانوادگي رو حتما پر کند . و اگر پر نکند به اون پيام بدهد که حتما این گزينه رو پر کنيد. شما مي توانيد این اعتبارسنجي رو با جی کوئري و جاوا اسکريپت هم انجام بديد ولي چون این برنامه ها سمت کلاینت هستند فرد ميتونه جاوا اسکريپ رو توي مرورگرش غير فعال کنه و حتي اگه يك فرم html بسازه ميتونه به راحتی اعتبارسنجي رو دور بزنه ولي اگه شما با php اعتبارسنجي بسازيد ديگه نمي تونه این کار رو انجام بده.

اعتبارسنجي با isset و empty :

اعتبارسنجي radio :

```

send.htm
<form action="8.php" method="POST">
مرد<input type="radio" name="gender" value="مرد">
زن<input type="radio" value="زن" name="gender"><br>
<input type="submit" value="ارسال">
</form>

```

```

8.php
<?php
if (isset($_POST['gender'])) {
echo " صحيح ";
}

```

```
} else {  
echo "گزینه ای انتخاب نکردید";  
}  
?>
```

isset میخواد بگه که آیا در gender مقداری هست یا نه؟ اگر هست پیام صحیح را چاپ کن اگر نیست پیام بده "گزینه ای انتخاب نکردید".

```
<?php  
if (!empty($_POST['gender'])) {  
echo "صحیح";  
} else {  
echo "گزینه ای انتخاب نکردید";  
}  
?>
```

اسکرپت بالا میگوید که اگر پیام ارسال شده خالی **نباشد** "صحیح" وگرنه پیام "گزینه ای انتخاب نکردید" چاپ می شود.

حالا اینطوری کاملترش کنید:

```
if(isset($_GET['gender']) && !empty($_GET['gender'])) {  
echo 'صحیح';  
}  
else {  
echo 'گزینه ای انتخاب نکردید';  
}
```

می تونید در قسمت echo از این کد جاوا اسکرپت هم برای پیغام استفاده کنید.

```
echo ("<script type='text/javascript'>alert('گزینه ای انتخاب نکردید')</script>");
```

این هم مثالی برای لیست:

```
<form action="get.php" method="post">  
<select name="tfmah" id="tfmah">  
<option value="1">1  
<option value="2">2  
<option value="3">3  
<option value="4">4  
<option value="5">5  
<option value="6">6  
<option value="7">7  
<option value="8">8
```



```

<option value="9">9
<option value="10">10
<option value="11">11
<option value="12">12
<option selected>ماه
</select>
<input type="submit" value="ارسال">
</form>

```

get.php

```

<?php
$values = array('1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12');
if(!in_array($_POST['tfmah'], $values)) {
    echo ("<script type='text/javascript'>alert('ماه تولد را وارد نكرديد')</script>");
}
?>

```

## شرطي يك خطي :

CONDITION(S) ? TRUE VALUE : FALSE FALSE

بايد اولش حتما مقداري برآش تعيين كنيم. اينطوري:

```

<?php
$t=55;
$b = ($t == 55 ? 'صحيح' : 'غلط');
echo $b;
?>

```

متغير \$t برابر با ۵۵ ، و دو تا مساوي هم يك نوع مقايسه ايجاد ميكنه. كه آيا اين متغير \$ برابر ۵۵ هست؟ اگه اينطوري باشه پيام صحيح رو چاپ ميكنه اگه اينطوري نيست غلط رو چاپ ميكنه.

يك مثال:

```

<form action="8.php" method="POST">
مرد<input type="radio" name="gender" value="مرد">
زن<input type="radio" value="زن" name="gender"><br>
<input type="submit" value="ارسال" name="submit">
</form>

```

8.php

```

<?php
isset($_POST['submit']) and isset($_POST['gender']) ? $ok = "صحيح":
$ok = "گزينه اي انتخاب نكرديد";

```

```
if(!empty($ok)){echo $ok;}
```

```
?>
```

## رمز گذاري با md5 :

```
<?php
```

```
$ravandi="ravandi";
```

```
echo md5($ravandi);
```

```
?>
```

خروجي که کد بالا ميده :

```
30293c297912a3c34a918e5d5a0357bd
```

در واقع کدي که اسکرپت بالا در خروجي ميده کد رمزگذاري شده ي رشته ي ravandi هست . مي تونيد با يك اعتبارسنجي به اين موضوع پي ببريد.

```
<?php
```

```
$ravandi = 'ravandi';
```

```
if (md5($ravandi) == '30293c297912a3c34a918e5d5a0357bd') {
```

```
    echo "صحيح است";
```

```
}else{
```

```
    echo "صحيح نيست";
```

```
}
```

```
?>
```

در خروجي پيام صحيح است را چاپ ميکند ولي اگر به جاي کد رمزگذاري شده کد ديگري رو قرار بديد پيام صحيح نيست رو چاپ ميکند.

کاربري که رمزش رو توي فرم لوگين وارد ميکند رمزش با تابع md5 به صورت يك رشته ي رمزگذاري شده تبديل ميشه و بعد در قسمت سمت سرور با توجه به اعتبارسنجي که انجام ميشه بررسي ميشه که اين کد رمزگذاري شده ارسالي به سرور با اون کد رمزگذاري شده که در ديتابيس ذخيره شده مطابقت ميکند يا نه؟ مثل مثال بالا به نمونه ي ساده هست.

## چرا بايد از md5 استفاده کرد؟

ممکنه ديتابيس هک بشه و کلمه رمز همه اعضاي سايت بيوفته دست هکر به همين دليل بايد کلمه رمز رو هش کنيد.

موقع ثبت نام کاربري و رمز عبور، براي ذخيره رمز ابتدا اون رو با MD5 کدگذاري کرده و بعد ذخيره ميکنيد.

## ۷ نکته که بايد رعايت شود : <http://www.ncis.ir>

۱- به هيچ عنوان نام کاربري و رمز عبور رو توي سيژن ذخيره نکنيد.

۲- چون توي ديتابيس رمز رو بصورت MD5 ذخيره کردين، موقع مقايسه هم اول بايد رمزي که کاربر تايپ کرده رو با MD5 کدگذاري کنيد و بعد مقايسه کنيد.

۳- براي اينکه نام کاربري به بزرگي و کوچکي حروف حساس نباشه، عبارت تايپ شده توسط کاربر رو به حروف کوچک تبديل کرده و با معادل حروف کوچک نام کاربري موجود در ديتابيس مقايسه کنيد.

۴- اینکه همه رکوردها رو استخراج کنید و بعد ببینید آیا رکوردی دارین که نام کاربری و رمز عبورش با اون چیزی که میخواین برابر هست، یک اشکال عمده در طراحی هست چون ممکنه تعداد کاربران زیاد باشه و این کار بشدت سرعت رو پایین میاره. علاوه حافظه زیادی هم اشغال میکنه. بجای اینکار، از خود MySQL بخواین فقط رکوردی که نام کاربری و رمز عبور اون با چیزی که کاربر وارد کرده یکسان هست رو برگردونه. بعد ببینید اگه چنین رکوردی وجود داشت (تعداد رکوردهای خروجی کوئری یک عدد مثبت بود)، یعنی کاربر اطلاعات رو درست وارد کرده.

۵- بجای قراردادن نام کاربری و رمز عبور، برای کاربری که لاگین کرده، فقط نام کاربری رو توی سیژن بگذارین تا بتونید بعداً از طریق اون به اطلاعات کاربر در هر صفحه ای که نیاز داشتن، دسترسی پیدا کنید.

۶- قبل از بررسی، چک کنید اگه قبلاً سیژن لاگین تعریف شده، اون رو حذف کنید تا کاربر نتونه از سیژن کاربر قبلی که احتمالاً از سیستم داشته استفاده میکرده و بدون بستن مرورگر و یا خارج شدن از سایت، سیستم رو ترک کرده، استفاده کنه.

۷- بعد از بررسی هم یا توسط توابع header کاربر رو به صفحه مناسب (صفحه اصلی یا فرم لاگین) هدایت کنید، یا لافل لینکی برای رفتن به صفحه مناسب در اختیارش بگذارین.

یه مثال عملی فرم ورود (login):

اول با md5 رمز کد گذاری شده می سازیم و در فیلد pass در دیتابیس ذخیره میکنیم به شکل زیر:

```
Database: `rams`
CREATE TABLE `pass` (
  `user` varchar(21) collate utf8_persian_ci NOT NULL,
  `pass` varchar(100) collate utf8_persian_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_persian_ci;
-INSERT INTO `pass` (`user`, `pass`) VALUES
('ravandi', '9683cc5f89562ea48e72bb321d9f03fb');
```

9683cc5f89562ea48e72bb321d9f03fb=1363

روش بالا مثل این می مونه که فرد ثبت نام کرده و یوزر و پسوردش ثبت شده. پسوردش ۱۳۶۳ هست.

login.php

```
<form action="chap.php" method="POST">
user<input type="text" name="user"><br>
password<input type="password" name="pass"><br>
<input type="submit" value="ارسال">
</form>
```

chap.php

```
<?php
session_start();
if(isset($_SESSION['login'])) {
  پاک کردن سیژن های قبلی //
  unset($_SESSION['login']);
}
mysql_connect('localhost', 'root', '') or die ('Connection error');
```

```

mysql_select_db('rams') or die ('Database error');
mysql_query('SET NAMES \'utf8\');
$user = mysql_real_escape_string(strtolower($_POST['user']));
$pass = md5($_POST['pass']);
$qry = "SELECT * FROM `pass` WHERE (LOWER(`user`)='{ $user}' AND `pass`='{ $pass}')";
$result = mysql_query($qry);
if($result && mysql_num_rows($result) > 0) {
    echo "<p>Welcome { $user}.</p>".PHP_EOL;
    echo '<p><a href="home.php">Go to main page</a></p>'.PHP_EOL;
    $_SESSION['login'] = $user;
}
else {
    echo '<p>نام کاربري و پسورد اشتباه است.</p>'.PHP_EOL;
    echo '<p><a href="login.php">Back to login page</a></p>'.PHP_EOL;
}
?>

```

LOWER حروف بزرگ رو به کوچک تبدیل میکند. چون اگه به اشتباه هم حروف بزرگ تایپ بشه ، به این صورت به حروف کوچک تبدیل بشه.

**در این عمل از تنظیمات کاراکتری موجود در پایگاه داده استفاده می کند.** **mysql\_real\_escape\_string : کاراکتر های ویژه را در یک رشته برای استفاده از پایگاه داده بهینه میکند و**

این تابع اینجا برای مسائل امنیتی استفاده می شود.  
توضیحات بیشتر در مورد این تابع در این آدرس می باشد:

[https://www.ircert.cc/upload/files/ar\\_mysql-wp-executing%2Bcve-2006-2753.pdf](https://www.ircert.cc/upload/files/ar_mysql-wp-executing%2Bcve-2006-2753.pdf)

شاید بازم صفحه ای داشته باشیم که بخوایم با این پسورد بازش کنیم . پس میایم این کد رو اول کدهای اون صفحه قرار میدیم:

```

<?php
session_start();
if(isset($_SESSION['login'])) {
?>

```

کد بالا میگه اگه سیژن مورد نظر وجود داشت اجازه ورود بده .  
کد پایین رو هم پایین همه ی کدهامون می نویسیم:

```

<?php
} else {
    header("Location:login.php");
exit;
}
?>

```

و کد بالا میگه اگه سیژن مورد نظر وجود نداشت و رمز رو اشتباه وارد کرده باشیم به آدرس login.php برگرد و از این صفحه خارج شو.

## ساخت تصویر امنیتی (captcha): وقتی در يك فرم ثبت نام میخواهید ثبت نام کنید یا در

قسمت نظر دهید سایتی می خواهید نظر بدید و ... با يك تصویری روبرو میشوید که باید قبل از ارسال مطالب فرم به سایت این تصویر رو پر کنید این تصویر در واقع برای جلوگیری از عمل اسپم‌هایی صورت میگیرد که به راحتی تعداد زیادی پیام برای سایت ارسال میکنند در واقع این تصاویر امنیتی برای این است که معلوم بشه که این فرم رو انسان پر کرده یا ربات.

با تابع Rand شما می تونید اعداد تصادفی تولید کنید:

```
<?php
$ravandi= Rand (100000,999999);
echo $ravandi;
?>
```

با هر بار فریش صفحه يك سری اعداد جدید ایجاد میشه .

ایجاد يك تصویر که با فریش صفحه اعداد تغییر می کنند :

```
<?php
session_start();
// ایجاد يك تصویر جدید
$img = imagecreatefrompng('test.png');
این تابع برای ایجاد اعداد تصادفی در تصویر است
$numero = rand(100, 999);
// تغییر اعداد تصادفی داخل سیژن ذخیره می شود.
$_SESSION["captcha"] = $numero;
اختصاص دادن رنگ RGB به اعداد
$white = imagecolorallocate($img, 255, 255, 255);
imagestring($img, 10, 8, 3, $numero, $white);
//mime aks
header ("Content-type: image/png");
// ایجاد عکس به فرمت png
imagepng($img);
?>
```

فایل بالا رو با نام captcha.php ذخیره کنید.

فایل test.png باید وجود داشته باشد.

RGB: یعنی Red, Green, Blue که اگه همه ي این رنگ ها ۲۵۵ باشد رنگ سفید تولید میکند و اگر همه ي این رنگ ها صفر باشد رنگ سیاه تولید میکند.

در خط بعدی عدد ده اندازه ي فونت عدد هست و ۸ اگر هر چي بزرگتر باشه عدد به سمت راست میره .  
و عدد ۳ هر چي بزرگتر باشه عدد به سمت پایین میره .

منبع تصویر امنیتی : <http://phpform.net/captcha.php>

دقت کنید که کد بالا اگه باعث ایجاد تصویر نشود. به گزینه ي Encoding در برنامه ي notepad++ رفته و گزینه ي Encode in UTF-8 without BOM رو انتخاب کنید و قبل از نگ <?php هم فاصله و کد html نباشد.

برای ایجاد تصویر امنیتی همیشه روش های دیگه ای رو هم استفاده کرد . مثلا استفاده از فونت ، تصویر متحرک ، و ...

فایل زیر رو هم با نام captcha.htm ذخیره کنید.

```
<form action="8.php" method="post">

<input type="text" name="captcha" id="captcha" size="3" maxlength="5" dir="ltr">
<input type="submit" value="ارسال">
</form>
```

8.php

```
<?php
session_start();
/*
اعتبارسنجی همیشه که معلوم بشه مقدار ارسال شده از فرم با
مقدار ذخیره شده در سیزن برابری میکنه یا نه؟
*/
if ($_POST["captcha"] != $_SESSION["captcha"] OR $_SESSION["captcha"]=="") {
    echo '<b>کد تایید اشتباه است</b><br>';
} else {
    echo '<b>تایید با موفقیت انجام شد</b><br>';
}
?>
```

همیشه برای قسمت html دکمه ی رفرش هم ساخت. و حتی همیشه این تصویر امنیتی رو امن ترش کرد و لی من فقط برای یادگیری این کدها رو گذاشتم. و در عمل نیاز همیشه شما از کدهای امن تری استفاده کنید.

## شی گرایی:

با شی در محیط زندگی خود هم روبرو هستید مثل کتاب ، دفتر، گریه ، اسب ، الاغ که هر کدام از این اشیاء دارای ویژگیهایی چون خواص ، رفتار ، و روابط هستند.

مثلاً الاغ دارای قد ، وزن ، رنگ و غیره است. رفتار هر شی هم طریقه ی پاسخ دادن به رویدادها و اتفاقاتی است که برایش پیش می آید. مثلاً الاغ رو آگه بهش لگد بزنی ممکنه جفتک در کنه. و روابطی هم که یک شی با دیگر اشیاء داره . مثلاً الاغ صاحب یک بچه الاغ است. مثلاً دکمه یک شی است که دارای ویژگی اندازه و رنگ هست و در مقابل رفتار کلیک ، کاری انجام میده مثلاً لینکی رو باز میکنه و با دیگر صفحات نوعی ارتباط برقرار میکنه.

```
<?php
class MyClass
{
    function printphp()
    {
        echo 'php';
    }
}

$my_var = new MyClass();
$my_var->printphp();
```

?>

در خط اول میام یک شی رو با class تعریف میکنیم .

بعد برای این شی نیاز به تابع و متغیر داریم .

**باید شی class رو به یک متغیر نسبت بدیم. الان توابع و متغیر های درون class در حافظه ی متغیر \$my\_var قرار گرفته.**

منبع اسکریپت : سایت مجید آنلاین.

در خط بعد باید اجازه ی دسترسی به تابع printphp () رو بدیم وقتی با >-اجازه ی دسترسی به تابع رو دادیم فرمان اجرای تابع داده میشه و تابع هم با دستور echo گزینه ی php رو چاپ میکنه.

نکته : در خط آخر دستور اجرای توابع صادر میشه. هر تابعی بالا باشه باید با دستور >- اجرای اون تابع صادر بشه.

پس دو بخش شد: بخش فرمان و بخش اجرا.

**برای اینکه اطلاعاتی رو از صفحه ای دیگر بگیرم به شکل زیر:**

```
<?php
require_once('class.Demo.php');
$objDemo = new Demo();
$objDemo -> sayHello('Masoud');
?>
```

```
class.Demo.php
<?php
class Demo
{
function sayHello($name)
{
print "Hello $name!";
} }
?>
```

**در آخر از شما میخوام که در نظر سنجی که در وبلاگم گذاشتم شرکت کنید و نظرتون رو در مورد این کتاب بگید:**

<http://barnamenevisi.blogspot.com/1390/12/06/post-94/>

## منابع

<http://www.w3schools.com>

<http://php.net>

<http://www.barnamenevis.org>

<http://www.barnamenevisi.blogspot.com>

در خوراک خوان سایت زیر عضو شوید در صورتی که کتاب جدیدی منتشر کردم در این آدرس پیغام خواهم گذاشت:

<http://pachenar.ir/rss.php>

## پایان