

آموزش کاربردی



سطح: مقدماتی و پیشرفته

گردآورنده: برادران ایمانی

۱۳۹۸

پیشگفتار

سخنی با خوانندگان

با سلام

جزوه آموزشی که هم اکنون در اختیار شماست حاصل تلاشها و زحمات ما دو برادر می‌باشد. اکثر جزوات آموزشی که در مورد زبان PHP وجود دارد دارای مطالب ناقص و پراکنده می‌باشند. در این راهنما سعی گردیده است که ترتیب دروس طوری باشد، فردی که هیچ اطلاعاتی در مورد این زبان ندارد بتواند با مطالعه دروس این راهنما روش کار با این زبان را فرا بگیرد. امیداست دوستان با راهنمایهای خود در مورد این جزوه ما را در هر چه بیشتر کاربردی کردن آن یاری فرمایند.

با تشکر

برادران کوچک شما: فرهاد و فرزاد ایمانی

Farhad.Imany@Im-Br.IrFazhad.Imany@Im-Br.Ir

فقط برای شروع کار شما باید قبلا راهنمای آموزش HTML-CSS-JAVASCRIPT را که توسط ما تهیه شده است را مطالعه کرده باشید.

نکته مهم

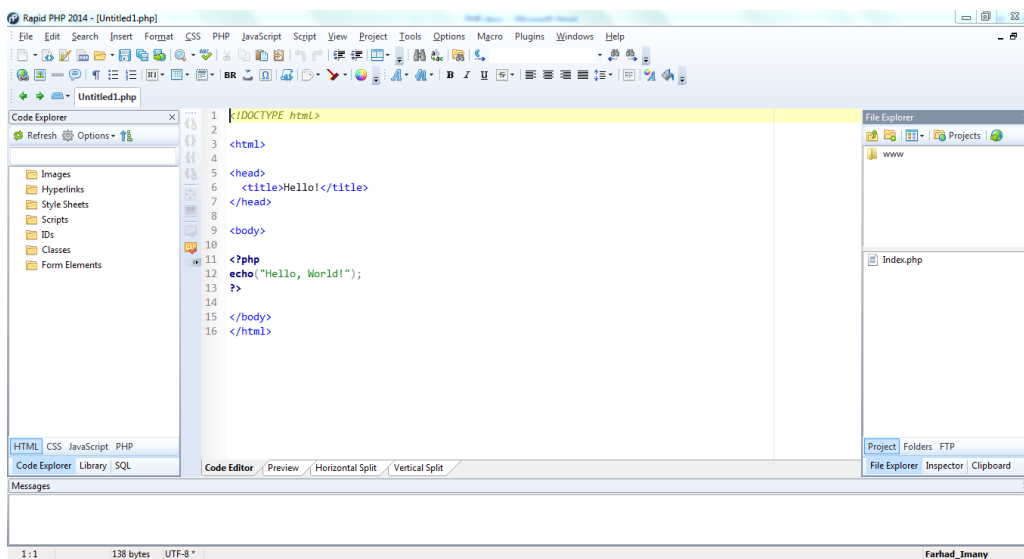
تمامی دستورات بکار رفته ابتدا یکبار تست و سپس در کتاب آمده است. بنابراین هیچ بخش از دستورات دارای خطا نمی‌باشد

فصل اول

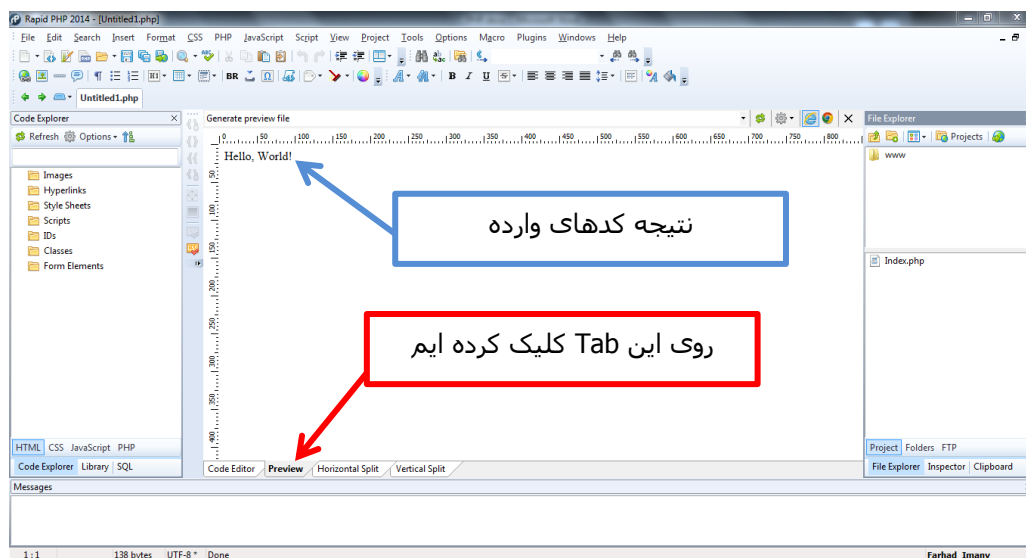
بررسی مفاهیم اولیه در برنامه نویسی PHP

آشنایی با اصول اولیه

در این کتاب آموزشی قصد داریم شما عزیزان را با تمام موارد مربوط به برنامه نویسی توسط زبان بسیار قدرتمند PHP آشنا سازیم. امید است با مطالعه گام به گام فصول به توانایی مورد انتظار در این زمینه برسید. شرط موفقیت شما در مطالعه این کتاب با انجام تمریناتی خواهد بود که به مرور با آنها مواجه خواهید شد. ما در این دوره آموزشی از نرم افزار Rapid PHP 2014 برای انجام تمرینات استفاده کرده ایم. بعد از اجرای نرم افزار مربوطه محیط زیر را مشاهده خواهید نمود:

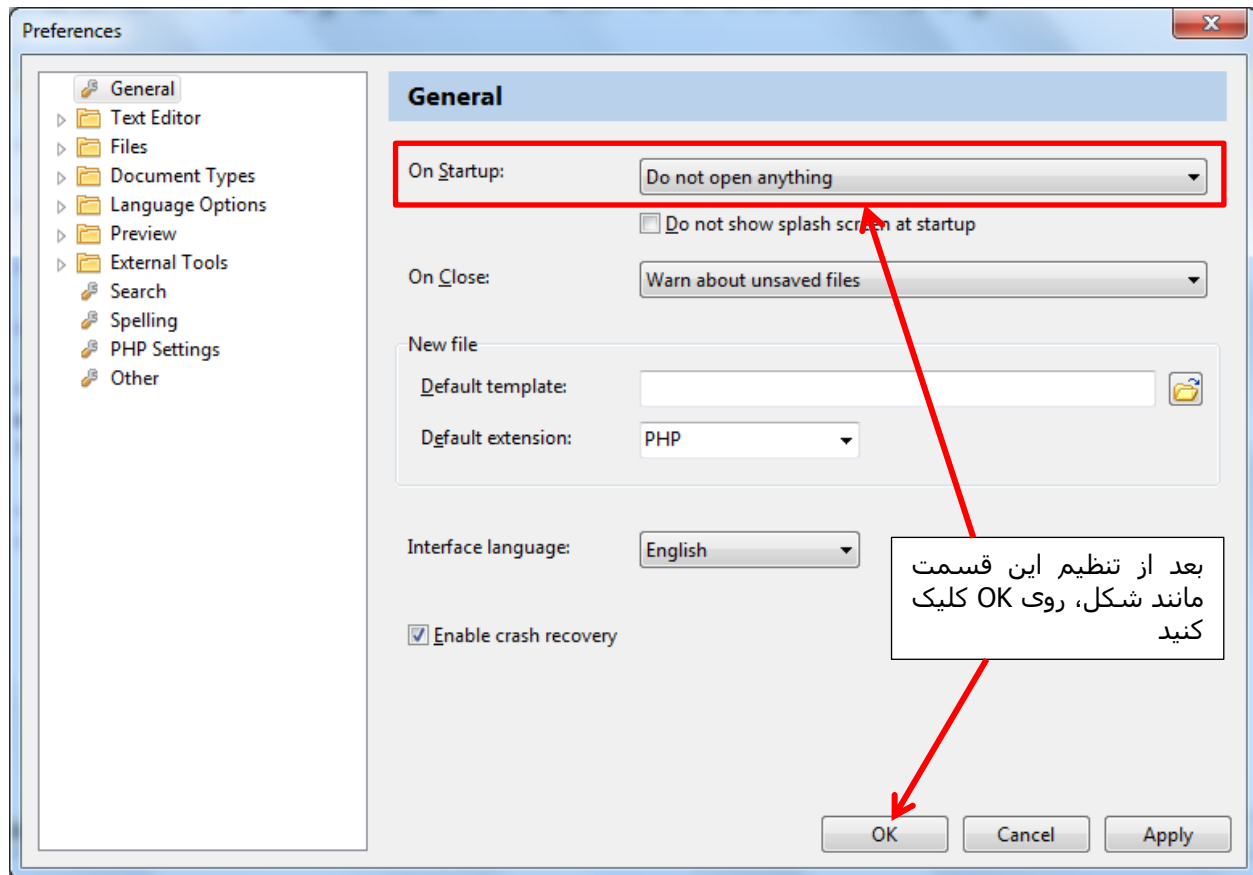


همانطور که ملاحظه می‌فرمایید بصورت پیش فرض یک فایل php بصورت اتوماتیک ایجاد شده و کدهایی ابتدایی در آن موجود است. کافیه برای دیدن نتیجه کدهای مربوطه روی Tab با عنوان Preview کلیک کنید. اگر خواستید نتیجه را در Browser ببینید از کلید F2 استفاده نمایید.



معمولا نرم افزار Rapid PHP را باید طوری تنظیم نمایید که هنگام اجرا، فایل نمونه را باز نکند. در غیر اینصورت همیشه مجبورید در ابتدا این فایل را ببندید. بدین منظور مسیر زیر را طی کنید:

Tools -> Preferences



در ادامه با سه روش برای نوشتن کدهای PHP آشنا می‌شویم:

روش اول:

<?

محل نوشتن کدهای PHP

?>

روش دوم:

<?php

محل نوشتن کدهای PHP

?>

روش سوم:

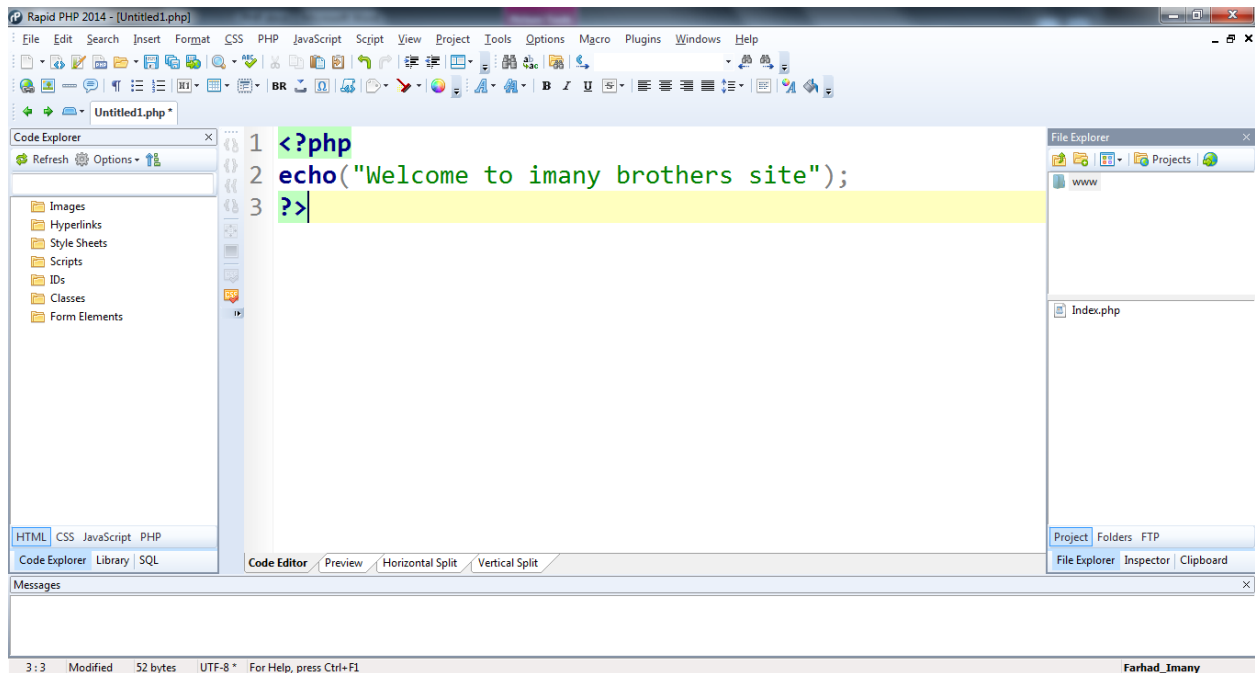
<script language="php">

محل نوشتن کدهای PHP

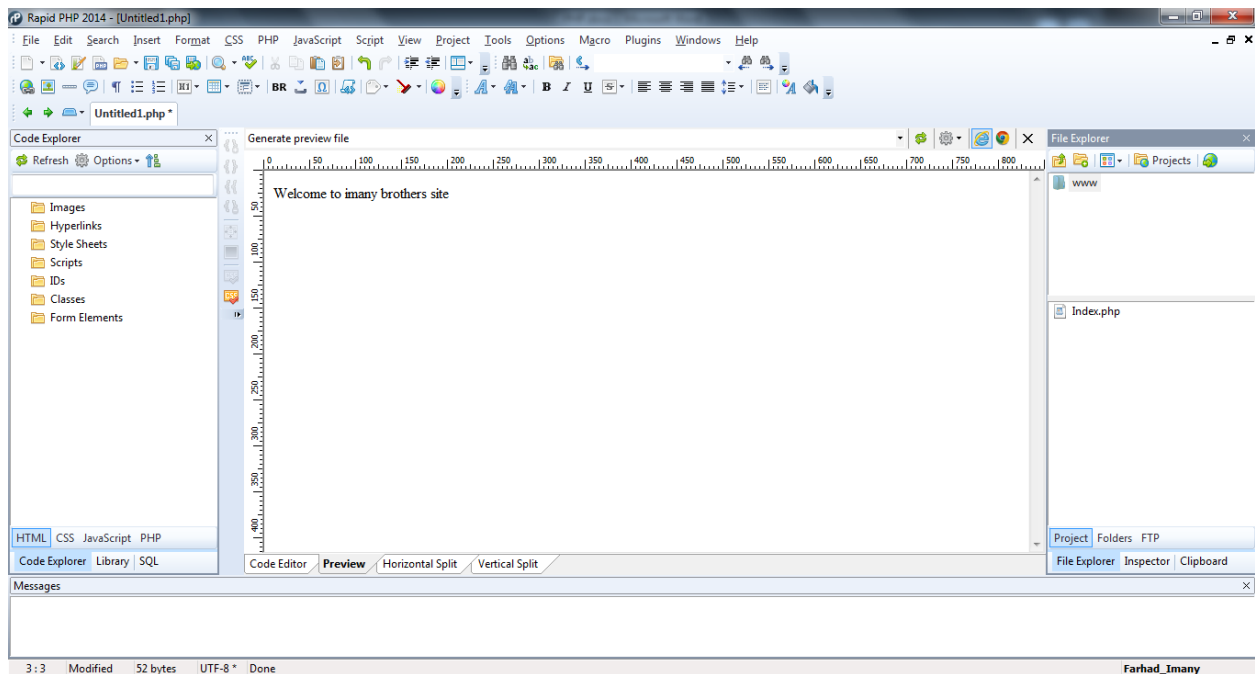
</script>

مثال) برنامه ای بنویسید که پیام Welcome to imany brothers site را داخل Browser شما نشان دهد.

پاسخ:



حال روی Tab با عنوان Preview کلیک کنید تا نتیجه را ملاحظه فرمایید. مانند شکل زیر:



نحوه پایان دادن به دستورات

برای اینکار به دو صورت میتوان عمل کرد:

حالت اول) در انتهای هر دستور باید از علامت ; استفاده نمایید. مانند زیر:

```
<?php
```

```
echo("Welcome to imany brothers site"); ←
?>
```

حالت دوم) نیازی به استفاده از علامت ; نیست. مانند زیر:

```
<?php echo("Welcome to imany brothers site") ?>
```

نحوه درج توضیحات به دستورات

بعضی اوقات نیاز است یکسری توضیحات را در متن برنامه بنویسید تا توسط آن بتوانید اطلاعاتی در مورد آن بخش ارایه نمایید. بنابراین باید حالتی را در مورد آن توضیحات بوجود بیاورید که به عنوان دستور در نظر گرفته نشوند و صرفاً حالت Comment داشته باشند. همچنین از این تکنیک میتوانید برای از دور خارج کردن یکسری از دستورات وارده استفاده نمایید و حالت اجرایی را از آن دستورات بگیرید. برای اینکار به یکی از دو صورت زیر عمل می‌کنیم:

روش اول: توضیح در انتهای یک خط دستوری

```
1 <?php
2 echo("Welcome to imany brothers site") //This is comment
3 ?>
```

روش دوم: توضیح در چند خط

```
1 <?php
2 /*
3 This is comment
4 This is comment
5 This is comment
6 */
7 echo("Welcome to imany brothers site")
8 ?>
```

نکته: از همین دو روش میتوانید برای از دور خارج کردن یک دستور یا محدوده‌ای از دستورات استفاده کنید.

به عنوان مثال برای از دور خارج کردن یک خط دستوری میتوانید ابتدای آن // را قرار دهید. خواهید دید که دستور به رنگ خاکستری درخواهد آمد.

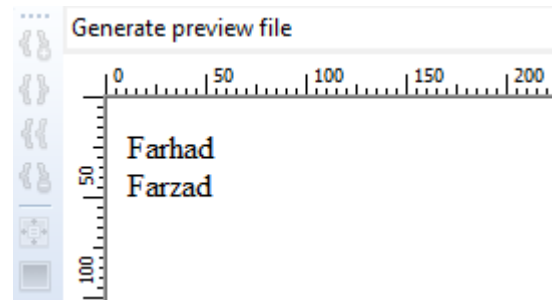
یا برای از دور خارج کردن یکسری از دستوراتی که پشت سرهم می‌باشند ابتدای اولین دستور مورد نظر علامت /* را قرار داده و سپس به انتهای آخرین دستور مورد نظر رفته و علامت */ را قرار دهید.

نحوه تغییر متغیر

همانطوریکه میدانید متغیرها بخشی از حافظه میباشند که میتوانند مقادیری را برای ما در داخل نگاه دارند. در php برای تعریف یک متغیر از علامت \$ استفاده می‌شود. در ضمن به خاطر داشته باشید زبان php به حروف بزرگ و کوچک حساس می‌باشد.

به مثال زیر توجه کنید:

```
1 <?php
2 $name='Farhad';
3 $Name='Farzad';
4 echo $name;
5 echo "<br />";
6 echo $Name;
7 ?>
```

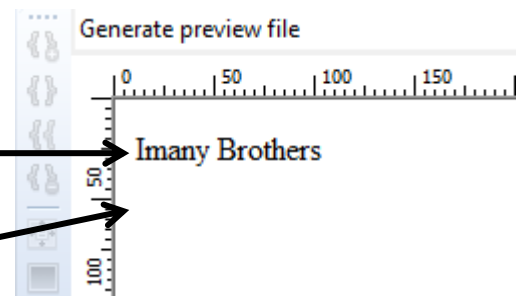


نحوه حذف مقدار متغیر

دو روش برای انجام اینکار وجود دارد:

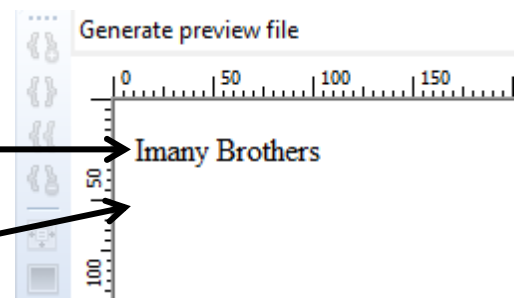
روش اول: استفاده از NULL

```
1 <?php
2 $name="Imany Brothers";
3 echo $name;
4 $name =NULL;
5 echo $name;
6 ?>
```



روش دوم: استفاده از unset

```
1 <?php
2 $name="Imany Brothers";
3 echo $name;
4 unset($name);
5 echo $name;
6 ?>
```



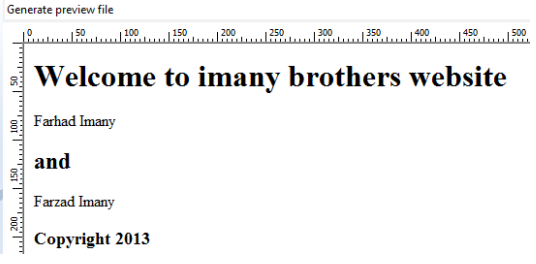
ترتیب کدهای HTML و PHP

برای اینکار محدودیتی نداریم و میتوانیم در ابتدا یا مابین برنامه یا انتها کدها را بنویسیم. به شکل دقت کنید:

```

1 <H1>Welcome to imany brothers website</H1>
2 <?php
3 echo "Farhad Imany";
4 echo "<h2>and</h2>";
5 echo "Farzad Imany";
6 ?>
7 <H3>Copyright 2013</H3>

```



1

2

3

نوشتن کد HTML قبل از PHP

1

نوشتن کد HTML مابین PHP

2

نوشتن کد HTML بعد از PHP

3

نحوه تعریف ثابتها

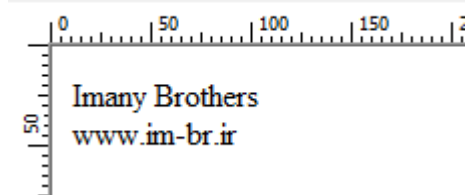
ثابتها متغیرهایی هستند که یک مقدار اولیه به آنها تعلق گرفته و در طول برنامه این مقدار را حفظ می‌کنند. برای تعریف ثابت از کلمه کلیدی define استفاده می‌کنیم. همچنین به خاطر داشته باشید که برای تعریف متغیری از نوع ثابت نیازی به علامت \$ نداریم.

```

<?php
define ("Name","Imany Brothers");
define ("Site","www.im-br.ir");
echo Name."<br />".Site;
?>

```

Generate preview file



Imany Brothers
www.im-br.ir

نوع دیگری از بکار بردن همزمان کدهای php و html

فصل دوم

بررسی دستورات شرطی و حلقه ها

بررسی دستور IF

ساختار دستور IF بصورت زیر است:

```
if (condition)
```

```
{
```

```
Commands
```

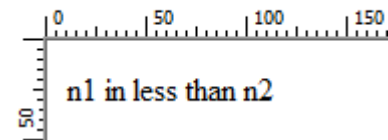
```
}
```

مثال) دو متغیر به نامهای n1 و n2 تعریف نموده و مقدار n1 را برابر ۱۰۰ و مقدار n2 را برابر ۲۰۰ قرار دهید. حال شرطی بنویسید که اگر n1 کوچکتر از n2 بود پیامی مناسب را نشان دهد.

```
1 <?php
2 $n1=100;
3 $n2=200;
4 if ($n1<$n2)
5 {
6 echo("n1 in less than n2");
7 }
8 ?>
```



Generate preview file



بررسی دستور Else If

از این دستور برای نقض شرطها استفاده می‌گردد. یعنی در صورت عدم برقراری یک شرط، با استفاده از این دستور می‌توان روند دیگری را برنامه ریزی نمود.

مثال) برنامه بنویسید که دو مقدار عددی متغیرهای n1 و n2 را بررسی کرده و در صورت بزرگتر بودن، مساوی یا کوچکتر بودن n1 نسبت به n2 پیامهای مناسب را نشان دهد.

پاسخ:

```
1 <?php
2 $n1=100;
3 $n2=200;
4 if ($n1>$n2)
5     {echo "n1 greather than n2";}
6 else if ($n1==$n2)
7     {echo "n1 equal n2";}
8 else if ($n1<$n2)
9     {echo "n1 less than n2";}
10 ?>
```

نحوه اجرای دستورات HTML در داخل شرط IF

مثال) برنامه ای بنویسید که اگر عدد n1 برابر 1 شد، یک جدول با دو عنوان نام و نام خانوادگی که مشخصات فرهاد ایمانی در آن باشد به نمایش درآید و در غیر اینصورت یک جدول با دو عنوان نام و نام خانوادگی که مشخصات فرزاد ایمانی در آن باشد به نمایش درآید.

پاسخ:

```

1 <?php $n1=1; ?>
2 <?php if($n1==1):?>
3     <table border="2">
4         <tr>
5             <td>Name</td>
6             <td>Family</td>
7         </tr>
8         <tr>
9             <td>Farhad</td>
10            <td>Imany</td>
11        </tr>
12    </table>
13 <?php else :?>
14     <table border="2">
15         <tr>
16             <td>Name</td>
17             <td>Family</td>
18         </tr>
19         <tr>
20             <td>Farzad</td>
21             <td>Imany</td>
22         </tr>
23     </table>
24 <?php endif ; ?>

```

Generate preview file

Name	Family
Farhad	Imany

در این برنامه n1=1 است

```

1 <?php $n1=2; ?>
2 <?php if($n1==1):?>
3     <table border="2">
4         <tr>
5             <td>Name</td>
6             <td>Family</td>
7         </tr>
8         <tr>
9             <td>Farhad</td>
10            <td>Imany</td>
11        </tr>
12    </table>
13 <?php else :?>
14     <table border="2">
15         <tr>
16             <td>Name</td>
17             <td>Family</td>
18         </tr>
19         <tr>
20             <td>Farzad</td>
21             <td>Imany</td>
22         </tr>
23     </table>
24 <?php endif ; ?>

```

در این برنامه n1=2 است

Generate preview file

Name	Family
Farzad	Imany

بررسی دستور While

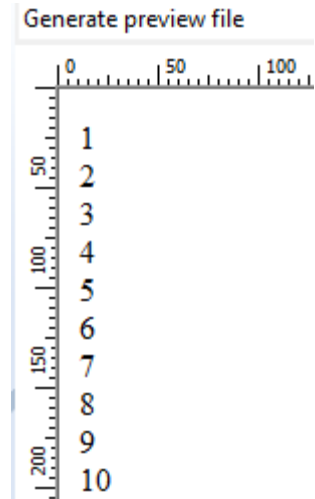
از این دستور برای ایجاد یک حلقه استفاده می‌شود. بدین صورت که: تا زمانی که شرط حلقه برقرار است، دستورات داخل حلقه تکرار می‌شوند.

(مثال) برنامه ای بنویسید که اعداد یک تا ده را زیر هم چاپ کند.

```

1 <?php
2 $n1=1;
3 while ($n1<=10):
4     echo $n1."<br/>";
5     $n1 ++;
6 endwhile;
7 ?>

```



نحوه اجرای دستورات HTML در داخل حلقه While

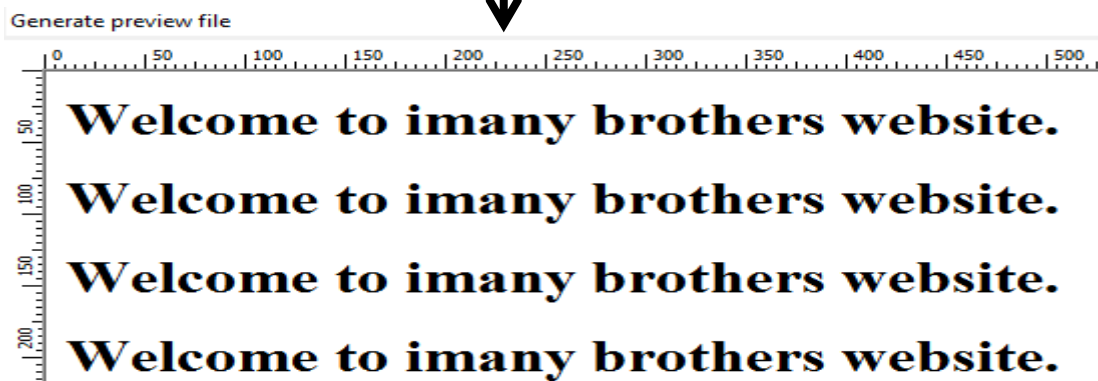
(مثال) برنامه ای بنویسید که عبارت Welcome to imany brothers website را با حالت h1 چهار بار تکرار کند.

پاسخ:

```

1 <?php
2 $n1=1;
3 while ($n1<=4): ?>
4 <h1> Welcome to imany brothers website.</h1>
5 <?php
6 $n1 ++;
7 endwhile; ?>

```



بررسی دستور Do While

از این دستور نیز برای ایجاد یک حلقه استفاده می‌شود با این تفاوت که شرط حلقه در انتهای آن خواهد آمد. بنابراین نتیجه می‌گیریم که دستورات داخل آن حداقل یکبار اجرا می‌گردند.

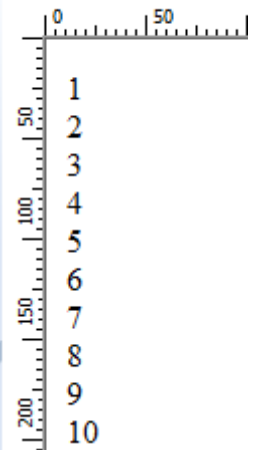
مثال) برنامه ای بنویسید که اعداد یک تا ده را زیر هم چاپ کند.

```

1 <?php
2 $n1=1;
3 do
4 {
5     echo $n1."<br/>";
6     $n1 ++;
7 } while ($n1<=10);
8 ?>
```



Generate preview file



دستور Break:

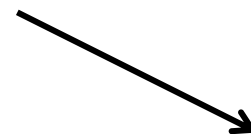
از این دستور در داخل حلقه زمانی استفاده می‌شود که در صورت برقرار شدن یک شرط در داخل حلقه، قصد خروج از حلقه را داشته باشیم.

مثال) یک حلقه بی نهایت تشکیل دهید که عبارت Welcome to imany brothers website را سه بار تکرار کند.

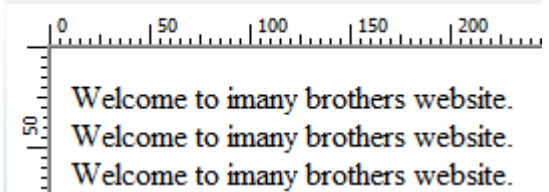
پاسخ:

```

1 <?php
2 $n1=1;
3 do
4 {
5     echo "Welcome to imany brothers website."<br/>;
6     $n ++;
7     if ($n==3)
8     {
9         break;
10    }
11 } while (True);
12 ?>
```



Generate preview file



بررسی دستور For

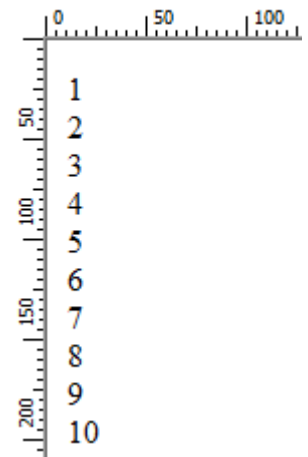
از این دستور نیز برای ایجاد حلقه استفاده می‌شود با این تفاوت که شمارنده آن داخل خود می‌باشد و مانند حلقه While نیازی نیست که ما شمارنده‌ای مانند ($n1++$) ایجاد نمائیم. از این دستور زمانی استفاده نماییم که تعداد دفعات اجرا مشخص می‌باشد.

```

1 <?php
2 for ($i=1; $i<=10; $i++)
3 {
4     echo $i."<br/>";
5 }
6 ?>

```

Generate preview file



بررسی دستور For Each

این دستور از PHP 4 به بعد ارایه شده است و برای کار با آرایه‌ها مناسب است.

نحوه تعریف یک آرایه که دارای دو مقدار رشته‌ای در داخل خود است:

```
$imbr = array("Farzad","Farhad");
```

مثال) با توجه به آرایه ذکر شده، مقادیر داخل آن را در خروجی نمایش دهید.

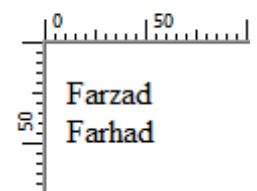
پاسخ:

```

1 <?php
2 $imbr = array("Farzad","Farhad");
3 foreach ($imbr as $value)
4 {
5     echo "$value."<br/>";
6 }
7 ?>

```

Generate preview file

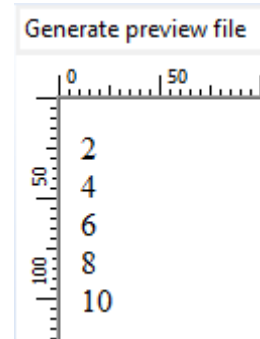


مثال) برنامه ای بنویسید که اعداد یک تا ده را با استفاده از یک حلقه داخل آرایه قرار دهد. سپس اعدادی را صرفاً در خروجی نمایش دهد که باقیمانده تقسیم آنها بر عدد ۲ صفر می‌باشد.

پاسخ:

```

1 <?php
2 $imbr = array();
3 for ($i=1; $i<=10; $i++)
4     $imbr[$i]=$i;
5 foreach ($imbr as $value)
6 {
7     if ($value%2==0)
8         echo "$value". "<br/>";
9
10 }
11 ?>
```

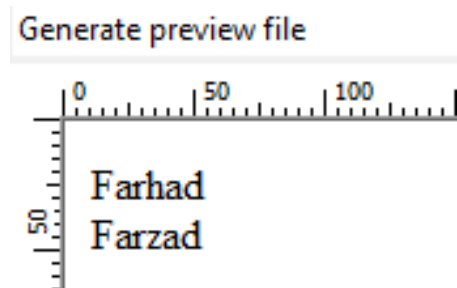
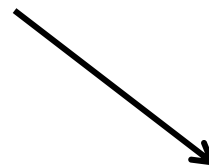


مثال) برنامه ای بنویسید که سه عبارت Farhad,im-br.ir,Farzad را داخل سه خانه آرایه قرار دهد.. سپس محتوای خانه‌هایی را نشان دهد که ایندکس آنها در آرایه زوج است. (منظور از ایندکس، شماره خانه آرایه می‌باشد که از صفر شروع می‌شود)

پاسخ:

```

1 <?php
2 $imbr = array("Farhad","im-br.ir","Farzad");
3 foreach ($imbr as $key =>$value)
4 {
5     if ($key%2==0)
6         echo "$value". "<br/>";
7 }
8 ?>
```



دستور Continue:

از این دستور در داخل حلقه زمانی استفاده می‌شود که قصد دارید در صورت برقراری یک شرط داخل حلقه، دستورات بعد از آن شرط اجرا نشده و به ابتدای حلقه بازگردید .

مثال) برنامه ای بنویسید که اعداد یک تا ده را به استثنای عدد ۵ چاپ کند.

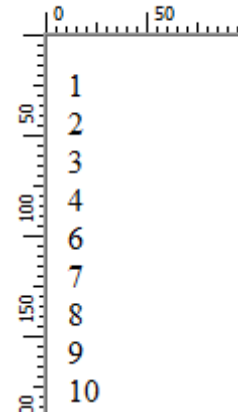
پاسخ:

```

1 <?php
2 for ($i=1; $i<=10; $i++)
3 {
4     if($i==5) continue;
5     echo $i."<br/>";
6 }
7 ?>

```

Generate preview file



بررسی دستور Switch

از این دستور زمانی استفاده کنید که قصد دارید برای مقدار یک متغیر چندین حالت را بررسی نمایید. در واقع جایگزینی بسیار مناسب برای If های تودرتو می‌باشد.

مثال) برنامه ای بنویسید مقدار متغیر n1 را بر اساس سه حالت دلخواه بررسی نموده و پیامی مناسب را نشان دهد.

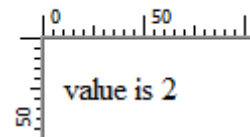
پاسخ:

```

1 <?php
2 $n1=2;
3 switch ($n1)
4 {
5     case 0:
6         echo "value is 0";
7         break;
8     case 1:
9         echo "value is 1";
10        break;
11    case 2:
12        echo "value is 2";
13        break;
14 }
15 ?>

```

Generate preview file



فصل سوم

عبارتها، عملگرها و عملوندها

عملگرهای ریاضی

PHP پنج نوع عملگر ریاضی را پشتیبانی می‌کند. در جدول زیر می‌توانید این پنج نوع عملگر را مشاهده فرمایید:

نام عملگر	عملگر	مثال	توضیح
جمع	+	$\$a + \b	جمع $\$a$ و $\$b$
تفریق	-	$\$a - \b	کسر $\$b$ از $\$a$
ضرب	*	$\$a * \b	ضرب $\$a$ در $\$b$
تقسیم	/	$\$a / \b	تقسیم $\$a$ بر $\$b$
باقیمانده	%	$\$a \% \b	باقیمانده تقسیم $\$a$ بر $\$b$

حال برای اینکه با خروجی این پنج عملگر بصورت عملی آشنا شویم، به مثال زیر توجه فرمایید:

```

1 <?php
2 $a = 5;
3 $b = 3;
4 echo ("a = $a<br>b = $b<br><br>");
5 echo ("a+b = " . ($a + $b) . "<br>");
6 echo ("a-b = " . ($a - $b) . "<br>");
7 echo ("a*b = " . ($a * $b) . "<br>");
8 echo ("a/b = " . ($a / $b) . "<br>");
9 echo ("a%b = " . ($a % $b) . "<br>");
10 ?>
```

→

Generate preview file

```

a = 5
b = 3
a + b = 8
a - b = 2
a * b = 15
a / b = 1.666666666666667
a % b = 2
```

نکته یک :

در PHP عبارت زیر به این معنی است که مقدار متغیر b برابر متغیر c می‌شود و مقدار متغیر a نیز برابر مقدار متغیر b می‌گردد.

```
$a=$b=$c
```

نکته دو :

در PHP از عملگرهای ادغامی نیز می‌توان استفاده نمود. به مثال زیر توجه کنید:

```
$a=4;
```

```
$a+=2;
```

در این مثال مقدار متغیر a نهایتاً برابر ۶ خواهد بود. در واقع:

$\$a+=2$ همان $\$a=\$a+2$ می‌باشد.

عملگرهای بیتی

در شکل زیر می‌توانید نحوه عملکرد عملگرهای بیتی را ملاحظه فرمایید:

توضیح	مثال	عملگر
اگر دو بیت برابر یک باشند، برابر یک و در غیر این صورت برابر صفر است.	$\$a \& \b	&
اگر یکی یا هر دو بیت برابر یک باشد، برابر یک و در غیر این صورت برابر صفر است.	$\$a \b	
اگر یکی از دو بیت و نه هر دو برابر یک باشند، برابر یک و در غیر این صورت برابر صفر است.	$\$a \wedge \b	^
بیتها را به سمت چپ حرکت می‌دهد.	$\$a \ll \b	<<
بیتها را به سمت راست حرکت می‌دهد.	$\$a \gg \b	>>
اگر دو بیت برابر یک باشند، برابر یک و در غیر این صورت برابر صفر است.	$\$a \text{ AND } \b	AND
اگر یکی یا هر دو بیت برابر یک باشد، برابر یک و در غیر این صورت برابر صفر است.	$\$a \text{ OR } \b	OR
اگر یکی از دو بیت و نه هر دو برابر یک باشند، برابر یک و در غیر این صورت برابر صفر است.	$\$a \text{ XOR } \b	XOR

در برنامه زیر تعدادی از آیتمهای جدول فوق را روی دو عدد ۷ و ۲ اعمال کرده ایم:

```

1 <?php
2 echo("(7 & 2) = " . (7 & 2) . "<br>");
3 echo("(7 | 2) = " . (7 | 2) . "<br>");
4 echo("(7 ^ 2) = " . (7 ^ 2) . "<br>");
5 echo("(2 << 2) = " . (2 << 2) . "<br>");
6 echo("(2 >> 1) = " . (2 >> 1) . "<br>");
7 ?>
```



Generate preview file

```

0
50
100
1
(7 & 2) = 2
(7 | 2) = 7
(7 ^ 2) = 5
(2 << 2) = 8
(2 >> 1) = 1
```

عملگرهای مقایسه‌های

از این عملگرهای در دستورات مقایسه‌ای استفاده می‌شود. به جدول زیر دقت فرمایید:

توضیح	مثال	اپراتور
تساوی	<code>\$a==\$b</code>	<code>==</code>
تساوی مقدار و نوع	<code>\$a===\$b</code>	<code>===</code>
عدم تساوی	<code>\$a!= \$b</code>	<code>!=</code>
عدم تساوی مقدار و نوع	<code>\$a!==\$b</code>	<code>!==</code>
بزرگتر	<code>\$a>\$b</code>	<code>></code>
کوچکتر	<code>\$a<\$b</code>	<code><</code>
بزرگتر یا مساوی	<code>\$a>=\$b</code>	<code>>=</code>
کوچکتر یا مساوی	<code>\$a<=\$b</code>	<code><=</code>

PHP دارای یک عملگری است که نیاز به سه بخش دارد. به دستور زیر دقت فرمایید:

; عبارت سوم: عبارت دوم? عبارت اول

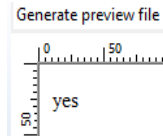
در این دستور عبارت اول بررسی می‌شود، اگر صحیح بود عبارت دوم محاسبه می‌گردد و اگر صحیح نبود عبارت سوم محاسبه می‌گردد.

(مثال)

```

1 <?php
2 $st=10;
3 $n=($st>0)?"yes":"no";
4 echo($n);
5 ?>

```



در مثال فوق مقدار متغیر `$st` بررسی می‌گردد. اگر بزرگتر از صفر بود مقدار `yes` در متغیر `$n` ذخیره خواهد شد و در غیر اینصورت مقدار `no` در نهایت مقدار `$n` چاپ می‌شود.

عملگرهای افزایشی و کاهششی

این عملگرها بصورت زیر می‌باشند:

توضیح	مثال	عملگر
مقدار متغیر <code>\$a</code> یک واحد افزایش می‌یابد و سپس مقدار برگشت داده می‌شود.	<code>++\$a</code>	<code><متغیر>++</code>
مقدار متغیر <code>\$a</code> برگشت داده شده و سپس یک واحد افزایش می‌یابد.	<code>\$a++</code>	<code>++<متغیر></code>
مقدار متغیر <code>\$a</code> یک واحد کاهش می‌یابد و سپس مقدار برگشت داده می‌شود.	<code>--\$a</code>	<code><متغیر>--</code>
مقدار متغیر <code>\$a</code> برگشت داده شده و سپس یک واحد کاهش می‌یابد.	<code>\$a--</code>	<code>--<متغیر></code>

عملگرهای منطقی

این عملگرها بصورت زیر می‌باشند:

توضیح	مثال	عملگر
اگر هر دو متغیر \$a و \$b برابر درست باشند، برابر درست خواهد بود.	\$a and \$b	And
اگر حداقل یکی از دو متغیر \$a یا \$b درست باشد، برابر درست خواهد بود.	\$a or \$b	Or
اگر یکی از دو متغیر برابر درست باشد، برابر درست است.	\$a xor \$b	Xor
اگر \$a برابر درست باشد، نادرست می‌شود.	!\$a	!
اگر هر دو متغیر \$a و \$b برابر درست باشند، برابر درست خواهد بود.	\$a && \$b	&&
اگر حداقل یکی از دو متغیر \$a یا \$b درست باشد، برابر درست خواهد بود.	\$a \$b	

تقدم عملگرها

در جدول زیر از بالا به پایین تقدم عملگرها کاهش می‌یابد:

[(
! ~ ++ --
* / %
+ - .
< <= > >=
== != === !==
&
^
&&
?:
= += -= *= /= .= &= = ^= ~=
Xor
or

فصل چهارم

ارتباط با کاربر

فرمها در زبان HTML

همانطور که در کتاب آموزش HTML-CSS-JAVASCRIPT بیان شد، برای ایجاد فرمها از زبان HTML استفاده می‌گردد. در این کتاب نحوه ایجاد فرم در HTML مورد بحث نیست (برای فراگیری آن به کتاب ذکر شده رجوع کنید)، بلکه می‌خواهیم یک فرم HTML را بسازیم و موارد داخل آن را به یک صفحه PHP ارجاع دهیم تا صفحه PHP آنها را مورد پردازش قرار دهد و نهایتاً نتیجه پردازش را در قالب یک فایل HTML بیان کند.

برای شروع کار یک فایل HTML با نام UserPass ایجاد می‌نماییم و کدهای زیر را داخل آن می‌نویسیم:

```

1 <html>
2   <head>
3     <title>Imany Brothers</title>
4   </head>
5   <body>
6     <h1>User Name and Password</h1>
7     <form action = "UserPass.php" method="post">
8       Username: <input type="text" name="username"><br>
9       Password: <input type="password" name="password">
10    <br><br>
11    <input type="submit">
12  </form>
13 </body>
14 </html>

```

در ادامه یک فایل PHP با نام UserPass ایجاد نموده و کدهای زیر را در آن می‌نویسیم:

```

1 <html>
2   <head>
3     <title>Imany Brothers</title>
4   </head>
5   <body>
6     <?php
7     echo ("Your username is : " . $_POST["username"] . "<br>");
8     echo ("Your password is : " . $_POST["password"]);
9     ?>
10  </body>
11 </html>

```

نکته بسیار مهم: هر دو فایل UserPass.php و UserPass.html باید در داخل فولدر www باشند. چرا که برای صحیح اجرا شدن برنامه باید سرور محلی PHP روی سیستم شما نصب شده باشد.

حال برای تست فایل‌های مربوطه باید Firefox یا Internet Explorer خود را باز نموده و عبارت Localhost/UserPass.html را در نوار ابزار آن وارد نمایید. کلمه کاربری و رمز عبور را وارد نموده و کلید Submit Query را کلیک کنید:

User Name and Password

Username:

Password:

Your username is : ImanyBrothers
Your password is : im-br.ir

تمرین) برنامه ای بنویسید که کلمه کاربری و رمز عبور را درخواست نماید. اگر کلمه کاربری برابر با ImanyBrothers و رمز عبور برابر با im-br.ir بود پیغام صحیح بودن آنها را نمایش داده و در غیر اینصورت با نمایش پیغام خطا، مجددا درخواست رمز عبور نماید. کلیه موارد برنامه نویسی داخل یک فایل php با عنوان Uptest.php نوشته شود.

پاسخ) فایل Uptest.php را داخل فولدر www ایجاد کرده و با وارد کردن Localhost/Uptest.php در Browser خود برنامه را تست خواهیم کرد.

```

1 <html>
2   <head>
3     <title>Welcome to ImanyBrothers website</title>
4   </head>
5   <body>
6     <?php
7       $isok=0;
8       if (isset($_POST["username"])) {
9         echo("username is : " . $_POST["username"]);
10        echo(" and password is : " . $_POST["password"] . "<br>");
11        if ($_POST["username"]=="ImanyBrothers" && $_POST["password"]=="im-br.ir") {
12          echo("Your username & password is Correct");
13          $isok=1;
14        }
15      }
16      else
17        echo("Your username & password is Incorrect");
18      }
19      if ($isok==0) {
20      ?>
21      <h1>User Name and password</h1>
22      <form action=" <?php echo($_SERVER["PHP_SELF"]) ?> " method="post">
23      Username : <input type="text" name="username"> <br>
24      password : <input type="password" name="password">
25      <br><br>
26      <input type="submit">
27      </form>
28      <?php
29      }
30      ?>
31 </body>
</html>

```

فصل پنجم

کار با آرایه ها

آرایه های دو بعدی

به شکل زیر توجه کنید که یک آرایه دو بعدی است:

		۰	۱	۲	۳
a2	۰	۸	۹۰	۷۸	۴۵
	۱	۷۴	۱۹	۲۱	۶۴
	۲	۶۹	۳۸	۴	۹۱

اینجا یک آرایه دو بعدی با نام a2 داریم که به عنوان مثال مقدار خانه a2[1][2] برابر ۲۱ میباشد. نحوه مقدار دهی و خواندن مقادیر آرایه دو بعدی نیز مانند آرایه تک بعدی بوده که در گذشته به آن پرداختیم.

در بعضی اوقات میتوانیم به جای عدد از یک رشته نیز برای نامگذاری آرایه استفاده کنیم.

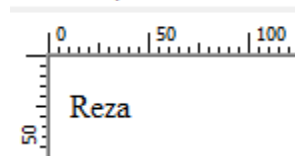
مثال) برنامه ای بنویسید با توجه به روز فراخوانی شده، نام فردی که در آن روز کار می کند را نشان دهد.

پاسخ:

```

1 <?php
2 $names["saturday"]="Ali";
3 $names["friday"]="Reza";
4 $names["sunday"]="Nima";
5 echo($names["friday"]);
6 ?>
```

Generate preview file



همانطور که ملاحظه می فرمایید نحوه فراخوانی یک آرایه توسط نام صورت گرفته که در این مثال خروجی برابر Reza می باشد.

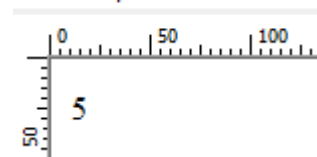
ادغام آرایه ها

به مثال زیر توجه کنید. دو آرایه به نامهای names2 و names تعریف شده و مقادیر آنها در آرایه newNames قرار گرفته است. سپس با تابع count تعداد خانه های آرایه newNames به نمایش درآمده است:

```

1 <?php
2 $names["saturday"]="Ali";
3 $names["friday"]="Reza";
4 $names["sunday"]="Nima";
5 $names2["monday"]="Amir";
6 $names2["tuesday"]="bijan";
7 $newNames = array_merge($names, $names2);
8 echo(count($newNames));
9 ?>
```

Generate preview file



Foreach

با استفاده از این دستور میتوان به تمام عناصر یک آرایه دست یافت. به مثال زیر دقت کنید:

```

1 <?php
2 $names["saturday"]="Ali";
3 $names["friday"]="Reza";
4 $names["sunday"]="Nima";
5 foreach($names as $name)
6     echo ("{$name <br>");
7 ?>

```



Generate preview file

مرتب سازی آرایه

در PHP به روشهای مختلفی میتوان آرایه ها را مرتب نمود.

- الف) با استفاده از تابع `sort()` عمل مرتب سازی **صعودی** به صورت فیزیکی در داخل آرایه صورت می پذیرد.
 - ب) با استفاده از تابع `rsort()` عمل مرتب سازی **نزولی** به صورت فیزیکی در داخل آرایه صورت می پذیرد.
 - ج) با استفاده از تابع `asort()` عمل مرتب سازی **صعودی** به صورت صعودی صرفاً در خروجی نشان داده میشود.
 - د) با استفاده از تابع `asort()` عمل مرتب سازی **صعودی** به صورت نزولی صرفاً در خروجی نشان داده میشود.
 - ه) با استفاده از تابع `suffle()` عمل جابجایی تصادفی را میتوان در مورد اعضای یک آرایه انجام داد.
- به یک نمونه از کاربرد توابع بالا دقت کنید.

```

1 <?php
2 $a[0]=10;
3 $a[1]=9;
4 $a[2]=8;
5 $a[3]=23;
6     sort($a);
7     foreach($a as $i)
8         echo($i . "<br>");
9 ?>

```



Generate preview file

نکته: در HTML یک فیلد از نوع `TextArea` وجود دارد که برای ارسال محتویات آن به PHP باید از آرایه استفاده نمود.

implode

با استفاده از این تابع میتوان محتوای یک آرایه را به یک رشته تبدیل نمود.

explode

با استفاده از این تابع میتوان محتوای یک رشته را داخل آرایه جای داد.

فصل ششم

پیاده سازی توابع

تابع چیست؟

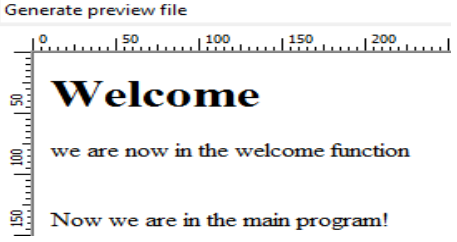
از توابع زمانی استفاده می‌شود که می‌خواهیم یک دستورالعمل خاص را در نقاط مختلفی از برنامه تکرار کنیم. به عنوان مثال می‌خواهیم تابعی بنویسیم که پیامی خاص را در جاهایی که فراخوانی می‌شود نمایش دهد. به تابع زیر دقت کنید:

```

1 <?php
2 welcome(); فراخوانی تابع
3 echo "<br>Now we are in the main program!";
4 function welcome( )
5 {
6     echo "<h1>Welcome</h1>";
7     echo "we are now in the welcome function";
8     echo "<br><br>";
9 }
10 ?>

```

نحوه نوشتن تابع

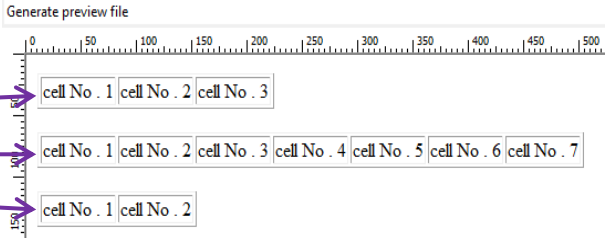


مثال) تابعی بنویسید که با دادن مقدار به آن یک جدول ترسیم نماید که تعداد خانه های آن به تعداد عدد وارده به تابع بوده و شماره هر سلول را داخل آن به همراه پیام مناسب نمایش دهد.

```

1 <?php
2 table(3);
3 table(7);
4 table(2);
5 function table($columns)
6 {
7     echo "<table border=1><tr>";
8     for ($i=0; $i<$columns; $i++)
9         echo "<td>cell No . " . ($i+1) . "</td>";
10    echo "</tr></table><br>";
11 }
12 ?>

```

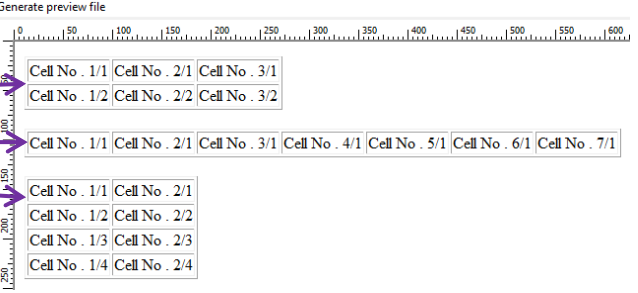


مثال) تابعی بنویسید که با دادن دو مقدار به آن جدولی ترسیم نماید که عدد اول وارده به تابع تعداد ستون و عدد دوم تعداد ردیف باشد. همچنین مختصات هر سلول را داخل آن به همراه پیام مناسب نمایش دهد.

```

1 <?php
2 table(3,2);
3 table(7,1);
4 table(2,4);
5 function table($columns,$rows)
6 {
7     echo "<table border=1>";
8     for ($j=0; $j<$rows; $j++){
9         echo "<tr>";
10        for ($i=0; $i<$columns; $i++)
11            echo "<td>Cell No . " . ($i+1) . "/" . ($j+1) . "</td>";
12        echo "</tr>";
13    }
14    echo "</tr></table><br>";
15 }
16 ?>

```



نکته مهم: متغیرهای داخل تابع کاملا مستقل از بیرون آن می‌باشند. به مثال زیر توجه کنید:

```

1  <?php
2  $test=10;
3  echo ("Before function, test is $test<br>");
4  change( );
5  echo ("After function, test is $test<br>");
6  function change( )
7  {
8      $test=5;
9      echo ("In function, test is $test<br>");
10 }
11 ?>

```

Generate preview file

0 50 100 150

Before function, test is 10
In function, test is 5
After function, test is 10

همانطور که ملاحظه می‌کنید متغیر test دارای مقدار ۱۰ بوده و نمایش داده می‌شود. با فراخوانی تابع مقدار آن به ۵ تغییر می‌یابد. سپس با فراخوانی مجدد آن در خارج تابع باز مقدار ۱۰ نمایش داده می‌شود.

نکته مهم: اگر متغیری را بیرون تابع تعریف و مقدار دهی کنید. سپس در داخل تابع روی آن متغیر عملیات ریاضی انجام دهید، باز هم آن نتیجه نهایی عملیات صرفا داخل تابع بوده و خارج تابع متغیر مقدار اولیه خود را حفظ خواهد نمود. به مثال زیر دقت فرمایید:

```

1  <?php
2  $test=10;
3  echo ("Before function, test is $test<br>");
4  change($test);
5  echo ("After function, test is $test<br>");
6  function change($test )
7  {
8      $test*=2;
9      echo ("In function, test is $test<br>");
10 }
11 ?>

```

Generate preview file

0 50 100 150

Before function, test is 10
In function, test is 20
After function, test is 10

نکته مهم: حال اگر خواستیم تابع مقدار متغیر را بصورت نهایی تغییر دهد و خارج از تابع نیز مقدار آن تغییر نماید باید هنگام فراخوانی آن در تابع از علامت & نیز استفاده نماییم. به مثال زیر توجه کنید:

```

1  <?php
2  $test=10;
3  echo ("Before function, test is $test<br>");
4  change($test);
5  echo ("After function, test is $test<br>");
6  function change(&$test )
7  {
8      $test*=2;
9      echo ("In function, test is $test<br>");
10 }
11 ?>

```

Generate preview file

0 50 100 150

Before function, test is 10
In function, test is 20
After function, test is 20

نحوه تعریف مقادیر پیش فرض برای تابع

بعضی اوقات لازم است برای تابع یکسری ورودی پیش فرض تعیین کرد. این کار زمانی مفید است که اگر ورودیهای تابع مشخص نگردند، مقادیر پیش فرض به عنوان ورودی تابع در نظر گرفته شوند:

```

1 <?php
2 ShowIt( );
3 ShowIt("Eniac Multimedia",3);
4 function showIt($text="welcome",$size=1)
5 {
6     echo("<h$size>$text</h$size><br>");
7 }
8 ?>

```

Generate preview file

نحوه تعریف تابع بصورت بازگشتی

اگر خواستید تابع یک مقدار را بصورت خروجی بازگرداند باید بصورت زیر عمل کرد:
 مثال) تابعی بنویسید که مقداری را دریافت و فاکتوریل آن را بصورت خروجی بازگرداند.

```

1 <?php
2 $k=fact(3);
3 echo("Factorial of 3 is " . $k);
4 echo("<br>Factorial of 5 is " . fact(5));
5 function fact($num)
6 {
7     $ans=1;
8     for ($i=2; $i<=$num; $i++)
9         $ans*=$i;
10    return $ans;
11 }
12 ?>

```

Generate preview file

نحوه استفاده از مقدار یک متغیر برنامه در داخل تابع

در بعضی اوقات نیاز است از مقدار یک متغیر که در بدنه اصلی برنامه موجود است، داخل تابع استفاده گردد. بنابراین برای فراخوانی آن متغیر در داخل تابع بصورتیکه مقدار خود را بیاورد باید از عبارت global استفاده کرد. به مثال زیر توجه کنید:

```

1 <?php
2 $a=10;
3 $b=5;
4 echo("Mohit=" . mohit( ) );
5 echo("<br>Masahat=" . masahat( ) );
6 function mohit ( )
7 {
8     global $a;
9     global $b;
10    return $a*2+$b*2;
11 }
12 function masahat( )
13 {
14     global $a,$b;
15     return $a*$b;
16 }
17 ?>

```

Generate preview file

فصل هفتم

کار با فایلها

بازکردن فایل

برای آنکه بتوان با یک فایل کار کرد ابتدا باید آن را با تابع `fopen()` باز کرد. این تابع دارای دو ورودی می‌باشد. ورودی اول نام فایل و ورودی دوم حالت دسترسی به فایل است. در جدول زیر حالت‌های مختلف دسترسی به فایل را ملاحظه می‌فرمایید:

عبارت	توضیح
r	دسترسی فقط خواندنی
r+	دسترسی خواندن و نوشتن
w	دسترسی فقط نوشتنی، در ابتدا تمامی محتویات فایل پاک می‌شود.
w+	دسترسی خواندن و نوشتن، در ابتدا تمامی محتویات فایل پاک می‌شود.
a	نوشتن در انتهای فایل
a+	خواندن و نوشتن در انتهای فایل

چند مثال:

برای مثال دستور زیر فایل `1.txt` را از شاخه اصلی سرویس دهنده باز می‌کند و می‌توانید اطلاعات داخل فایل را بخوانید.

```
$fp1 = fopen("1.txt", "r");
```

با استفاده از دستور زیر فایل `test.txt` در شاخه `temp` واقع در شاخه `wwwroot` سرویس دهنده ایجاد شده و دسترسی نوشتن به آن را نیز خواهید داشت.

```
$fp2 = fopen("temp/test.txt", "w");
```

با اجرای دستور زیر فایل `price.txt` با استفاده از یک اتصال `ftp` از سایت `eniamedia` بصورت خواندنی باز می‌شود.

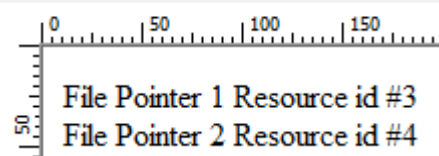
```
$fp3 = fopen("ftp://eniamedia.com/price.txt", "r");
```

بعد از اتمام کار با فایل باید با دستور `close` آن را ببندید.

مثال) دو فایل با نامهای `1.php` که حاوی عبارت `This is 1.php file` و `2.php` که حاوی عبارت `This is 2.php` میباشد را ایجاد نمایید. سپس در همان مسیر یک فایل دیگر با عنوان `read.php` ایجاد نمایید و کدهای زیر را در آن وارد نمایید.

```
1 <?php
2 $fp1=fopen("1.php", "r");
3 $fp2=fopen("2.php", "r");
4 echo ("File Pointer 1 $fp1<br>");
5 echo ("File Pointer 2 $fp2<br>");
6 fclose($fp1);
7 fclose($fp2);
8 ?>
```

Generate preview file



همانطور که ملاحظه می‌فرمایید در متغیرهای `$fp1` و `$fp2` دو اشاره گر ذخیره شده است. اشاره گر به فایل `1.php` دارای id به شماره ۳ و اشاره گر به فایل `2.php` دارای id به شماره ۴ می‌باشد.

خواندن فایل

برای خواندن یک سطر از فایل از تابع fgets استفاده می‌کنیم. به دستور زیر دقت کنید:

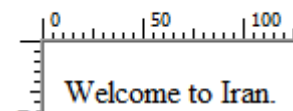
```
$c=fgetc($fp1)
```

این دستور اولین سطر از فایلی که توسط \$fp1 به آن اشاره شده است را خوانده و در متغیر \$c قرار می‌دهد. پس نتیجه می‌گیریم که اگر بخواهیم تمام فایل را بخوانیم، باید از یک حلقه استفاده نماییم.

مثال) یک فایل با نام 3.php ایجاد نمایید و در آن چند سطر متن را به دلخواه بنویسید. اگر بخواهید فقط سطر اول آن را نمایش دهید باید کد زیر را وارد نمایید:

```
1 <?php
2 $fp3=fopen("3.php","r");
3 $s=fgets($fp3);
4 echo($s);
5 fclose($fp3);
6 ?>
```

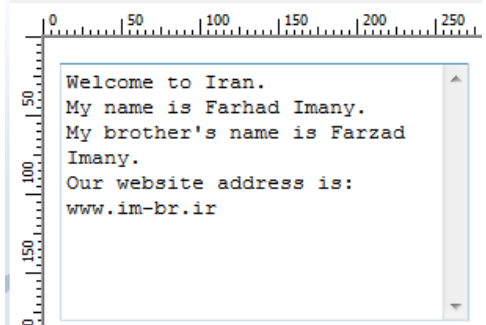
Generate preview file



حال اگر بخواهیم تمام فایل را بخوانیم، ابتدا باید یک text area را بوجود آورده و محتویات فایل را با استفاده از حلقه while داخل آن نمایش دهیم:

```
1 <html>
2 <body>
3 <?php
4     $fp3=fopen("3.php","r");
5     echo("<form> <textarea cols=30 rows=10>");
6     while(!feof($fp3))
7     {
8         $s=fgets($fp3);
9         echo($s);
10    }
11    fclose($fp3);
12 ?>
13 </textarea>
14 </form>
15 </body>
16 </html>
```

Generate preview file



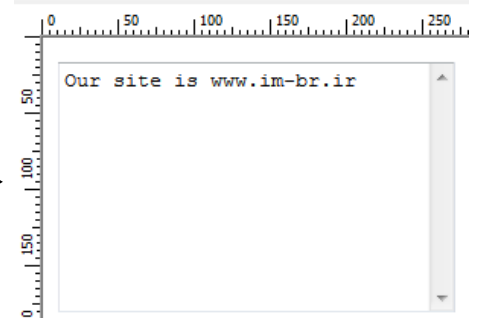
نوشتن در فایل

برای نوشتن در داخل فایل از تابع fputs استفاده می‌شود. به مثال زیر دقت کنید:

```
<html>
<body>
<?php
    $fp1=fopen("1.php","w");
    fputs($fp1,"Our site is www.im-br.ir");
    fclose($fp1);

    $fp1=fopen("1.php","r");
    echo("<form> <textarea cols=30 rows=10>");
    while (!feof($fp1))
    {
        $s=fgets($fp1);
        echo ($s);
    }
    fclose($fp1);
?>
</textarea>
</form>
</body>
</html>
```

Generate preview file

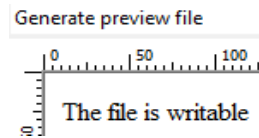


تابع `is_writable()`

این تابع بررسی می‌کند آیا یک فایل قابل نوشتن می‌باشد یا نه. به مثال زیر دقت کنید

```

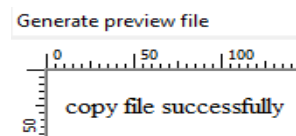
1 <?php
2 $filename = '1.php';
3 if (is_writable($filename)) {
4     echo 'The file is writable';
5 } else {
6     echo 'The file is not writable';
7 }
8 ?>
```

تابع `copy()`

وظیفه این تابع کپی یک فایل موجود، با نامی جدید است. به مثال زیر دقت کنید:

```

1 <?php
2 $file = '1.php';
3 $newfile = '2.php';
4 if (!copy($file, $newfile)) {
5     echo "failed to copy $file...\n";
6 }else{
7     echo "copy file successfully";
8 }
9 ?>
```

تابع `rename()`

وظیفه این تابع تغییر نام یک فایل موجود است. به مثال توجه فرمایید:

```

1 <?php
2 rename("2.php", "3.php")
3 ?>
```

تابع `filesize()`

وظیفه این تابع نمایش سایز یک فایل بر حسب بایت است. به مثال توجه فرمایید:

```

1 <?php
2 echo filesize("3.php");
3 ?>
```

تابع `isfile()`

با این تابع می‌توانید بررسی نمایید آیا یک فایل وجود دارد یا نه؟ به مثال زیر دقت کنید:

```

1 <?php
2 if(is_file("2.php")){
3     echo "File is exists";
4 }else{
5     echo "File not exists";}
6 ?>
```

طرز خواندن فایل‌های یک پوشه و نمایش فایل‌های داخل آن

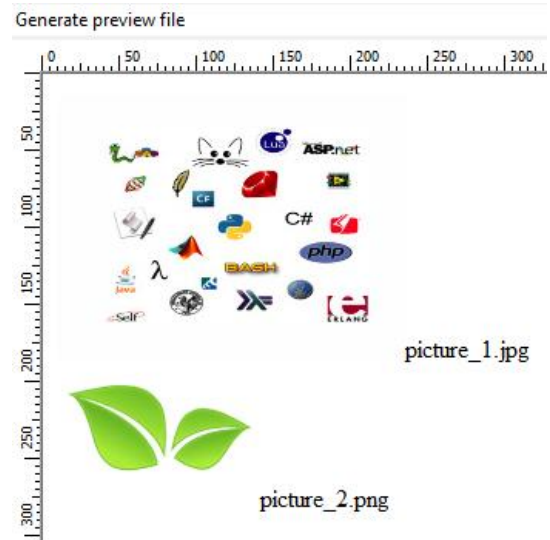
یک پوشه با نام pic ایجاد نمایید و چند تصویر را در آن ذخیره نمایید. سپس دستورات زیر را نوشته و نتیجه را ملاحظه نمایید.

در این مثال یک پوشه با نام pic ایجاد شده و دو فایل picture_1.jpg و picture_2.png داخل آن ایجاد گشته است. حال خارج از پوشه یک فایل با نام read.php ایجاد کرده و دستورات زیر را در آن نوشته ایم. به خروجی دقت کنید:

```

1 <html>
2   <body>
3     <?php
4       $df1 = opendir("pic");
5       readdir($df1);
6       readdir($df1);
7       $picname=readdir($df1);
8       while( $picname!="") {
9         echo("<img src=\"pic\\\" . $picname . \">");
10        echo($picname . "<br>");
11        $picname=readdir($df1);
12      }
13      closedir($df1);
14    ?>
15  </body>
16 </html>

```



انتمام دوره آموزش PHP

موفق باشید

WWW.IM-BR.IR