

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 می باشد. اکثر جزوات آموزشی که در مورد زبان PHP وجود دارد دارای مطالب ناقص و پراکنده می باشند. 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 راهنماییهای 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.Ir

Fazhad.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 نمی باشد

فصل اول

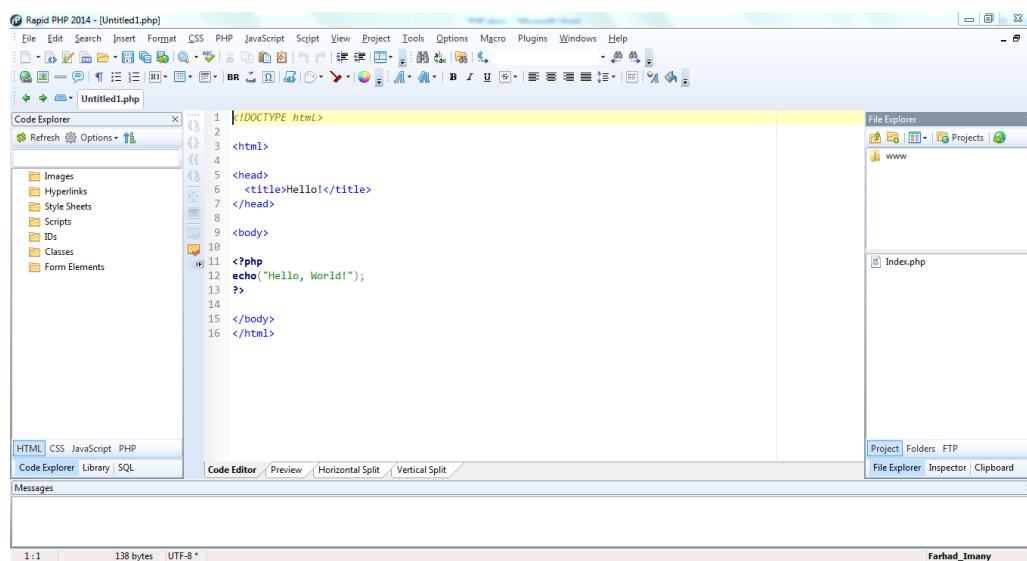
بررسی مفاهیم اولیه در برنامه نویسی PHP

آشنایی با اصول اولیه

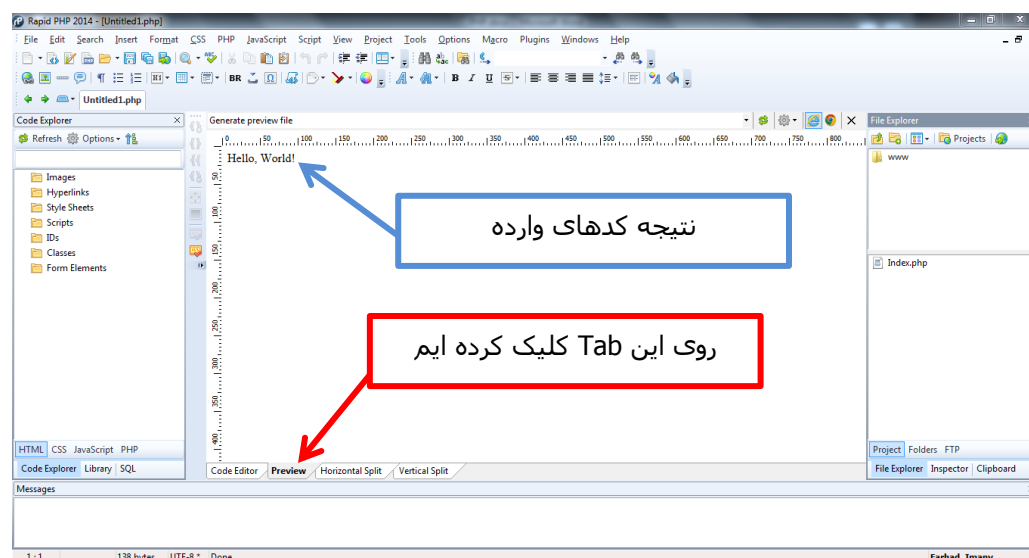
در این کتاب آموزشی قصد داریم شما عزیزان را با تمام موارد مربوط به برنامه نویسی توسط زبان بسیار قدرتمند PHP آشنا سازیم. امید است با مطالعه گام به گام فصول به توانایی مورد انتظار در این زمینه برسید.

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:

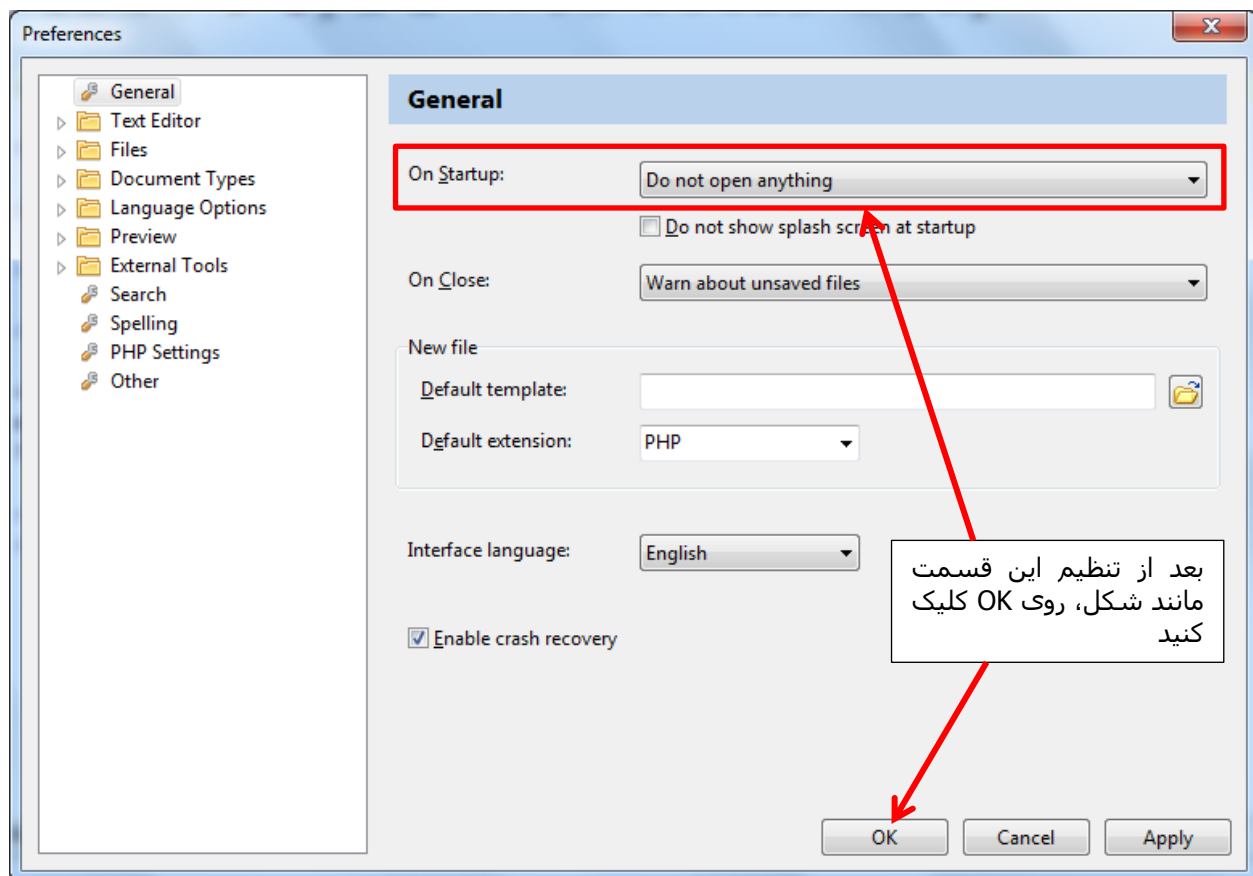


همانطور که ملاحظه می‌فرمایید بصورت پیش فرض یک فایل php بصورت اتوماتیک ایجاد شده و کدهایی ابتدایی در آن موجود است. کافیه برای دیدن نتیجه کدهای مربوطه روی Tab با عنوان Preview کلیک کنید. اگر خواستید نتیجه را در Browser ببینید از کلید F2 استفاده نمایید.



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:

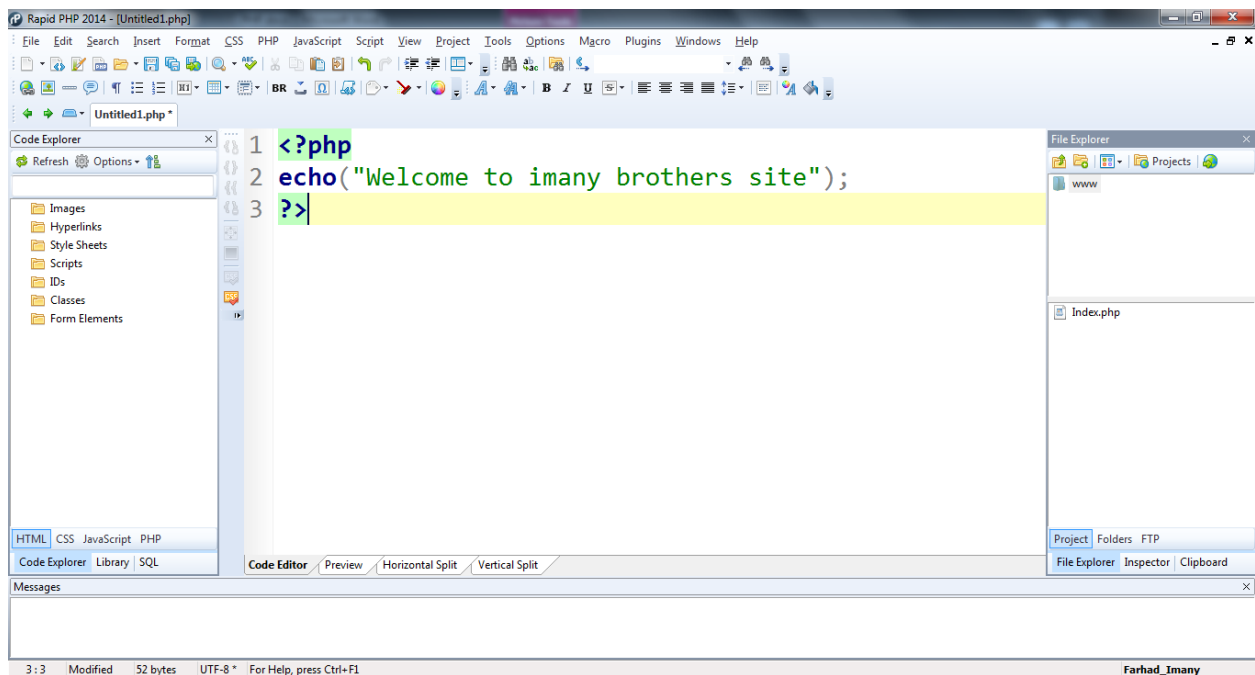
<script language="php">

محل نوشتن کدهای PHP

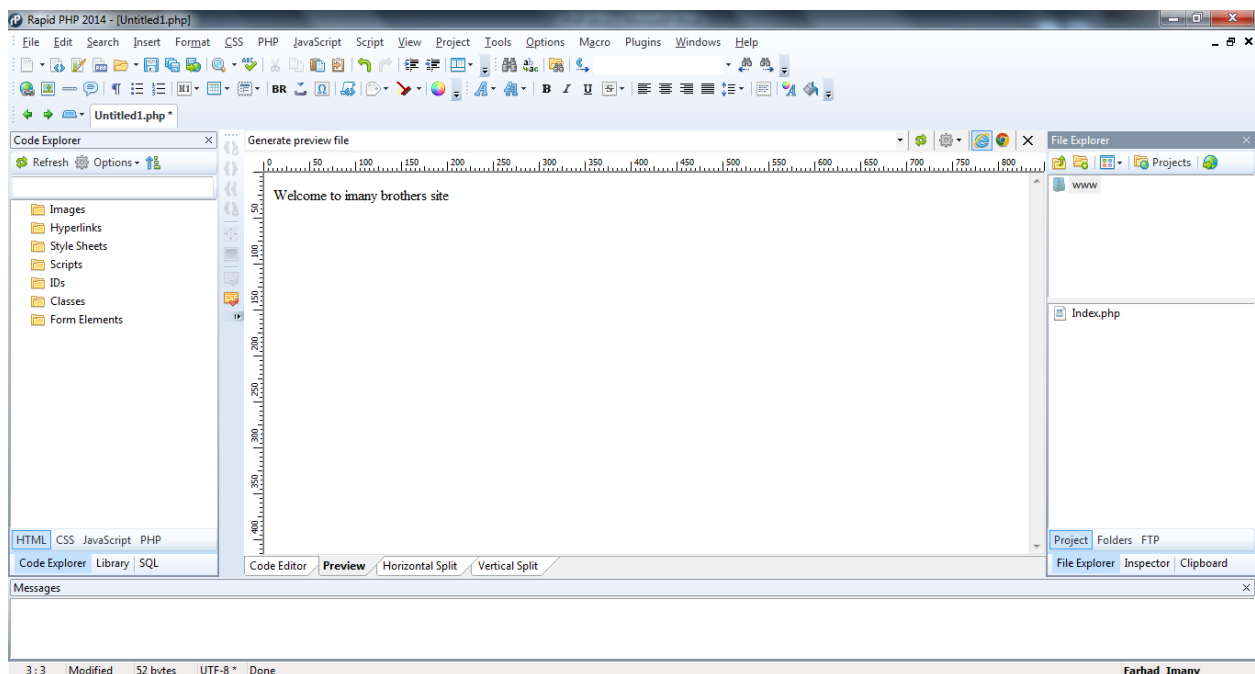
</script>

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.

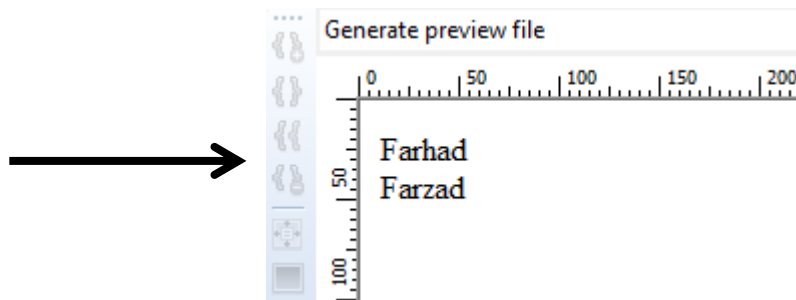
یا برای از دور خارج کردن یکسری از دستوراتی که پشت سرهم می‌باشند ابتدای اولین دستور مورد نظر علامت /* را قرار داده و سپس به انتهای آخرین دستور مورد نظر رفته و علامت */ را قرار دهید.

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 ?>
```

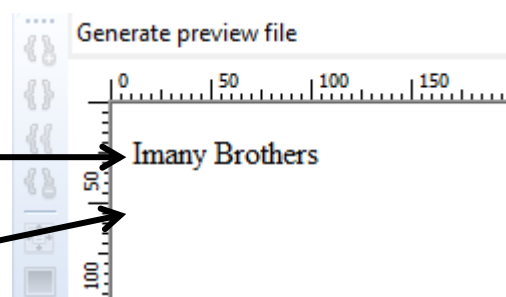


نحوه حذف مقدار متغیر

دو روش برای انجام اینکار وجود دارد:

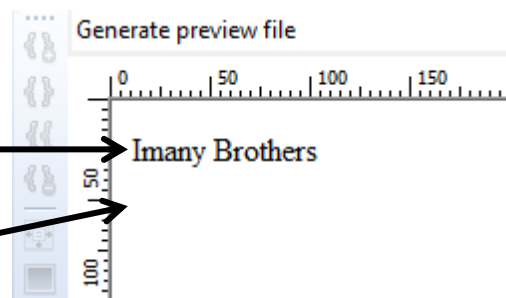
روش اول: استفاده از NULL

```
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 ?>
```



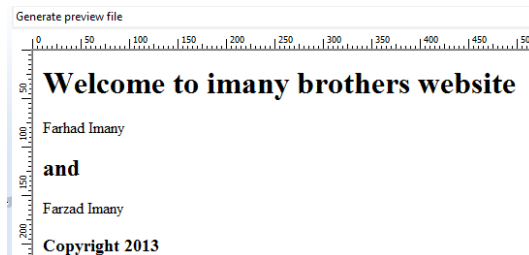
ترتیب کدهای HTML و PHP

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>

```



نوشتن کد HTML قبل از PHP

1

نوشتن کد HTML مابین PHP

2

نوشتن کد HTML بعد از PHP

3

نحوه تعریف ثابتها

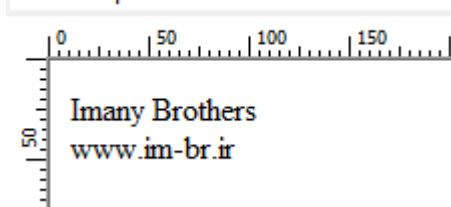
ثابتها متغیرهایی هستند که یک مقدار اولیه به آنها تعلق گرفته و در طول برنامه این مقدار را حفظ می‌کنند. برای تعریف ثابت از کلمه کلیدی define استفاده می‌کنیم. همچنین به خاطر داشته باشید که برای تعریف متغیری از نوع ثابت نیازی به علامت \$ نداریم.

```

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

```

Generate preview file



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
```

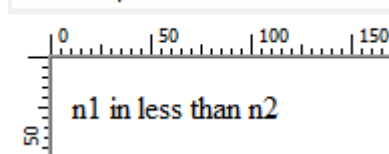
```
}
```

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 ۱ 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 ; ?>

```



Generate preview file

Name	Family
Farzad	Imany

در این برنامه $n1=2$ است

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.

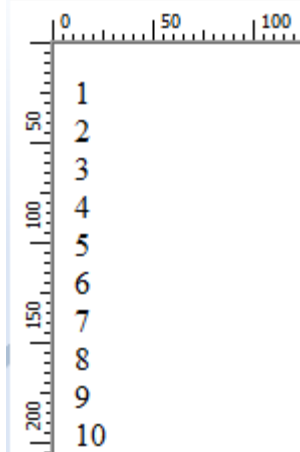
```

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

```



Generate preview file



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; ?>

```



Generate preview file

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.

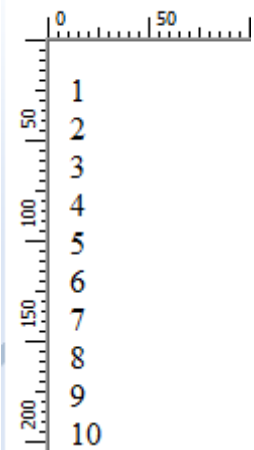
```

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.

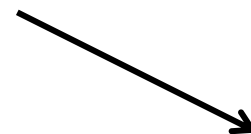
مثال) یک حلقه بی نهایت تشکیل دهید که عبارت Welcome to imany brothers website را سه بار تکرار کند.

پاسخ:

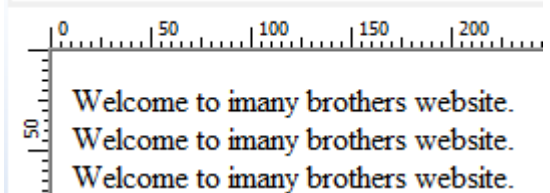
```

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

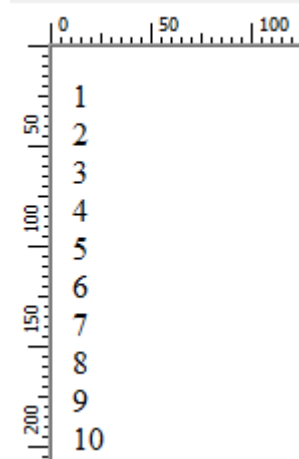
از این دستور نیز برای ایجاد حلقه استفاده می‌شود با این تفاوت که شمارنده آن داخل خود می‌باشد و مانند حلقه While نیازی نیست که ما شمارنده‌ای مانند (\$n1 ++;) ایجاد نمائیم. از این دستور زمانی استفاده نمایید که تعداد دفعات اجرا مشخص می‌باشد.

```

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.

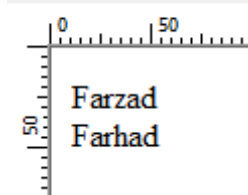
پاسخ:

```

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

```

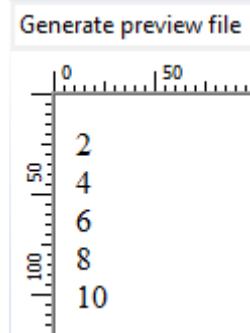
Generate preview file



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:

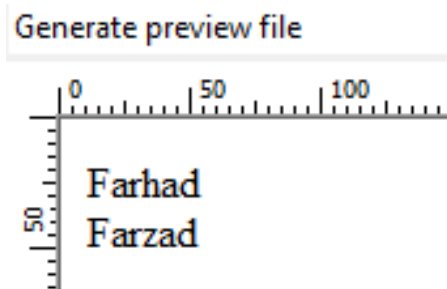
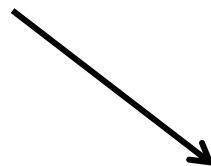
```
1 <?php
2 $imbr = array();
3 for ($i=1; $i<=10; $i++)
4     $imbr[$i]=$i;
5 foreach ($imbr as $value)
6 {
7     if ($value%2==0)
8         echo "$value". "<br/>";
9
10 }
11 ?>
```



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.

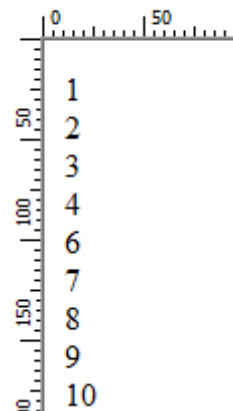
پاسخ:

```

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.

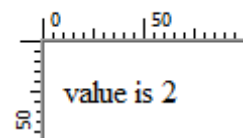
پاسخ:

```

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	+	<code>\$a + \$b</code>	جمع <code>\$a</code> و <code>\$b</code>
تفریق	-	<code>\$a - \$b</code>	کسر <code>\$b</code> از <code>\$a</code>
ضرب	*	<code>\$a * \$b</code>	ضرب <code>\$a</code> در <code>\$b</code>
تقسیم	/	<code>\$a / \$b</code>	تقسیم <code>\$a</code> بر <code>\$b</code>
باقیمانده	%	<code>\$a % \$b</code>	باقیمانده تقسیم <code>\$a</code> بر <code>\$b</code>

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 = 1.6666666666667
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:

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

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

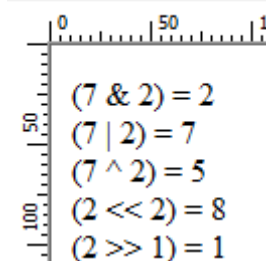
```

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



عملگرهای مقایسه های

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

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

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.

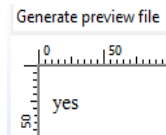
(مثال)

```

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:

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

عملگرهای منطقی

این عملگرها بصورت زیر می‌باشند:

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

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>
```

Generate preview file

User Name and Password

Username:

Password:

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>
```

Generate preview file

Your username is :
Your password is :

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.