

کار با متغیرها، عملگرها و ساختارهای کنترلی

واحد کارششم

پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:

- 1 - تگ‌های آغاز و پایان PHP را بشناسد.
- 2 - اصول نحوه نوشتن دستورات در PHP را بشناسد.
- 3 - بتواند کدهای HTML و PHP را در کنار یکدیگر استفاده کند.
- 4 - با نحوه درج توضیحات درون کدهای PHP آشنا باشد.
- 5 - انواع متغیرهای PHP را بشناسد.
- 6 - بتواند متغیرهای از پیش تعریف شده را در برنامه به کار ببرد.
- 7 - با ثابت‌ها و عملکرد آن‌ها آشنا باشد.
- 8 - با اصول نامگذاری متغیرها آشنا باشد.
- 9 - انواع عملگرهای PHP را بشناسد.
- 10 - اصول کار با انواع ساختارهای کنترلی را بداند.

زمان (ساعت)	
نظری	عملی
6	12

مقدمه

برای ایجاد فایل‌های PHP نیازی نیست حتماً برنامه خاصی روی سیستم نصب شود و کدهای PHP را میتوان در یک ویرایشگر ساده متنی مانند Notepad نوشت. نرم‌افزار Dreamweaver نیز می‌تواند فایل‌های PHP را پشتیبانی کرده و محیط مناسبی برای کدنویسی فراهم کند.

1 - 6 شروع برنامه‌نویسی در PHP

برای نوشتن دستورات PHP دو روش اصلی وجود دارد که در ادامه به توضیح آن‌ها خواهیم پرداخت. شما به صورت دلخواه می‌توانید هر کدام از این ساختارها را در برنامه، مورد استفاده قرار دهید.

• روش اول: دستورات و کدهای PHP درون علائم `<? ... ?>` قرار می‌گیرند.

`<?`

دستورات و کدهای PHP

`?>`

• روش دوم: دستورات و کدهای PHP درون علائم `<?php ... ?>` قرار می‌گیرند.

`<?php`

دستورات و کدهای PHP

`?>`

توصیه: به منظور تسهیل در کدنویسی و حداکثر سازگاری کدها در مرورگرهای مختلف، توصیه می‌شود برای نوشتن کدهای PHP از روش دوم استفاده نمایید.

برای ایجاد یک فایل اجرایی بسیار ساده در PHP متن `hello` را در مرورگر نمایش می‌دهیم. در انتهای هر خط از دستورات نیز باید از علامت «؛» استفاده شود. برای نمایش اطلاعات در مرورگر، دستورات مختلفی به‌کار می‌رود، دستور `Print` یکی از پرکاربردترین آن‌هاست. کد زیر میتواند متن `hello` را در مرورگر نمایش دهد.

```
<? PHP
```

```
Print ("hello!");
```

```
?>
```

این کد را در Notepad بنویسید، فایل را با پسوند PHP در مسیر نصب WAMP و در شاخه www ذخیره کنید.

سپس مرورگر را باز کرده و در نوار آدرس این متن را تایپ کنید: (به جای نام فایل، عنوانی را که برای ذخیره فایل وارد کردید درج نمایید).

http://localhost/نام.فایل.php



شکل 6-1

نکته اگر این فایل را با نام index.php یا default.php ذخیره کنید، لازم نیست در نوار آدرس نام فایل را ذکر نمایید، فقط کافی است عبارت http://localhost را تایپ کرده و کلید Enter را فشار دهید.

در تمام سایت‌ها، صفحه اصلی (home page) سایت را با نام index یا default ذخیره میکنند، هنگام درج آدرس یک سایت، همیشه صفحه‌ای که در مسیر اصلی سایت با نام index یا default باشد، به طور پیش فرض برای اجرا انتخاب میشود، سایر صفحات از طریق همین صفحه با یکدیگر مرتبط میشوند.

دستور اصلی برای چاپ اطلاعات در خروجی هر زبان PHP، دستور echo است. برنامه قبل را با این دستور می‌توانید به این صورت بنویسید:

```
<? Php
```

```
echo ("hello!");
```

```
?>
```

اگر بخواهید چند عبارت را توسط یک دستور echo چاپ کنید، باید هر عبارت را با
" از عبارت بعدی جدا کنید. مانند (echo ("hello!" "welcome")
حال اگر بخواهید هر کدام از عبارات در یک سطر جدا چاپ شود باید بین هر عبارت
از "
" استفاده کنید.

```
echo ("hello!" . "<br>" . "welcome")
```

مثال:

1-1-6 استفاده از کدهای PHP و HTML در کنار یکدیگر

یکی از قابلیت‌های زبان PHP این است که میتواند درون کدهای HTML به کار برده شود.

```
<html>
```

```
<head>
```

```
<title>first program</title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
echo("hello!");
```

```
?>
```

```
</body>
```

```
</html>
```

همانطور که ملاحظه می‌کنید، برای درج کد PHP درون HTML فقط کافی است آن را
درون تگ <body> ... </body> بنویسید، اما پس از نوشتن کدها، باید فایل با پسوند php.
ذخیره شود و در صورت ذخیره فایل با پسوند .htm یا .html، کدهای PHP اجرا نخواهند شد.

2-1-6 درج توضیحات درون کدهای PHP

توضیحات در زبان‌های برنامه‌نویسی به منظور راهنمایی برای برنامه‌نویس یا فردی

است که میخواهد از کدها استفاده کند، در PHP نیز مانند سایر زبانهای برنامه‌نویسی دستورالعمل‌هایی برای درج توضیحات وجود دارد. در PHP برای توضیحات یک خطی از علائم // و # و برای درج توضیحات چند خطی از علامت /* در ابتدا و از علامت */ در انتهای توضیحات استفاده میشود.

```
<html>
<head>
<title>first program</title>
</head>
<body>

<?php
/* Line number1: These 3 lines will not either!
Line number2
Line number3: End of comment */

echo("hello!");

// This line will not be parsed in PHP

echo("welcome!");

# This line will not be parsed in PHP like the line above

?>

</body>
</html>
```

2 - 6 متغیرها

یکی از مهمترین ابزارهایی که برای تبدیل صفحات ساده و ثابت به صفحات پویا لازم است، این است که سرویس دهنده بتواند اطلاعات را از کاربر سرویس گیرنده دریافت کند و با توجه به اطلاعات پاسخ مناسبی به کاربر بدهد. متغیرها ابزاری هستند که با استفاده از آنها میتوان اطلاعات را از کاربر دریافت کرد. در واقع متغیرها به منزله ظرفهایی هستند که اطلاعات کاربر را به طور موقت ذخیره کرده و در مواقع لزوم آنرا برای سرویس دهنده ارسال میکنند.

برای تعریف متغیر در PHP باید از حرف \$ استفاده کرد.

مثال: \$a;

\$b=10;

نوع متغیر در PHP اهمیتی ندارد و لازم نیست هنگام تعریف متغیر، نوع آن ذکر شود. متغیرهای PHP نسبت به حروف کوچک و بزرگ حساس هستند. به عنوان مثال متغیرهای a و A با یکدیگر متفاوت بوده و دو متغیر جدا محسوب می شوند. در نامگذاری متغیرها علاوه بر حروف کوچک و بزرگ می توان از علائمی مانند "-" یا خط فاصله پایین (Underline) استفاده کرد. استفاده از اعداد نیز در نامگذاری متغیرهای PHP مجاز است در صورتی که اولین حرف از متغیر نباشد.

مثال: متغیرهای با نامهای \$ab_1 و \$abc مجاز هستند.

مثال: متغیرهای با نامهای \$1ab و \$1_a غیرمجاز هستند.

 **نکته** نامگذاری متغیرها به طوری که "-" (Underline) در ابتدای آنها باشد، مجاز است اما توصیه می شود از نامگذاری متغیرها به این شیوه استفاده نکنید و حتی الامکان از "-" در ابتدای نام متغیرها استفاده نکنید.

در متغیرها نشان دادن نوع توسط مقداردهی تعیین می شود. دستورالعمل تعیین نوع برای هر متغیر به صورت زیر است:

- اعداد صحیح به طور ساده نوشته می شوند.

مثال: `$a=15;`

• اعداد اعشاری توسط علامت "." تعیین می شوند.

مثال:

`$b=7.0;`

`$C=1.295;`

• اعداد کسری توسط علامت "/" تعیین می شوند.

مثال: `$f=1/2;`

نکته  اعداد کسری به صورت اعشاری ذخیره می شوند، در مثال فوق متغیر f با 0.5 مقداردهی می شود.

رشته ها درون علامت "" یا '' درج می شوند.

مثال: `$Str = "Hello" ;`

`$H= "2914" ;`

`$M = 'Welcome' ;`

اگر یک عدد بین علامت "" قرار گیرد، مقدار رشته ای محسوب می شود.

مثال: می خواهیم ابتدا دو متغیر را با اعداد 35 و 79 مقداردهی کنیم، سپس در خروجی حاصل جمع این دو عدد را چاپ کنیم:

`<html>`

`<head>`

`<title>first program</title>`

`</head>`

`<body>`

`<?php`

`$num1 = 35;`

`$num2 = 79;`

```
$sum = $num1 + $num2;
```

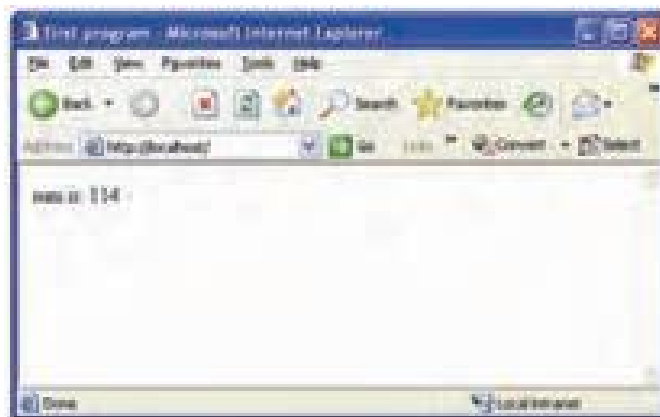
```
echo("sum is: ");
```

```
echo("$sum");
```

```
?>
```

```
</body>
```

```
</html>
```



شکل 6-2

1-2-6 آرایه‌ها

همان‌طور که می‌دانید آرایه‌ها از لحاظ ساختاری پیچیده‌تر از انواع داده‌های دیگر هستند، برخلاف انواع داده‌های رشته‌ای و عددی که فقط می‌توانند یک مقدار را بپذیرند، آرایه‌ها می‌توانند شامل لیستی از مقادیر مختلف باشند.

در زبان PHP آرایه‌ها را می‌توان به دو دسته اندیس‌دار¹ و انجمنی² تقسیم کرد، روش دسترسی به عناصر در این دو روش متفاوت است. در آرایه‌های اندیس‌دار، دسترسی به هریک از عناصر توسط اندیس آن صورت می‌گیرد.

اولین اندیس آرایه نیز مانند زبان جاوا اسکریپت «0» است. ساختار کلی آرایه اندیس‌دار به صورت زیر است:

مثال: "مقدار عنصر آرایه" = [] نام متغیر آرایه‌ای;

```
$name[]="Maryam";
```

```
$name[]="zahra";
```

```
$name[]="Meisam";
```

روش دوم برای معرفی آرایه‌ها استفاده از شناسه array است.

(... و "مقدار عنصر دوم" و "مقدار عنصر اول") array = نام متغیر آرایه‌ای

مثال:

```
$name = array ("Maryam" و "Zahra" و "Meisam" );
```

1- Indexed

2- Association

```
$name [0] = "Maryam"
$name [1] = "Zahra"
$name [2] = "Meisam"
```

The screenshot shows a Microsoft Internet Explorer window titled 'first program'. The address bar contains the URL 'http://www.marysals.com/'. The main content area displays the text 'marysals.com'.

```
$name= array("maryam","zahra","meisam");
```

```
echo("<br>."<br>." $name[0] "<br>." $name[1] "<br>." $name[2] ");
```

```
</body>
</html>
```

یک آرایه عددی با 4 عنصر تعریف کنید، عناصر را با اعداد 9 تا 5 مقداردهی کرده، سپس با استفاده از اندیس آرایه‌ها، آن‌ها را با هم جمع کرده، نتیجه حاصل جمع را در خروجی نمایش دهید.

2-2-6 متغیرهای از پیش تعریف شده

متغیرهای از پیش تعریف شده متغیرهایی هستند که در یکی از برنامه‌های کاربردی سرویس دهنده وب استفاده شده‌اند و برنامه‌نویس مجاز نیست از نام آن‌ها برای متغیرهای خود استفاده کند.

مهم‌ترین متغیرهای از پیش تعریف شده عبارتند از:
OSTYPE: نوع سیستم عاملی را که روی سرویس دهنده در حال اجراست در بر می‌گیرد.
HOSTNAME: نام سرویس دهنده را در بر می‌گیرد.
PHP-SELF: نام صفحه جاری را در بر می‌گیرد.

3-2-6 ثابت‌ها

همان‌گونه که از نام ثابت‌ها پیداست، مقادیری هستند که در طول اجرای برنامه ثابت بوده و مقدار آن‌ها تغییر نمی‌کند.

ثابت‌ها به‌طور سراسری و در همه جای برنامه قابل استفاده و در دسترس هستند. برای تعریف ثابت‌ها از تابع `define` استفاده می‌شود و نیاز به درج علامت `$` ندارند. هنگام استفاده از ثابت‌ها در طول برنامه نیز به علامت `$` نیاز نداشته و تنها نوشتن نام متغیر کافی است.

ساختار کلی تعریف ثابت‌ها به‌صورت زیر است:
مثال:

```
define ("مقدار ثابت","نام ثابت");  
define("PI","3.14");
```

مثال زیر عدد 3,14 را در متغیری به نام `pi` به‌طور ثابت قرار می‌دهد، شعاع دایره را متغیری صحیح و با عدد 5 مقداردهی می‌کند، سپس محیط دایره را محاسبه کرده و در خروجی نشان می‌دهد.

```
<?php
```

```
define ("pi","3.14");  
$r=5;
```

```
$e=pi*2*"$r";
```

```
echo ("environment : ".$e")
```

```
?>
```



شکل 6-4

6 - 3 عملگرها

6 - 3 - 1 عملگرهای محاسباتی

عملگرها در تمام زبان‌ها برای اجرای محاسبات روی داده‌ها به کار می‌روند.

جدول 6-1

مثال	عملگر	
$a+b$	جمع	+
$a-b$	تفریق	-
$a*b$	ضرب	*
a/b	تقسیم	/
$a \% b$	باقیمانده تقسیم	%
$a++$ یا $++a$	افزایش	++
$--a$ یا $a--$	کاهش	--

2-3-6 عملگرهای مقایسه‌ای

جدول 6-2

مثال	عملگر	
$a == b$	تساوی	$==$
$a === b$	تساوی	$===$
$a != b$	نامساوی	$!=$
$a <> b$	نامساوی	$<>$
$a < b$	کوچک‌تر از	$<$
$a > b$	بزرگ‌تر از	$>$
$a <= b$	کوچک‌تر یا مساوی	$<=$
$a >= b$	بزرگ‌تر یا مساوی	$>=$

اگر بخواهید مشخص کنید که آیا دو متغیر a و b با هم مساوی هستند یا خیر، می‌توانید از عملگر $==$ استفاده کنید.

مثال 1:

- $a = 5;$
- $b = 5;$
- $c = 10;$
- $a == b;$
- $b == c;$

در دستور سطر 4 چون a با b برابر هستند مقدار $true$ برگردانده می‌شود. اما در سطر 5 به دلیل این که c و b با هم مساوی نیستند مقدار $false$ برگردانده می‌شود.

مثال 2:

- $a = 3;$
- $b = "3";$
- $a == b;$

در دستور سوم ارزش $true$ برگردانده می‌شود زیرا ابتدا متغیر b نوع رشته‌ای به نوع عددی تبدیل می‌شود و سپس مقایسه صورت می‌گیرد.

اگر بخواهیم تبدیل نوع صورت نگیرد و مقایسه با انواع یکسان صورت گیرد، باید از عملگر `===` استفاده کنیم:

```
$a = 7;
```

```
$b = "7";
```

```
$a === $b;
```

دستور سوم مقدار `false` را برمی گرداند، زیرا `$a` و `$b` هم‌نوع نیستند.

3 - 3 - 6 عملگرهای منطقی

عملگرهای منطقی روی عبارات منطقی عمل می‌کنند. به عنوان مثال اگر بخواهیم تشخیص دهیم مقدار متغیر `$a` بین 1 و 100 قرار دارد باید از عملگرهای منطقی استفاده کنیم. این عملگرها را در جدول زیر مشاهده می‌نمایید:

جدول 6-3

عملگر	نام	مثال	نتیجه
	نقیض (NOT)	<code>\$b</code>	اگر <code>\$b</code> نادرست باشد، نتیجه درست است و برعکس.
&&	و (AND)	<code>\$a && \$b</code>	اگر <code>\$a</code> و <code>\$b</code> هر دو درست باشند نتیجه درست است در غیر این صورت نادرست است.
	یا (OR)	<code>\$a \$b</code>	اگر حداقل یکی از متغیرهای <code>\$a</code> یا <code>\$b</code> درست باشند، نتیجه درست است و در غیر این صورت نادرست است.

4 - 3 - 6 عملگرهای ترکیبی

تعدادی از عملگرها در PHP وجود دارند که ترکیبی از دو عملگر دیگر هستند و به نام عملگرهای ترکیبی خوانده می‌شوند. این عملگرها در جدول زیر توضیح داده شده‌اند:

جدول 6-4

عملگر	نام	مثال	عبارت معادل
<code>+=</code>	انتساب جمع	<code>\$a += \$b</code>	<code>\$a = \$a + \$b</code>
<code>-=</code>	انتساب تفریق	<code>\$a -= \$b</code>	<code>\$a = \$a - \$b</code>
<code>*=</code>	انتساب ضرب	<code>\$a *= \$b</code>	<code>\$a = \$a * \$b</code>
<code>/=</code>	انتساب تقسیم	<code>\$a /= \$b</code>	<code>\$a = \$a / \$b</code>
<code>%=</code>	انتساب باقیمانده تقسیم	<code>\$a %= \$b</code>	<code>\$a = \$a % \$b</code>

5 - 3 - 6 عملگرهای رشته‌ای

یکی از پرستفاده‌ترین عملیاتی که روی رشته‌ها انجام می‌گیرد، عمل الحاق کردن رشته‌هاست. به عمل چسباندن دو رشته به یکدیگر عمل الحاق رشته‌ها گفته می‌شود. به عنوان مثال دو رشته زیر را در نظر بگیرید.

```
$a="web";
```

```
$b="design";
```

برای الحاق رشته‌ها در PHP از عملگر . استفاده می‌شود:

```
$c=$a . $b;
```

به این ترتیب حاصل رشته '\$c'، 'webdesign' خواهد بود. در PHP دو نوع رشته وجود دارد:

رشته‌هایی که بین دو علامت "" قرار می‌گیرند. مثل 'webdesign'

رشته‌هایی که بین دو علامت ' ' قرار می‌گیرند. مثل 'webdesign'

اگر رشته‌ای در نقل قول دوتایی قرار گیرد، PHP سعی می‌کند آن را مثل یک عبارت ارزیابی کند، ولی اگر رشته در نقل قول یکانی قرار گیرد، ارزیابی نمی‌شود. برای مثال به دستورات زیر نگاه کنید:

```
1.$a=10;
```

```
2.echo "value of $a";
```

```
3.echo 'value of $a';
```

دستور سطر اول مقدار 10 را در متغیر \$a قرار می‌دهد. خروجی دستور سطر دوم به‌صورت زیر است:

```
Value of 10
```

در حالی که دستور سطر سوم، به‌صورت زیر است:

```
Value of $a
```

4 - 6 کار با ساختارهای کنترلی

در مبحث جاوا اسکریپت با ساختارهای کنترلی آشنا شدید، در این قسمت ضمن بیان نکات مختصری جهت یادآوری مطالب، به بررسی ساختار هر کدام از دستورات در زبان

PHP می‌پردازیم:

1 - 4 - 6 دستور if

یک عبارت را بررسی می‌کند، در صورت داشتن پاسخ true وارد بلوک کد مربوط به if می‌شود، در غیر این صورت از بلوک if می‌گذرد.
 ساختار دستور if به صورت زیر است:

(عبارت شرطی) if

```
{
```

در پرانتز مقابل دستور if باید یک عبارت شرطی گذاشته شود. دستورات بدنه

```
}
```

مثال:

```
<html>
<head>
<title>Listing 5.1</title>
</head>
<body>
<?php
$a = "hello";
if ( $a == "hello" )
{
    echo "hello... welcome";
}

?>
</body>
</html>
```

اگر بخواهیم دستور if را طوری بنویسیم که در صورت برقرار نبودن شرط اول، شرط دوم، سوم و ... بررسی شود و در صورت صحیح بودن هر کدام کدهای مخصوص آن

اجرا شود می‌توانیم از دستور `elseif` و `else` استفاده کنیم.
ساختار کلی این دستورات عبارتند از:

```
if(شرط اول)
{
    دستورات بدنه
}
elseif(شرط دوم)
{
    دستورات بدنه
}
elseif(شرط سوم)
{
    دستورات بدنه
}
...
else
{
    دستورات بدنه
}
```

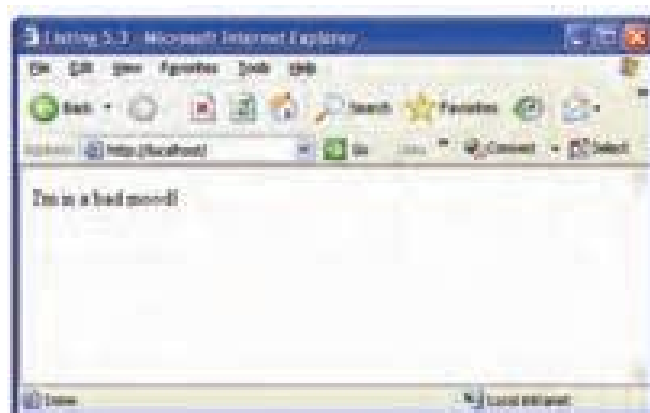
در ساختار فوق ابتدا شرط مقابل `if` چک می‌شود، اگر درست بود، دستورات بدنه آن اجرا می‌شود، اما اگر نادرست بود شرط مقابل `elseif` اول اجرا می‌شود، در صورت صحیح بودن، کدهای آن اجرا شده، اما اگر صحیح نباشد، دستورات `elseif` بعدی را بررسی می‌کند و این عمل به تعداد دستورات `elseif` ادامه پیدا می‌کند و در نهایت اگر هیچ‌کدام از شرط‌های `if` و `elseif` صحیح نبودند، دستورات مربوط به بدنه `else` اجرا می‌گردد.

مثال:

```
<html>
<head>
<title>Listing 5.3</title>
</head>
```

```
<body>
<?php
$mood = "sad";
if ( $mood == "happy" )
{
    echo "Hooray, I'm in a good mood";
}
elseif ( $mood == "sad" )
{
    echo " I'm in a bad mood!";
}
else
{
    echo "Neither happy nor sad but $mood";
}
?>
</body>
</html>
```

پس از اجرای کد مورد نظر مشاهده می‌کنید که به دلیل درست بودن شرط مقابل `elseif`، دستورات مربوط به این بلوک اجرا می‌شود.



شکل 5-6



تمرین:

دستورات فوق را طوری تغییر دهید که بلوک مربوط به else اجرا شود.

2 - 4 - 6 دستور Switch

ساختار کلی این دستور مانند جاوا اسکریپت است:

```
Switch (نام متغیر)
{
    مقدار اول Case:
        دستورات بدنه
    break;
    مقدار دوم Case:
        دستورات بدنه
    break;
    :
    default
        دستورات بدنه
}
```

در ساختار Switch مقدار متغیر مقابل Switch با مقادیر مقابل Case ها مقایسه می شود، اگر با مقدار مقابل Case اول برابر باشد، دستورات بدنه آن اجرا می شود و سپس از بلوک Switch خارج می شود، اگر برابر با مقدار مقابل Case دوم باشد، دستورات مربوط به آن اجرا شده و از بلوک Switch خارج می شود.

به همین ترتیب چک کردن مقادیر ادامه پیدا می کند و اگر هیچ کدام از مقادیر برابر با متغیر مقابل Switch نبودند، دستور بدنه default اجرا می شود.

مثال قبل مربوط به شرط if ... elseif را با استفاده از دستور Switch بازنویسی می کنیم:

<html>

<head>

<title>Listing 5.4</title>

```
</head>
<body>
<?php
$mood = "sad";
switch ( $mood )
{
case "happy":
echo "Hooray, I'm in a good mood";
break;
case "sad":
echo "I'm in a bad mood!";
break;
default:
echo "Neither happy nor sad but $mood";
}
?>
</body>
</html>
```

پس از اجرای کد فوق خروجی این برنامه نیز مشابه شکل 5-5 خواهد بود.

5 - 6 حلقه‌ها

حلقه‌ها یکی دیگر از امکانات زبان‌های برنامه‌نویسی هستند که اجازه می‌دهند دستور یا دستوراتی چند بار اجرا شوند.

1- 5- 6 دستور for

ساختار کلی دستور for به صورت زیر است:

```
for (میزان افزایش یا کاهش متغیر ; عبارت شرطی ; مقدار اولیه حلقه)
{
    بدنه حلقه
}
```

ساختار حلقه‌ها در PHP نیز شباهت زیادی به جاوا اسکریپت دارد. مقدار اولیه حلقه، مقداری است که برای متغیر شمارنده تعداد دفعات اجرا تعیین می‌شود.

عبارت دوم، عبارت کنترلی است که آخرین بار اجرای حلقه را معین می‌کند. عبارت سوم مقدار اضافه یا کم شدن متغیر اولیه در هر بار اجرای حلقه را تعیین می‌کند. مثال: می‌خواهیم اعداد 1 تا 10 را در خروجی چاپ کنیم.

```
<html>
<head>
<title>for</title>
</head>
<body>
<?php
for ( $i=1; $i<=10; $i++ )
{
echo"$i"."<br>";
}
?>
</body>
</html>
```



شکل 6-6

تمرین:

با استفاده از حلقه for برنامه‌ای بنویسید که اعداد 100 تا 50 را به صورت نزولی در خروجی چاپ کند.

در ساختار حلقه for مانند جاوا اسکریپت می‌توان مقدار اولیه حلقه را قبل از مقداردهی کرد و از نوشتن آن مقابل for صرف‌نظر کرد، هم‌چنین از میزان افزایش یا کاهش متغیر که معمولاً در پرانتز مقابل دستور for انجام می‌شود می‌توان صرف‌نظر کرده و عمل کاهش یا افزایش متغیر را درون حلقه for انجام داد.

مثال:

```
<html>
<head>
<title>for</title>
</head>
<body>
<?php
$i=1;
for ( ; $i<=10; )
{
    echo "$i". "<br>";
    $i++;
}
?>
</body>
</html>
```

2 - 5 - 6 حلقه while

ساختار کلی این حلقه به صورت زیر است:

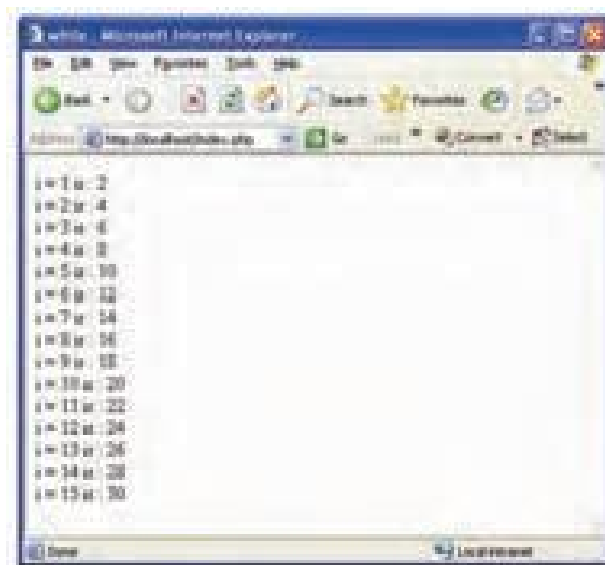
```
while (شرط خاتمه)
{
    دستورات بدنه
}
```

مثال: می خواهیم توسط دستور while مضارب 2 مربوط به اعداد کوچک تر از 15 را

در خروجی نمایش دهیم.

```
<html>
<head>
<title>while</title>
</head>
<body>
```

```
<?php
$i = 1;
while ( $i <= 15 )
{
    echo "i = ".$i. " is : " . ($i*2). "<br>";
    $i=$i+1;
}
?>
</body>
</html>
```



شکل 6-7

3 - 5 - 6 حلقه while ... do

ساختار این حلقه نیز کاملاً شبیه به do ... while در جاوا اسکریپت است. این حلقه در تمام زبان‌ها حداقل یک‌بار اجرا می‌شود، چون شرط بررسی ادامه در انتهای حلقه نوشته شده است.

```
do {
    دستورات بدنه
}
while(عبارت شرطی) ;
```

مثال:

```
<html>
<head>
<title>do...while</title>
</head>
<body>
<?php
$n1 = 8;
do
```

```
{
print "number: ." $n1."<br>";
$n1++;
}
while ( $n1 > 10 && $n1 < 15 );
?>
</body>
</html>
```



شکل 6-8

در کد قبلی متغیر اولیه $n1$ با عدد 8 مقداردهی اولیه شده است و در عبارت شرطی مقابل while، بررسی شده است که تا زمانی که متغیر $n1$ بزرگ‌تر از 10 و کوچک‌تر از 15 است حلقه ادامه یابد، اما در شکل 6-8 ملاحظه می‌کنید با این که این شرط هرگز صحیح نیست، یکبار اجرا شده است.

6 - 6 دستورات break و continue

عملکرد این دستورات مانند جاوا اسکریپت است.
 دستور break برای خروجی از بلوک جاری و ادامه برنامه استفاده می‌شود.
 دستور continue برای خروج از حلقه با پارامتر فعلی و ادامه بلوک با مقدار پارامتر بعدی انجام می‌شود.

7 - 6 متغیرهای سراسری

اگر در برنامه بخواهید یک متغیر را بین تمام قسمت‌ها به‌طور یکسان استفاده کنید، می‌توانید آن را از نوع سراسری (global) تعریف نمایید:
شیوه تعریف یک متغیر سراسری به فرم‌های زیر است:

; نام متغیر \$global

; مقدار متغیر = نام متغیر \$global

واژه‌نامه

Association	انجمن
Break	شکستن
Continue	ادامه دادن
Default	پیش فرض
Define	مقداردهی
HomePage	صفحه خانگی
Indexed	دارای زیرنویس

خلاصه مطالب

- برای ایجاد فایل‌های PHP نیازی به محیط مخصوص این زبان نیست، حتی در محیط متنی ساده Notepad نیز می‌توان کدهای آن را نوشت.
- فایل HomePage صفحات سایت باید با نام index یا default ذخیره شود.
- مهم‌ترین ابزار تولید صفحات پویا استفاده از متغیرهاست.
- متغیرهای PHP همه منظوره هستند و نیازی نیست هنگام معرفی، نوع متغیر تعیین شود.
- هنگام مقداردهی متغیرهای PHP، نوع آن‌ها به‌طور ضمنی تعیین می‌شود.
- آرایه‌های PHP دارای دو نوع اصلی اندیس دار و انجمنی هستند.
- اندیس مقادیر آرایه‌های PHP از صفر شروع می‌شود یعنی اندیس اولین عنصر آرایه صفر و اندیس آخرین عنصر آرایه $n-1$ است.
- در PHP متغیرهایی وجود دارند که به‌طور از پیش تعریف شده برای اهداف خاصی در نظر گرفته شده‌اند، در نامگذاری متغیرها باید دقت کنید، متغیری که تعریف می‌کنید، هم‌نام با متغیرهای از پیش تعریف شده نباشد.
- عملگرهای زبان PHP نیز مانند سایر زبان‌ها برای انجام عملیات محاسباتی، مقایسه‌ای، منطقی و ... به کار می‌روند.
- در تمام زبان‌های برنامه‌نویسی از جمله PHP، برای کنترل روند اجرای برنامه از

ساختارهای کنترلی استفاده می‌شود که مهم‌ترین آن‌ها دستورات if، if ... elseif و Switch ... Case هستند.

• اگر در نظر داشته باشیم یک یا چند دستورالعمل را چند بار اجرا کنیم بهتر است از دستورات حلقه استفاده نماییم.

• مهم‌ترین دستورات حلقه عبارتند از: for، while و do ... while