

ساختار PHP

فصل دوم

هدف‌های رفتاری:

پس از آموزش این فصل هنرجو می‌تواند:

- ۱- برچسب‌های معین‌کننده زبان PHP را به شکل‌های مختلف بنویسد.
- ۲- برنامه‌های ساده PHP را در Notepad بنویسد.
- ۳- برنامه‌های PHP را در مرورگر اجرا کند.
- ۴- انواع داده‌ها و متغیرها را تعریف و آن‌ها را در برنامه به کار گیرد.
- ۵- با استفاده از دستورهای مختلف برنامه‌نویسی متغیرها را ارزیابی کند.
- ۶- از عملگرهای مقایسه‌ای در برنامه استفاده کند.

مقدمه

در جلد اول این کتاب با شکل نوشتن دستورات در فایل HTML آشنا شدید. همانطور که می‌دانید هر زبان برنامه نویسی از استانداردها و قواعد مشخصی پیروی می‌کند. طراحان زبان PHP نیز مانند سایر زبان‌های اسکریپت نویسی و برنامه نویسی ساختارهای مشخصی برای نوشتن دستورات، پیشنهاد داده‌اند که در ادامه با هر کدام از آن‌ها آشنا خواهید شد.

۱-۲ شکل نوشتن دستورات PHP

برای نوشتن دستورات PHP چهار روش وجود دارد که در این جا به معرفی دو روش مرسوم می‌پردازیم. شما به صورت دلخواه می‌توانید هر کدام از این ساختارها را در برنامه، مورد استفاده قرار دهید.

روش اول:

در این روش که به روش کوتاه (Short) معروف است، دستورات و کدهای PHP درون علائم <?...?> قرار می‌گیرند.

<?

دستورات و کدهای PHP

?>

روش دوم:

در این روش که به روش XML معروف است، 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. معمولاً کدهای PHP درون تگ‌های HTML نوشته می‌شود و با این تگ‌ها ترکیب می‌شود، اما پس از نوشتن کدها، باید فایل با پسوند PHP ذخیره شود و در صورت ذخیره فایل با پسوند HTML یا HTM، کدهای PHP اجرا نخواهند شد.

۲-۲ نوشتن اولین برنامه به زبان PHP

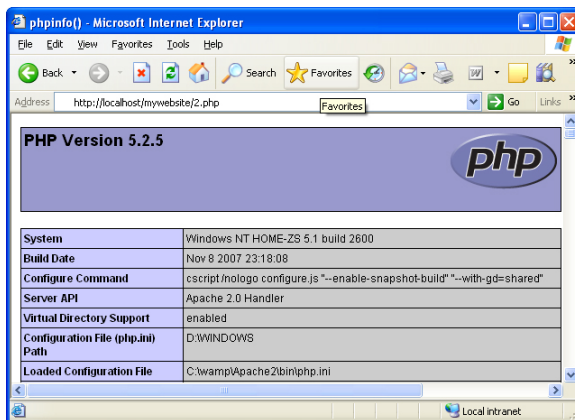
به منظور نوشتن اولین برنامه PHP و اجرای آن کدهای زیر را در برنامه Notepad به منظور نوشتن اولین برنامه PHP و اجرای آن کدهای زیر را در برنامه Notepad بنویسید، سپس فایل را بپسوند. PHP در مسیری که پیش از این برای ریشه WAMP تعیین کرده بودید، ذخیره کنید:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
<?php
phpinfo();?>
</body>
</html>

```

برای اجرای این فایل در مرورگر، با کلیک روی آیکن WAMP و انتخاب گزینه Localhost مسیرپوشه‌ها و فایل‌های موجود در مسیر سرویس دهنده نشان داده می‌شود، اگر روی عنوان فایل کلیک کنید پنجره مرورگر مشخصات کامل PHP مورد استفاده شما را نشان خواهد داد (شکل ۲-۱)



شکل ۲-۱

در زبان PHP برای نمایش یک عبارت در خروجی می‌توان یکی از دستورات echo و یا print را به کار گرفت.

قطعه کد زیر عبارت Hello Word را در خروجی نمایش می‌دهد:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
<?php
echo "Hello Word";
?>
</body>
</html>
```

در انتهای هر کدام از دستورات PHP باید علامت «؛» درج شود و در واقع علامت سمی کولن به عنوان جداکننده دو دستور مختلف از همدیگر به کار می‌رود. به مثال زیر توجه کنید:

```
<?php
echo "Hello Word"; echo "Welcome";
?>
```

در کد برنامه فوق، با استفاده از دستور echo عبارت Hello Word و پس از آن Welcome چاپ می‌شود، اما این دو دستور در یک سطر نوشته شده‌اند، یعنی درج علامت «؛» در انتهای هر دستور echo به معنای خاتمه دستور جاری و شروع دستور بعدی است و نیازی نیست برای درج دستور جدید، سطر دیگری ایجاد کنید، اما توصیه می‌شود برای بالا بردن خوانایی برنامه از درج دو دستور مختلف در یک سطر اجتناب کنید.

۲-۳ درج توضیحات در کدهای PHP

در زبان PHP برای درج توضیحات یک خطی از علامت // یا # و برای درج توضیحات چندخطی از علامت /* در شروع و از علامت */ برای خاتمه توضیحات استفاده می‌شود. توضیحات درون کدهای برنامه خوانایی آن را بالایی برد، از این رو سعی کنید هنگام نوشتن یک برنامه، توضیحات مناسبی را به همراه کدهای آن ارائه دهید.

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD XHTML 1.0 Strict//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
<?php
//This is a single line comment
/*
This is
a comment
block
*/
# This is a single line comment
?>
</body>
</html>
```

کدهای PHP هنگام اجرا در سرویس گیرنده به کدهای HTML تبدیل می‌شوند به این معنا که اگر کد فایل PHP ای که در حال اجرا توسط مرورگر است را با کلیک راست در صفحه و انتخاب گزینه View Source مشاهده کنید، خواهید دید که تمام کدها به برچسب‌های HTML تبدیل شده‌اند و دستورات PHP در آن مشاهده نمی‌شود.

۴-۲ متغیرها در PHP

همانطور که می‌دانید، هر زبان برنامه نویسی برای انعطاف بیشتر، به انواع مختلفی از داده‌ها نیازمند است. یکی از ویژگی‌های اصلی زبان‌های برنامه نویسی، توانایی تعریف متغیرها است. متغیرها مکان‌هایی در حافظه هستند که برای ذخیره مقادیر انواع داده‌های مختلف به کار می‌روند. در برنامه به هر متغیر یک نام منحصر به فرد اختصاص داده می‌شود که شناسایی آن را از سایر متغیرها امکان پذیر می‌کند.

پس از معرفی یک متغیر می‌توان بارها از آن در جاهای مختلف برنامه و بدون محدودیت استفاده کرد.

برای تعریف متغیرها در زبان PHP، علامت \$ در ابتدای نام متغیرها درج می‌شود. شکل صحیح و کلی تعریف متغیر به صورت زیر است:

مقدار متغیر = نام متغیر \$

مثال زیر یک متغیر از نوع رشته‌ای با مقدار Hello Word و یک متغیر از نوع عددی با مقدار ۱۶ را تعریف می‌کند:

```
<?php
$txt="Hello Word!";
$x=16;
?>
```

زبان PHP به نوع متغیرها وابسته نیست و لازم نیست هنگام تعریف متغیر، نوع آن نیز معرفی شود، بلکه پس از مقداردهی به طور خودکار نوع متغیر تشخیص داده می‌شود، این توانایی موجب انعطاف این زبان نسبت به بسیاری از زبان‌های برنامه‌نویسی دیگر شده است.

۱-۴-۲ قواعد نامگذاری متغیرها در PHP

قوانین نام گذاری متغیرها در PHP، همانند Visual Basic است و برای نامگذاری متغیرهای PHP می‌توانید از حروف الفبای کوچک و بزرگ، اعداد و Underline (_) استفاده کنید. نام هر متغیر باید با یکی از حروف الفبا یا علامت " _ " شروع شود و متغیرهایی که با عدد یا علامتی غیر از حروف الفبا و " _ " شروع می‌شوند، نامعتبر هستند. لازم به ذکر است که از اعداد فقط در ابتدای نام متغیرها نمی‌توان استفاده کرد و در هر جای دیگری از نام متغیرها، استفاده از اعداد بلامانع است.

به طور مثال متغیرهایی با نام‌های زیر معتبرند:

```
$_txt
$_txt
$a5Bc
$a
$a
$str23
```

و متغیرهایی با نام‌های زیر نامعتبر هستند:

```
$2a
$#txt
$+abc
$my var
```

برنامه زیر نحوه استفاده از متغیرها را نشان می‌دهد. در این برنامه با استفاده از متغیرها، محیط دایره‌ای به شعاع دلخواه r محاسبه شده است:

```
$PI = 3.14;
$r = 5;
$circumference = $PI * 2 * $r;
```

متغیرها در زبان PHP محدوداً کاملاً محلی دارند و فقط درون تابعی که تعریف شده‌اند قابل دسترسی هستند. به همین دلیل اغلب برنامه‌نویسان متغیرهای مهم را که ممکن است در جاهای مختلف برنامه مورد استفاده قرار گیرند، درون اسکریپت اصلی می‌نویسند.

مطالعه آزاد

روش دیگری به منظور تعریف حوزه سراسری برای متغیرها، استفاده از آرایه از پیش تعریف شده `$GLOBALS` است.

استفاده از این آرایه به این صورت است که نام متغیر مورد نظر را به عنوان اندیس آرایه درج می‌کنید. به این ترتیب با استفاده از آرایه مخصوص `$GLOBALS` می‌توان برنامه فوق را به شکل زیر بازنویسی کرد:

```
$PI = 3.14;
$r = 5;
$circumference = $GLOBALS["PI"] * 2 * $GLOBALS["r"];
```

۵-۲ انواع داده‌ها

در PHP هشت نوع داده مختلف قابل استفاده است. در مباحث گذشته با روش مقداردهی متغیرها آشنا شدید. هر متغیر براساس مقداری که می‌گیرد، نوع داده‌ای متفاوتی را می‌تواند داشته باشد. شناخت انواع داده‌ها در به کارگیری متغیرهای استفاده شده در برنامه بسیار مؤثر و مفید است.

انواع داده قابل پشتیبانی در PHP عبارتند از: `Integer`، `double`، `Boolean`، `String`، `Array`، `Float` و `Float` که در ادامه به معرفی ساختار هر یک خواهیم پرداخت.

نوع داده	مثال
Integer	240000 0xABCD 007 -100
float	3.14 +0.9e-2 -170000.5 54.6E42
Boolean	true , false
string	در ادامه این نوع داده به طور دقیق بررسی شده است.
array	در ادامه این نوع داده به طور دقیق بررسی شده است.

۱-۵-۲ String

به چند کاراکتر که در کنار یکدیگر قرار می‌گیرند، اصطلاحاً رشته گفته می‌شود. برای معرفی و مقداردهی یک مجموعه کاراکتر به یک متغیر از علائم " و ' یا ' استفاده می‌شود.

بین علائم نقل قول جفتی (" ") و نقل قول تکی (' ') یک تفاوت مهم وجود دارد. اگر به هنگام انتساب یک مقدار به یک متغیر از " استفاده کنید، در داخل نقل قول می‌توانید از متغیرهای مختلف استفاده نمایید، اما در صورت استفاده از متغیرها در علائم ' نام متغیر به عنوان یک رشته ثابت در نظر گرفته شده، و بدون در نظر گرفتن محتوای آن چاپ می‌شود، با دقت در مثال زیر این موضوع به خوبی آشکار می‌شود:

```
<?php
$txt="Hello Word!";
$x=16;
echo "string is: $txt".<br/>";
echo "-----".<br/>";
echo 'my variable is : $x';
?>
```

خروجی

```
string is: Hello Word!
-----
my variable is : $x
```


درج
 مقابل دستور echo یک سطر جدید در خروجی ایجاد می کند. نوشتن مثال فوق بدون
 خروجی را به صورت زیر نمایش خواهد داد:

خروجی:

```
string is: Hello Word!----- my variable is : $x
```


در یک دستور echo می توان دو یا چند عبارت را در خروجی نمایش داد، برای مشخص کردن چند متغیر یا رشته می توان از کاراکتر نقطه (.) بین هر دو مورد از آن ها استفاده نمود. کاراکتر نقطه برای الحاق (چسباندن) دو رشته به یکدیگر به کار می رود و یکی از اصلی ترین عملگرهای مربوط به رشته ها به شمار می رود. مثال زیر شکل به کارگیری این کاراکتر را به خوبی نشان می دهد:

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

خروجی:

```
Hello Word! What a nice day!
```

در مثال فوق بین دو متغیر رشته ای \$txt1 و \$txt2، تعدادی کاراکتر فاصله (space) به منظور جدا کردن مقدار دو متغیر قرار داده شده و به متغیرهای \$txt1 و \$txt2 الحاق شده است. در فصل های آینده با تعدادی از مهمترین توابع رشته ای آشنا خواهید شد.

 **تمرین:** سه متغیر جداگانه تعریف کنید و به هر کدام از آنها به ترتیب روز، ماه و سال تولد خود را نسبت دهید، سپس با استفاده از یک دستور echo تاریخ تولد خود را به صورت روز / ماه / سال در خروجی نمایش دهید.

۲-۵-۲ array

در PHP هر آرایه مجموعه ای اندیس دار به همراه مقدار است و به عبارت دیگر هر آرایه

گروهی از متغیرهای کنارهم است که ارجاع به آنها از طریق یک نام مشترک صورت می‌گیرد. آرایه‌ها روش بسیار مفیدی برای گروه بندی اطلاعات مرتبط به هم هستند. در PHP آرایه‌هایی از انواع مختلف را به صورت یک یا چند بعدی می‌توان ایجاد نمود. اگر یک لیست از متغیرهای هم نوع را داشته باشید و بخواهید آنها را ذخیره کنید، می‌توانید از ساختار آرایه‌ای استفاده نمایید.

آرایه‌ها قادرند تعداد زیادی متغیر را با یک نام یکسان ذخیره کنند و پس از آن برای شناسایی تک تک عناصر آرایه می‌توان از اندیس هر عنصر استفاده نمود. به طور مثال برای ذخیره اسامی ۲۰ نفر از هم کلاسی‌های خود می‌توانید یک آرایه تعریف کنید که دارای ۲۰ عنصر از نوع رشته باشد:

رشته ۱	رشته ۲	رشته ۳	...	رشته ۲۰
--------	--------	--------	-----	---------

ساختار آرایه ای از نوع string با ۲۰ عنصر

شکل کلی تعریف یک آرایه به صورت زیر است:

("مقدار عنصر آخر" ... و "مقدار عنصر دوم" و "مقدار عنصر اول") array = نام آرایه \$

مثال:



در مثال زیر یک آرایه با نام cars تعریف شده است. این آرایه دارای چهار عنصر است.

که مقدار دهی عناصر آن هنگام تعریف آرایه، صورت گرفته است:

```
<?php
```

```
$cars=array("Samand","Pars","L90","pride");
```

```
?>
```

در PHP پس از ایجاد آرایه، برای هر کدام از عناصر آن به طور خودکار یک اندیس اختصاص داده می‌شود، اندیس‌های عناصر به طور خودکار از عدد صفر آغاز می‌شود. با توضیحات فوق می‌توان گفت اندیس عناصر آرایه مثال فوق به صورت زیر در نظر گرفته می‌شود:

```
$cars[0]="Samand";
```

```
$cars[1]="Pars";
```

```
$cars[2]="L90";
```

```
$cars[3]="Pride"
```

مثال:



می‌خواهیم یک مورد از عناصر آرایه cars را در خروجی نمایش دهیم:

```
<?php
$scars=array("Samand","Pars","L90","Pride");
echo $scars[0]
?>
```

خروجی:

Samand

تمرین: نام عناصر آرایه cars به مثال قبل را در خروجی نشان دهید

دسترسی به کاراکترهای تشکیل دهنده یک رشته، مانند دسترسی به عناصر آرایه است. یک رشته همانند آرایه‌ای از کاراکترها ذخیره می‌شود و برای دسترسی به هر کدام از عناصر آن می‌توان نام متغیر رشته‌ای به همراه اندیس کاراکتر را به کار برد، در این صورت اندیس کاراکتر را باید درون { } یا [] قرا داد.

مثال:



مثال زیر اولین و هفتمین کاراکتر یک رشته را در خروجی چاپ می‌کند.

```
<?php
$str="Hello Word";
echo "first character is: ".$str{0}."and";
echo "seventh character is: ".$str[6];
?>
```

خروجی:

first character is: H and seventh character is: W

نکته: اندیس عناصر رشته مانند آرایه‌ها از صفر شروع می‌شود.



نکته: PHP به حروف بزرگ و کوچک حساس است. یعنی تغییری که در ابتدا با ترکیب مشخصی از حروف کوچک و بزرگ معرفی می‌شود، در همه جای برنامه باید به همان صورت استفاده شود؛ در غیر این صورت به عنوان متغیر دیگری در نظر گرفته می‌شود. به طور مثال متغیرهای \$name و \$Name را متمایز از یکدیگر در نظر می‌گیرد.



مطالعه آزاد

علاوه بر انواع داده‌ای که مطرح شدند، سه نوع داده دیگر در PHP کاربرد دارد که عبارتند از:

object

این نوع داده در مباحث مربوط به شیء گرایی کاربرد دارد و همانند نوع داده آرایه، یک نوع ترکیبی به شمار می‌رود.

Resource

این نوع داده اختصاصی بوده و در نسخه‌های جدید PHP تعریف شده است. متغیر Resource می‌تواند یک فایل باز شده، یک اتصال به پایگاه داده و یا هر نوع خارجی دیگری را شامل شود.

در برنامه برخلاف انواع داده دیگر، نمی‌توان این نوع داده را مستقیماً مقداردهی کرد و نسبت به سایر انواع، کاربرد محدودتری دارد.

Null

اگر تغییری از این نوع داده تعریف شده باشد، شامل هیچ مقداری نیست، به این معنا که یا مقدار Null به آن اختصاص داده شده و یا اینکه مقدار آن توسط توابع ویژه‌ای در PHP حذف شده است.

۶-۲ ثابت‌ها

یکی از امکانات اغلب زبان‌های برنامه نویسی قابلیت استفاده از ثابت‌ها است. ثابت‌ها مقادیری هستند که در طول اجرای برنامه بدون تغییر باقی می‌مانند. به منظور تعریف ثابت‌ها از کلمه کلیدی define استفاده می‌شود. فقط داده‌هایی از نوع integer، float، boolean و string قابل استفاده در ثابت‌ها هستند.

شکل کلی تعریف ثابت‌ها بصورت زیر است:

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

ثابت‌ها در همه جای برنامه قابل دسترسی بوده و اصطلاحاً به صورت سراسری (global) در نظر گرفته می‌شوند.

برای دسترسی به ثابت‌ها در طول برنامه، نیازی به علامت \$ نیست و دسترسی به آن توسط نام ثابت صورت می‌گیرد.

در مثال زیر یک ثابت تعریف و توسط دستور echo در خروجی چاپ می‌شود:

```
<?php
define("name","Mahdi");
echo "His name is : " . name;
?>
```

خروجی:

His name is : Mahdi

قوانین نامگذاری ثابت‌ها همانند متغیرهاست و باید با یکی از حروف الفبا و یا کاراکتر «_» شروع شود.

۷-۲ انواع عملگرها

عملگر نمادی است که عملیات خاصی را روی متغیرها انجام می‌دهد. زبان PHP بسته به نوع متغیرها، عملگرهای متعددی را پشتیبانی می‌کند. عملگرهای اصلی PHP عبارتند از:

- ۱- عملگرهای ریاضی (محاسباتی)
- ۲- عملگرهای مقایسه‌ای

۱-۷-۲ عملگرهای ریاضی (محاسباتی)

در جدول زیر لیست انواع عملگرهای ریاضی که در PHP پشتیبانی می‌شوند ارائه شده است.

عملگرهای ریاضی

عملگر	عملکرد	مثال به ازای $y = 5$ $x = 2$	نتیجه
+	جمع	$x+2$	4
-	تفریق	$5-x$	3
*	ضرب	$x*5$	10
/	تقسیم	$y/x=5/2$	2,5
%	باقیمانده تقسیم	$y\%x=5\%2$	1 2 0
++	یک واحد افزایش	$x++$	$x=3$
--	یک واحد کاهش	$x--$	$x=2$

عملگرهای افزایش و کاهش به اندازه ۱ واحد به ترتیب به متغیر اضافه کرده یا از آن کم می‌کنند.

مثال:



$\$a=5;$

$\$a++;$

$++\$a;$

در مثال بالا ابتدا متغیر a از نوع عددی با مقدار ۵ تعریف شده است، در خط دوم با استفاده از عملگر ++ یک واحد به آن اضافه شده، یعنی مقدار a به ۶ تغییر پیدا می‌کند، در خط سوم نیز مجدداً یک واحد دیگر به a افزوده می‌شود که در نهایت مقدار a برابر با ۷ می‌شود. این دو دستور معادل هم هستند و فقط در یک مورد با هم تفاوت دارند، به مثال بعد توجه کنید:

$\$a;$

$\$b;$

$\$a=5;$

$\$b=\$a++;$

در اینجا پس از اعلان دو متغیر a و b مقدار دادن به a در خط سوم به دلیل این که عملگر ++ بعد از نام متغیر آمده، مقدار فعلی a یعنی ۵ در b ریخته می‌شود و بعد به متغیر a یک واحد افزوده می‌شود. یعنی در پایان برنامه متغیر a برابر با مقدار ۶ و متغیر b برابر با

مقدار ۵ خواهد بود.

\$a , \$b;

\$a=5;

\$b=++\$a;

به مجموعه دستورات فوق دقت کنید، خط اول و دوم این دستورات همانند دستورات قبل هستند و تفاوت در خط سوم است. در خط سوم به دلیل این که عملگر ++ قبل از نام متغیر آمده، در ابتدا یک واحد به a اضافه می شود یعنی مقدار a به ۶ تغییر کرده و بعد مقدار a که در حال حاضر ۶ است، در b هم ریخته می شود و در پایان خط سوم متغیرهای a و b حاوی عدد ۶ هستند.

شیوه عملکرد -- نیز مشابه ++ است با این تفاوت که در -- عمل کاهش به اندازه ۱ واحد انجام می شود.

۲-۷-۲ عملگرهای مقایسه‌ای

این عملگرها برای مقایسه بین اعداد و متغیرها به شکل‌های مختلف صورت می گیرد و نتیجه این مقایسه یک مقدار boolean (درست یا غلط) است.

مثال	عملگر
$8=5$ مقدار false برگردانده می شود. $2=2$ مقدار true برگردانده میشود.	==
$8!=5$ مقدار true را برمی گرداند.	!=
$8<>5$ مقدار true را برمی گرداند.	<>
$8<5$ مقدار false برگردانده می شود.	<
$8>5$ مقدار true را برمی گرداند.	>
$8<=5$ مقدار false برگردانده می شود.	>=
$8>=5$ مقدار true را برمی گرداند.	<=

$5===5$ مقدار true برگردانده می شود. $2===2$ مقدار False برگردانده می شود.	$===$
$5!=5$ مقدار false برگردانده می شود. $2!=2$ مقدار true برگردانده می شود.	$!===$

رفتار عملگرهای $! =$ و $<>$ مشابه است و با یکدیگر تفاوتی ندارند. عملگرهای $===$ و $==$ برای بررسی مساوی یا نامساوی بودن پارامترهای مختلف به کار می‌روند، اما این دو تفاوت‌های ساختاری با یکدیگر دارند. در عملگر $===$ مقایسه بر اساس نوع و مقدار انجام می‌شود، اما در عملگر $==$ مقایسه فقط بر اساس مقدار صورت می‌گیرد. برای روشن شدن موضوع، به مثال‌های زیر دقت کنید.

مثال ۱:



در دستور سطر ۴ چون a با b برابر هستند مقدار true برگردانده می‌شود اما در سطر ۵ به دلیل این که c و b با هم مساوی نیستند مقدار false برگردانده می‌شود.

مثال ۲:



$a = 3;$

$b = "3";$

$a == b;$

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

همان طور که اشاره شد، 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 هم نوع نیستند.

۳-۷-۲ عملگرهای منطقی

نتیجه برگشت داده شده در این دسته از عملگرها مشابه عملگرهای مقایسه‌ای است. (عملگرهای منطقی نیز نتیجه true یا false را برمی گردانند.)

عملگر	عملکرد	مثال به ازای $x=6$ و $y=3$
&&	and (معادل حرف «و» در جملات محاوره‌ای فارسی)	$x < 10 \&\& y > 1$ مقدار true را برمی گرداند.
	or (معادل کلمه «یا» در جملات محاوره‌ای فارسی)	$(x == 5 \ \ y == 5)$ مقدار false را برمی گرداند.
!	not (نقیض - نفی کننده)	$(x == y)$ مقدار true را برمی گرداند.

زبان php در عملگر منطقی &&، دو عملوند && را بررسی می‌کند و اگر هر دو true بودند، نتیجه true را برمی گرداند، در غیر این صورت، مقدار برگشت داده شده در این عملگر، false خواهد بود، همچنین در دستور || دو عملوند || را بررسی می‌کند حتی اگر یکی از دو طرف true بودند، نتیجه true را برمی گرداند و تنها در صورتی نتیجه false می‌شود که هر دو متغیر یا عبارت منطقی، false باشند.

نکته: به جای && می‌توانید از عملگر and و به جای || می‌توانید از عملگر or استفاده کنید.



مطالعه آزاد

عملگرهای تجمعی

یکی از مسائلی که مورد توجه برنامه نویسان حرفه‌ای قرار می‌گیرد، سرعت نوشتن و سرعت اجرای دستورات توسط پردازنده است.

در اغلب زبان‌های برنامه‌نویسی که امروزه به کار می‌روند، امکان استفاده از عملگرها به شکل تجمعی است.

عملگرهای تجمعی با سرعت بالاتری توسط پردازنده اجرا می‌شوند و برنامه نویسان

حرفه‌ای به جای شکل ساده عبارات توسط عملگرهای محاسباتی، آنها را به صورت تجمعی به کار می‌برند.

عملگرهای تجمعی

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

در هر ردیف جدول فوق، یکی از عملگرهای تجمعی به همراه مثال بیان شده است، در ستون "صورت مشابه عملگر"، شکل ساده آن بیان شده است به طور مثال $x+=y$ مانند $x=x+y$ عمل می‌کند.

نوشتن عملگرها به صورت تجمعی، سرعت اجرای برنامه را بالا می‌برد،

چکیده فصل

پراستفاده‌ترین روش برای نوشتن کدهای PHP به شکل زیر است:

```
<?PHP
```

دستورات و کدهای PHP

```
?>
```

در زبان PHP برای نمایش یک عبارت در خروجی یکی از دستورات echo و یا print استفاده می‌شود و در انتهای هر کدام از دستورات PHP باید علامت ";" درج شود.

در زبان PHP برای درج توضیحات یک خطی از علامت // یا # و برای درج توضیحات چند خطی از علامت /* در شروع، و از علامت */ برای خاتمه توضیحات استفاده می‌شود. متغیرها مکان‌هایی در حافظه هستند که برای ذخیره مقادیر انواع داده‌های مختلف به کار می‌روند

برای تعریف متغیرها در زبان PHP، علامت \$ در ابتدای نام متغیرها درج می‌شود. متغیرها در زبان PHP محدوده کاملاً محلی دارند و فقط درون تابعی که تعریف شده‌اند قابل دسترسی هستند.

انواع داده قابل پشتیبانی در PHP عبارتند از: Integer، double، Boolean، String، Float و

در PHP هر آرایه مجموعه‌ای اندیس‌دار به همراه مقدار است و آرایه‌ها قادرند تعداد زیادی متغیر را با یک نام یکسان ذخیره کنند و پس از آن برای شناسایی تک تک عناصر آرایه می‌توان از اندیس هر عنصر استفاده نمود.

به منظور تعریف ثابت‌ها از کلمه کلیدی define استفاده می‌شود و فقط داده‌هایی از نوع integer، float، boolean و string قابل استفاده در ثابت‌ها هستند.

عملگر نمادی است که عملیات خاصی را روی متغیرها انجام می‌دهد و زبان PHP تمام عملگرهای اصلی محاسباتی و منطقی را پشتیبانی می‌کند.

خودآزمایی

۱. یک برنامه برای نمایش نام و نام خانوادگی خود بنویسید و در کنار هر سطر برنامه توضیحات مربوط به عملکرد آن را درج کنید.
۲. با استفاده از متغیرها، مساحت دایره‌ای به شعاع ۵cm را محاسبه کنید.
۳. نام ۴ شهر ایران را در یک آرایه تعریف کنید، سپس با استفاده از روش دسترسی به عناصر آرایه، عنوان شهر اول و سوم را با استفاده از یک دستور echo در خروجی نمایش دهید.