



آموزش برنامه نویسی PHP

مدرس آقای مشکانی

قسمت دوم

ILikephp.ir





بنام خدا

متغیرها در PHP

در این بخش می خواهیم طریقه ی استفاده از متغیرها در PHP را آموزش دهیم. اگر تا امروز از نرم افزار برنامه نویسی دیگری استفاده کرده باشید با نام متغیرها آشنایی دارید. در برنامه نویسی با استفاده از متغیرها یک سری داده تعریف می شود و ما می توانیم برای این داده ها اسم گذاری کنیم و در ادامه نوشتن برنامه هر جا که نیاز شد آن را صدا بزنیم و از تکرار کردن آن در طول برنامه بپرهیزیم.

در پی اچ پی متغیرها را با علامت \$ یا دلار نمایش م ی دهند.

دو قانون کلی برای متغیرها در PHP

1- نام متغیرها تنها میتواند حروف انگلیسی (a-z یا A-Z)، اعداد ۰-۹ و زیر خط (آندرلاین) _ را شامل شود.

2- نام متغیر هیچ گاه نمیتواند با عدد شروع شود.

به مثال زیر دقت کنید

```
<html>
```

```
<body>
```

```
<?php
```

```
$x=5;
```

```
$y=6;
```



```
$z=$x+$y;
```

```
echo $z;
```

```
?>
```

```
</body>
```

```
</html>
```

همان طور که در مثال بالا مشاهده م یکنید سه متغیر Y, X و Z تعریف شده اند. هر یک از این متغیرها در PHP یک داده را معرفی می کند. X عدد ۵، Y عدد ۶ و Z حاصل جمع این دو عدد را بیان می کند و در نهایت اطلاعاتی که ما می خواهیم به چاپ رسانیم متغیر Z است.

اگر دقت کنید متوجه می شوید که متغیر Z بین علامت های “ “ قرار نگرفته است، تنها دلیل این است که Z یک متغیر رشته ای و یا در حقیقت متن نیست و جنس عدد است.

خروجی دستور بالا به صورت زیر تنها یک عدد که حاصل جمع ۵ و ۶ است نمایش داده می شود و برابر است با:

11

حال یک مثال دیگر را بررسی م یکنیم.

```
<html>
```

```
<body>
```

```
<?php
```

```
$txt="Hello world!";
```



مدرس : آقای مشکانی

```
$x=5;
```

```
$y=10.5;
```

```
echo $txt;
```

```
echo "<br>";
```

```
echo $x;
```

```
echo "<br>";
```

```
echo $y;
```

```
?>
```

```
</body>
```

```
</html>
```

خروجی دستور بالا به صورت زیر است:

!Hello world

5

10.5

دستور اول چاپ متغیر txt را خواستار شده است و در ادامه متغیر X و y را نمایش می دهد.



Global در PHP

Local متغیرهای

متغیرهای Local یا متغیرهایی محلی، جزو دسته ای از متغیرها در PHP هستند که باید در داخل تابع تعریف می شوند، در غیر این صورت در هیچ داده ای نمایش داده نمی شود و متغیرهای Global یا متغیرهای جهانی، متغیرهایی هستند که با تعریف آن ها در خارج از تابع م بتوانیم به آن ها دسترسی داشته باشیم.

در باره ی تابع یا Function در ادامه بیشتر توضیح م بدهیم، برای آشنایی م بتوانیم به این موضوع اشاره کنیم که تابع دارای یک سری ورودی و خرجی است. در PHP همانند نرم افزارهای دیگر از تابع برای جلوگیری از روند تکرار توابع مثلا ضرب، جمع یا ... یک بار آن ها را در Function نوشته و هر بار که نیاز شد آن را صدا م میکنیم

یک مثال را بررسی م میکنیم، به متغیرهای Local و Global دقت کنید

```
<html>
```

```
<body>
```

```
<?php
```

```
$x=5;
```



```
$y=10;

functionmyTest()

{

$GLOBALS['y']=$GLOBALS['x']+$GLOBALS['y'];

}

myTest();

echo $y;

?>

</body>

</html>
```

خروجی دستور بالا عدد ۱۵ است و تنها مدل نوشتن دستورها تغییر کرده ولی خروجی دستور تغییری نمی کند.

متغیر static در php

هر گاه به متغیرها در PHP بخواهیم یک مقدار اولیه بدهیم و تنها یک بار این متغیر اولیه اجرا شود و در ادامه که آن را صدا میزنیم مقدار آخری که از تابع بدست آمده را جایگزین مقدار اولیه و بعد دستور را اجرا کند از متغیر static استفاده م یکنیم.



در روش استفاده از Static خواسته شده که متغیر Global حذف شود.

به مثال زیر دقت کنید.

```
<html>
```

```
<body>
```

```
<?php
```

```
functionmyTest()
```

```
{
```

```
static $x=0;
```

```
echo $x;
```

```
$x++;
```

```
}
```

```
myTest(); echo "<br>";
```

```
myTest(); echo "<br>";
```

```
myTest(); echo "<br>";
```

```
myTest(); echo "<br>";
```

```
myTest();
```

```
?>
```



</body>

</html>

در این مثال به متغیر استاتیک X مقدار اولیه ۰ را داده ایم و خواستیم که X را چاپ کند. در ادامه متغیر ++ X را نمایش دهد.

متغیر ++X بیان کننده این است که متغیر X را در هر بار چاپ با عدد ۱ جمع کن.

۰

۱

۲

۳

۴

داده ها در PHP

داده ها در php به انواع مختلف تقسیم می شوند:

1- رشته ای String

2- اعداد صحیح، Integer

3- Floating point number، اعداد ممیز شناور

5- Boolean، بولین

6- Array، آرایه



7 - Object، شی

8 - Null، تهی

حال به بررسی تک تک این داده ها م یپردازیم.

داده های string در php

یک رشته، یک سری کاراکتر است. یک استرینگ م میتواند هر نوع متنی باشد.

مانند: "Hello World!"

یک مثال در PHP مشاهده کنید.

```
<html>
```

```
<body>
```

```
<?php
```

```
$x = "Hello world!";
```

```
echo $x;
```

```
echo "<br>";
```



مدرس : آقای مشکانی

```
$x = 'Hello world!';
```

```
echo $x;
```

```
?>
```

```
</body>
```

```
</html>
```

خروجی این دستور متن زیر است:

!Hello world

!Hello world

داده های integer در php

تمام اعداد به غیر از اعداد اعشاری شامل اعداد صحیح می شوند.

در مثال زیر مدل های مختلف اعداد صحیح را مشاهده م یکنیم. که برای نمایش نوع داده در خروجی از دستور

var_dump() استفاده م یکنیم.

```
<html>
```

```
<body>
```



```
<?php
```

```
$x = 5985; var_dump($x);  
echo "<br>";
```

```
$x = -345; // negative number  
var_dump($x); echo "<br>";
```

```
$x = 0x8C; // hexadecimal number
```

```
var_dump($x);
```

```
echo "<br>";
```

```
$x = 047; // octal number
```

```
var_dump($x);
```

```
?>
```

```
</body>
```

```
</html>
```

خروجی دستور بالا به صورت زیر نمایش داده می شود:

```
int(5985)
```

```
int(-345)
```

```
int(140)
```



```
int(39)
```

در این خروجی دستور) `var_dump` باعث نمایش `int` که نوع داده است و همچنین عدد مقابل آن را نمایش می دهد.

PHP در Floating Point Number بداده های

عدد با ممیز شناور م میتواند به صورت اعشاری یا نمایی نوشته شود.

در نوشتن این داده ها در PHP از) `var_dump` استفاده می شود.

مثال زیر را برای اعداد با ممیز شناور مشاهده کنید: `<html>`

```
<body>
```

```
<?php
```

```
$x = 10.365;
```

```
var_dump($x); echo
```

```
"<br>"; $x = 2.4e3;
```

```
var_dump($x); echo
```

```
"<br>"; $x = 8E-5;
```

```
var_dump($x);
```

```
?>
```



</body>

</html>

در مثال ذکر شده همانند مثال قبل `var_dump` برای نمایش نوع داده و عدد مقابل آن است. در خروجی این مثال هم اعداد اعشاری م یبینید و هم عدد نمایی. به خروجی مثال دقت کنید:

`)float(10.365``)float(2400``)float(8.0E-5`

PHP در Boolean داده های

بولین یک دستور منطقی است که تنها درست یا غلط بودن داده ها در PHP نشان می دهد

`var x=true;``var y=false`

دستور بولین در بیشتر مواقع، شرایط را بررسی می کند که در درس های آتی بیشتر در مورد آن توضیح می دهیم.

داده های Array در PHP



آرایه ها چند مقدار را در یک متغیر نمایش م ی دهند. هنگامی که مقادیر ما بسیار زیاد است، استفاده از آرایه بهترین گزینه است.

در مثال زیر یک آرایه ایجاد کردیم و با استفاده از (`var_dump`) نوع و مقدار داده ها را نمایش می دهیم.

```
<html>
```

```
<body>
```

```
<?php
```

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

```
var_dump($cars);
```

```
?>
```

```
</body>
```

```
</html>
```

حال خروجی مثال بالا را بررسی کنیم.

```
} "array(3) , *0+=> string(5) "Volvo" *1+=> string(3) "BMW" *2+=> string(6) "Toyota"
```



اگر به خروجی دقت کنید، مشاهده می‌کنید ابتدا دستور `var_dump` نوع داده‌ها در PHP مشخص و سپس دستور آرایه، تعداد آن را بیان کرده است، در ادامه نوع و تعداد داده‌های استفاده شده در آرایه یعنی `Volvo,BMW,Toyota` را نیز بیان می‌کند.

تنها یک سوال باقی می‌ماند، اعداد واقع شده در علامت‌های [] بیان‌کننده چه هستند؟

در آرایه‌ها جایگاه عناصر از ۰ شمارش می‌شود، به همین دلیل در علامت‌های [] اعداد صفر و سپس ۱ و ۲ را چاپ نموده است.

در اینجا تنها اشاره کوچکی به قوانین آرایه‌ها کردیم تا در بخش‌های آتی بیشتر راجع به آرایه‌ها توضیح دهیم.

PHP در Object داده‌های

یک شی نوعی داده است که اطلاعاتی را درباره چگونگی پردازش داده‌ها ذخیره می‌کند.



در پی اچ پی، شی باید به صورت واضح بیان شده باشد. ابتدا ما باید یک کلاس برای یک شی اعلام کنیم، که با کلمه کلیدی `class` بیان می شود که شامل خواص، ویژگی ها و روش ها است. سپس با معرفی نوع داده در کلاس شی به استفاده از آن در مواقعی که می خواهیم، م پردازیم. در ادامه تدریس با مدل نوشتن `class` و `Obj` (آبجکت) بیشتر آشنا می شویم.

```
<?php

class Car

{

var $color;

function Car($color="green")

{

$this->color = $color;

}

function what_color()

{

return $this->color;

}

}

?>
```




داده های NULL در PHP

تهی نشان دهنده این است که آن داده ها در PHP ارزش ندارد و NULL تنها روش نشان دادن این است که آن مقدار ارزشی برابر با تهی دارد. این دستور برای تمایز بین رشتههایی که خالی هستند و ارزش های پوچ در پایگاه داده استفاده می شود.

```
<html>
```

```
<body>
```

```
<?php
```

```
$x="Hello world!";
```

```
$x=null;
```

```
var_dump($x);
```

```
?>
```

```
</body>
```

```
</html>
```

خروجی دستور بالا به صورت زیر است:



NULL

تنها `var_dump` نوع داده ها در PHP را بیان می کند ولی برای آن هیچ ارزشی قائل نمی شود

داده تابع رشته ای `addslashes`

این تابع با توجه به اسم تابع رشته ای `addslashes` باید اسلشی را در بین کاراکترها اضافه نماید. این دستور تمام حروف خواسته شده را در متن شناسایی می کند و قبل از آن علامت را م یگذارد. با یک مثال بیشتر با این تابع آشنا شویم:

در این مثال می خواهیم قبل از `W` از یک اسلش استفاده کنیم.

```
<html>
```

```
<body>
```

```
<?php
```



```
$str = addslashes("Hello World!","W");
```

```
echo($str);
```

```
?>
```

```
</body>
```

```
</html>
```

خروجی دستور بالا را مشاهده کنید:

```
!Hello \World
```

در این متن رشته ای تنها یک حرف W داشتیم به همین دلیل تنها یک علامت در خروجی مشاهده م یکنید.

یک مثال دیگر ببینید، در این مثال یک جمله داریم ولی می خواهیم در بخش های مختلف از بک اسلش را استفاده کنیم.

```
<html>
```

```
<body>
```

```
<?php
```



```
$str = "Welcome to my humble Homepage!";
```

```
echo $str."<br>";
```

```
echoaddslashes($str,'m')."<br>";
```

```
echoaddslashes($str,'H')."<br>";
```

```
?>
```

```
</body>
```

```
</html>
```

همان طور که در مثال م ببینید یک بار قبل از حروف m و بار دیگر قبل از حروف H استفاده از خواسته شده است.

در این متن ۴ حرف m مشاهده می شود که قبل هر یک از آنها، بک اسلش استفاده شده است.

نکته: این دستور تنها برای حروف کاربرد دارد.

نکته: این دستور برای بعضی حروف عمل نمی کند زیرا در php علامت قبل از این حروف خود یک سری دستور هستند.

این نکته شامل حروف زیر است:



دوستان یه مطلب در مورد متغیر ها یادمرفت بگم مهم بود اینم ببینید

تغییر نوع داده ها به صورت مستقیم در PHP :

همانطور که در بخش های پیشین گفته بودیم ، زبان PHP شامل چندین نوع ، نوع داده ای میباشد ، از قبیل : integer , array , object , string و... در این بخش قصد داریم پس از تعریف متغیر ، در هنگام نیاز ، نوع داده ای خود را با استفاده از تابع های از پیش آماده شده در زبان php تبدیل نمائیم.

داده ها در زبان php ، بسیار ساده هستند ، و البته هوشمند ، اما شاید از خودتان بپرسید چرا از کلمه هوشمند استفاده کردم ؟؟؟ برای درک هوشمندی داده ها در php به مثال زیر توجه نمائید :

```
<?php
```

```
$result=50; // integer
```

```
$string = "100"; // string
```

```
echo $output = $result + $string;
```

```
?>
```

عد از اجرا کردن کد فوق ، حتما خروجی را مشاهده نموده اید ، که برابر با 150 می باشد ، اما چرا 150 ؟؟؟



در قسمت اول کد فوق ، ما یک متغیر با مقدار 50 تعریف نموده ایم و در خط دوم یک متغیر دیگر با مقدار رشته 100 ، توجه کنید رشته 100 تعریف نموده ایم (این متغیر در حال حاضر یک رشته محسوب می شود و یک عدد نیست!) و در خط آخر کد ، ما دو متغیر را با یکدیگر جمع نموده ایم ، سپس در یک متغیر قرار داده ایم و آن را چاپ کرده ایم.

اما نکته قابل اینجاست که در زمان جمع کردن دو متغیر ، php به صورت هوشمندانه تبدیل نوع را انجام می دهد و متغیر رشته ای ما را به یک integer تبدیل می کند.

حال به مثال زیر نیز توجه نمائید :

به نظر شما خروجی کد فوق چیست؟؟؟ در خط اول این کد ، ما یک متغیر با مقدار 7learn تعریف نموده ایم ، سپس یک متغیر دیگر با مقدار 20 ایجاد نموده ایم اما وقتی که دو متغیر پس از اعمال جمع ، چاپ نموده ایم

خروجی به ما 27 نمایش داده است ، پس شاید تا به حال متوجه شده باشید که php از رشته صرف نظر نموده و فقط اعداد را با یکدیگر جمع می کند.

www.ILikePHP.ir