

آموزش PHP (پی اچ پی) – مقدمه

PHP (پی اچ پی) یک زبان اسکریپت نویسی سمت سرور است و می تواند، ابزاری قدرتمند برای ساخت صفحات پویا و داینامیک بحساب آید

- ✚ گستردگی استفاده
- ✚ رایگان بودن (بدون اهمیت در ایران)
- ✚ جایگزینی مناسب برای ASP میکروسافت

آموزش آسان PHP (پی اچ پی) با مثال های فراوان

مثال های فراوان در آموزش PHP، باعث می شود که PHP را به آسانی فراگیرید، در این مثال ها، هم کدهای PHP و هم خروجی آن همزمان نمایش داده می شود:

```
<!DOCTYPE html>
<html>
<body>

<?php
echo "My first PHP script!";
?>

</body>
</html>
```

خروجی کد بالا:

```
My first PHP script!
```

پیشنیازهای آموزش PHP (پی اچ پی)

پیشنهاد می شود اگر با آموزش های زیر آشنا نیستید، قبل از ادامه آموزش PHP، آنها را مطالعه فرمایید:

- آموزش HTML
- آموزش CSS
- آموزش JavaScript

آموزش PHP (پی اچ پی): چیست؟

- PHP سرنام وازگان "PHP Hypertext Preprocessor" بحساب می آید.
- PHP یک نرم افزار کد باز است. (Open Source)
- PHP زبان اسکریپتی سمت سرور می باشد. (server-side)
- PHP را می توانید بصورت رایگان دانلود و استفاده کرد.

PHP یک زبان جالب و معروف

- PHP آنقدر قدرتمند هست که بتواند هسته بزرگترین سیستم محتوای وب (Joomla) را تشکیل دهد.
- PHP به اندازه کافی توانایی دارد که یکی از بزرگترین جوامع مجازی (Facebook) را اجرا کند.
- همچنین PHP بقدری آسان است که بتواند مورد استفاده تازه کاران قرار بگیرد.

آموزش PHP (پی اچ پی): فایل PHP چیست؟

- فایل های PHP می تواند شامل متن، تگ های HTML، خصوصیت های CSS، دستورات JavaScript و دستورات PHP باشد.
- دستورات PHP سمت سرور اجرا می شوند و نتیجه بصورت HTML به مرورگر کاربر ارسال می شود.
- فایل های PHP با پسوند .php ذخیره می شوند.

آموزش PHP (پی اچ پی): قابلیت های PHP

- PHP می تواند صفحات با محتوای داینامیک را ایجاد کند.
- PHP می تواند سمت سرور فایل ها را ایجاد کند، حذف کند، باز کند، بخواند، ویرایش کند و ببندد.
- PHP می تواند داده های فرم را جمع آوری کند.
- PHP می تواند Cookie ها را ارسال و دریافت کند.
- PHP می تواند داده ها را در پایگاه داده اضافه، حذف و ویرایش کند.
- PHP می تواند دسترسی کاربران وب سایتان را به بعضی از صفحات محدود کند.
- PHP می تواند داده ها را رمزنگاری کند.
- با PHP هیچ محدودیتی در تولید HTML ندارید. شما می توانید تصاویر، فایل های PDF و حتی فایل های Flash را ایجاد نمایید. همچنین می توانید هر متنی مانند XHTML و XML را در خروجی چاپ نمایید.

آموزش PHP (پی اچ پی): MySQL چیست؟

- MySQL محصول گروه MySQL AB و یک پروژه Open Source است.
- قابلیت حمل بر روی سیستم عامل های مختلف از قبیل Linux و windows و غیره را دارد.
- MySQL یک سیستم چند کاربره است، یعنی چندین کاربر به طور همزمان می توانند به یک بانک اطلاعاتی MySQL دسترسی داشته باشند.
- MySQL هم برای برنامه های بزرگ و هم کوچک کاربرد دارد.
- استفاده و دانلود MySQL رایگان است

آموزش PHP (پی اچ پی): چرا PHP؟

- یکی از برترین مزایای PHP (پی اچ پی) سازگاری آن با اکثر سیستم عامل ها و نرم افزارهای وب سرور مانند IIS و Apache است.
- PHP (پی اچ پی) ساختار و ترکیبی بسیار شبیه C دارد.
- استفاده و دانلود PHP (پی اچ پی) رایگان است.
- PHP (پی اچ پی) یکی از سریع ترین زبان ها در نوع خود است.

در آموزش PHP از کجا شروع کنم؟

- PHP (پی اچ پی) برای اجرا نیاز به یک Web-Server دارد، ساده ترین Web-Server برای شما احتمالاً IIS خواهد بود. (PHP & IIS)
- طبیعتاً برای اجرای فایل های PHP (پی اچ پی) نیاز به یک مرورگر وب دارید که مسلماً رایجترین آن IE است.
- برای ایجاد فایل های PHP (پی اچ پی)، نیاز به یک ویرایشگر ساده متنی مثل Notepad یا Wordpad دارید.
- PHP (پی اچ پی) و در صورت نیاز MySQL را نصب کنید.
- در ادامه آموزش PHP، خواهید دید که با نصب بسته XAMPP، همزمان PHP و MySQL و Apache نصب خواهد شد.

آموزش PHP-نصب PHP

به چه چیزی نیاز دارم؟

- برای شروع کار با PHP می توانید یکی از موارد زیر را استفاده نمایید:
- می توانید از یک وب هاست که PHP و MySQL را پشتیبانی می کند، استفاده نمایید.
- یک web server مانند Apache، روی کامپیوترتان نصب کنید و سپس PHP و MySQL را نصب نمایید.
- می توانید از بکج هایی مانند Xampp یا Wamp استفاده نمایید.

1- وب هاستی که PHP را پشتیبانی می کند

اگر سرور شما PHP را پشتیبانی می کند به چیز دیگری نیاز ندارید.
فقط باید یک فایل PHP ایجاد نمایید و آنرا در دایرکتوری وب قرار دهید. سرور بطور اتوماتیک آنرا ترجمه خواهد کرد.
بخاطر اینکه PHP رایگان است و اغلب سرورها آنرا پشتیبانی می کنند نیاز به کامپایل یا نصب ابزار های اضافی ندارید.
به هر حال اگر سرور شما PHP را پشتیبانی نمی کند، باید PHP را نصب نمایید.

2- نصب جداگانه Web Server و PHP و MySQL

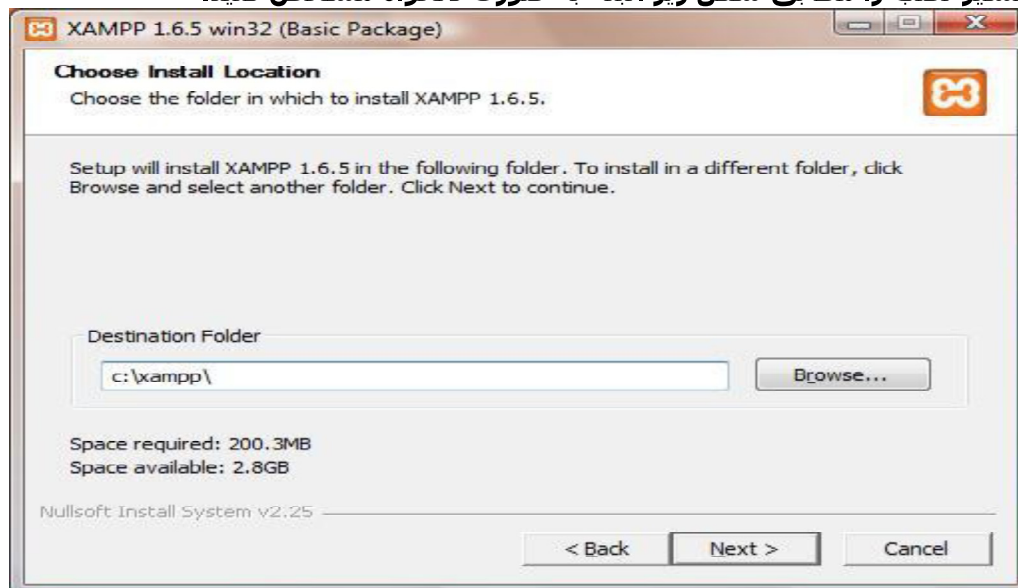
اگر کامپیوترتان، PHP را پشتیبانی نمی کند باید کارهای زیر را انجام دهید:
نصب وب سرور (Apache)
نصب PHP
نصب بک پایگاه داده، مانند MySQL
در لینک روبرو، آموزش چگونگی نصب PHP به خوبی توضیح داده شده است:
<http://php.net/manual/en/install.php>
دانلود Apache Server:
برای دانلود رایگان Apache از لینک روبرو استفاده کنید: <http://httpd.apache.org/download.cgi>
دانلود PHP:
برای دانلود رایگان PHP از لینک روبرو استفاده کنید: <http://www.php.net/downloads.php>
دانلود MySQL :
برای دانلود رایگان MySQL از لینک روبرو استفاده کنید: <http://www.mysql.com/downloads>
توجه: به جای نصب جداگانه Apache, MySQL, PHP می توانید از پکیج هایی مانند Xampp یا Wamp استفاده نمایید که کار شما را بسیار ساده تر می نمایند، البته در مطلب PHP & IIS چگونگی اجرای فایل های PHP روی IIS توضیح داده شده است.

3- استفاده از Xampp یا Wamp

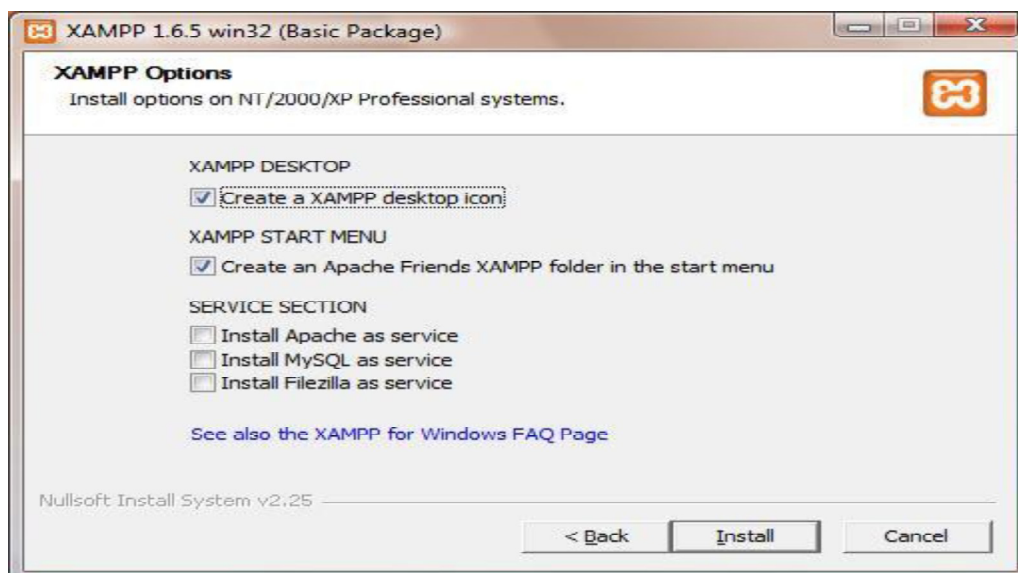
همانطور که قبلاً گفته شد برای شروع کار با php نیاز به نصب مفسر آن داریم که پکیج Xampp برای این منظور مناسب می باشد.
مراحل نصب XAMPP:
برای دانلود رایگان XAMPP از لینک روبرو استفاده کنید:
<http://www.apachefriends.org/en/xampp.html>
1- در اولین مرحله، خوش آمدگویی داریم، پس Next را مطابق شکل زیر کلیک کنید.



2- مسیر نصب را مطابق شکل زیر البته به صورت دلخواه مشخص کنید.



3- در صورتی که میخواهید Apache و MySQL به عنوان سرویس های ویندوز نصب بشوند گزینه های 'Install MySQL as Service' و 'Install Apache as Service' را تیک بزنید، در غیر این صورت اگر تیک 'Install as Service' را نزده باشید باید هر بار که ویندوز بالا می آید MySQL و Apache را فعال کنید، یعنی باید دکمه Start کنارشون را در کنترل پانل XAMPP کلیک کنید، در نهایت کلید Install را مطابق شکل زیر کلیک کنید.



نکته 1: در صورتیکه IIS روی سیستم شما نصب است و XAMPP را نصب کنید، در نهایت برای اجرای فایل های PHP به مشکل برخورد، بنابراین باید IIS را Stop کنید.

نکته 2: اگر SQL Server روی سیستم شما نصب است باید سرویس Reporting Services Configuration Manager آنرا Stop کنید تا XAMPP به درستی فایل های PHP را اجرا کند. در شکل زیر کنترل پنل Xampp را می بینید که می توانید Apache و MySQL را Start یا Stop کنید.

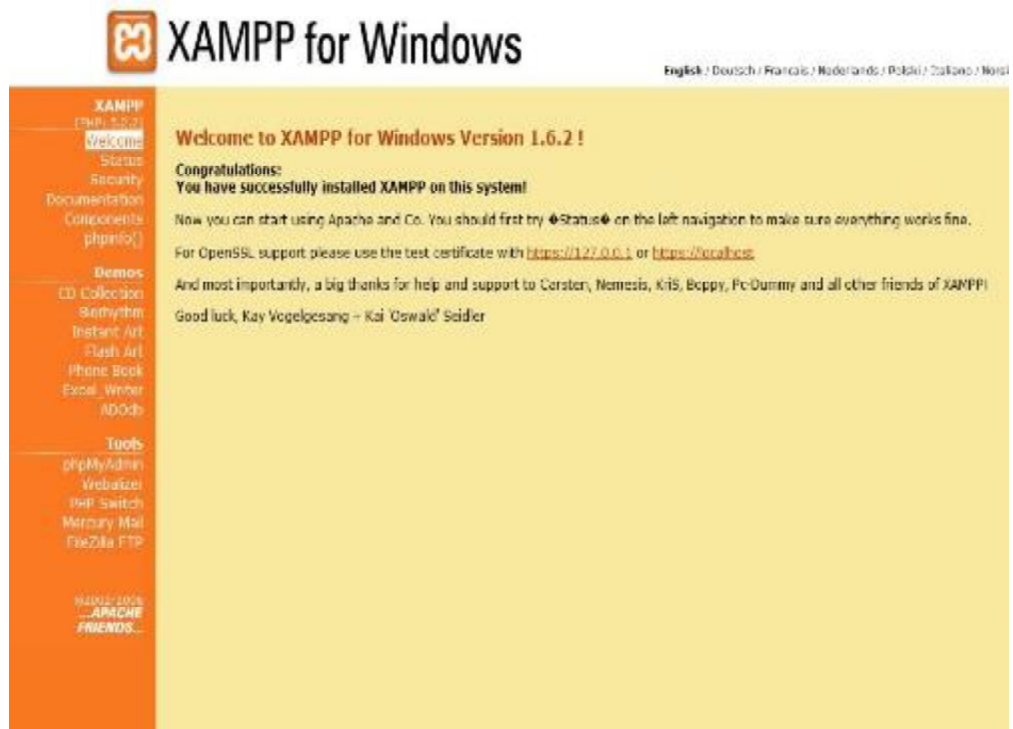


حالا مرورگرها را باز کنید و در آدرس بار مرورگر localhost یا 127.0.0.1 را تایپ کنید و یا می توانید در شکل بالا دکمه Admin را کلیک کنید اگر همه چیز خوب پیش رفته باشه صفحه زیر را خواهید دید که می توانید زبان صفحات Xampp را انتخاب کنید پس روی English کلیک کنید!



[English](#) / [Deutsch](#) / [Français](#) / [Nederlands](#) / [Polski](#) / [Italiano](#) / [Norsk](#) / [Español](#) / [中文](#) / [Português \(Brasil\)](#) / [日本語](#)

در نهایت صفحه ای مانند شکل زیر نمایش داده می شود.



ایجاد و اجرای یک فایل PHP:

یک فایل PHP ساده ایجاد کنید، به عنوان مثال در ادیتور NOTEPAD کد زیر را تایپ کنید و با نام hello و با پسوند php ذخیره کنید(hello.php):

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


فایل ایجاد شده در بالا را در مسیر C:\xampp\htdocs کپی کنید. (مسیر نصب XAMPP خواهد بود)
در آدرس بار مرورگر، آدرس localhost/hello.php را وارد کنید.
اگر همه چیز درست پیش رفته باشد، خروجی زیر را خواهید داشت:

```
Hello
```

ساختار دستورات در PHP

دستورات PHP همیشه با `<?php` آغاز و با `>` خاتمه می یابد. یک بلوک اسکریپتی PHP می تواند در هر جایی از کد HTML قرار بگیرد. در یک سرور که به تکنیک مختصر نویسی مجهز است می توان بلوک اسکریپتی php را با `<?>` شروع و با `>` خاتمه داد.

توجه: برای داشتن حداکثر سازگاری، ما حالت استاندارد (`<?PHP`) را پیشنهاد می کنیم.
یک فایل php به طور معمول ترکیبی از تگ های HTML و تعدادی کد های اسکریپتی PHP است.
مثال: در مثال زیر، متن "Hello world" توسط دستور echo چاپ می شود:

```
<html>
<body>
<?php
echo "Hello World";
?>
</body>
</html>
```

هر خط در php باید با علامت (;) پایان پذیرد. این علامت یک جدا کننده است و برای تمیز دادن یک مجموعه از دستورات عمل ها از دیگری استفاده می شود.

دو دستور پایه ای برای ارسال خروجی متنی وجود دارد:

echo

print

در مثال بالا از دستور echo برای فرستادن رشته ی "hello world" به خروجی استفاده شده است.
نکته: این فایل باید دارای پسوند php باشد. اگر پسوند فایل html باشد کد php اجرا نخواهد شد.

توضیحات در PHP (Comment)

در php از // برای یک خط comment و از /* */ برای یک بلوک بزرگ از توضیحات استفاده می کنیم.

```
<html>
```



```
<body>
<?php
//This is a comment
/*
This is
a comment
block
*/
?>
</body>
</html>
```

PHP یک زبان غیرحساس به حروف بزرگ و کوچک (case-insensitive)

در PHP، هیچکدام از توابع، کلاس ها و کلمات کلیدی (مانند if و while و ...) به حروف بزرگ و کوچک، حساس نیستند.

در مثال زیر، هر سه دستور echo، صحیح و یکسان است:

```
<!DOCTYPE html>
<html>
<body>
<?php
ECHO "Hello World!<br>";
echo "Hello World!<br>";
EcHo "Hello World!<br>";
?>
</body>
</html>
```

خروجی کد بالا:

```
Hello World!
Hello World!
Hello World!
```

اما، در PHP تمام متغیرها به حروف بزرگ و کوچک حساس اند. (case-sensitive)

در مثال زیر، تنها دستور اول مقدار متغیر \$color را نمایش می دهد: (دلیل این اتفاق، این است که با سه متغیر \$color و \$COLOR و \$coLOR مانند سه متغیر متفاوت برخورد می شود)

```
<!DOCTYPE html>
<html>
<body>
<?php
$color="red";
echo "My car is " . $color . "<br>";
echo "My house is " . $COLOR . "<br>";
echo "My boat is " . $coLOR . "<br>";
?>
</body>
</html>
```

خروجی کد بالا:

My car is red
My house is
My boat is

متغیرها، ظرفی برای ذخیره اطلاعات هستند.

```
<?php
$x=5;
$y=6;
$z=$x+$y;
echo $z;
?>
```

خروجی کد بالا:

11

آیا خبر را از زمان مدرسه به یاد دارید؟

$x=5, y=6, z=x+y$

آیا به یاد می آورید که یک حرف (مانند x) می تواند برای نگهداری یک مقدار (مثل 5) به کار رود و شما می توانید طبق اطلاعات بالا مقدار z را برابر 11 ارزیابی کنید.

این حروف، متغیر نامیده می شوند و آنها را می توان برای نگهداری مقادیر ($x=5$) و یا عبارات ($z=x+y$) استفاده کرد.

$x=5$

$y=6$

$z=x+y$

نکته : به متغیرها مانند طرفی برای نگهداری داده ها نگاه کنید.

متغیرها در PHP

- متغیرها برای ذخیره ی مقادیر مانند رشته های متنی، اعداد، یا آرایه ها استفاده می شوند.
- زمانی که یک متغیر را تعریف می کنید، می توانید آنرا در کدتان بارها و بارها استفاده کنید.
- در PHP متغیرها برخلاف C نوع خاصی ندارند، و از متغیرها بدون تعریف قبلی می توان استفاده نمود.
- هر متغیر با علامت \$ در ابتدای آن مشخص می شود.

روشی صحیح برای تعریف یک متغیر در PHP:

```
$var_name = value;
```

توجه: اغلب برنامه نویسان تازه کار در PHP فراموش می کنند که در ابتدای نام متغیر علامت \$ را درج کنند، در این صورت برنامه کار نخواهد کرد.

مثال 1: در زیر یک متغیر با محتویات رشته و یک متغیر دیگر با محتویات عدد تعریف شده است:

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

قواعد نامگذاری متغیرها در PHP

- نام یک متغیر باید با یک حرف یا یک "_" آغاز شود.
- نام یک متغیر می تواند شامل اعداد، حروف کوچک و بزرگ و "_" باشد. (A-Z,a-z,9-0)
- نام متغیرها نباید شامل فضای خالی باشد.
- نام متغیرها به حروف بزرگ و کوچک حساس است. (\$Y و \$y دو متغیر متفاوت اند)

نکته : بیاد داشته باشید که متغیرها در PHP به حروف کوچک و بزرگ حساس هستند.

ایجاد یا اعلان متغیرها در PHP

در PHP دستوری برای ایجاد متغیرها وجود ندارد.

یک متغیر زمانی که مقداری را به آن اختصاص می دهید ایجاد می شود.

```
<? Php
$txt="Hello world!";
$x=5;
$y=10.5;
Echo $txt;
Echo "<br>";
Echo $x;
Echo "<br>";
Echo $y;
?>
```

خروجی کد بالا:

```
Hello world!
5
10.5
```

PHP یک زبان بی ربط و بی قاعده (Loosely Type Language)

یک متغیر نیازی به تعریف پیش از استفاده ندارد PHP در
به طور PHP بگویند که متغیر از چه نوع داده ای است، PHP در مثال بالا، می بینید که مجبور نیستید به
خودکار متغیرها را به نوع داده ای مناسب شان تبدیل می کند که بستگی دارد چگونه مقدار دهی
شده باشند.

شما باید نوع و اسم متغیر را پیش از استفاده تعیین (Java یا C در یک زبان برنامه نویسی قوی مانند
متغیرها زمانی که شما از آنها استفاده می کنید به طور خود کار تعریف می شوند PHP کنید، در

محدوده متغیرها در PHP

، متغیرها در هر جایی از کد می توانند اعلان شوند PHP در
منظور از محدوده، بخشی از اسکریپت است که متغیر می تواند در آن محدوده استفاده شود.

سه محدوده مختلف برای متغیرها وجود دارد:

local 

global 🚩

static 🚩

متغیر های محلی (local) در PHP

متغیرهایی که داخل یک تابع اعلان شده اند، تنها داخل همان تابع، قابل دسترسی اند. (متغیرهای با قلمرو محلی)

متغیرهای محلی به محض اینکه عملیات تابع تکمیل شد، حذف می شوند.

نکته : شما می توانید در توابع مختلف، متغیرهای محلی با نام های یکسان داشته باشید. بخاطر اینکه متغیرهای محلی، تنها بوسیله تابعی که آنها را اعلان کرده است شناخته می شوند. در پیامورهای بعدی، درباره توابع بیشتر خواهید آموخت.

متغیر های عمومی (global) در PHP

متغیرهایی که خارج از توابع اعلان شده اند، عمومی می شوند، و در تمام اسکریپت بجز توابع، به آنها دسترسی خواهد بود.

زمانی که یک صفحه وب را ببینید، متغیرهای عمومی حذف خواهند شد.

در مثال زیر، می توانید محدوده متغیرهای محلی و عمومی را مشاهده نمایید:

```
<? php
$x=5; // global scope
Function myTest()
{
$y=10; // local scope
Echo "<p>Test variables inside the function :<p>";
Echo "Variable x is: $x";
Echo "<br>";
Echo "Variable y is: $y";
}
myTest();
echo "<p>Test variables outside the function:<p>";
echo "Variable x is: $x";
echo "<br>";
echo "Variable y is: $y";
?>
```

خروجی کد بالا:

Test variables inside the function:

Variable x is:

Variable y is: 10Test

variables outside the function:

Variable x is: 5

Variable y is:

در مثال بالا، دو متغیر \$x و \$y و یک تابع myTest() وجود دارد. متغیر \$x، بخاطر اینکه خارج از تابع اعلان شده، عمومی است و متغیر \$y، بخاطر اینکه داخل تابع اعلان شده، محلی است. زمانی که داخل تابع myTest()، مقادیر متغیرها را چاپ می کنیم، مقدار متغیر \$y بخاطر اینکه محلی است چاپ می شود، اما مقدار متغیر \$x، چونکه خارج از تابع اعلان شده است نمی تواند چاپ شود. سپس، زمانی که مقادیر متغیرها را خارج از تابع myTest() چاپ می کنیم، مقدار متغیر \$x چاپ می شود، اما مقدار متغیر \$y، چونکه داخل تابع mytest() اعلان شده است نمی تواند چاپ شود.

کلمه کلیدی global در PHP

با استفاده از کلمه کلیدی global، می توانید به متغیرهای عمومی در داخل یک تابع دسترسی داشته باشید.

برای انجام این کار، از کلمه کلیدی global، قبل از نام متغیر استفاده نمایید:

```
<?php
$x=5;
$y=10;

function myTest()
{
global $x,$y;
$y=$x+$y;
}

myTest();
echo $y; // outputs 15
?>
```

خروجی کد بالا:

15

در PHP، با استفاده از آرایه \$GLOBALS نیز می‌توانید به متغیرهای عمومی دسترسی داشته باشید. در واقع تمام متغیرهای عمومی در این آرایه ذخیره می‌شوند و با استفاده از نام متغیر بعنوان ایندکس آرایه می‌توانید به متغیرها، دسترسی داشته باشید.

بازنویسی مثال بالا، با استفاده از آرایه \$GLOBALS:

```
<?php
$x=5;
$y=10;

function myTest()
{
    $GLOBALS['y']=$GLOBALS['x']+$GLOBALS['y'];
}

myTest();
echo $y; // outputs 15
?>
```

خروجی کد بالا:

15

کلمه کلیدی static در PHP

بطور معمول زمانی که اجرای یک تابع به پایان می‌رسد، تمام متغیرهای آن نیز حذف خواهند شد. اما بعضی مواقع نیاز می‌شود که یک متغیر محلی برای استفاده‌های بعدی حذف نشود.

برای انجام این کار، از کلمه کلیدی static در ابتدای اعلان متغیر، استفاده نمایید:

```
<?php

function myTest()
{
    static $x=0;
```



```
echo $x;  
echo "<br />";  
$x++;  
}  
  
myTest();  
myTest();  
myTest();  
  
?>
```

خروجی کد بالا:

```
0  
1  
2
```

هر زمان که تابع صدا زده می شود، آخرین مقدار متغیر static، حفظ می شود. توجه: در مثال بالا، متغیر \$x محلی است و فقط در داخل تابع به آن دسترسی وجود دارد.

آموزش PHP-دستور echo/print

در PHP دو روش اصلی برای چاپ وجود دارد: echo و print تقریباً در هر مثالی از این بیاموز از echo (و print) استفاده شده است. بنابراین در این مطلب سعی شده است تا در این باره مقداری بیشتر توضیح داده شود.

دستور echo و print در PHP

تفاوت های echo و print:

- ✚ echo: یک یا بیشتر از یک آرگومان را می گیرد و مقداری را برنمی گرداند. برای اتصال دو رشته هم می توان از "." و هم از "," استفاده کرد.
- ✚ print: تنها یک آرگومان را می گیرد و همیشه مقدار 1 را برمی گرداند. برای اتصال دو رشته فقط می توان از "." استفاده کرد.

شباهت های echo و print:

- ✚ هر دو تابع نیستند.
- ✚ جزئی از ساختار زبانی هستند و می توان از آنها با یا بدون پرانتز استفاده نمود.

دستور echo در PHP

نمایش رشته ها

در مثال زیر، نحوه نمایش رشته های مختلف با استفاده از دستور echo نشان داده شده است:
(همچنین توجه داشته باشید که رشته چاپ شده می تواند شامل تگ های HTML باشد)

```
<?php
echo "<h2>PHP is fun!</h2>";

echo "Hello world!<br>";

echo "I'm about to learn PHP!<br>";

echo "This", " string", " was", " made", " with multiple parameters.";

?>
```

خروجی کد بالا:

```
PHP is fun! Hello world!
I'm about to learn PHP!
This string was made with multiple parameters.
```

نمایش متغیرها

در مثال زیر، نحوه چاپ رشته ها و متغیرها با دستور echo نشان داده شده است:

```
<?php
$txt1="Learn PHP";
$txt2="hojat.com";
$cars=array("Volvo","BMW","Toyota");

echo $txt1;
echo "<br>";
echo "Study PHP at $txt2";
echo "My car is a {$cars[0]}";

?>
```

خروجی کد بالا:

Learn PHP

Study PHP at hojat.com

My car is a Volvo

توجه: اگر متغیری درون علامت " قرار بگیرد مقدار متغیر به جای آن قرار می گیرد. اما اگر درون علامت ' قرار بگیرد مقدار متغیر جایگزین نمی شود.

دستور print در PHP

نمایش رشته ها

در مثال زیر، نحوه نمایش رشته های مختلف با استفاده از دستور print نشان داده شده است: (همچنین توجه داشته باشید که رشته چاپ شده می تواند شامل تگ های HTML باشد)

```
<?php
print "<h2>PHP is fun!</h2>";
print "Hello world!<br>";
print "I'm about to learn PHP!";
?>
```

خروجی کد بالا:

PHP is fun! Hello world!

I'm about to learn PHP!

نمایش متغیرها

در مثال زیر، نحوه ی چاپ رشته ها و متغیرها با دستور print نشان داده شده است:

```
<?php
$txt1="Learn PHP";
$txt2="hojat.com";
$cars=array("Volvo","BMW","Toyota");

print $txt1;
print "<br>";
print "Study PHP at $txt2";
print "My car is a {$cars[0]}";
```

```
?>
```

خروجی کد بالا:

```
Learn PHP
```

```
Study PHP at hojat.com
```

```
My car is a Volvo
```

آموزش PHP-انواع داده ها

رشته ها، اعداد صحیح، اعداد با ممیز شناور، بولین، آرایه ها، اشیاء، تهی (NULL)

رشته ها در PHP

رشته، یک توالی از کاراکترهاست، مانند "Hello world!"

هر متنی داخل کوتیشن، می تواند رشته باشد. می توانید از کوتیشن (') یا دابل کوتیشن (") استفاده کنید:

```
<?php
$x = "Hello world!";
echo $x;
echo "<br>";
$x = 'Hello world!';
echo $x;
?>
```

خروجی کد بالا :

```
Hello world!
```

```
Hello world!
```

اعداد صحیح در PHP

اعداد بدون ممیز، صحیح هستند.

قوانین اعداد صحیح:

- ✚ یک عدد صحیح، حداقل باید یک رقم داشته باشد.
- ✚ یک عدد صحیح، نمی تواند شامل کاما یا فاصله باشد.
- ✚ یک عدد صحیح، نباید ممیز داشته باشد.
- ✚ یک عدد صحیح، می تواند منفی یا مثبت باشد.
- ✚ یک عدد صحیح، می تواند در سه فرمت مشخص شود: decimal (بر مبنای 10)، hexadecimal (بر مبنای 16 - با پیشوند x0) و octal (بر مبنای 8 - با پیشوند 0)

در مثال زیر، فرمت های مختلف اعداد صحیح نشان داده شده است. با استفاده از تابع `var_dump()` می توانید مقدار و نوع متغیر را مشخص نمایید:

```
<?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);
?>
```

خروجی کد بالا:

```
int(5985)
int (-345)
int(140)
int(39)
```

اعداد با ممیز شناور در PHP

اعداد با ممیز شناور، اعدادی هستند که شامل یک ممیز هستند یا اعدادی که در قالب نماد ریاضی نشان داده می شوند.

در مثال زیر، اعداد اعشاری در فرمت های مختلف، نشان داده شده است. با استفاده از تابع `var_dump()`، می توانید مقدار و نوع متغیر را مشخص نمایید:

```
<?php
$x = 10.365;
```

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

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

```
$x = 2.4e3;
```

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

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

```
$x = 8E-5;
```

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

```
?>
```

خروجی کد بالا:

```
float(10.365)
```

```
float(2400)
```

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

داده های Boolean در PHP

مقدار داده های Boolean می تواند TRUE یا FALSE باشد.

```
$x=true;
```

```
$y=false;
```

داده ی Boolean، معمولاً در تست مشروط استفاده می شود. در پیامورهای بعدی درباره ی تست مشروط بیشتر خواهید آموخت.

آرایه ها در PHP

آرایه ها انواع خاصی از متغیرها به حساب می آیند که می توانند چندین داده را در قالب یک نام ذخیره کنند.

در مثال زیر، ابتدا یک آرایه ایجاد شده و سپس با استفاده از تابع `var_dump()`، مقدار و نوع هر سلول آرایه مشخص شده است:

```
<?php
```

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

خروجی کد بالا:

```
array(3) { [0]=> string(5) "Volvo" [1]=> string(3) "BMW" [2]=> string(6) "Toyota" }
```

درباره آرایه ها در پیاموزهای بعدی توضیحات بیشتری را خواهید دید.

اشیاء (object) در PHP

یک شیء نوع داده ای است که هم داده ها و هم اطلاعات مربوط به نحوه پردازش آنها را ذخیره می کند.

یک شیء در PHP، باید بطور صریح اعلان شود.

ابتدا باید کلاس شیء اعلان شود، برای این کار، از کلمه کلیدی class استفاده نمایید. یک کلاس، ساختاری است که می تواند شامل چندین property و method باشد.

برای دسترسی به property و method های یک کلاس، باید از آن کلاس یک نمونه بسازید:

```
<?php  
class My_class  
{  
    $name="Amir";  
    function SayHello()  
    {  
        print "Hello My Name is $this->name";  
    }  
}  
  
$obj=new My_Class();  
$obj->SayHello();  
?>
```

خروجی کد بالا:

```
Hello My Name is Amir
```

در پیاموزهای بعدی، درباره ی اشیاء بیشتر توضیح داده خواهد شد.

مقدار NULL در PHP

با استفاده از مقدار NULL، می توان نشان داد که یک متغیر مقدار ندارد. NULL تنها مقدار ممکن از نوع داده NULL است.

مقدار NULL، خالی یا پر بودن یک متغیر را نشان می دهد. همچنین بهتر است بدانید که در پایگاه داده بین NULL و رشته خالی تفاوت وجود دارد.

می توان متغیرها را با تنظیم مقدار NULL خالی کرد:

```
<?php
$x="Hello world!";
$x=null;
var_dump($x);
?>
```

خروجی کد بالا:

```
NULL
```

متغیرهای رشته ای در PHP

متغیرهای رشته ای برای ذخیره مجموعه ای از کاراکترها استفاده می شود.

در این فصل قصد داریم متداولترین توابعی که برای دستکاری رشته ها (String) در PHP استفاده می شود را نشان دهیم.

بعد از اینکه یک متغیر رشته ای (String Variable) ایجاد کردید می توانید آنرا دستکاری کنید، یک رشته می تواند مستقیماً در یک تابع استفاده شود یا می توانید آنرا در یک متغیر ذخیره کنید.

در کد PHP زیر کلمه "سلام" را به یک متغیر رشته ای (String Variable) بنام \$TXT انتساب داده و سپس آنرا چاپ می کنیم:

```
<? php
$txt = "سلام";
echo $txt;
?>
```

خروجی کد بالا:

```
سلام
```

حالا اجازه دهید تا بعضی از توابع و عمل کننده هایی (Operators) که برای دستکاری رشته ها بکار می رود را نشان دهیم:

یک ثابت، در واقع یک شناسه یا یک نام برای یک مقدار ساده است. در طول اسکریپت، مقدار ثابت نمی تواند تغییر نماید.

نام یک ثابت باید با یک حرف یا یک "_" آغاز شود. به علامت \$ قبل از نام ثابت نیازی نیست. توجه: برخلاف متغیرها، یک ثابت بطور اتوماتیک در طول تمام اسکریپت عمومی است.

تنظیم یک ثابت در PHP

با استفاده از تابع `define()`، می توانید یک ثابت را تنظیم نمایید. این تابع سه پارامتر می گیرد: پارامتر اول، نام ثابت را تعیین می کند، پارامتر دوم، مقدار ثابت و پارامتر اختیاری سوم، تعیین می کند که آیا نام ثابت به حروف بزرگ و کوچک حساس است یا خیر. (مقدار پیشفرض `false` است)

در مثال زیر، یک ثابت با نام `GREETING` و مقدار `"Welcome to hojat.com!"` تعریف شده است:

```
<?php
define("GREETING", "Welcome to hojat.com!");
echo GREETING;
?>
```

خروجی کد بالا:

```
Welcome to hojat.com!
```

در مثال زیر، نام ثابت به حروف بزرگ و کوچک حساس نیست (`greeting` و `GREETING` یکسان است):

```
<?php
define("GREETING", "Welcome to hojat.com!", true);
echo greeting;
echo GREETING;
?>
```

خروجی کد بالا:

```
Welcome to hojat.com!
```

```
Welcome to hojat.com!
```

آموزش PHP-عملگرها

عملگرها در PHP

عملگرها برای انجام عملیات و محاسبات روی مقادیر استفاده می شوند، در زیر لیستی از عملگرهای مختلف که در PHP استفاده می شود، آورده شده است:

عملگرهای ریاضی

این عملگرها عملی را روی دو متغیر یا دو عدد انجام می دهند.

نتیجه	مثال	توضیحات	عملگر
4	$x=2$ $x+2$	جمع	+
3	$x=2$ $5-x$	تفریق	-
20	$x=4$ $x*5$	ضرب	*
3 2.5	$15/5$ $5/2$	تقسیم	/
1 2 0	$5\%2$ $10\%8$ $10\%2$	باقیمانده تقسیم	%
$x=6$	$x=5$ $x++$	فرايش به میزان یک واحد	++
$x=4$	$x=5$ $x--$	کاهش به میزان یک واحد	--

عملگرهای انتسابی

هم ارز با	مثال	عملگر
$x=x+y$	$x+=y$	$+=$
$x=x-y$	$x-=y$	$-=$
$x=x*y$	$x*=y$	$*=$
$x=x/y$	$x/=y$	$/=$
$x=x.y$	$x.=y$	$.=$
$x=x\%y$	$x\%=y$	$\%=$
$x=y$	$x=y$	$=$

عملگرهای مقایسه ای

این عملگرها مقایسه ای بین دو متغیر انجام می دهند و نتیجه را به صورت true یا false برمی گردانند.

مثال	توضیحات	عملگر
$5==8$ returns false	برابری	$==$
$5!=8$ returns true	نابرابری	$!=$
$5>8$ returns false	بزرگتر از	$>$
$5<8$ returns true	کوچکتر از	$<$
$5>=8$ returns false	بزرگتر یا مساوی با	$>=$
$5<=8$ returns true	کوچکتر یا مساوی با	$<=$

عملگرهای منطقی

این عملگرها با مقدارهای true و false کار می کنند و آنها را با هم ادغام می کنند.

مثال	توضیحات	عملگر
$x=6$	and	$\&\&$