

<?PHP?>



«پیش نمایش»

کتاب جامع آموزش PHP

همراه با صدها مثال

- ◀ PHP سرنام واژگان Hypertext Preprocessor بمعنی "پردازنده ابرمتن" و یکی از محبوبترین زبانهای برنامه‌نویسی سمت سرور محسوب می‌شود.
- ◀ با استفاده از زبان برنامه‌نویسی PHP می‌توان صفحات وب را طراحی و توسعه داد.
- ◀ این زبان بدلیل سادگی، یادگیری آسان، و انعطاف‌پذیری بالا، به سرعت جایگاه خود را در صنعت نرم‌افزار پیدا کرده است.

نسخه 1.2

۱-۱ - مقدمه PHP	۶
۱-۲ - آموزش نصب PHP	۹
۱-۳ - ساختار دستورات در PHP	۱۶
۱-۴ - متغیرها در PHP	۲۰
۱-۵ - تفاوت دستور Echo و Print	۲۹
۱-۶ - انواع داده ها در PHP	۳۲
۱-۷ - متغیرهای رشته ای در PHP	در فایل اصلی
۱-۸ - ثابت ها (Constant) در PHP	در فایل اصلی
۱-۹ - عملگرها در PHP	در فایل اصلی
۱-۱۰ - جملات شرطی در PHP	در فایل اصلی
۱-۱۱ - دستور Switch در PHP	در فایل اصلی
۱-۱۲ - آرایه ها در PHP	در فایل اصلی
۱-۱۳ - حلقه ها در PHP	در فایل اصلی
۱-۱۴ - حلقه for در PHP	در فایل اصلی
۱-۱۵ - تعریف توابع در PHP	در فایل اصلی
۱-۱۶ - فرم های HTML در PHP	در فایل اصلی
۱-۱۷ - متغیر \$_GET در PHP	در فایل اصلی
۱-۱۸ - متغیر \$_POST در PHP	در فایل اصلی

۲-۱ - کار با فرم های PHP	۳۹
--------------------------------	----

۲-۲ - اعتبارسنجی فرم در PHP	۴۱
۲-۳ - فیلدهای الزامی در PHP	در فایل اصلی
۲-۴ - اعتبارسنجی ایمیل در PHP	در فایل اصلی
۲-۵ - حفظ مقادیر فرم در PHP	در فایل اصلی

فصل ۳: آموزش پیشرفته PHP ۵۱

۳-۱ - تابع Date در PHP	۵۱
۳-۲ - دستور include در PHP	۵۴
۳-۳ - فایل ها در PHP	۵۹
۳-۴ - ارسال یا آپلود فایل در PHP	۶۳
۳-۵ - متغییر Cookie در PHP	در فایل اصلی
۳-۶ - متغیرهای جلسه (Session) در PHP	در فایل اصلی
۳-۷ - ارسال ایمیل در PHP	در فایل اصلی
۳-۸ - امنیت در کد ارسال ایمیل	در فایل اصلی
۳-۹ - رسیدگی یا مدیریت خطاها در PHP	در فایل اصلی
۳-۱۰ - بررسی استثنایا در PHP	در فایل اصلی
۳-۱۱ - فیلتر ورودی ها در PHP	در فایل اصلی

فصل ۴: آموزش پایگاه داده MySQL ۶۸

۴-۱ - مقدمه MySQL	۶۸
۴-۲ - ارتباط با پایگاه داده MySQL در PHP	۷۰
۴-۳ - ایجاد پایگاه داده در PHP	۷۵
۴-۴ - ایجاد جدول در MySQL	۷۹
۴-۵ - دستور Insert در PHP	در فایل اصلی

۴-۶ - آخرین ID درج شده.....	در فایل اصلی
۴-۷ - درج چند رکورد در MySQL.....	در فایل اصلی
۴-۸ - دستورات Prepare در PHP.....	در فایل اصلی
۴-۹ - دستور Select در PHP.....	در فایل اصلی
۴-۱۰ - فیلتر کردن رکوردها در PHP.....	در فایل اصلی
۴-۱۱ - مرتب کردن داده ها در PHP.....	در فایل اصلی
۴-۱۲ - دستور Update در PHP.....	در فایل اصلی
۴-۱۳ - دستور Delete در PHP.....	در فایل اصلی
۴-۱۴ - صفحه بندی داده ها در PHP.....	در فایل اصلی
۴-۱۵ - ایجاد یک ارتباط ODBC در PHP.....	در فایل اصلی

۸۱

فصل ۵: آموزش XML در PHP

۵-۱ - تجزیه کننده DOM چیست؟.....	۸۱
۵-۲ - تجزیه کننده Expat چیست؟.....	در فایل اصلی
۵-۳ - معرفی SimpleXML.....	در فایل اصلی

۸۵

فصل ۶: آموزش Ajax در PHP

۶-۱ - مقدمه AJAX.....	۸۵
۶-۲ - استفاده از AJAX در PHP.....	۸۷
۶-۳ - واکنشی اطلاعات پایگاه داده با AJAX.....	در فایل اصلی
۶-۴ - واکنشی اطلاعات فایل XML با AJAX.....	در فایل اصلی
۶-۵ - جستجوی پیشرفته در Ajax.....	در فایل اصلی
۶-۶ - سیستم نظرسنجی در AJAX.....	در فایل اصلی

۹۰	۷-۱ - کلاس ها در PHP
۹۲	۷-۲ - متغیرهای (Properties) کلاس در PHP
	۷-۳ - متد های (Method) کلاس در PHP..... در فایل اصلی
	۷-۴ - سازنده (Constructor) در PHP..... در فایل اصلی
	۷-۵ - ارث بری یک کلاس از کلاس دیگر..... در فایل اصلی
	۷-۶ - متغیرها و توابع Static در PHP..... در فایل اصلی
	۷-۷ - تجرد (Abstract) در PHP..... در فایل اصلی

فصل ۱: آموزش مقدماتی PHP

۱-۱ - مقدمه PHP

PHP یک زبان اسکریپت نویسی سمت سرور است و ابزاری قدرتمند برای ساخت صفحات پویا و داینامیک بحساب می آید.

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

بیاموز آسان PHP با مثال های فراوان

با استفاده از مثال های فراوانی که در این کتاب ارائه شده است می توانید PHP را به آسانی فراگیرید، در این مثال ها، هم کدهای PHP و هم خروجی آن همزمان نمایش داده می شود:

مثال - بیاموز PHP

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

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

</body>
</html>
```

My first PHP script!



خروجی کد بالا

پیش نیازهای بیاموز PHP

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

- [بیاموز HTML](#)
- [بیاموز CSS](#)
- [بیاموز JavaScript](#)

PHP چیست؟

- **PHP** سرنام واژگان "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.



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.

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.

چرا PHP ؟

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

آموزش PHP را از کجا شروع کنم؟

- **PHP** 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** نیاز به یک مرورگر وب دارید که مسلماً رایجترین آن IE است.
- برای ایجاد فایل های **PHP**، نیاز به یک ویرایشگر ساده متنی مثل Notepad یا Wordpad دارید.
- **PHP** و در صورت نیاز MySQL را نصب کنید.

در ادامه بیاموز **PHP**، خواهید دید که با نصب بسته XAMPP، همزمان PHP و MySQL و Apache نصب خواهد شد.

۱-۲- آموزش نصب PHP

به چه چیزی نیاز دارم؟

برای شروع کار با PHP می توانید یکی از موارد زیر را استفاده نمایید:

۱. می توانید از یک وب هاست که PHP و MySQL را پشتیبانی می کند، استفاده نمایید.
۲. یک web server مانند Apache، روی کامپیوترتان نصب کنید و سپس PHP و MySQL را نصب نمایید.
۳. می توانید از پکیج 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

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:** بجای 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

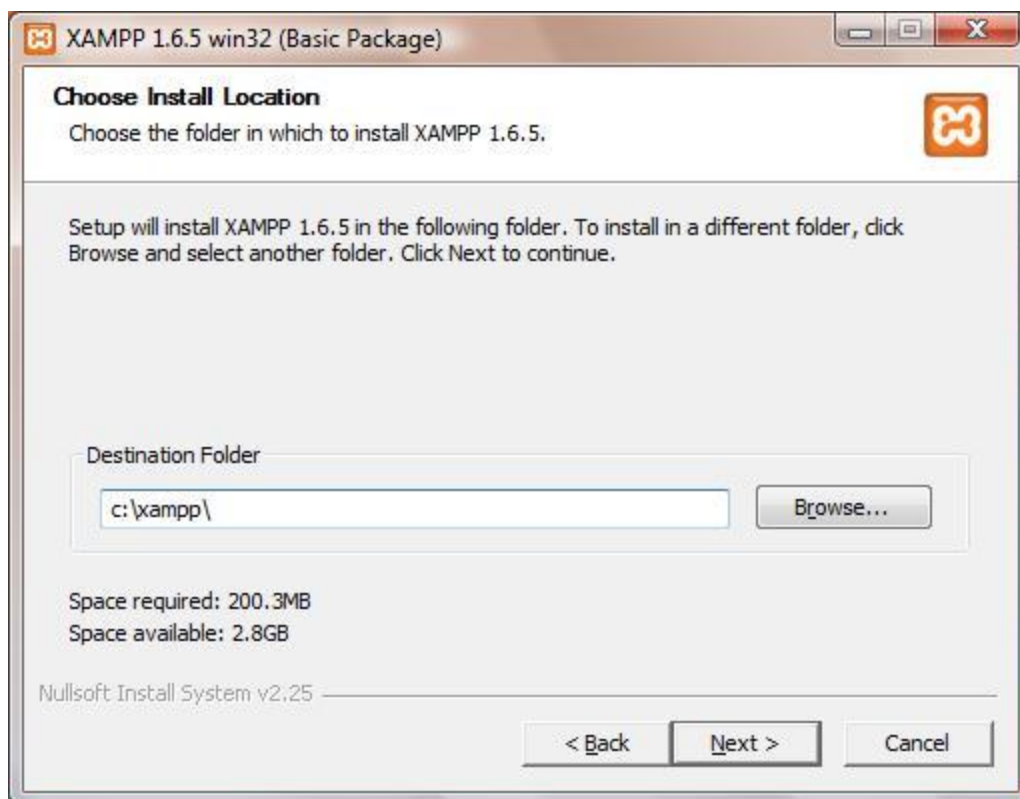
برای دانلود رایگان XAMPP از لینک روبرو استفاده کنید:

<https://www.apachefriends.org/download.html>

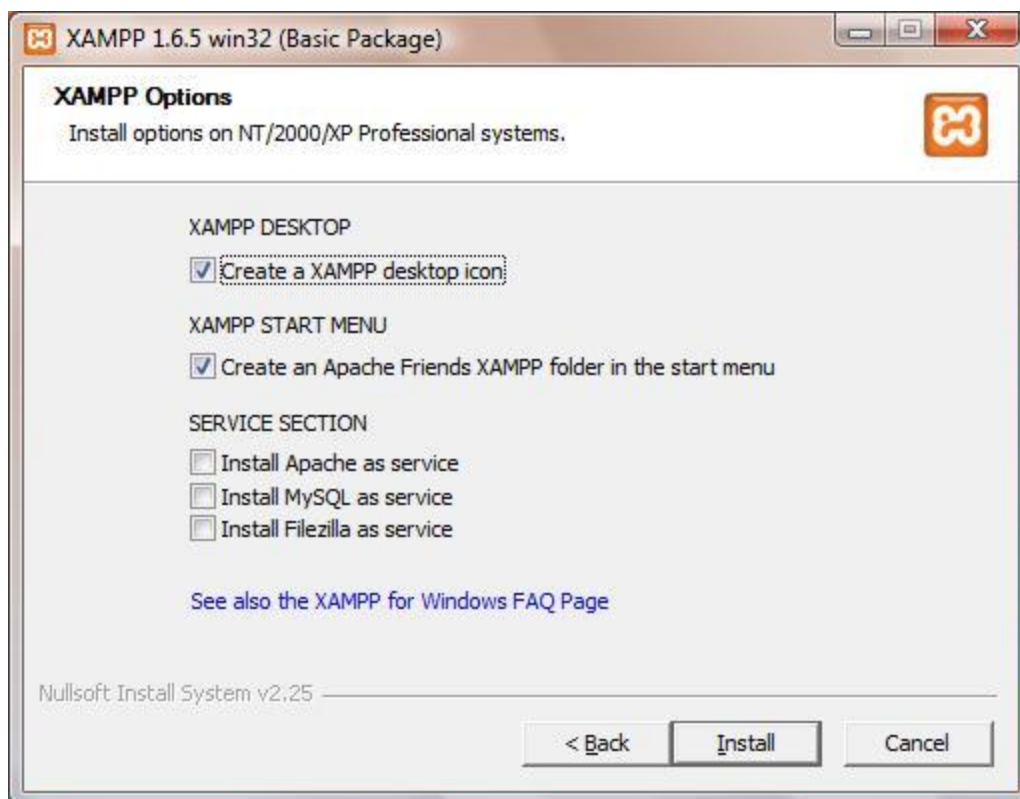
۱- در اولین مرحله، خوش آمدگویی داریم، پس Next را مطابق شکل زیر کلیک کنید.



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



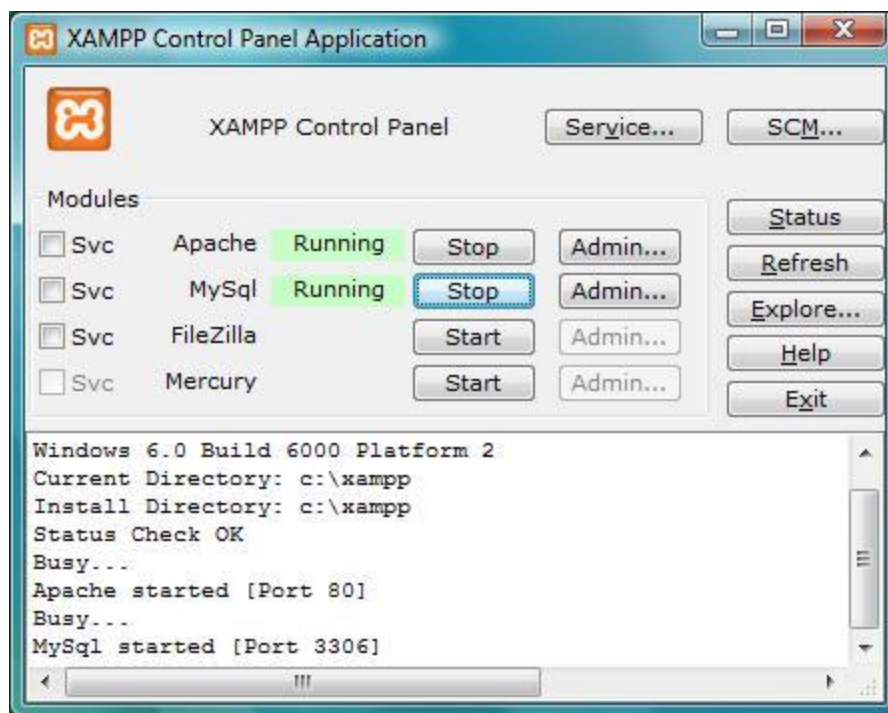
۳- در صورتی که می‌خواهید Apache و MySQL به عنوان سرویس های ویندوز نصب بشوند گزینه های Install MySQL as Service و Install Apache as Service را تیک بزنید، در غیر این صورت اگر تیک Install as Service را نزده باشید باید هر بار که ویندوز بالا می آید Apache و MySQL را فعال کنید، یعنی باید دکمه Start کنارشون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



نکته ۱: در صورتیکه IIS روی سیستم شما نصب است و XAMPP را نصب کنید، در نهایت برای اجرای فایل های PHP به مشکل برخورد خواهید خورد، بنابراین باید IIS را Stop کنید.

نکته ۲: 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 مرورگرتان 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 باشد صفحه زیر را خواهید دید که می توانید زبان صفحات Xampp را انتخاب کنید پس روی English کلیک کنید!



[English](#) / [Deutsch](#) / [Francais](#) / [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:

۱. 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";  
?>
```

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

Hello

۱-۳- 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.

```
<?php
?>
```

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:

۱. echo
۲. 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.

PHP (Comment) توضیحات در

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

```
<html>
<body>

<?php
//This is a comment

/*
This is
a comment
block
*/
?>
```

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>
```

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



خروجی کد بالا

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

در مثال صفحه ی بعد ، 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>
```

My car is red
My house is
My boat is



خروجی کد بالا

۱-۴- متغیرها در PHP

متغیرها، ظرفی برای ذخیره اطلاعات هستند.

مثال

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

11



خروجی کد بالا

آیا جبر 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 ۵) به کار رود و شما می توانید طبق اطلاعات بالا مقدار z را برابر ۱۱ 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.

$$x=5$$

$$y=6$$

$$z=x+y$$

به متغیرها مانند ظرفی 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 ۱: 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, ۰-۹)
- نام متغیر ها نباید شامل فضای خالی باشد.
- نام متغیرها به حروف بزرگ و کوچک حساس است (\$y و \$Y دو متغیر متفاوت اند)

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.

مثال

```
<?php
$txt="Hello world!";
$x=5;
$y=10.5;

echo $txt;
echo "<br>";
echo $x;
echo "<br>";
echo $y;
?>
```

```
Hello world!
5
10.5
```



خروجی کد بالا

بعد از اجرای مثال بالا، متغیر txt مقدار "Hello world!"، متغیر x مقدار "۵" و متغیر y مقدار "۱۰.۵" را در خود نگهداری می کنند.

💡 **توجه:** زمانی که می خواهید یک مقدار رشته ای را به یک متغیر اختصاص دهید از علامت ' استفاده نمایید.

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

در PHP یک متغیر نیازی به تعریف پیش از استفاده ندارد.

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

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

محدوده متغیرها در PHP

در PHP، متغیرها در هر جایی از کد می توانند اعلان شوند.

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
-

متغیرهای 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.



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 بیاموزهای بعدی، درباره توابع بیشتر خواهید آموخت.

متغیرهای 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 ببندید، متغیرهای عمومی حذف خواهند شد.

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

مثال

```
<?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";
?>
```

```
Test variables inside the function:
Variable x is:
Variable y is: 10
```

```
variables outside the function:
Variable x is: 5
Variable y is:
```



خروجی کد بالا

در مثال صفحه ی قبل ، 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:

مثال

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

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

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

15



خروجی کد بالا

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

بازنویسی مثال صفحه ی قبل، با استفاده از آرایه GLOBALS :

مثال

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

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

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

15



خروجی کد بالا

کلمه کلیدی static در PHP

بطور معمول زمانی که اجرای یک تابع به پایان می رسد، تمام متغیرهای آن نیز حذف خواهند شد. اما بعضی مواقع نیاز می شود که یک متغیر محلی برای استفاده های بعدی حذف نشود.

هر زمان که تابع صدا زده می شود، آخرین مقدار متغیر static، حفظ می شود.

💡 **توجه:** در مثال زیر، متغیر \$x محلی است و فقط در داخل تابع به آن دسترسی وجود دارد.

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

مثال

```
<?php  
  
function myTest()  
{  
    static $x=0;  
    echo $x;  
    echo "<br />";  
    $x++;  
}  
  
myTest();  
myTest();  
myTest();  
  
?>
```

0
1
2



خروجی کد بالا

۱-۵- تفاوت دستور Echo و Print

در PHP دو روش اصلی برای چاپ وجود دارد echo : و print

تقریباً در هر مثالی از این پیاموز 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 ۱ 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.

💡 نکته: echo سرعت بیشتری نسبت به print دارد.

دستور echo در PHP

نمایش رشته ها

در مثال صفحه ی بعد، 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)

مثال

```
<?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.";
?>
```

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



خروجی کد بالا

نمایش متغیرها

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

مثال

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

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

Learn PHP
Study PHP at Beyamooz.com
My car is a Volvo



خروجی کد صفحه
قبل

💡 **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)

مثال

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

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



خروجی کد بالا

نمایش متغیرها

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

مثال

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

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

```
Learn PHP
Study PHP at Beyamooz.com
My car is a Volvo
```



خروجی کد بالا

۱-۶- انواع داده ها در PHP

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;
?>
```

Hello world!
Hello world!



خروجی کد بالا

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 :

۱- Decimal (بر مبنای ۱۰)

۲- Hexadecimal (بر مبنای ۱۶)

۳- Octal (بر مبنای ۸)

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);
?>
```

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



خروجی کد بالا

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 صفحه ی بعد ، 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:

مثال

```
<?php
$x = 10.365;
var_dump($x);
echo "<br>";
$x = 2.4e3;
var_dump($x);
echo "<br>";
$x = 8E-5;
var_dump($x);
?>
```

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



خروجی کد بالا

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 بیاموزهای 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:

مثال

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

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



خروجی کد بالا

درباره آرایه ها در پیاموهای 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
{
    public $name="Amir";
    function SayHello()
    {
        print "Hello My Name is $this->name";
    }
}

$obj=new My_Class();
$obj->SayHello();
?>
```

Hello My Name is Amir



خروجی کد بالا

در بیاموزهای 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:

مثال

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

NULL



خروجی کد بالا

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

۱. [کتاب آموزش PHP](#) (کتاب پیش رو)

۲. [کتاب آموزش SQL](#)

۳. [کتاب آموزش Bootstrap](#)

۴. [کتاب آموزش JavaScript](#)

۵. [کتاب آموزش HTML](#)

۶. [کتاب آموزش CSS](#)

موفق و پاینده باشید

فصل ۲: آموزش کار با فرم ها

۲-۱ - کار با فرم های PHP

با استفاده از متغیرهای \$_GET و \$_POST ، می توانید سمت سرور مقادیر فرم HTML را دریافت نمایید.

یک فرم ساده HTML

در مثال زیر، یک فرم ساده HTML با دو فیلد ورودی و یک دکمه submit نشان داده شده است:

مثال

```
html>
<body>

<form action="welcome.php" method="post">
نام: <input type="text" name="name"><br>
ایمیل: <input type="text" name="email"><br>
<input type="submit" value="ارسال اطلاعات">
</form>

</body>
</html>
```

خروجی کد بالا:

نام:	
ایمیل:	
<input type="submit" value="ارسال اطلاعات"/>	

زمانی که کاربر، اطلاعات فرم بالا را پر می کند و روی دکمه "submit" کلیک می کند، اطلاعات فرم برای پردازش به یک فایل PHP بنام "welcome.PHP" ارسال می شود. اطلاعات فرم با متد POST ارسال می شود.

فایل welcome.PHP

برای نمایش اطلاعات ارسال شده به سرور، بصورت ساده می توانید همه متغیرها را چاپ نمایید:

مثال

```
<html>
<body>

نام:<?php echo $_POST["name"]; ?><br>
ایمیل : <?php echo $_POST["email"]; ?>

</body>
</html>
```

نام: beyamooz
ایمیل: admin@beyamooz.com



خروجی کد بالا

💡 **توجه:** مثال بالا را می توان از طریق متد `get` نیز انجام داد. که البته نحوه انجام کار در پیامزهای قبلی ذکر شده است.

کد بالا، یک مثال کاملاً ساده است. اما توجه داشته باشید که یک چیز بسیار مهم در آن از قلم افتاده است و آن اعتبار سنجی یا `Validate` اطلاعات ارسالی است. از این طریق اسکریپتتان امن تر خواهد بود.

در مثال بالا، اعتبار سنجی انجام نشده است، توجه داشته باشید که در این مثال فقط نحوه ی ارسال و دریافت اطلاعات نشان داده شده است.



اما در پیامز بعدی، نحوه ی پردازش اطلاعات فرم ارسال شده بصورتی که امنیت حفظ شود، نشان داده خواهد شد. اعتبار سنجی مناسب اطلاعات فرم، شما را از حمله هکرها حفظ خواهد کرد.

۲-۲- اعتبارسنجی فرم در PHP

در این پیاموز و همچنین پیاموز بعدی، نحوه ی اعتبارسنجی داده های فرم با استفاده از PHP آموزش داده خواهد شد.

اعتبارسنجی فرم در PHP

زمان پردازش فرم های PHP، به امنیت بیاندیشید!



در این پیاموز نحوه ی پردازش فرم های PHP بصورتی که حداقل موارد امنیتی که در ذهن دارید حفظ شود، نشان داده خواهد شد. اعتبار سنجی مناسب اطلاعات فرم، بمنظور جلوگیری از حمله هکرها دارای اهمیت است.

فرم HTMLی که در این پیاموز روی آن کار خواهیم کرد شامل فیلدهای مختلفی است: فیلدهای متنی الزامی و اختیاری، radio button و یک دکمه submit

در جدول زیر قوانین اعتبارسنجی ذکر شده است:

فیلد	قوانین اعتبارسنجی
نام	الزامی است. فقط باید شامل حروف الفبا و خط فاصله باشد.
ایمیل	الزامی است. باید شامل فرمت صحیح ایمیل باشد (همراه با علامت @ و .)
وب سایت	اختیاری است. اگر پر شد، باید شامل فرمت صحیح URL باشد.
توضیحات	اختیاری است. می تواند شامل چند خط متن باشد (textarea)
جنسیت	الزامی است. یکی از دو مورد باید انتخاب شود.

اجازه دهید تا در ابتدا نگاهی به کد HTML فرم بیاندازیم.

فیلدهای متنی

برای فیلدهای متنی "نام"، "ایمیل" و "وب سایت" از عنصر متنی `<input>` و برای فیلد "توضیحات" از عنصر `<textarea>` استفاده می کنیم:

```
نام: <input type="text" name="name">
ایمیل: <input type="text" name="email">
وب سایت: <input type="text" name="website">
توضیحات: <textarea name="comment" rows="5" cols="40"></textarea>
```

فیلد "جنسیت"

برای فیلد "جنسیت" از Radio Button استفاده می کنیم: (عنصر `<input type="radio">`)

```
جنسیت:
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male
```

عنصر `<form>`

کد HTML فرم شبیه زیر است:

```
<form method="post" action="<?php echo htmlspecialchars
($_SERVER["PHP_SELF"]);?>">
```

زمانی که روی دکمه submit کلیک می شود، اطلاعات با متد "post" به سرور ارسال می شود.

متغیر `SERVER_$`



متغیر `SERVER_$` یکی از متغیرهای عمومی از پیش تعریف شده است (آرایه `"PHP_SELF"` نام فایل اسکریپت جاری را برمی گرداند).

بنابراین زمانی که کاربر روی دکمه submit کلیک می کند، بجای اینکه داده ها به یک فایل متفاوت دیگر ارسال شود به خودش ارسال خواهد شد. در این روش، کاربر خطاها را در همان صفحه ای که فرم وجود دارد مشاهده خواهد کرد.

تابع `htmlspecialchars()`



با استفاده از تابع `htmlspecialchars()`، می توان کاراکترهای خاص را به HTML entity تبدیل نمود. منظور این است که کاراکترهایی مثل علامت کوچکتر (<) و بزرگتر (>) در پارامتر ورودی را به `<$` و `>$` تبدیل می کند. با این کار از حمله ی هکریایی که می خواهند از طریق تزریق HTML یا JavaScript اخلال ایجاد کنند، جلوگیری می شود.

یک تذکر مهم درباره امنیت فرم PHP

`PHP_SELF` می تواند توسط هکرها مورد استفاده قرار گیرد.

یک هکر می تواند در آدرس بار مرورگرش بعد از آدرس فایل، یک اسلش (/) قرار دهد و سپس دستورات XSS را برای اجرا تایپ کند.

یا Cross-site scripting



XSS یک نوع قابلیت آسیب پذیری امنیت کامپیوتر است. معمولاً در برنامه های کاربردی web بکار می رود. XSS، هکرها را قادر می سازد تا صفحات وب را از طریق تزریق اسکریپت سمت client هک کنند. دلیل به وجود آمدن این آسیب پذیری عدم اعتبارسنجی ورودی های کاربر می باشد، و مهاجم می تواند با تزریق اسکریپت های مخرب در سایت از این آسیب پذیری سو استفاده کند.

فرض کنید، فرم زیر را در یک فایل بنام "test_form.PHP" داریم:

```
<form method="post" action="<?php echo $_SERVER["PHP_SELF"];?>">
```

حالا اگر یک کاربر در آدرس بار مرورگرش "http://www.example.com/test_form.PHP" را وارد کند، کد بالا بصورت زیر ترجمه خواهد شد:

```
<form method="post" action="test_form.php">
```

خوب تا اینجا همه چیز خوب است.

اما درنظر بگیرید که کاربری URL زیر را در آدرس بار وارد کند:

```
http://www.example.com/test_form.php/%22%3E%3Cscript%3Ealert('hacked')%3C/script%3E
```

در این صورت کد بالا بصورت زیر ترجمه خواهد شد:

```
<form method="post"
action="test_form.php"/><script>alert('hacked')</script>
```

صفحه لود می شود، کد JavaScript اجرا می شود (کاربر یک جعبه پیغام خواهد دید). این فقط یک مثال ساده و بی ضرر است که نحوه هک کردن متغیر PHP_SELF را نشان می دهد.

توجه داشته باشید که هر کد JavaScript دیگری را می توان در تگ <script> قرار داد...! یک هکر می تواند کاربر را به یک فایل دیگر روی سروری دیگر redirect کند، و از طریق آن فایل، اطلاعات کاربر را ذخیره کند.

نحوه مقابله با هک از طریق "PHP_SELF"

با استفاده از تابع `htmlspecialchars()` ، می توان با هک از طریق "PHP_SELF" مقابله نمود.

کد آن شبیه زیر است:

```
<form method="post" action="<?php echo htmlspecialchars  
($_SERVER["PHP_SELF"]);?>">
```

تابع `htmlspecialchars()` ، کاراکترهای خاص را به HTML entity تبدیل می کند. حالا اگر کاربر بخواهد از متغیر "PHP_SELF" سوء استفاده کند، با نتیجه زیر روبرو خواهد شد:

```
<form method="post"  
action="test_form.php/&quot;&gt;&lt;script&gt;alert('hacked')&lt  
/script&gt;">
```

و از این طریق هیچ آسیبی وارد نخواهد شد!...

اعتبارسنجی داده های فرم با PHP

اولین کاری که باید انجام دهید این است که تمام متغیرها را به تابع `htmlspecialchars()` پاس دهیم.

حالا اگر کاربر تلاش کند که متنی مانند زیر را ارسال کند:

```
<script>location.href('http://www.hacked.com')</script>
```

در اینصورت اسکریپت بالا اجرا نخواهد شد، چونکه کاراکترهای خاص در متن بالا به HTML entity معادلشان تبدیل شده اند:

```
<script>location.href('http://www.hacked.com')</script>
```

حالا این کد برای نمایش در یک صفحه یا داخل یک ایمیل، امن شده است.

همچنین ما دو کار دیگر را هنگام ارسال داده ها به سرور انجام می دهیم:

۱. با استفاده از تابع `trim()` کاراکترهای غیرضروری (مثل: فاصله های اضافی، `tab` و خطوط خالی) را حذف می کنیم.

۲. با استفاده از تابع `stripslashes()`، بک اسلش ها (`\`) را حذف می کنیم

گام بعدی، ایجاد یک تابع، برای انجام تمام کارهای بالاست (بجای اینکه کدهای مربوط به این قسمت را بارها و بارها بنویسیم، مناسب تر است که از یک تابع استفاده کنیم)

ما این تابع را `test_input()` می نامیم.

حالا ما می توانیم به ازای هر متغیر `POST_$` مقدار آنرا با تابع `test_input()` چک کنیم و کد آن شبیه زیر است:

```
<!DOCTYPE HTML>
<html>
<head>
<style>
    span{min-width: 200px;float: right;}
</style>
</head>
<body style="direction:rtl;">

<?php
$name = $email = $gender = $comment = $website = "";
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    $name = test_input($_POST["name"]);
    $email = test_input($_POST["email"]);
    $website = test_input($_POST["website"]);
    $comment = test_input($_POST["comment"]);
    $gender = test_input($_POST["gender"]);
}
```

```

function test_input($data)
{
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>
<h2>PHP مثال اعتبارسنجی فرم ها در</h2>
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
<div><span>نام:</span><input type="text" name="name"></div>
<div><span>ایمیل:</span><input type="text" name="email"></div>
<div><span>وب سایت:</span><input type="text"
name="website"></div>
<div><span>توضیحات:</span><textarea name="comment" rows="5"
cols="40"></textarea>
</div>
<div><span>جنسیت:</span>
<input type="radio" name="gender" value="female">زن
<input type="radio" name="gender" value="male">مرد
</div>
</div><input type="submit" name="submit" value="ارسال
اطلاعات"></div>
</form>
<?php
if (isset($name) || isset($email) || isset($gender) ||
isset($comment) || isset($website))
{
    echo "<br /><h2>خروجی کدتان</h2>";
    echo "نام:$name";
    echo "<br />";
    echo "ایمیل: $email";
    echo "<br />";
    echo "وب سایت: $website";
    echo "<br />";
    echo "توضیحات: $comment";
    echo "<br />";
    echo "جنسیت: $gender";
}
?>
</body>
</html>

```


خروجی مثال صفحه ی قبل

مثال اعتبارسنجی فرم ها در PHP

نام:

ایمیل:

وب سایت:

توضیحات:

جنسیت: ☐ زن ☐ مرد

توجه داشته باشید که در ابتدای اسکریپت، با استفاده از متغیر "REQUEST_METHOD" نحوه ارسال داده های فرم را چک می کنیم. اگر نحوه ی ارسال داده های فرم، از طریق متد "POST" است، اطلاعات فرم پردازش خواهد شد وگرنه با یک صفحه خالی روبرو خواهیم شد.

🔗 **توجه:** برای بار اول که کاربر درخواست مشاهده فایل مثال بالا را به سرور ارسال می کند، بدلیل اینکه متد پیش فرض برای مشاهده صفحات از نوع get است، بنابراین شرط

`$_SERVER["REQUEST_METHOD"] == "POST"` درست نخواهد بود و دستورات داخل شرط اجرا

نخواهد شد. اما بعد از اینکه کاربر اطلاعات فرم را پر کرده و روی دکمه ارسال (submit) کلیک کرد، چون

ویژگی method فرم را با مقدار "post" تنظیم کرده ایم، شرط ذکر شده درست خواهد بود و دستورات

داخل آن اجرا خواهد شد.

اما در مثال بالا، تمام فیلدهای ورودی اختیاری است. حتی اگر کاربر هیچ کدام از فیلدها را پر نکند، اسکریپت بالا باز هم کار خواهد کرد.

در پیام بعدی برای فیلدهای الزامی، یک پیغام مناسب نمایش خواهیم داد.

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

۱. [کتاب آموزش PHP \(کتاب پیش رو\)](#)

۲. [کتاب آموزش SQL](#)

۳. [کتاب آموزش Bootstrap](#)

۴. [کتاب آموزش JavaScript](#)

۵. [کتاب آموزش HTML](#)

۶. [کتاب آموزش CSS](#)

موفق و پاینده باشید

فصل ۳: آموزش پیشرفته PHP

۳-۱ - تابع Date در PHP

این تابع، رشته ای از داده ها را به عنوان پارمتر می گیرد و نتایج آنها را به صورت زمان یا تاریخ نمایش می دهد.

در حقیقت با استفاده از این تابع می توان تاریخ یا زمان را در شکل های مختلف نشان داد.

نحوه استفاده:

```
date(format, timestamp)
```

پارامتر	توضیحات
format	الزامی است، فرمت تاریخ یا زمان را مشخص می کند.
timestamp	اختیاری است، یک برچسب زمان که باید به فرمتی که در پارامتر اول مشخص کرده ایم تبدیل شود (به صورت پیشفرض تاریخ جاری در نظر گرفته می شود).

بعضی از کاراکترهایی که می توان به عنوان پارامتر الزامی، در تابع date استفاده کرد:

- a - صبح یا عصر، نماینده am و pm است.
- A - صبح یا عصر، نماینده AM و PM است.
- B - زمان اینترنت سوئچ (swatch) که یک زمان جهانی است.
- C - تاریخ ISO 8601، تاریخ به صورت YYYY_MM_DD این کد فرمت در نسخه PHP 5 ارائه شده است.
- d - روز به صورت عدد دو رقمی از ۰۱ تا ۳۱
- D - روز هفته به صورت ۳ کاراکتر مانند: Mon و Sun
- F - ماه در فرمت متنی . محدوده آن از January تا December است.
- g - ساعت در فرمت ۱۲ ساعت از ۱ تا ۱۲
- G - ساعت در فرمت ۲۴ ساعت از ۰ تا ۲۳

- h - ساعت در فرمت ۱۲ ساعت از ۰۱ تا ۱۲
- H - ساعت در فرمت ۲۴ ساعت از ۰۰ تا ۲۳
- i - دقیقه به صورت عدد دو رقمی از ۰۰ تا ۵۹
- l - این مقدار بولی ساعت را یک ساعت به جلو و عقب می برد (در آخر تابستان و اول فروردین)
- j - روز را در یک ماه نشان می دهد. از ۱ تا ۳۱
- l - روز را در هفته از Sunday تا Monday مشخص می کند.
- L - این مقدار بولی سال کبیسه را مشخص می کند و اگر سال کبیسه باشد مقدار یک را بر می گرداند.
- m - ماه را در فرمت دو رقمی از ۰۱ تا ۱۲ مشخص می کند.
- M - ماه در فرمت ۳ کاراکتر مانند Jan : و Dec
- n - ماه در فرمت عددی بدون صفر از ۱ تا ۱۲
- o - تفاوت زمان منطقه جغرافیایی موجود و GMT را در واحد ساعت بیان می کند.
- r - زمان و تاریخ در فرمت RFC822
- s - ثانیه در محدوده ۰۰ تا ۵۹
- S - پسوند های ترتیبی اعداد در فرمت دو کاراکتر. مانند: st۱ یا th۲۳
- t - تعداد روز های ماه را از ۲۸ تا ۳۱ مشخص می کند.
- T - تنظیمات منطقه زمانی سرور در فرمت ۳ کاراکتر مانند EST
- U - تعداد کل ثانیه ها از اول ژانویه ۱۹۷۰ تاکنون را بیان می کند.
- w - روز هفته در یک رقم. ۰ برای یکشنبه و ۶ برای شنبه
- W - شماره هفته از اول سال. مانند ۲۱ برای نشان دادن هفته ۲۱ام سال
- Y - سال در فرمت دورقمی، ۰۵ برای ۲۰۰۵
- Y - سال در فرمت چهار رقمی، مانند ۲۰۰۶
- Z - روز را مشخص می کند. محدوده آن ۰ تا ۳۶۵ است.
- Z - افسست منطقه زمانی را بر حسب ثانیه بیان می کند. محدوده آن بین ۴۳۲۰۰- و ۴۳۲۰۰+ است.

💡 نکته: کاراکتر هایی که در لیست بالا نباشند، مستقیماً چاپ می شوند. مانند: / , و , و ...

مثال

```
<?php
echo date("Y/m/d") . "<br />";
echo date("Y.m.d") . "<br />";
echo date("Y-m-d") ;
?>
```

```
2009/05/11
2009.05.11
2009-05-11
```



خروجی کد بالا

پارامتر timestamp در تابع date

تابع `mktime()` یک برچسب زمان یونیکسی برمی گرداند که می توان از آن، به عنوان پارامتر اختیاری تابع `date()` استفاده کرد.

برچسب زمان یونیکسی: منظور تعداد ثانیه هایی است که از زمان (January 1 1970 00:00:00 GMT) تا زمانی که در تابع `mktime` مشخص شده است.

نحوه استفاده:

```
mktime(hour,minute,second,month,day,year,is_dst)
```

مثال: در مثال زیر با استفاده از تابع `mktime` و `date` تاریخ فردا را چاپ می کنیم:

مثال

```
<?php
$tomorrow = mktime(0,0,0,date("m"),date("d")+1,date("Y"));
echo "Tomorrow is ".date("Y/m/d", $tomorrow);
?>
```

Tomorrow is 2025/05/12



خروجی کد بالا
(البته هنگام اجرای
مثال)

۳-۲- دستور include در PHP

درج کردن یک فایل PHP در یک فایل PHP دیگر

شما می توانید با استفاده از دو تابع include یا require محتویات یک فایل PHP را داخل یک فایل PHP دیگر درج کنید، البته قبل از اینکه فایل دوم اجرا شود.

این دو تابع در همه موارد یکسان عمل می کنند، بجز در چگونگی برخورد با خطا:

- include: پیغام خطا را تولید می کند، اما ادامه اسکریپت اجرا خواهد شد.
- require: پیغام خطا را تولید می کند و اجرای اسکریپت متوقف خواهد شد.

موارد استفاده دو تابع include و require می تواند شامل موارد زیر باشد:

- توابع
- Header
- Footer
- المانهایی که در چندین صفحه استفاده می شود مانند منوها، کامپوننت تاریخ و ...

شما می توانید یک فایل استاندارد Header یا Footer و یا منو ایجاد کنید و در همه صفحات پروژه تان از آنها استفاده کنید، و موقعی که نیاز به تغییر داشت، تنها با تغییر یک فایل، کل پروژه تان را تازه سازی کنید.

💡 دستور `include_once` و `require_once`:

دو تابع بالا دقیقاً مثل `include` و `require` عمل می کنند، با این تفاوت که اگر فایل مورد نظر قبلاً خوانده شده است، دیگر خوانده نمی شود.

مثال ۱:

فرض کنید یک فایل Header با نام "header.PHP" دارید، برای اضافه کردن آن به صفحه ای از پروژه از کد صفحه ی بعد استفاده کنید:

```
<html>
<body>

<?php include("header.php"); ?>
<h1>Welcome to my home page!</h1>
<p>Some text.</p>

</body>
</html>
```

💡 دستورات `include`, `include_once`, `require` و `require_once` را می توان بدون پرانتز نیز بکار برد.

مثال ۲:

فرض کنید یک فایل منو طبق زیر با نام "menu.PHP" داریم، که باید به تمام صفحات پروژه اضافه شود:

```
<a href="/default.php">Home</a>
<a href="/tutorials.php">Tutorials</a>
<a href="/references.php">References</a>
<a href="/examples.php">Examples</a>
<a href="/about.php">About Us</a>
<a href="/contact.php">Contact Us</a>
```

در زیر فایل "menu.PHP" را به صفحه دلخواهمان اضافه کرده ایم:

```
<html>
<body>
<div class="leftmenu">
<?php include("menu.php"); ?>
</div>
<h1>Welcome to my home page.</h1>
<p>Some text.</p>
</body>
</html>
```


اگر بعد از اجرای فایل صفحه قبل در IE ، روی صفحه راست کلیک کرده و View Source را کلیک کنید، کد html شبیه زیر خواهیم داشت:

```
<html>
<body>

<div class="leftmenu">
<a href="/default.php">Home</a>
<a href="/tutorials.php">Tutorials</a>
<a href="/references.php">References</a>
<a href="/examples.php">Examples</a>
<a href="/about.php">About Us</a>
<a href="/contact.php">Contact Us</a>
</div>

<h1>Welcome to my home page!</h1>
<p>Some text.</p>

</body>
</html>
```

مثال ۳: خطا در تابع include:

در مثال زیر فایل wrongFile.PHP وجود ندارد و همان طور که مشاهده می کنید بعد از چاپ پیغام خطا ادامه اسکریپت اجرا می شود.

```
<html>
<body>

<?php
include("wrongFile.php");
echo "Hello World!";
?>

</body>
</html>
```

پیغام خطا:

```
Warning: include(wrongFile.php) [function.include]:  
failed to open stream:  
No such file or directory in C:\home\website\test.php on line 5
```

```
Warning: include() [function.include]:  
Failed opening 'wrongFile.php' for inclusion  
(include_path='.;C:\php5\pear')  
in C:\home\website\test.php on line 5
```

Hello World!

مثال ۴: پیغام خطا در تابع **require**:

```
<html>  
<body>  
  
<?php  
require("wrongFile.php");  
echo "Hello World!";  
?>  
  
</body>  
</html>
```

پیغام خطا:

```
Warning: require(wrongFile.php) [function.require]:  
failed to open stream:  
No such file or directory in C:\home\website\test.php on line 5  
  
Fatal error: require() [function.require]:  
Failed opening required 'wrongFile.php'  
(include_path='.;C:\php5\pear')  
in C:\home\website\test.php on line 5
```

بعد از ایجاد خطا، ادامه اجرای اسکریپت متوقف می شود، و در حقیقت خطی که قرار است Hello World را چاپ کند، اجرا نمی شود.

💡 پیشنهاد می شود از تابع `require` بجای `include` استفاده شود، چون اسکریپت ها بعد از برخورد با خطا نباید ادامه پیدا کنند.

۳-۳- فایل ها در PHP

تابع `fopen()` برای باز کردن فایل ها در PHP مورد استفاده قرار می گیرد.

پارامتر اول این تابع شامل نام فایلی است که می خواهیم باز شود و پارامتر دوم مشخص می کند که فایل در چه حالتی باز شود:

```
<html>  
<body>  
  
<?php  
$file=fopen("welcome.txt","r");  
?>  
  
</body>  
</html>
```

فایل ممکن است در یکی از حالت های زیر باز شود:

حالت	توضیحات
r	فقط خواندنی. از ابتدای فایل آغاز می شود
r+	خواندنی/نوشتنی. از ابتدای فایل آغاز می شود
w	فقط نوشتنی. فایل را باز و محتویات آن را پاک می کند، یا اگر فایلی وجود نداشت آن را ایجاد می کند
w+	خواندنی/نوشتنی. فایل را باز و محتویات آن را پاک می کند، یا اگر فایلی وجود نداشت آن را ایجاد می کند
a	افزودن(الحاق). فایل را باز و در انتهای آن می نویسد، یا اگر فایلی وجود نداشت آن را ایجاد می کند
a+	خواندنی/افزودنی. محتویات فایل را با نوشتن در انتهای آن حفظ می کند
x	فقط نوشتنی. یک فایل جدید ایجاد می کند. اگر فایل در حال حاضر وجود داشته باشد false و یک خطا برمیگرداند
x+	خواندنی/نوشتنی. یک فایل جدید ایجاد می کند. اگر فایل در حال حاضر وجود داشته باشد false و یک خطا برمیگرداند

🔑 نکته: اگر تابع `fopen()` قادر به باز کردن فایل مشخص شده نباشد، مقدار "۰" یا `false` را برمی گرداند.

مثال:

در مثال زیر اگر تابع `fopen()` قادر به باز کردن فایل مشخص شده نباشد یک پیغام تولید می کند:

```
<html>
<body>

<?php
$file=fopen("welcome.txt","r") or exit("Unable to open file!");
?>

</body>
</html>
```

بستن فایل در PHP

تابع `fclose()` برای بستن یک فایل باز استفاده می شود:

```
<?php
$file = fopen("test.txt","r");

//some code to be executed

fclose($file);
?>
```

چک کردن پایان فایل

تابع `feof()` بررسی می کند که آیا به پایان فایل (`eof`) رسیده ایم یا نه.

تابع `feof()` برای ایجاد حلقه در داده های با طول نامشخص مناسب است.

💡 **توجه:** شما نمی توانید از فایل هایی که در حالت `w` و `a` و `x` باز شده اند، بخوانید!

```
if (feof($file)) echo "End of file";
```

خواندن خط به خط یک فایل

تابع `fgets()` برای خواندن یک خط از یک فایل، مورد استفاده قرار می گیرد.

💡 **نکته:** بعد از هر بار فراخوانی این تابع، اشاره گر فایل، به خط بعدی جابه جا می شود.

مثال:

مثال زیر یک فایل را تا پایان فایل، خط به خط می خواند:

```
<?php
$file = fopen("welcome.txt", "r") or exit("Unable to open
file!");
//Output a line of the file until the end is reached
while(!feof($file))
{
    echo fgets($file). "<br />";
}
fclose($file);
?>
```

خواندن کاراکتر به کاراکتر یک فایل

تابع `fgetc()` برای خواندن یک کاراکتر تنها از یک فایل، مورد استفاده قرار می گیرد.

نکته: بعد از فراخوانی این تابع، اشاره گر فایل به کاراکتر بعدی جابه جا می شود.

مثال:

مثال زیر یک فایل را تا پایان فایل، کاراکتر به کاراکتر می خواند:

```
<?php
$file=fopen("welcome.txt","r") or exit("Unable to open file!");
while (!feof($file))
{
    echo fgetc($file);
}
fclose($file);
?>
```

۳-۴- ارسال یا آپلود فایل در PHP

برای ارسال فایل از Client به Server مراحل زیر را طی کنید:

1: ایجاد یک فرم HTML ارسال فایل

به فرم HTML زیر برای ارسال فایل توجه کنید:

```
<html>
<body>

<form action="upload_file.php" method="post"
enctype="multipart/form-data">
  <label for="file">Filename:</label>
  <input type="file" name="file" id="file" />
  <br />
  <input type="submit" name="submit" value="Submit" />
</form>

</body>
</html>
```

- خاصیت **enctype** در تگ **<form>**: هنگامی که یک فرم، داده های دودویی، مثل محتوای یک فایل را برای ارسال شدن نیاز دارد، این خاصیت با مقدار "multipart/form-data" پر می شود، در حقیقت مشخص می کند که چه نوع محتوایی هنگام ارائه ی فرم استفاده می شود.
- خاصیت **action** در تگ **<form>**: اگر submit اتفاق افتاد، اطلاعات فرم HTML به صفحه ای که در این خصوصیت مشخص شده ارسال می شود.(به صورت پیش فرض صفحه جاری در نظر گرفته می شود)
- خاصیت **type** در تگ **<input>**: با مقدار "file" پر شده است و مشخص می کند که ورودی کاربر، باید به عنوان یک فایل پردازش شود.(یک تکس باکس به همراه دکمه Browse، جهت جستجوی فایل از کامپیوتر کاربر)

💡 **نکته:** دادن سطح دسترسی ارسال فایل به کاربران ریسک بزرگی برای امنیت است، بنابراین تنها به کاربران مورد اعتماد اجازه ی ارسال فایل بدهید.

2: ایجاد یک اسکریپت ارسالی

فایل "upload_file.PHP" که در فرم HTML بالا در تگ <form> مشخص کردیم، می تواند چیزی شبیه زیر باشد:

```
<?php
if ($_FILES["file"]["error"] > 0)
{
    echo "Error: " . $_FILES["file"]["error"] . "<br />";
}
else
{
    echo "Upload: " . $_FILES["file"]["name"] . "<br />";
    echo "Type: " . $_FILES["file"]["type"] . "<br />";
    echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
    echo "Stored in: " . $_FILES["file"]["tmp_name"];
}
?>
```

با استفاده از متغیر سراسری \$_FILES می توانید فایل های مشخص شده توسط کاربر را از client به Server ارسال کنید، در حقیقت \$_FILES یک آرایه دوبعدی از پیش تعریف شده است که با استفاده از ایندکس مقداری آن می توانید به مقادیر زیر دسترسی داشته باشید:

- \$_FILES["file"]["name"]: نام فایل ارسالی
- \$_FILES["file"]["type"]: نوع فایل ارسالی
- \$_FILES["file"]["size"]: اندازه فایل ارسالی با واحد بایت
- \$_FILES["file"]["tmp_name"]: نام کپی موقت از فایل که روی سرور ذخیره شده
- \$_FILES["file"]["error"]: کد خطا که از ارسال فایل به وجود آمده

💡 **توجه:** به دلایل امنیتی، باید محدودیتی را روی آنچه که کاربر ارسال می کند قرار دهیم.

3: محدودیت های ارسال

دراین اسکریپت چند شرط را برای ارسال فایل اضافه می کنیم. کاربر فقط قادر است فایل های "jpg." یا "gif." را ارسال کند و اندازه ی فایل باید کمتر از "۲۰ KB" باشد:

```
<?php
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/pjpeg"))
&& ($_FILES["file"]["size"] < 20000))
{
    if ($_FILES["file"]["error"] > 0)
    {
        echo "Error: " . $_FILES["file"]["error"] . "<br />";
    }
    else
    {
        echo "Upload: " . $_FILES["file"]["name"] . "<br />";
        echo "Type: " . $_FILES["file"]["type"] . "<br />";
        echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
        echo "Stored in: " . $_FILES["file"]["tmp_name"];
    }
}
else
{
    echo "Invalid file";
}
?>
```

نکته: در کد بالا برای اینکه IE فایل های با فرمت jpg را بشناسد، باید نوع pjpg باشد و برای Firefox باید jpeg باشد.

4: ذخیره فایل های ارسالی

مثال های بالا یک کپی موقت از فایل های ارسال شده در پوشه ی موقت PHP روی سرور ایجاد می کنند و وقتی که اسکریپت به پایان رسید، از بین می روند. بنابراین باید آن ها را به یک مکان دیگر منتقل کنیم.

```
<?php
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/pjpeg"))
&& ($_FILES["file"]["size"] < 20000))
{
    if ($_FILES["file"]["error"] > 0)
    {
        echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
    }
    else
    {
        echo "Upload: " . $_FILES["file"]["name"] . "<br />";
        echo "Type: " . $_FILES["file"]["type"] . "<br />";
        echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
        echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";

        if (file_exists("upload/" . $_FILES["file"]["name"]))
        {
            echo $_FILES["file"]["name"] . " already exists. ";
        }
        else
        {
            move_uploaded_file($_FILES["file"]["tmp_name"],
            "upload/" . $_FILES["file"]["name"]);
            echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
        }
    }
}
else
{
    echo "Invalid file";
}
?>
```

در اسکریپت بالا در قسمت if چک می شود که آیا فایل ارسال شده در پوشه "upload" قبلاً وجود داشته یا نه، اگر موجود نباشد، آن را به پوشه مشخص شده منتقل یا کپی می کند.

💡 **نکته:** این مثال فایل را در یگ پوشه ی جدید به نام "upload" ذخیره می کند، احتمالاً شما نیاز خواهید داشت که نام فایل و مسیر آن را در پایگاه داده خود ذخیره کنید بنابراین در ادامه مثال بالا به راحتی می توانید با استفاده از دستور insert اینکار را انجام دهید.

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

۱. [کتاب آموزش PHP](#) (کتاب پیش رو)

۲. [کتاب آموزش SQL](#)

۳. [کتاب آموزش Bootstrap](#)

۴. [کتاب آموزش JavaScript](#)

۵. [کتاب آموزش HTML](#)

۶. [کتاب آموزش CSS](#)

موفق و پاینده باشید

فصل ۴: آموزش پایگاه داده MySQL

۴-۱ - مقدمه MySQL

با استفاده از PHP به پایگاه داده متصل می شوید و داده ها را دستکاری می کنید.

MySQL، یکی از معروف ترین سیستم های پایگاه داده است که همراه با PHP استفاده می شود.

MySQL چیست؟

- پایگاه داده MySQL، روی وب استفاده می شود.
- پایگاه داده MySQL، روی سرور اجرا می شود.
- پایگاه داده MySQL، خیلی سریع، قابل اطمینان و استفاده از آن آسان است.
- پایگاه داده MySQL، از استاندارد SQL استفاده می کند.
- قابلیت حمل بر روی سیستم عامل های مختلف از قبیل Linux و windows و غیره را دارد.
- استفاده و دانلود MySQL رایگان است.
- پایگاه داده MySQL، توسط شرکت Oracle توسعه و پشتیبانی می شود.
- پیادبود دختر بنیانگذار MySQL یعنی Monty Widenius در ابتدای نام آن از کلمه My استفاده شده است.

به زبان ساده، در MySQL داده ها در جداول ذخیره می شوند، جدول یک مجموعه ای از داده های مرتبط با هم است و از ستون ها و ردیفهایی تشکیل شده است.

موقعی که بخواهیم داده ها را به صورت همیشگی ذخیره کنیم، پایگاه های داده مفید واقع می شوند، به طور مثال یک سایت خرید و فروش می تواند شامل جدول های زیر باشد:

۱. جدول کالاها
۲. جدول مشتری ها
۳. جدول سفارشات
۴. جدول کارمندان

PHP + پایگاه داده MySQL

- PHP به همراه MySQL قابلیت cross-platform را داراست. (یعنی پروژه یا سایت تان را می توانید روی Windows پیاده کنید و روی Unix به کاربران ارائه دهید و برعکس! ...)
-

جداول پایگاه داده

یک پایگاه داده معمولاً از یک یا چند جدول تشکیل شده است، هر جدول را با یک نام مشخص می کنیم، مثلاً برای جدول مشتری ها می توان از نام Customer استفاده کرد. جداول شامل رکوردهایی از داده هاست.

مثال: جدول مشتری ها می تواند شامل ستون ها و ردیف های زیر باشد:

City	Address	FirstName	LastName
Esfahan	...	ali	mohammadi
Esfahan	...	reza	Nayeb
Tehran	...	sara	Najafi

جدول بالا شامل سه رکورد اطلاعاتی و چهار ستون City, Address, FirstName, LastName می باشد.

کوئری ها (Queries)

کوئری را می توان پرسیدن یا درخواست کردن معنی کرد.

با کوئری ها ما می توانیم یک درخواست از پایگاه داده برای بدست آوردن اطلاعاتی خاص داشته باشیم.

مثال: لطفاً به کوئری یا درخواست زیر توجه کنید:

```
SELECT LastName FROM Customer
```

کوئری بالا همه ی داده های ستون LastName از جدول Customer را انتخاب می کند و جدول زیر را نمایش می دهد:

LastName
mohammadi
Nayeb
Najafi

۴-۲- ارتباط با پایگاه داده MySQL در PHP

در PHP نسخه ۵ به بالا، برای کار با پایگاه داده MySQL می توانید یکی از روش های زیر را استفاده نمایید:

- افزونه **MySQLi** (کاراکتر i مخفف improved بمعنی بهبود یافته است).
- **PDO** که سرنام واژگان PHP Data Objects است

در نسخه های قبلی PHP از افزونه MySQL استفاده می شده، اما استفاده از این افزونه، از ۲۰۱۲ دیگر توصیه نمی شود.

باید از MySQLi استفاده کنم یا PDO ؟

اگر بخواهیم خیلی خلاصه جواب دهیم، این انتخاب بستگی به نیاز شما دارد !...

هر کدام از گزینه های MySQLi و PDO، مزیت های خودشان را دارند.

گزینه PDO با ۱۲ پایگاه داده مختلف کار می کند، اما MySQLi تنها با پایگاه داده MySQL کار خواهد کرد.

بنابراین اگر بخواهید زمانی به یک پایگاه داده دیگر سوییچ کنید، گزینه PDO کار را آسان تر خواهد کرد و فقط کافی است که ارتباط یا connection به پایگاه داده را تغییر دهید و احتمالاً چند تغییر کوچک در کوئری ها را خواهید داشت.

هر دو گزینه شیء گرا هستند، اما MySQLi رویه های API را نیز ارائه می دهد.

هر دو گزینه، دستورات آماده برای مقابله با تزریقات SQL یا SQL injection را پشتیبانی می کنند، که البته این موضوع، برای حفظ امنیت برنامه های کاربردی وب بسیار حیاتی است.

مثال های MySQL به هر دو روش MySQLi و PDO

در این مطلب و بیاموز های بعدی، سه روش مختلف برای کار کردن با پایگاه داده را نشان خواهیم داد:

- MySQLi (شیء گرا یا Object-Oriented)
 - MySQLi (رویه ای یا Procedural)
 - PDO
-

نصب MySQLi

برای سیستم عامل Linux و Windows، افزونه MySQLi در اغلب موارد بصورت اتوماتیک بعد از نصب پکیج PHP5 mysql نصب خواهد شد.

برای کسب اطلاعات بیشتر به لینک روبرو مراجع فرمایید :

<http://PHP.net/manual/en/mysql.installation.PHP>

نصب PDO

برای کسب اطلاعات بیشتر به لینک روبرو مراجع فرمایید :

<http://PHP.net/manual/en/pdo.installation.PHP>

ارتباط یا Connection به پایگاه داده MySQL

قبل از اینکه به داده های پایگاه داده دسترسی داشته باشید، باید یک ارتباط یا Connection تعریف کنید:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

توجه: توجه داشته باشید که در مثال شیء گرای بالا، `$connect_error` تا PHP 5.2.9 و PHP 5.3.0 کار نمی کند، بنابراین اگر می خواهید که کدتان با نسخه های قدیمی PHP نیز سازگار باشد، بجای آن از کد زیر استفاده نمایید:

```
// Check connection
if (mysqli_connect_error()) {
    die("Database connection failed: " .
mysqli_connect_error());
}
```


در مثال زیر با استفاده از تابع `mysqli_connect()` یک `connection` به پایگاه داده تعریف شده است:

مثال (MySQLi Procedural)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = mysqli_connect($servername, $username, $password);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>
```

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
    $conn = new PDO("mysql:host=$servername;dbname=myDB",
$username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
}
catch(PDOException $e)
{
    echo "Connection failed: " . $e->getMessage();
}
?>
```

توجه داشته باشید که در مثال PDO بالا، با تنظیم گزینه dbname با مقدار "myDB" نام پایگاه داده را نیز مشخص کرده ایم. اگر PDO موفق به اتصال به پایگاه داده نشود یک استثنا یا exception پرتاب (thrown) می شود.



نکته: یکی از بزرگترین مزیت های PDO وجود کلاس exception برای مدیریت خطاها است. همان طور که می دانید این خطاها ممکن است در کوئری های پایگاه داده رخ دهد و اگر یک استثنا یا exception از داخل بلاک { }try پرتاب (thrown) شود، اجرای اسکریپت متوقف شده و جریان کار به اولین بلاک catch { } منتقل می شود. در واقع استثنای که در قسمت try رخ داده در قسمت catch به دام می افتد.

بستن یک ارتباط

ارتباط یا Connection به صورت اتوماتیک وقتی که اسکریپت پایان یافت، بسته می شود، اما اگر مایل باشید می توانید این کار را قبل از پایان اسکریپت انجام دهید:

مثال (MySQLi Object-Oriented)

```
$conn->close();
```

مثال (MySQLi Procedural)

```
mysqli_close($conn);
```

مثال (PDO)

```
$conn = null;
```

۴-۳- ایجاد پایگاه داده در PHP

پایگاه داده شامل یک یا چند جدول است.

باید بدانید که برای ایجاد یا حذف پایگاه داده، باید دسترسی لازم را داشته باشید.

توجه: زمانی که می خواهید یک پایگاه داده جدید ایجاد کنید، در شیء `mysqli` تنها سه آرگومان اول را باید مشخص نمایید (`password` و `username` و `servername`)

نکته: اگر برای اتصال به پایگاه داده باید از یک `Port` مشخص استفاده نمایید، برای تنظیم این آرگومان در شیء `mysqli` باید آرگومان چهارم یعنی `database-name` را با یک رشته خالی تنظیم نمایید و سپس آرگومان پنجم را برای تنظیم `Port` استفاده نمایید:



ایجاد پایگاه داده با استفاده از PDO و MySQLi

دستور `CREATE DATABASE` برای ایجاد یک پایگاه داده در `MySQL` استفاده می شود.

نحوه استفاده:

مثال (MySQLi Procedural)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . $conn->error;
}
$conn->close();
?>
```

```
mysqli("localhost", "username", "password", "", port)
```

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = mysqli_connect($servername, $username, $password);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Create database
$sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql)) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

در مثال زیر یک پایگاه داده به نام "myDBPDO" ایجاد کرده ایم:

مثال (PDO)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
    $conn = new PDO("mysql:host=$servername;dbname=myDB",
$username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
    $sql = "CREATE DATABASE myDBPDO";
    // use exec() because no results are returned
    $conn->exec($sql);
    echo "Database created successfully<br>";
}
catch(PDOException $e)
{
    echo $sql . "<br>" . $e->getMessage();
}

$conn = null;
?>
```

نکته: یکی از بزرگترین مزیت های PDO وجود کلاس **exception** برای مدیریت خطاها است. همان طور که می دانید این خطاها ممکن است در کوئری های پایگاه داده رخ دهد و اگر یک استثنا یا **exception** از داخل بلاک **try{ }** پرتاب (**thrown**) شود، اجرای اسکریپت متوقف شده و جریان کار به اولین بلاک **catch{ }** منتقل می شود. در واقع استثنای که در قسمت **try** رخ داده در قسمت **catch** به دام می افتد. در بلاک **catch**، با استفاده از دستور **echo**، دستور SQL و متن خطای تولید شده را چاپ می کنیم.

۴-۴- ایجاد جدول در MySQL

یک جدول در پایگاه داده شامل یک نام منحصر بفرد می باشد و شامل تعدادی ستون و ردیف است.

ایجاد یک جدول در MySQL با استفاده از MySQLi و PDO

دستور CREATE TABLE برای ایجاد یک جدول در MySQL استفاده می شود.

در اینجا قصد داریم یک جدول با نام "MyGuests" که شامل ۵ فیلد "id" و "firstname" و "lastname" و "email" و "reg_date" است را ایجاد نماییم:

```
CREATE TABLE MyGuests (  
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
firstname VARCHAR(30) NOT NULL,  
lastname VARCHAR(30) NOT NULL,  
email VARCHAR(50),  
reg_date TIMESTAMP  
)
```

نکاتی درباره جدول بالا:

نوع داده مشخص می کند که چه داده هایی در فیلدها نگهداری شوند، اگر مایلید در این زمینه اطلاعات بیشتری کسب کنید به لینک [انواع داده در SQL](#) مراجعه نمایید. در مثال بالا، موقعی که نوع فیلدها را varchar در نظر می گیرید باید حداکثر طول آنرا در پرانتز مشخص کنید (تا ۲۵۵ کاراکتر)، اگر به اندازه ای بزرگتر از ۲۵۵ نیاز دارید از نوع text استفاده کنید (تا ۶۵,۵۳۵ کاراکتر).

بعد از مشخص کردن نوع داده یا Data Type، گزینه های اختیاری دیگری نیز برای هر ستون وجود دارد:

- NOT NULL: محدودیت NOT NULL یک ستون را مجبور می کند که مقدار خالی را قبول نکند.
- مقدار DEFAULT: محدودیت DEFAULT برای وارد کردن مقداری به صورت پیش فرض در یک ستون استفاده می شود.

- **UNSIGNED**: این محدودیت تنها برای نوع عددی استفاده می شود و زمان اضافه کردن رکورد تنها اجازه وارد کردن صفر و اعداد مثبت وجود دارد.
- **AUTO INCREMENT**: مقدار فیلد بصورت اتوماتیک به ازای هر رکورد جدید ۱ واحد اضافه می شود. (خود افزا)
- **PRIMARY KEY**: هر جدول می تواند یک کلید اصلی (PRIMARY KEY) داشته باشد، کلید اصلی برای منحصر به فرد کردن ردیفهای یک جدول استفاده می شود، بنابراین مقدار فیلدی را که کلید در نظر می گیرد در کل نباید تکراری باشد. کلید اصلی معمولاً از نوع عددی و خودافزا است، همچنین باید با مقدار NOT NULL تنظیم شود.

مثال های زیر، نشان می دهد که چگونه می تون در PHP یک جدول را ایجاد کرد:

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

۱. [کتاب آموزش PHP \(کتاب پیش رو\)](#)

۲. [کتاب آموزش SQL](#)

۳. [کتاب آموزش Bootstrap](#)

۴. [کتاب آموزش JavaScript](#)

۵. [کتاب آموزش HTML](#)

۶. [کتاب آموزش CSS](#)

موفق و پاینده باشید

فصل ۵: آموزش XML در PHP

۵-۱ - تجزیه کننده DOM چیست؟

یک سری اشیاء استاندارد برای دسترسی و دستکاری فایل های HTML و XML فراهم می کند و در سه بخش زیر مجزا شده است:

- هسته DOM: یک سری اشیاء استاندارد برای فایل های ساخت یافته تعریف می کند.
- XML DOM: یک سری اشیاء استاندارد برای فایل های XML تعریف می کند.
- HTML DOM: یک سری اشیاء استاندارد برای فایل های HTML تعریف می کند.

تجزیه کننده XML

برای خواندن، ویرایش کردن، ایجاد و خلاصه دستکاری یک فایل XML، به یک تجزیه کننده XML نیاز داریم. به صورت کلی دو نوع تجزیه کننده XML وجود دارد:

۱. تجزیه کننده درختی: این تجزیه کننده فایل XML را به یک ساختار درختی منتقل می کند و تمام فایل را تحلیل کرده و دسترسی به عناصر درخت را فراهم می کند.
۲. تجزیه کننده رویدادی: یک فایل XML را به صورت یک سری از رویدادها می بیند و زمانی که یک رویداد خاص اتفاق می افتد، یک تابع را برای بررسی آن صدا می زند.

تجزیه کننده DOM از نوع درختی است.

به فایل XML زیر توجه کنید:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<from>Amir</from>
```

تجزیه کننده DOM فایل XML صفحه ی قبل را به صورت زیر می بیند(سه سطح):

- سطح اول درخت: یک فایل XML را مشخص می کند.
- سطح دوم درخت: ریشه عنصر را مشخص می کند که تگ <from> است.
- سطح سوم درخت: متن عنصر که کلمه "Amir" است.

نصب تجزیه کننده XML

توابع تجزیه کننده DOM قسمتی از هسته PHP هستند و نیازی به نصب وجود ندارد.

فایل XML زیر در مثال های آینده استفاده خواهد شد:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

لود و چاپ یک فایل XML

مثال: در مثال صفحه ی بعد می خواهیم یک تجزیه کننده XML تعریف کرده و فایل XML بالا را داخل آن لود کنیم و سپس در خروجی چاپ کنیم:

```
<?php
$xmlDoc = new DOMDocument();
$xmlDoc->load("note.xml");
print $xmlDoc->saveXML();
?>
```

ToveJaniReminderDon't forget me this weekend!



خروجی کد بالا

اگر در Browser ویندوزتان گزینه "View source" را انتخاب کنید، فایل HTML زیر را خواهید دید:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

مثال بالا یک شی از DOMDocument ایجاد می کند و فایل "note.xml" را داخل آن لود می کند.

سپس تابع saveXML() فایل XML لود شده را داخل یک رشته قرار می دهد، حالا می توانیم آنرا در خروجی چاپ کنیم.

حلقه زدن در یک فایل XML

مثال: در مثال زیر می خواهیم یک تجزیه کننده XML تعریف کنیم و فایل XML قبلی را در آن لود کرده و بین عناصر آن یک حلقه بزنیم:

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

۱. [کتاب آموزش PHP](#) (کتاب پیش رو)

۲. [کتاب آموزش SQL](#)

۳. [کتاب آموزش Bootstrap](#)

۴. [کتاب آموزش JavaScript](#)

۵. [کتاب آموزش HTML](#)

۶. [کتاب آموزش CSS](#)

موفق و پاینده باشید

فصل ۶: آموزش Ajax در PHP

۶-۱ - مقدمه AJAX

AJAX (آجاکس)، هنر تبادل داده ها با سرور و بروز رسانی بخش هایی از یک صفحه وب، بدون بارگذاری مجدد کل صفحه است.

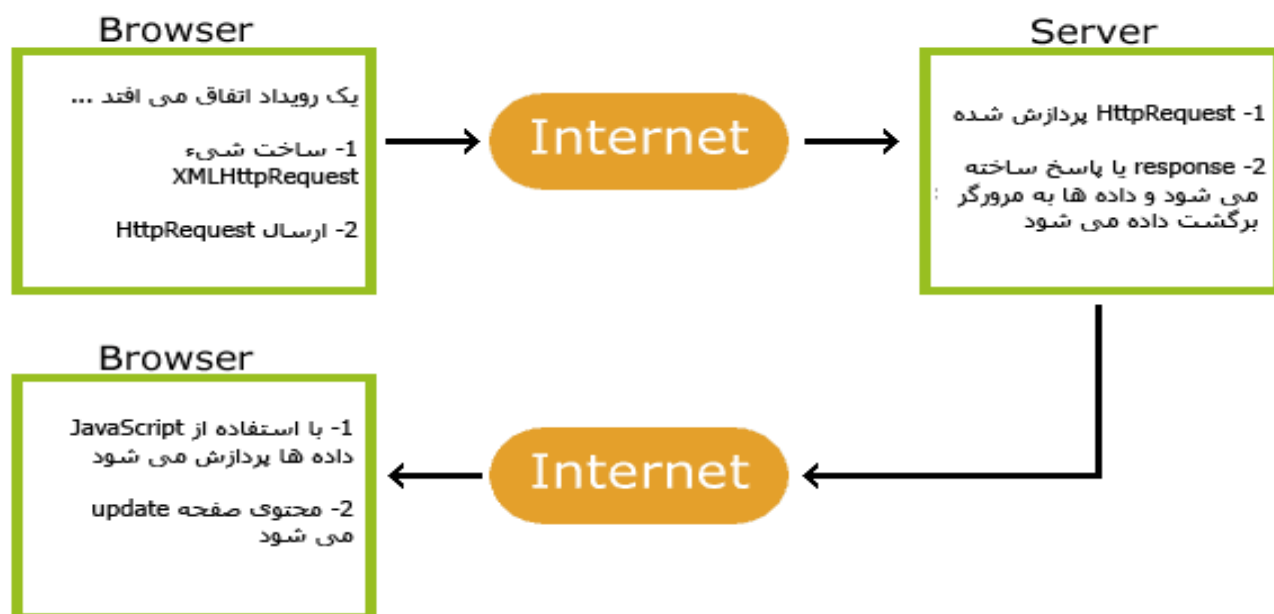
AJAX سرنام کلمات **Asynchronous JavaScript And XML** است.

به طور خلاصه، AJAX در مورد بارگذاری داده ها در پس زمینه و نمایش آن بر روی صفحه وب بدون بارگذاری مجدد کل صفحه است.

مثال هایی از استفاده AJAX در برنامه ها:

- Gmail
- Google Maps
- Youtube
- Facebook

AJAX چگونه کار می کند؟



AJAX براساس استانداردهای اینترنت

AJAX براساس استانداردهای اینترنت است و ترکیبی از موارد زیر را استفاده می کند:

- شیء XMLHttpRequest: برای تبادل داده با سرور، بصورت غیرهمزمان (asynchronously) استفاده می شود.
- JavaScript/DOM: برای نمایش اطلاعات و یا تعامل با داده ها
- CSS: برای مشخص کردن نحوه نمایش داده ها
- XML: اغلب بعنوان یک فرمت برای انتقال داده ها استفاده می شود.



برنامه های کاربردی AJAX مرورگرند و مستقل از پلتفرم عمل می کنند! ...

امکان Google Suggest

AJAX در سال ۲۰۰۵ توسط Google با امکان Google Suggest این سایت معروف شد.

این امکان ([Google Suggest](#)) با استفاده از AJAX یک واسط کاربری بسیار پویا ایجاد کرده است. زمانی که در کادر جستجوی Google شروع به تایپ می کنید، یک کد JavaScript حروف را به سرور ارسال می کند و سرور لیستی از پیشنهادات برای کلمه مورد جستجوی شما را برمی گرداند.

شروع به استفاده از AJAX

در پیامز PHP، ما نحوه update کردن بخشی از صفحه وب، بدون بارگذاری کل صفحه را با استفاده از AJAX پیامز خواهیم داد.

اسکرپت های سمت سرور، با PHP نوشته خواهد شد.

۶-۲ - استفاده از AJAX در PHP

با استفاده از AJAX (آجاکس)، می توانید برنامه ها کاربری خود را تعاملی تر کنید.

مثال AJAX (آجاکس) در PHP

مثال زیر نشان خواهد داد شد که چگونه یک صفحه وب می تواند با یک وب سرور گفتگو کند (ارتباط داشته باشد). تا زمانی که کاربر در کادر ورودی، کلمه ای را تایپ می کند، این ارتباط وجود دارد:

مثال

لطفاً نامی را در کادر زیر تایپ نمایید:

نام:

پیشنهادهای:

توضیح مثال بالا:

زمانی که کاربر در کادر ورودی بالا، کلمه ای را تایپ کند، تابع "showHint()" اجرا می شود. این تابع زمانی که رویداد "onkeyup" فیلد ورودی رخ دهد، فراخوانی می شود:

```

<html>
<head>
<script>
function showHint(str)
{
if (str.length==0)
{
document.getElementById("txtHint").innerHTML="";
return;
}
var xmlhttp=new XMLHttpRequest();
xmlhttp.onreadystatechange=function()
{
if (xmlhttp.readyState==4 && xmlhttp.status==200)
{

document.getElementById("txtHint").innerHTML=xmlhttp.responseText;
}
}
xmlhttp.open("GET","gethint.php?q="+str,true);
xmlhttp.send();
}
</script>
</head>
<body>

<p><b>لطفاً نامی را در کادر زیر تایپ نمایید</b></p>
<form>
نام: <input type="text" onkeyup="showHint(this.value)">
</form>
<p>پیشنهاها: <span id="txtHint"></span></p>

</body>
</html>

```

توضیح کد Javascript:

اگر فیلد ورودی خالی باشد (str.length==0)، محتوای عنصر با شناسه "txtHint" خالی و از تابع خارج می شود.

اگر فیلد ورودی خالی نباشد، تابع `showHint()` کارهای زیر را انجام می دهد:

- شیء `XMLHttpRequest` ایجاد می شود.
- زمانی که پاسخ سرور آماده باشد، یک تابع ایجاد و اجرا می شود. (وظیفه این تابع این است که پاسخ دریافت شده از سرور را در عنصر با شناسه `"txtHint"` قرار دهد)
- درخواست یک فایل به سرور ارسال می شود.
- توجه داشته باشید که یک پارامتر `(q)` به URL اضافه شده است (با محتوای فیلد ورودی تنظیم می شود)

فایل PHP مثال بالا:

فایلی که توسط کد JavaScript بالا صدا زده می شود یک فایل PHP بنام `"gethint.PHP"` است.

در فایل `"gethint.PHP"`، یک آرایه از نام ها چک می شود، و آنهایی که با پارامتر `(q)` متناظر است، چاپ می شود: (پاسخ ارسال می شود)

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

۱. [کتاب آموزش PHP](#) (کتاب پیش رو)

۲. [کتاب آموزش SQL](#)

۳. [کتاب آموزش Bootstrap](#)

۴. [کتاب آموزش JavaScript](#)

۵. [کتاب آموزش HTML](#)

۶. [کتاب آموزش CSS](#)

موفق و پاینده باشید

فصل ۷: آموزش شیء گرایی در PHP

۷-۱ - کلاس ها در PHP

در این فصل به بررسی یکی از زیباترین، و در عین حال خطرناکترین مباحث برنامه نویسی می پردازیم، این مبحث از این جهت خطرناک هست که اگر شما اصول اولیه را یاد بگیرید و این نوع برنامه نویسی بر شما تاثیر بگذارد، از آن پس دیگر به همه چیز به چشم یک شیء نگاه خواهید کرد و تمام روشهای برنامه نویسی گذشته خود را در دنیای واقعی کنار خواهید گذاشت.

در اینجا قصد نداریم شروع کنیم به آوردن مثالهایی از Object در دنیای واقعی و فرض می کنیم که شما مثالهایی مثل رنگ ماشین و یا تلفن را بلد باشید! در حالت کلی یک شیء شامل یک سری متغیرها و توابع می باشد که درون یک قالب کلی به نام کلاس قرار دارند، به متغیرهای درون کلاسها Properties و به توابع موجود در آن Method می گویند.

دلیل استفاده از اشیاء در زبان های برنامه نویسی: در برنامه های حجیم، استفاده از کلاس ها موجب می شود تا کدهای تکراری نوشته نشوند لذا مقدار کدها کمتر می شود و در نتیجه فشار کمتری به سیستم اجرا کننده دستورات می آید و همچنین سرعت پردازش کد ها مقدار زیادی افزایش می یابد.

به عبارت دیگر ویژگی کلاسها در این است که می توان یک کلاس را بعداً با نام های مختلف و متغیرهای مختلف برای استفاده های مشابه استفاده نمود.

object یا شیء چیست؟

مجموعه ای از متغیرها و توابع است که از یک الگوی خاص به نام کلاس ساخته شده است. اما کلاس ها چه هستند؟ فرض کنید ما یک شرکت داریم که این شرکت از بخش های مختلفی تشکیل شده است حال ما در هر بخش احتیاج داریم که هر ماه یک گزارش مالی بگیریم اما اینجا دو کار میتونیم بکنیم هم میتونیم برای هر بخش چند نفر بزاریم و آنها گزارش مالی را تهیه کنند و هر ماه تحویل دهند در این صورت در هر بخش شلوغی و همچنین کارمند بیشتری نیازمندیم! راه دیگر اینست که یک قسمت به عنوان اتاق گزارش مالی درست کنیم و هر بخش داده های خود را به این قسمت بدهد و گزارش مالی خود را دریافت کند در این روش هم بخش ها منظم تر خواهند بود و هم دیگر احتیاج به کارمند اضافی نداریم.

در اینجا آبجکتها نقش کارمند در بخش گزارش مالی را بر عهده دارند.

نحوه تعریف کلاس:

```
<?php
//تعریف کلاس
class class_name
{
//Properties
//Method
}

//ساختن یک شی از کلاس
$obj=new class_name();
?>
```

توضیح:

۱. ساخته شدن یک کلاس توسط کلمه کلیدی `class` صورت می گیرد.
۲. شما در هر جای کلاس قادر به تعریف متغیرها یا `Property`های کلاس هستید، اما بهتر است که آنها را در ابتدای کلاس تعریف کنید.
۳. بعد از تعریف متغیرها، توابع یا `Method`ها را تعریف می کنیم، توابع درون کلاس همانند توابع معمولی تعریف می شوند.
۴. تمام موارد بالا بین دو علامت "`{ }`" قرار می گیرد.
۵. در برنامه نویسی شیء گرا یک شیء چندین بار با چندین اسم متفاوت و ویژگی های متفاوت (متغیرهای مختلف) می تواند مورد استفاده قرار گیرد، برای ساختن یک شیء از کلمه کلیدی `new` استفاده می کنیم.

🔗 **توجه:** با بزرگتر شدن سیستم ها، ضروری خواهد بود که از یک ساختار دایرکتوری درخت مانند، برای نگهداری تمامی کلاسهایی که در برنامه وبتان بکار برده اید، استفاده کنید. در ادامه شما می توانید با استفاده از

تابع `include_once` یا `require_once` برای اضافه کردن کلاسهای تعریف شده به صفحات دلخواهتان استفاده کنید.

۷-۲- متغیرهای (Properties) کلاس در PHP

اشیاء یا `Object`ها به متغیرهای خاصی دسترسی دارند که به آنها `Property` می گویند این `Property`ها می توانند در هر جای بدنه کلاس تعریف شوند اما برای اینکه اسکریپتتان مرتب باشد بهتر است که در بالای کلاس تعریف شوند.

مثال: در `my_class` یک `property` به نام `name` با مقدار اولیه "Amir" تعریف شده است:

مثال

```
<?php
class my_class
{
    public $name="Amir";
}

$obj1=new my_class();
$obj1->name="Reza";
$obj2=new my_class();
print "$obj1->name". "<br />";
print "$obj2->name". "<br />";
?>
```

Reza
Amir



خروجی کد بالا

توضیح مثال صفحه ی قبل:

۱. در مثال بالا متغیر یا Property تعریف شده، در ابتدا با مقدار "Amir" تنظیم شده است، ولی در ادامه با ساختن شیء obj1 و دسترسی به این متغیر مقدار آنرا به "Reza" تغییر می دهیم.
۲. به این نکته توجه داشته باشید که بعد از اسم کلاس از دو پرانتز خالی استفاده نمودیم چون کلاس را ساده و بدون سازنده (Constructor) ساختیم.
۳. علامت ">" به شما اجازه می دهد تا به متغیر های یک کلاس دسترسی داشته باشید.

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

۱. [کتاب آموزش PHP](#) (کتاب پیش رو)

۲. [کتاب آموزش SQL](#)

۳. [کتاب آموزش Bootstrap](#)

۴. [کتاب آموزش JavaScript](#)

۵. [کتاب آموزش HTML](#)

۶. [کتاب آموزش CSS](#)

موفق و پاینده باشید

همراهان خوبی باشید و با

ری پست کردن ( Repost) این محتوی در [لینکدین پیاموز](#)

ما را در تولید دوره های آموزشی مفید

حمایت کنید.

از طریق صفحه لینکدین پیاموز، ما را **فالو** کنید.

+ Follow

برای باخبر شدن از آخرین دوره های آموزشی، کافیه در

لینکدین پیاموز روی **آیکن زنگ** کلیک کنید.



Beyamooz

