
بسم الله الرحمن الرحيم

چکیده

این پروژه آموزشی توسط دانشجو کارشناسی دانشگاه شمسی پور محسن رجبی انجام شده و در مورده آموزش زبان PHP است که امروزه پر استفاده ترین زبان برای طراحی صفحات نیز می باشد . با توجه به تحقیقات انجام شده در سال 2013 نزدیک به 60% سایت های ایجاد شده به وسیله ی این زبان بوده اند . این زبان از محبوبیت بسیار بالایی برخوردار می باشد . در این پروژه سعی شده تا این زبان از مبتدی تا پیشرفته با استفاده از مثال ها و توضیحات آموزش داده شود و در آخر نیز چندین مثال کاربردی در پروژه های کاری زده شده است . این آموزش در 37 مثال (Example) که هر مثال دارای یک عدد ورد که دارای چندین مثال می باشد تشکیل شده است . در هر ورد پیرامون چند دستور PHP توضیح داده شده است و مثال هایی برای آن ها آورده شده است که فایل مثال ها درون پوشه مثال موجود می باشد . برای شروع باید از مثال 0 شروع نمود که در این ورد جزئیاتی در مورده PHP و دستورات آن و نرم افزار های مورده نیاز آمده است . تمامی نرم افزار های مورده نیاز درون پروژه موجود می باشد . مثال های پروژه از ابتدا دستوررات PHP شروع شده است و شرح کامل آن ها آمده است و این دستورات ادامه پیدا نموده اند تا استفاده از پایگاه داده MySql و ذخیره اطلاعات در آن به علاوه انجام عملیاتی همچون درج , حذف , ویرایش و ... . همچنین در میان مثال ها نحوه استفاده از تکنولوژی Ajax با انجام مثال های کاربردی به خوبی آموزش داده شده است . با مطالعه دقیق این پروژه شخص خواننده به خوبی با زبان PHP آشنایی کامل پیدا می کند و به مباحث آن مانند شئ گرایی , کار با فایل ها , استفاده از Ajax و .. تسلط لازم را پیدا می نمایید .

فهرست

پیشگفتار

معرفی PHP

PHP یک زبان برنامه نویسی اسکریپتی اپن سورس است که برای طراحی برنامه های تحت وب سرور به کار می رود . سمت سرور بودن به این معناست که صفحات PHP ابتدا توسط سرور (که می تواند از نوع Apache یا IIS) باشد , پردازش شده و سپس خروجی به صورت کد های HTML و جاوا اسکریپت برای مرورگر کاربر ارسال می شود . به عبارت دیگر وظیفه اجرای صفحات PHP به عهده سرور وب هاست سایت می باشد بر خلاف HTML یا جاوا اسکریپت .

PHP مخفف عبارت Hypertext PreProcessor به معنا پیش پردازنده فرا متن می باشد که در سال 1994 توسط رسموس لردورف ایجاد شد و سپس تسط سایرین توسعه و گسترش پیدا کرد . اولین نگارش عمومی آن در اوایل سال 95 ارائه شد و با نام Personal Home Page Tools روانه بازار شد البته بسیار ساده بود .

ساختار زبان PHP بسیار شبیه به زبان C و در نسخه های جدید شبیه به جاوا می باشد و به همین دلیل از محبوبیت فراوانی بر خوردار است .از مشهور ترین نرم افزار های ساخته شده با PHP می توان به جوملا , وردپرس و.... اشاره کرد . سایت های فراوانی در جهان بر اساس زبان PHP نوشته شده اند و هر روز نیز بر تعداد آن ها اضافه می شود . بر طبق آمار منتشر شده 60% از سایت های موجود در سرور ها با PHP نوشته شده است که از مهم ترین آن ها می تواند به ویکی پدیا و فیسبوک اشاره کرد. PHP امکان استفاده از انواع مختلفی از پایگاه داده را از جمله MySql , SqlLite , اوراکل , IBM DB2 , Microsoft Sql Server و ... را با دستور هایی ساده فراهم می سازد . پی اچ پی روی بیشتر سیستم عامل های معروف از جمله ویندوز , لینوکس , یونیکس , مک و با اغلب سرور های معروف قابل اجراست . پیش از آغاز به یادگیری PHP شما باید آشنایی کافی با زبان های HTML و جاوا اسکریپت و کار با MySql را بدانید . دستورات این زبان می توانند به صورت مستقیم در درون کد های HTML قرا بگیرند . زبان PHP از نسخه 4.3 به بعد قابلیت پشتیبانی از واسط خط فرمان را نیز به امکانات خود اضافه کرد که این قابلیت می تواند برای ایجاد نرم افزار های غیر وب و یا نرم افزار هایی با واسط گرافیکی کاربر مورد استفاده قرار بگیرد .

مزیت های PHP

- PHP یک ابزار اپن سورس و رایگان است به همین دلیل هاست هایی که میزبانی آن را انجام می دهند بسیار ارزان تر از هاست های .net هستند .
- پی اچ پی بر روی تمامی پلتفورم های معروف مثل ویندوز , لینوکس , مک قابل اجراست .
- PHP یک زبان ساخت یافته بوده و یادگیری آن بسیار ساده است .
- ابزار کار با PHP همگی اپن سورس بوده و استفاده از آن رایگان هستند .
- سرعت بالا . اجرای یک اسکریپت PHP به طور متوسط تا سه برابر یک اسکریپت asp است .

در زمان نوشتن این مقاله آخرین نسخه PHP موجود نسخه 5.5 می باشد . در کنار نسخه 5 پی اچ پی یک نسخه اصلی دیگر در حال توسعه است . با توجه به تغییرات عمده موجود در این نسخه از جمله پشتیبانی کامل از یونیکد بود که قرار بود این نسخه به عنوان نسخه 6 پی اچ پی منتشر شود . اما پیاده سازی پشتیبانی از یونیکد بیش از آنچه انتظار می رفت به طول انجامید این امر باعث شده تا در سال 2010 این نسخه به بخش در حال توسعه منتقل شود و دیگر به آن نسخه 6 گفته نمی شود . مفسر PHP تنها کد هایی که در درون جدا کننده های PHP قرار بگیرند را تفسیر میکنند , معرف ترین این جدا کنند ها <?PHP برای ابتدا و ?> برای انتها استفاده می شوند . نام متغییر در زبان PHP حتما باید با نماد $ آغاز شود و نیازی به مشخص کردن آن نیست . بر خلاف نام توابع و کلاس ها نام متغییر ها به بزرگی و کوچکی حروف حساس هستند . خطوط جدید و فاصله نادیده گرفته می شوند البته به جزء فاصله هایی که در درون رشته ها قرار داشته باشد و تمامی دستورات این زبان با علامت سمی کولن (;) پایان می یابد . در زبان PHP به 3 صورت متفاوت می توان کامنت (توضیحات) گذاشت .

از علامت // یا # برای کامنت های یک خطی و از /**/ برای کامنت های چند خطی می توان استفاده کرد به تصویر زیر دقت کنید .

```
<?php
//Test Comment
#Test Comment
/*
This is a test comment
*/
?>
```

زبان PHP در ابتدا به صورت یک زبان مفسری پیاده سازی شد و امروزه نیز این پیاده سازی پر کاربرد ترین نسخه مورد استفاده است . تعدادی مترجم نیز برای این زبان ایجاد شده است که این زبان را از مفسر ها دور می کند .

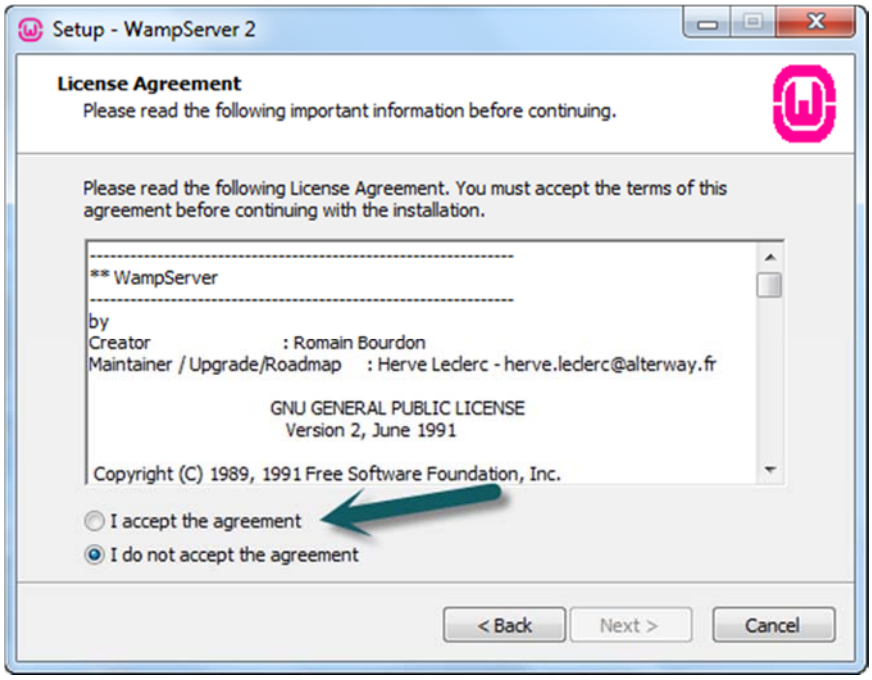
نحوه نصب و اجرای PHP

برای برنامه نویسی و استفاده از PHP به 3 چیز کلی نیاز است :

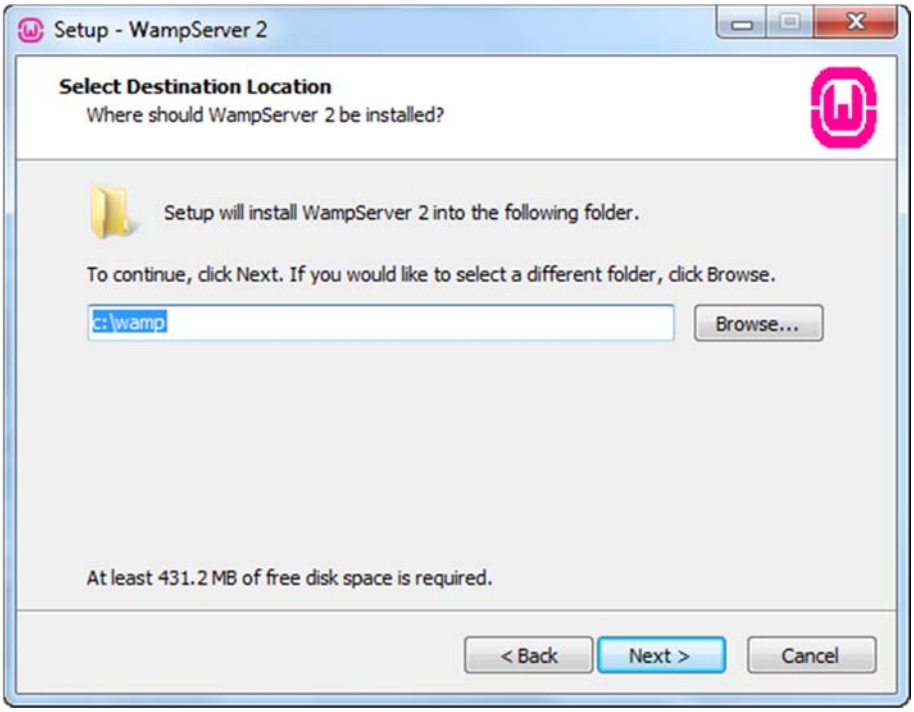
- برنامه ی یا ابزاری برای کد نویسی : در ساده ترین حالت می توانید از برنامه Notepad برای کد نویسی استفاده کنید . اما نرم افزار های حرفه ای مانند Dreamweaver , NuSphere PhpED و NetBeans و ... هستند که در این آموزش از نرم افزار Dreamweaver استفاده شده است .
- مرورگر وب : برای مشاهده صفحات نیاز به یک مرورگر می باشد که البته می شود از مرورگر پیش فرض استفاده کرد ولی پیشنهاد می شود از مرورگر Firefox استفاده شود .
- سرور اجرا کننده PHP : چنانچه بخواهید برنامه ها و صفحات PHP را بر روی کامپیوتر خود اجرا کنید نیاز به نصب PHP و MySql می باشد که می توان آن ها را به صورت تکی دانلود و نصب کرد ولی دو برنامه وجود دارند که نقش وب سرور را ایفا می کنند و همه برنامه ها و کتابخانه های مورد نیاز را دارا می باشند : WampServer و XamppServer که در این آموزش از Wamp استفاده شده است . که می توانید این برنامه رو از سایت خودش یعنی www.wampserver.com دانلود کنید .

وقتی این برنامه رو دانلود نمودید و فایل آن را اجرا کنید صفحه ایی مانند صفحه زیر نمایان می شود .

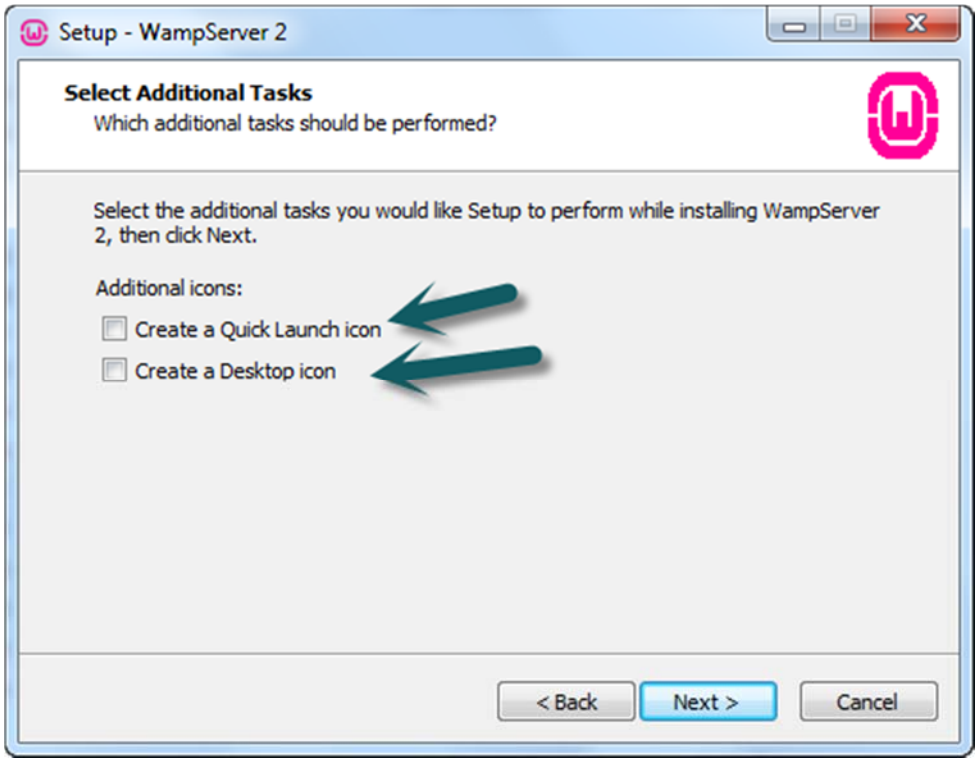
در این صفحه بر روی دکمه Next کلیک نمایید . در مرحله بعد تصویر زیر نمایان می شود .



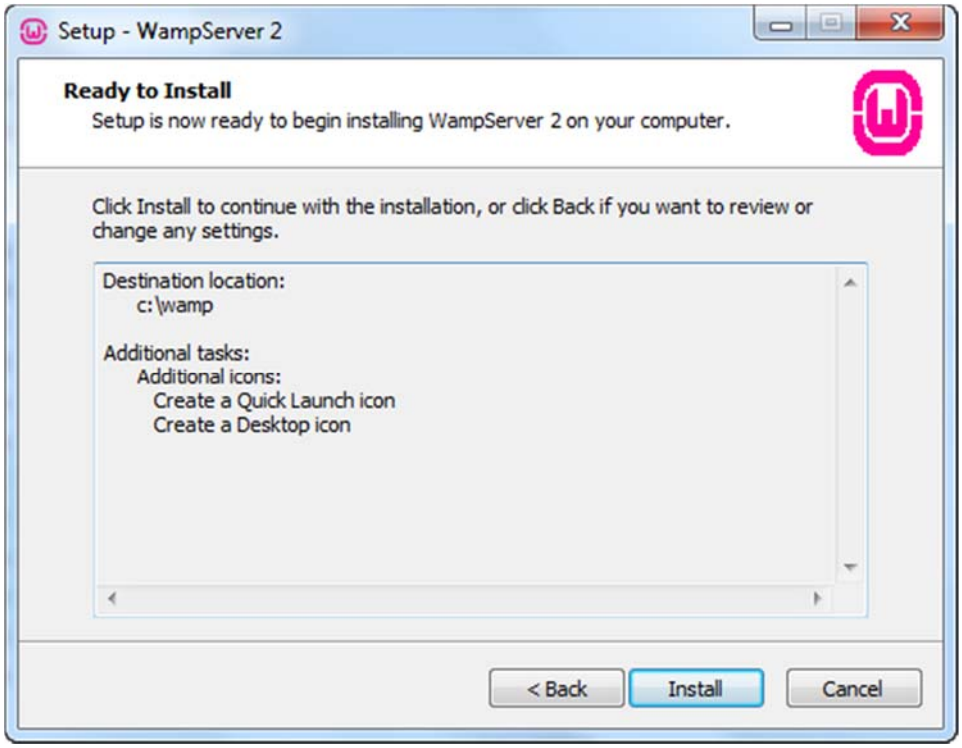
در این مرحله روی گزینه ایی که با فلش نشان داده شده است کلیک کرده وسپس دکمه Next را بزنید.



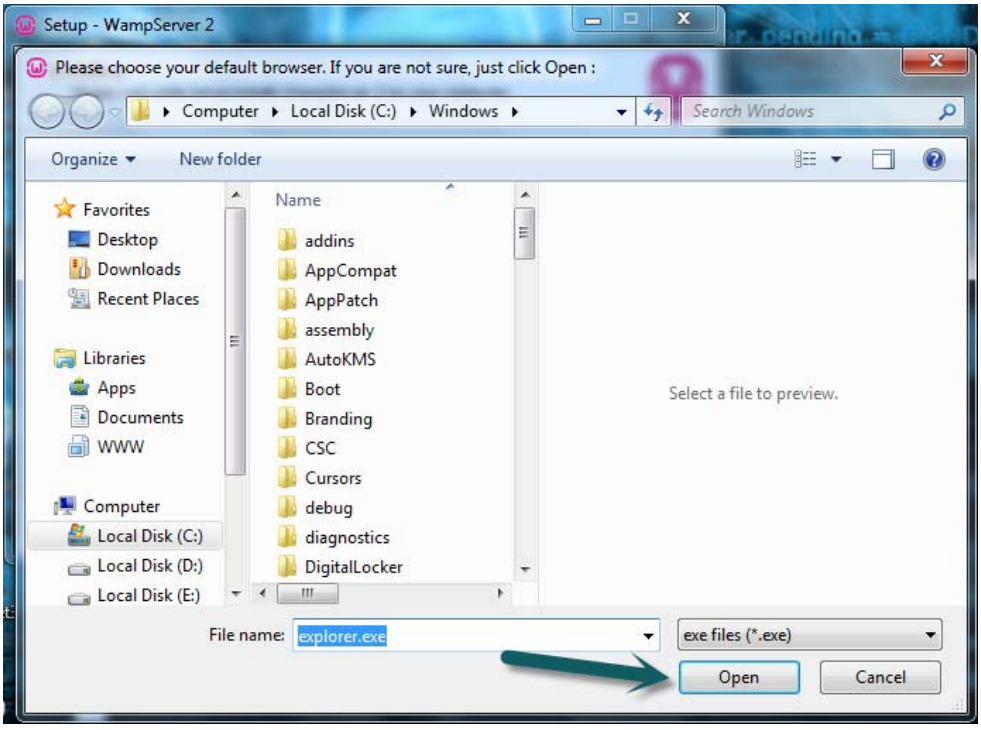
در این مرحله آدرس را تغییر ندهید و بر روی دکمه Next کلیک نمایید .



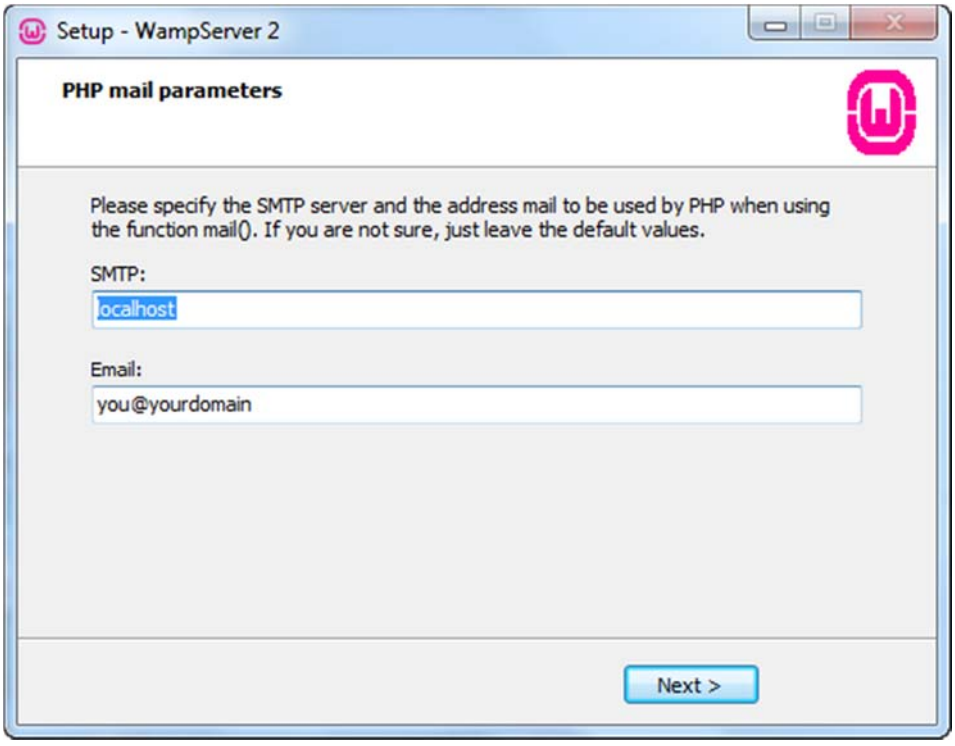
در این مرحله بر روی هر دوی چک باکس ها کلیک کرده و در نهایت بر روی گزینه Next کلیک کرده .



در این مرحله بر روی گزینه ی Install کلیک رده و صبر نمایید تا برنامه نصب شود .



در این مرحله نصب پایان یافته و پنجره ایی باز می شود که در تصویر بالا مشخص می باشد . فقط کافی است روی دکمه Open کلیک نمایید .

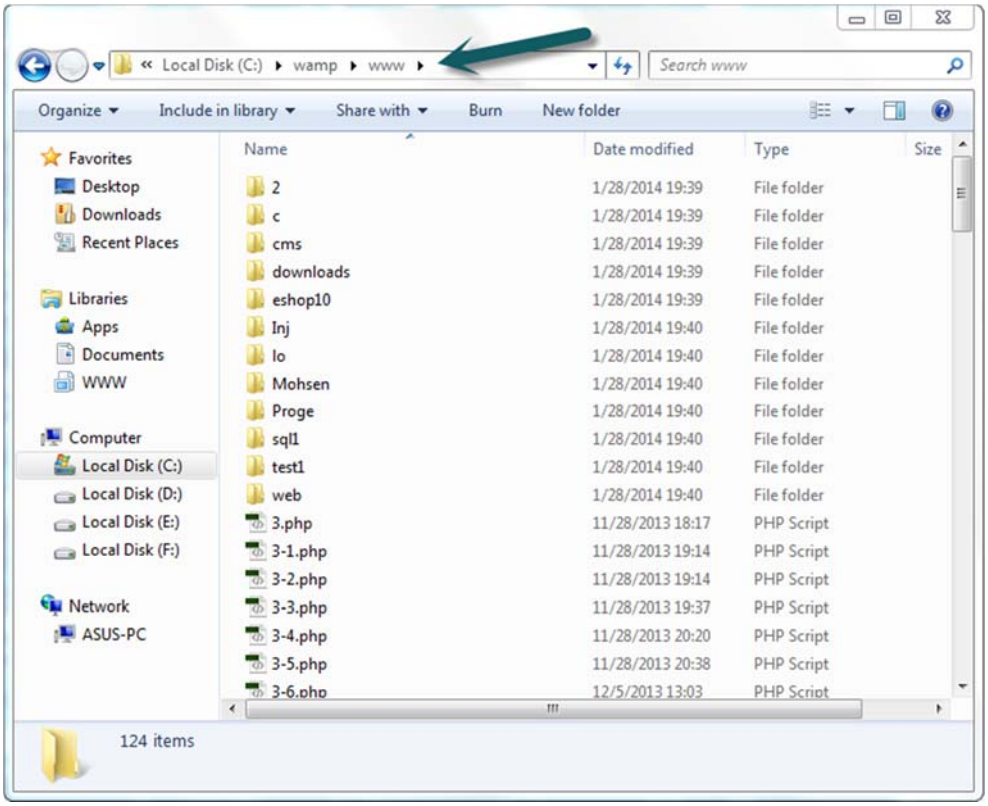


در این مرحله نباید به متغییر های بالا دست زد و فقط کافی است روی دکمه Next کلیک نمایید .

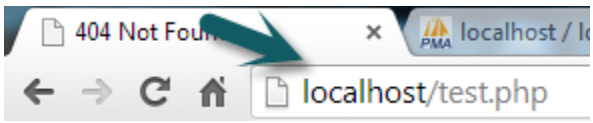
این آخرین محله نصب می باشد و فقط کافی است روی دکمه Finish کلیک نمایید . بعد از کلیک برنامه خودش اجرا می شود . و در کنار ساعت ویندوز ظاهر می شود . تصویر این نرم افزار 3 حالت دارد اگر

قرمز باشد نمی تواند برنامه نویسی کرد , اگر زرد باشد بعضی از قابلیت های آن اجرا می شود ولی وقتی سبز باشد تمام قابلیت های آن را می توان استفاده کرد .

خب برای برنامه نویسی PHP ابتدا بعد اینکه برنامه WampServer را نصب و اجرا کردیم باید برنامه خود را بنویسیم و بعد اینکه برنامه را نوشتیم آن برنامه را ذخیره و در مسیر زیر قرار دهیم تا بتوان آن را اجرا کرد .



در تصویر بالا مسیر نشان داده شده است این مسیر در داخل درایو c و درون پوشه wamp و دوباره درون پوشه www می باشد که ما باید برنامه های خود را در اینجا ذخیره کنیم . بعد از ذخیره کردن فایل درون این آدرس ابتدا مرورگر را باز کرده و عبارت localhost را می نویسیم این آدرس صفحه index ما می باشد . فرض کنید ما برنامه نوشته ایم که ای برنامه را به اسمه test.php در مسیری که گفته شد ذخیره کرده ایم برای اجرا این برنامه باید آدرس localhost/test.php را درون مرورگر وارد کنیم . به تصویر زیر نگاه کنید .



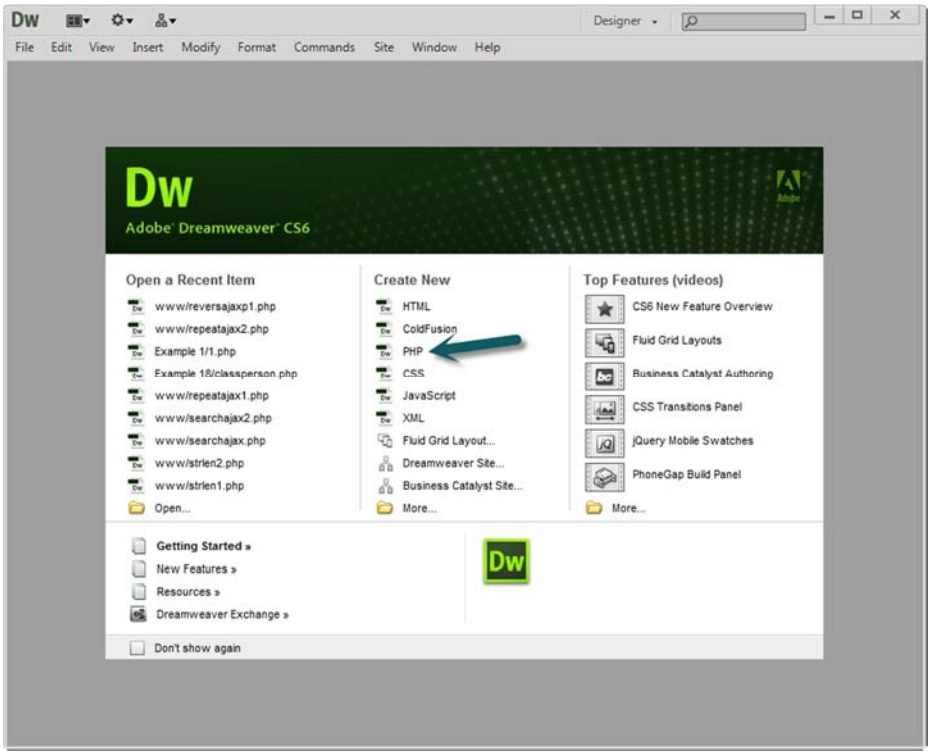
با اینگونه آدرس دهی می توان برنامه های نوشته شده را اجرا نمود .

برای کار با پایگاه داده MySql باید آدرس زیر را در مرورگر وارد کنید : localhost/phpmyadmin/

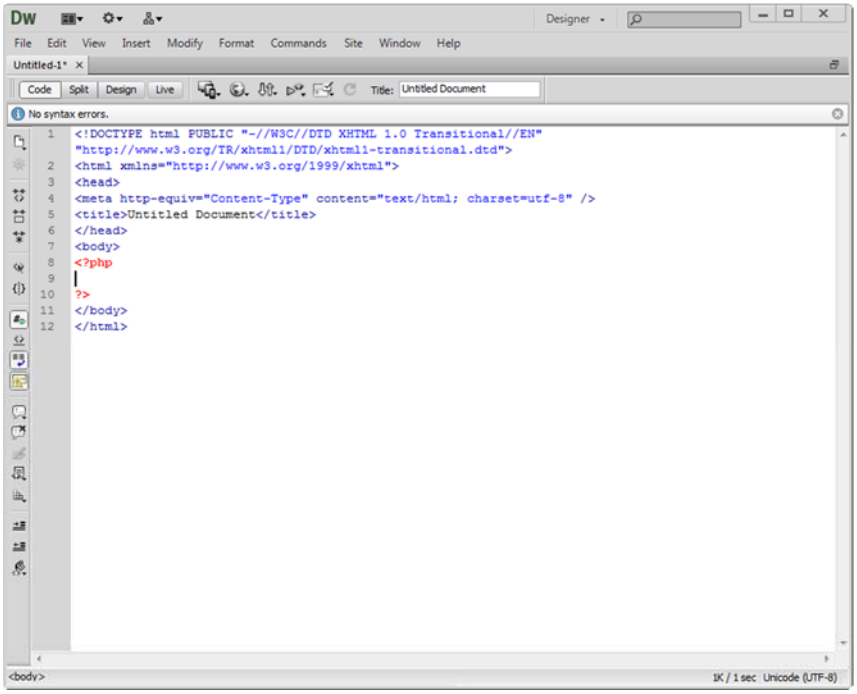
با وارد کردن این آدرس تصویر زیر نمایان می شود .

یوزر پیشفرض پایگاه داده root می باشد و رمز عبور پیش فرض هم دارا نمی باشد . برای ورود روی دکمه Go کلیک کنید تا وارد شوید و پایگاه داده مورد نظر خود را طراحی کنید یا با دیتابس مورد نظر کار کنید که در این آموزش مبحث کار با MySql مد نظر ما نیست .

برنامه Dreamweaver را نصب نمایید . بعد از نصب آن را اجرا نمایید . در صفحه ورود آن زبان PHP را مانند تصویر زیر انتخاب نمایید .



بعد از انتخای زبان PHP صفحه ایی مانند شکل زیر باز می شود که می توانیم کد های PHP خود را در آن بنویسیم و بعد در مسیر گفته شده ذخیره نامیید .



فصل اول

دستور Print و Echo

در مقدمه خواندیم که کد های PHP چگونه نوشته می شود و با تگ های آغاز و پایان آن آشنا شدیم ولی حالا میخواهیم با چکونگی چاپ کردن متن مورد نظر خود در صفحه را بیاموزیم, برای این کار از دو دستور زیر استفاده میکنیم که هر کدام رو با توضیحات کامل معرفی میکنم .

1. Echo
2. Print

قبل از اینکه این دو دستور رو توضیح دهیم باید فرق تک کوتیشن (') با دو کوتیشن (") رو بدانید.زبان PHP هر چیزی که داخل تک کوتیشن هست را به عنوان رشته میشناسد و هر چیزی که اعم از اعداد یا متغییر ها رو داخل آن بنویسیم رشته در نظر میگیرد ولی وقتی متنی داخل دو کوتیشن مینویسیم و متغییری داخل آن مینویسیم PHP آن را میفهمد و محاسبه می کند . به مثال echo توجه کنید.

```php
<?php
$a=10;
echo "This Is $a";
echo "<br>";
echo "This Is $a+10";
echo "<br>";
echo 'This Is $a';
?>
```

خروجی

This Is 10
This Is 10+10
This Is $a

در مثال بالا دستور echo "
" برای رفتن به خط بعدی است.حالا فرق ' و " رو فهمیدیم . اما دو برای چاپ بر روی صفحه وب به کار می رود . دستور echo نسبت به print سریع تر است و هیچ خروجی ندارد

ولی دستور print بعد از چاپ عدد 1 را بر میگرداند که اختیاری است یعنی نمی خواهد حتما خروجی دستور print را در متغییری بریزید.به مثال شماره echo&print در زیر توجه کنید.

```php
<?php
echo "This Is Number .";
echo "<br>";
print("This Is Number .");
//
echo "<hr>";
echo "This Is Number ."." This Is Good";
echo "<br>";
$b=print("This Is Number ."." This Is Good");
echo "<br>";
print("This Is Number ."." This Is Good");

//
echo "<hr>";
echo $b;
?>
```

خروجی

This Is Number .
This Is Number .

This Is Number . This Is Good
This Is Number . This Is Good
This Is Number . This Is Good

1

در مثال بالا دستور echo “<hr>” برای انداختن خط در صفحه می باشد .

فصل دوم

متغییر ها

فرض کنید شما می خواهید یک مقدار عددی یا رشته ایی مثلا نمره یه دانشجو یا اسم یک دانشجو رو نگه دارید و بعد از یک سری کار ها دوباره استفاده کنید برای این کار از متغییر ها استفاده میکنیم و یک متغییر را تعریف میکنیم.در زبان های دیگر مانند C , C++ , C# , Pascal ما هنگام تعریف متغییر باید نوع آن را هم تعریف کنیم اما در PHP نیازی به تعریف نوع متعییر نمی باشد خود PHP نوع آن را تشخیص می دهد . PHP به حروف بزرگ و کوچک حساس است.برای تعریف متغییر دو نوع متغییر موجود است اولی ثابت ها هستند که بعد از تعریف آن ها از آن ها فقط استفاده میکنیم و مقدار آن ها را نمی توان تغییر داد و مقدار آن ها همیشه ثابت هستد نوع دومی متغییر ها معمولی هستند که مقدار آن ها می توانند در داخل برنامه تغییر کنند .

قوانین نام گذاری متغییر ها

1. نام متغییر می تواند از حروف الفبای انگلیسی و بزرگ و کوچک و عدد و علامت _ (همان Underline) تشکیل شود .
2. نام متغییر حتما باید با یک حرف یا علامت _ شروع شود. مثال نام Intnumber صحیح و نام 1int غلط است .
3. نام یک متغییر نمی تواند شامل فاصله باشد .

ثابت ها

ثابت ها در کل برنامه به صورت سراسری هستند و در برنامه با دستور define تعریف می شوند و در نام گذاری آن ها لازم نیست که کاراکتر $ اول آن قرار بگیرد . دستور define سه ورودی دارد که اولین ورودی آن نام ثابت هست و دومین ورودی آن مقدار آن ثابت هست و ورودی سومی که اختیاری هم هست مقدار true یا false است که اگه در آن true قرار دهیم نام ثابت ما در کل برنامه حساس به حروف بزرگ و کوچک نیست و فرقی نمی کند که در برنامه با حروف بزرگ نوشته شوند یا کوچک ولی اگر false قرار دهیم در کل برنامه حساس به حروف بزرگ و کوچک است و ما باید در کل برنامه نام

ثابت را دقیقا همان نامی که در دستور define نوشتیم بنویسیم پیشفرض دستور define کلمه false است. به مثال define توجه کنید .

```php
<?php
define("name","Alireza");
echo name;
echo "<hr>";
define("family","Zamani",true);
echo family;
echo "<br>";
echo FAMILY;
?>
```

خروجی

Alireza

Zamani
Zamani

متغییر ها

در PHP برای تعریف متغییر باید از کاراکتر $ اول اسم متغییر استفاده کرد که خود PHP نوع داده ایی آن را تشخیص می دهد . وقتی متغییری را تعریف می کنید محلی (یا Local) است این به این معنی است که وقتی متغییری در داخل یک تابع تعریف شود فقط در همان تابع قابل دسترسی هست نه در کل برنامه برای این منظور ما در PHP چهار دامنه کاربرد متغییر داریم که به شرح زیر است .

1. Local
2. Global
3. Static
4. Parameter

متغییر های Local

متغییری که درون تابع تعریف می شود فقط برای آن تابع قابل استفاده است و محلی است وبعد از اتمام کار تابع متغییر های محلی درون آن از درون حافظه حذف می شوند . به مثال local توجه کنید در این

مثال یک تابع نوشته شده است البته شما هنوز با نوشتن آن آشنا نشدید نگران نباشید در مباحث بعد با آن آشنا می شوید .

```php
<?php
$x=5;
function test() {
echo $x;
}
test();
?>
```

این برنامه خطا می دهد چون متغییر x به صورت Local است و نمی توان همینطوری درون تابع از آن استفاده نمود .

متغییر های Global

هر متغییری که خارج از تمام توابع تعریف شده باشد دارای دامنه کاربردی Global خواهد بود . متغییر های Global را می توان در هر قسمت برنامه از جمله درون توابع بطور مشترک استفاده کرد . برای استفاده از متغییر درون تابع از کلمه کلیدی global استفاده می کنیم که به دو صورت است . به مثال global توجه کنید .

نوع اول

```php
<?php
$x=25;
$y=75;
function test() {
    global $x,$y;
    $y=$x+$y;
}
test();
echo $y;
?>
```

خروجی

100

نوع دوم مثال global2

```php
<?php
$x=25;
$y=75;
function test() {
    $GLOBALS['y']=$GLOBALS['x']+$GLOBALS['y'];
}
test();
echo $y;
?>
```

خروجی

100

متغییر های Static

گاهی ما می خواهیم بعد از تمام شدن دستورات تابع متغییر های محلی آن تابع و مقادیر درون آن ها از بین نروند تا بار دیگر که آن تابع اجرا شد مقادیر قبلی درون آن ها محفوظ مانده باشند و بتوان از آن ها استفاده کرد برای این کار وقتی که برای اولین بار بخواهیم متغییر را تعریف کنیم از عبارت static استفاده میکنیم اما فراموش نکنید این متغییر ها نیز به صورت محلی هستند یعنی فقط برای آن تابع شناخته می شوند به مثال static دقت نمایید .

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

خروجی

```
0
1
2
3
```

متغییر های Parameter

این یک متغییر محلی است که مقدار آن در زمان فراخوانی تابع برای آن متغییر ارسال می شود و محل تعریف آن درون پرانتز جلوی تابع است.به مثال parameter توجه کنید .

```php
<?php
function Test($x) {
    echo $x;
}
Test(5);
?>
```

خروجی

```
5
```

فصل سوم

انواع متغییر ها

PHP از هشت نوع داده اصلی پشتیبانی می نماید , البته توجه داشته باشید نوع داده ایی توسط برنامه نویس مشخص نمی شود بلکه در زمان اجرا توسط خود PHP مشخص می شود . با استفاده از تابع ()Var_dump می توانیم نوع داده ایی متغییر را ببینیم . این انواع داده عبارتند از :

چهار نوع داده اسکالر :

- Boolean
- Integer
- Float
- String

دو نوع داده ترکیبی :

- Array
- Object

و در نهایت دو نوع داده مخصوص :

- Resource
- Null

نوع داده ایی Boolean

نوع داده ایی Boolean شامل دو مقدار True و False است و فقط می تواند یکی را دریافت کند .

هر داده ایی در Boolean مقدار True رو بر میگرداند مگر اینکه

- عدد صفر (0)
- عدد صفر ممیز صفر (0.0)
- رشته خالی و یا رشته "0 "

- یک آرایه بدون عنصر
- یک شئ بدون متغییر عضو
- مقدار ویژه Null

با استفاده از تابع ()Is_boll می توانیم بفهمیم یک متغییر از نوع Boolean هست یا خیر . به مثال boolean توجه کنید .

```php
<?php
$st=True;
$st1=10.2;
$st2="A";
$st3=False;
$st4=0;
$st5="";
var_dump((bool)($st));
var_dump((bool)($st1));
var_dump((bool)($st2));
var_dump((bool)($st3));
var_dump((bool)($st4));
var_dump((bool)($st5));
?>
```

خروجی

```
boolean true
boolean true
boolean true
boolean false
boolean false
boolean false
```

نوع داده ایی Integer

از نوع داده ایی صحیح پشتیبانی می کند و شما می توانید اعداد صحیح را به صورت دسیمال(مبنای 10) , هگزا دسیمال (مبنای 16) , اوکتال (مبنای 8) به کار ببرید اندازه نوع Integer به سخت افزار و سیستم عامل بستگی دارد .

برای مثال دو عدد 123 و 123– هر دو از نوع Integer هستند برای نوشتن اعداد در مبنای 16 در ابتدای عدد علامت 0x را میگذاریم مثلا عدد 123 اینگونه می شود 0x123 و برای نوشتن اعداد ر مبنای 8 ابتدای عدد 0 میگذاریم .

با استفاده از تابع ()Is_int می توانیم بفهمیم که یک تابع از نوع Integer هست یا نه . به مثال int توجه کنید .

```php
<?php
$st=123;
$st1=-123;
$st2=0123;  // مبنای 8
$st3=0x1AB; //مبنای 16
$st4=0;
$st5="";
$st6=12.5;
var_dump((int)($st));
var_dump((int)($st1));
var_dump((int)($st2));
var_dump((int)($st3));
var_dump((int)($st4));
var_dump((int)($st5));
var_dump((int)($st6));
?>
```

خروجی

```
int 123

int -123

int 83

int 427

int 0

int 0

int 12
```

نوع داده ایی Float

این نوع داده ایی در PHP از نوع عددی اعشاری پشتیبانی می کند . اندازه این اعداد به نوع سیستم عامل و پردازنده بستگی دارد ولی حداکثر عدد 1.8e308 با 14 رقم اعشاری را می توان در نظر گرفت.

با استفاده از تابع is_float() می توانیم بفهمیم که یک تابع از نوع Float هست یا نه . به مثال float توجه کنید .

```php
<?php
$st=1.234;
$st1=-1.2e4;
$st2=7E-89;
var_dump((float)($st));
var_dump((float)($st1));
var_dump((float)($st2));
?>
```

خروجی

```
float 1.234

float -12000

float 7.0E-89
```

نوع داده ایی String

رشته مجموعه ایی از کاراکترها می باشد و در PHP از این نوع داده ایی برای ذخیره رشته استفاده می شود PHP هیچ نوع محدودیتی در طول رشته ایجاد نمی کند .

برای ذخیره رشته در PHP به روش می توانیم عمل کنیم :

- با کاراکتر های نقل قول تکی ' '
- با کاراکتر های نقل قول جفتی " "
- به روش Heredoc

ساده ترین روش برای ایجاد رشته استفاده از کاراکتر های نقل قول تکی است و عیب آن این است که در آن نمی توانیم از متغییر استفاده کنیم اگر در نوشته مان نیازمند استفاده از کاراکتر \ باشیم میتوانیم آن را یک بار یا دوبار بنویسیم در هر صورت PHP برای ما کاراکتر \ را یکبار چاپ می کند . به مثال string توجه کنید .

```php
<?php
$point=17;
$st='This is a simple string';
$st1='You deleted c:\\*.*';
$st2='You deleted c:\*.*';
$st3='You point $point';
echo $st."<br>";
echo $st1."<br>";
echo $st2."<br>";
echo $st3;
?>
```

خروجی

This is a simple string
You deleted c:*.*
You deleted c:*.*
You point $point

یکی دیگر از روش های ایجاد رشته استفاده از کاراکترهای نقل قول جفتی است که با استفاده از آن ما دیگر مشکل کاراکتر نقل قولی تکی را نداریم و میتوانیم از متغییر در آن استفاده کنیم در این حالت نیز ما می توانیم از کدهای ویژه ایی برای تولید کاراکتر های خاص استفاده کنیم . به جدول زیر توجه کنید .

کد های ویژه مورد استفاده در دستور echo	
\n	کاراکتر 10 اسکی را تولید می کند
\r	کاراکتر 13 اسکی را تولید می کند
\t	معادل کاراکتر تب افقی می باشد
\\	کاراکتر \ را ایجاد می نماید
\$	کاراکتر $ را ایجاد می نماید
\”	کاراکتر " را ایجاد می نماید

به مثال string2 توجه کنید .

```php
<?php
$name="\n\t\tMohammad";
echo "<pre>"."My Name Is : $name"."</pre>";
?>
```

خروجی

```
My Name Is :
                Mohammad
```

یکی دیگر از روش های ایجاد رشته در PHP استفاده از هردوک است . این روش برای ایجاد رشته های طولانی است در این روش برای ایجاد رشته باید مراحل زیر را طی کرد

- ابتدا کاراکترهای <<< را نوشته و یک شناسه دلخواه را ذکر کنید
- رشته مورد نظر را بنویسید
- شناسه استفاده شده در ابتدای رشته را نیز بنویسید

به مثال string3 توجه کنید در این مثال از شناسه EOD برای مشخص کردن ابتدا و انتهای نوشته استفاده شده است نام این شناسه اختیاری است و می تواند هر نام دیگیری باشد ولی نباید در متن اصلی باشد و از قوانین نام گذاری متغییر ها پیروی می کند .

```php
<?php
$point=17;
$str=<<<EOD
Example for string
<br>
Your point=$point
EOD;
echo $str;
?>
```

خروجی

```
Example for string
Your point=17
```

نوع داده ایی Array

تا کنون با متغییر ها و نقش آنها در PHP آشنا شدید , همان گونه که دیدید یک متغییر تنها قابلیت ذخیره یک مقدار را دارا می باشد ولی در مقایسه آرایه ها می توانند چندین مقدار را در خود ذخیره کنند . یه آرایه دارای چند عنصر است هر عنصر دارای یک مقدار است . در PHP هر آرایه به دو دسته آرایه های ایندکس دار و آرایه های انجمنی تقسیم می شوند این دو نوع در آرایه در روش دسترسی به عناصرشان متفاوت هستند .

در آرایه ایندکس دار هر یک از عناصر آرایه بر اساس ایندکس مورد دسترسی قرار می گیرند که به صورت پیشفرض عدد صحیح است و از صفر شروع می شود به مثال array توجه کنید .

```php
<?php
$car=array("BMW","Toyota","Volvo");
echo $car[0]."<br>";
echo $car[1]."<br>";
echo $car[2]."<hr>";
var_dump($car);
?>
```

خروجی

BMW
Toyota
Volvo

```
array (size=3)
  0 => string 'BMW' (length=3)
  1 => string 'Toyota' (length=6)
  2 => string 'Volvo' (length=5)
```

آرایه انجمنی در PHP مجموعه ایی مرتب از نگاشت ها می باشد , در یک نگاشت برای دسترسی به هر یک از مقادیر ذخیره شده در آرایه یک کلید دلخواه انتساب داده می شود . از این نوع آرایه می توان به صورت آرایه عادی , لغت نامه , پشته , صف , ... استفاده کرد به مثال array2 توجه کنید .

```php
<?php
$age=array("Mohsen"=>"19","Sina"=>"18","Amir"=>"15");
echo 'Mohsen Age :'.$age["Mohsen"]."<br>";
echo 'Sina Age :'.$age["Sina"]."<br>";
echo 'Amir Age :'.$age["Amir"]."<hr>";
var_dump($age);
?>
```

خروجی

Mohsen Age :19
Sina Age :18
Amir Age :15

```
array (size=3)
  'Mohsen' => string '19' (length=2)
  'Sina' => string '18' (length=2)
  'Amir' => string '15' (length=2)
```

نوع داده ایی Object

متغییر هایی که در داخل کلاس تعریف می شوند گفته می شود به مثال object توجه کنید .

```php
<?php
class Car
{
  var $color;
  function Car($color="green")
  {
    $this->color = $color;
  }
  function what_color()
  {
    return $this->color;
  }
}
?>
```

در این مثال کلاسی به اسم car تعریف شده که در آن متغییره color یه متغییر از نوع object است .

نوع داده ایی NULL

این نوع داده ایی مخصوص می باشد و تنها یک مقدار یعنی NULL را می گیرد . NULL مشخص میکنند که متغییر دارای مقدار نمی باشد , یک متغییر در صورتی NULL به حساب می آید که یکی از شرایط زیر را دارا باشد :

- مقدار NULL به آن اختصاص داده شده باشد
- هیچ مقداری به آن اختصاص نداده شده باشد
- تابع ()unset بر روی آن اعمال شده باشد

به مثال null توجه کنید .

```php
<?php
$a=100;
$a=NULL;
$b=NULL;
var_dump($a);
var_dump($b);
?>
```

خروجی

```
null

null
```

فصل چهارم

توابع کار با رشته

معرفی چند تابع که مخصوص کار با رشته می باشند .PHP توابع زیادی برای کار با رشته دارد .

تابع ()strlen

این تابع یک ورودی میگیرد و تعداد کاراکتر موجود در این ورودی را به ما می دهد . به مثال strlen توجه کنید .

```
<?php
$a='Hello Word!';
echo strlen($a);
?>
```

خروجی

```
11
```

تابع ()strpos

این تابع دو ورودی میگیرد و ورودی دومی را در ورودی اولی جستجو می کند و شماره کاراکتر را بر می گرداند. به مثال strpos توجه کنید .

```
<?php
echo strpos("Hello Word!","Word");
?>
```

خروجی

```
6
```

تابع ()strtolower

این تابع یک ورودی میگیرد و تمام حروف ورودی را به حروف کوچک تبدیل میکند . به مثال strtolower توجه کنید .

```php
<?php
echo strtolower("HeLlo WoRD!");
?>
```

خروجی

hello word!

تابع strtoupper

این تابع دقیقا برعکس تابع strtolower() عمل میکند و تمام حروف ورودی خود را به حروف بزرگ تبدیل می کند . به مثال strtoupper توجه کنید .

```php
<?php
echo strtoupper("Hello WoRd!");
?>
```

خروجی

HELLO WORD!

تابع ord()

این تابع یک ورودی میگرد و کد اسکی حرف اول آن را چاپ می کند . به مثال ord نوچه کنید .

```php
<?php
echo ord("H");
echo "<br>";
echo ord("Hello WoRd!");
?>
```

خروجی

72
72

تابع trim()

این تابع فضا خالی سمته چپ و راست پارامتر ورودی را حذف می کند این تابع بیشتر مواقعی که پارامتری از کاربر دریافت میکنیم کاربرد دارد . به مثال trim توجه کنید ابتدا دو ورودی به نام a و b داریم که داخل آن دو یک کلمه نوشته شده است ولی در مقدار a سمته چپ و راست آن فضا خالی وجود دارد بخاطر همین در شرط اول (دستور if) میبینم که این دو با هم فرق دارند .

```php
<?php
$a=' Messi!  ';
$b='Messi!';
if ($a==$b)
    echo 'No Difference $a and $b';
else
    echo 'Yes Difference $a and $b';
//trim
$a=trim($a);
$b=trim($b);
echo "<hr>";
if ($a==$b)
    echo 'NO Difference $a and $b';
else
    echo 'Yes Difference $a and $b';
?>
```

خروجی

Yes Difference $a and $b

NO Difference $a and $b

این تابع همچنین نیز دو تابع دیگر هم دارد که یکی ltrim() که فضا خالی سمته چپ را حذف می کند و همچنین تابع rtrim() که فضا خالی سمته راست را از بین می برد .

فصل پنجم

عملگر ها

کار اصلی یک عملگر انجام برخی کارها بر روی مقدار یک متغییر است . آن کار ها می توانند اختصاص یک مقدار , تغییر دادن یک مقدار , یا مقایسه دو یا چند مقدار باشند .

عملگر ها به 4 قسمت تقسیم می شوند که در ادامه به تشریح کامل آن ها می پردازیم .

- عملگر های تخصیص
- عملگر های ریاضی
- عملگر های مقایسه
- عملگر های منطقی

عملگر های تخصیص

شما قبلا یک عملگر تخصیص را عملا دیدید , علامت مساوی عملگر اصلی تخصیص می باشد . به جدول زیر توجه کنید .

عملگر	مثال	عملی که انجام می دهد
+=	$a+=3;	$a=$a+3;
-=	$a-=3;	$a=$a-3;
.=	$a.="string";	$a=$a."string";
*=	$a*=3;	$a=$a*3;
/=	$a/=3;	$a=$a/3;
%=	$a%=2;	$a=$a%2;

عملگر / برای تقسیم و عملگر % برای مد (با قیمانده) به کار می رود .

عملگر های ریاضی

عملگر های ریاضی بسادگی برای انجام وظایف ریاضی پایه می باشند . به توجه مقابل توجه کنید .

عملگر	مثال	عملی که انجام می دهد
+	$b=$a+3;	مقادیر را جمع می کند
-	$b=$a-3;	مقادیر را تفریق می کند
*	$b=$a*3;	مقادیر را ضرب می کند
/	$b=$a/3;	مقادیر را تقسیم می کند
%	$b=$a%3;	با قیمانده را بر میگرداند
++	$a++;	متغییر $a را برمیگرداند و سپس مقدار $a را یک واحد افزایش می دهد
++	++$a;	متغییر $a را یک واحد افزایش می دهد و سپس مقدار $a را بر میگرداند
--	$a--;	متغییر $a را برمیگرداند و سپس مقدار $a را یک واحد کاهش می دهد
--	--$a;	متغییر $a را یک واحد کاهش می دهد و سپس مقدار $a را بر میگرداند

عملگر های مقایسه ایی

این عملگر ها برای مقایسه دو مقدار به کار می روند . به جدول زیر توجه کنید .

عملگر	مثال	عملی که انجام می دهد
==	$b==$a;	اگر مقدار $a و $b برابر باشد مقدار true را بر میگرداند
===	$b===$a;	اگر مقدار $a و $b برابر بوده و هر دو هم نوع باشند آنگاه مقدار true را برمیگرداند
!=	$b!=$a;	اگر مقدار $a و $b برابر نباشد مقدار true را بر میگرداند
<>	$b<>$a;	اگر مقدار $a و $b برابر نباشد مقدار true را بر میگرداند
!==	$b!==$a;	اگر مقدار $a و $b برابر نبوده یا هر دو هم نوع باشند آنگاه مقدار true را برمیگرداند
<	$b<$a;	اگر مقدار $b کوچکتر از $a باشد مقدار true را بر میگرداند
>	$b>$a;	اگر مقدار $b بزرگتر از $a باشد مقدار true را بر میگرداند
<=	$b<=$a;	اگر مقدار $b کوچکتر مساوی از $a باشد مقدار true را بر میگرداند
>=	$b>=$a;	اگر مقدار $b بزرگتر مساوی از $a باشد مقدار true را بر میگرداند

عملگر های منطقی

این عملگر ها برای انجام عملیات منطقی مانند and و or به کار می روند و کاربرد گسترده ای در دستورات مختلف یک زبان برنامه نویسی دارند .

عملگر	مثال	عملی که انجام می دهد
and	$a and $b;	اگر متغییر $a و $b هر دو مقدار true را داشته باشند آنگاه true را باز میگرداند
or	$a or $b;	اگر حداقل یکی از متغییر های $a و $b مقدار true را داشته باشند آنگاه true را باز میگرداند
xor	$a xor $b;	اگر یکی از متغییر های $a و $b مقدار false و دیگری مقدار true را داشته باشند آنگاهtrue می شود
!	! $a;	مقدار متغییر $a را نقیض می کند
&&	$a && $b;	اگر متغییر $a و $b هر دو مقدار true را داشته باشند آنگاه true را باز میگرداند
||	$a || $b;	اگر حداقل یکی از متغییر های $a و $b مقدار true را داشته باشند آنگاه true را باز میگرداند

فصل ششم

دستورات شرطی

زبان های برنامه نویسی ساختار یافته دارای دستورات شرطی می باشند . دستورات شرطی با برقراری شرط خاصی مجموعه ای از دستورالعمل ها را اجرا می نمایند . PHP دو نوع دستور شرطی را ارائه می نماید .

دستور شرطی if

این دستور بر اساس شرط خاصی مجموعه ای از دستورات را اجرا می کند . به مثال if توجه کنید .

```
<?php
$a=10;
$b=16;
if($a<=$b) echo 'b bozorg tar az a ast'; //دستور شرطی یک خطی
?>
```

خروجی

b bozorg tar az a ast

برنامه بالا یک حالت ساده if است و فقط یک حالت دارد اگر a کوچکتر از b باشد متن چاپ می شود در غیر اینصورت هیچ مقداری چاپ نمی شود .

نکته : در if اگر شرط ما یک خط باشد نیاز به باز و بسته کردن بلاک نیست و برای خوانایی برنامه از بلاک باز و بسته استفاده می شود . به مثال if2 توجه کنید .

```
<?php
$a=150;
$b=120;
if($a<=$b)
{
echo 'b bozorg tar az a ast';
}
else // اگر
{
echo 'a bozorg tar az b ast';
}
?>
```

خروجی

```
a bozorg tar az b ast
```

در مثال بالا دو حالت بیشتر پیش نمی آید یا a بزگتر است یا b . دستور else در if به معنی اگر است در مثال بالا اگرa کوچکتر از b باشد متن اول چاپ می شود ولی اگر a بزرگتر از b باشد متن دومی چاپ می شود .

دستور شرطی elseif

دستور elseif این امکان را فراهم می کند تا در صورت عدم برقراری شرط دستور if شرط های دیگری را بررسی کنیم . به مثال elseif توجه کنید .

```php
<?php
$a=8;
$b=7;
if($a<=$b)
{
    echo $a-$b;
}
elseif($a==$b)
{
    echo $a+$b;
}
else
{
    echo $a*$b;
}
?>
```

خروجی

```
56
```

در مثال بالا ابتدا بررسی می شود که آیا $a کوچکتر از $b است یا نه اگر این شرط درست باشد مقدار تفاضل a و b را چاپ میکنیم و اگر شرط درست نباشد دوباره بررسی میکنیم که آیا a و b با هم مساوی هستند اگر مساوی بودند مجموع دو را چاپ می کنیم و اگر مساوی هم نبودند ضرب دو مقدار را چاپ می کنیم .

دستور شرطی switch

در بعضی مواقع ما میخواهیم تا تساوی مقدار یک متغییر را با مجمو عه ای از مقادیر متفاوت بررسی کنیم و در صورت تساوی با مقدار متغییر با هر یک از این مقدار ها مجموعه ای از دستورات را اجرا کنیم , در این مواقع از دستود switch استفاده میکنیم . دستور switch معادل مجموعه ای از دستورات if است که بر روی یک شرط یا عبارت یکسان اجرا می شوند . به مثال switch توجه کنید .

```php
<?php
$a=5;
switch($a)
{
    case 1 :
    echo 'a = 1';
    break;
    case 2 :
    echo 'a = 2';
    break;
    case 3 :
    echo 'a = 3';
    break;
    default:
    echo 'a = ?!';
}
?>
```

خروجی

```
a = ?!
```

در مثال بالا دستور switch مقدار متغییر a را دریافت کرده و آن را با مقادیر case های خود مقایسه می کند وقتی هیچکدام از case های دستور Switch با مقدار دریافتی مساوی نباشد دستور داخل default اجرا می شود .

فصل هفتم

حلقه های تکرار

حلقه ها از عناصر اصلی برنامه نویسی ساختار یافته می باشند . با کمک حلقه ها می توانیم مجموعه ایی از دستورات را به دفعات مشخص یا تا برقراری شرط خاصی اجرا نمائیم.

دستور while

دستور while یکی از دستورات لازم برای ایجاد حلقه می باشد . در ساختار این دستور از یک شرط استفاده می شود و تا برقرار بودن این شرط , مجموعه دستورالعمل های حلقه while تکرار می گردند . به مثال while توجه کنید .

```php
<?php
$a=20;
while($a>=1)
{
echo $a."<br>";
$a=$a-2;
}
?>
```

خروجی

20
18
16
14
12
10
8
6
4
2

دستور do while

دستور do while مشابه حلقه while است . البته با این تفاوت که شرط حلقه در انتهای بدنه حلقه چک می شود. بدین ترتیب حلقه حداقل یکبار اجرا می شود . به مثال dowhile توجه کنید .

```php
<?php
$a=20;
do
{
echo $a."<br>";
$a=$a-2;
} while ($a<=1);
echo "<hr>";
echo $a;
?>
```

خروجی

20

18

دستور for

از دستور for بیشتر زمانی استفاده می کنیم که تعداد دفعات تکرار را بدانیم . این حلقه دارای سه قسمت است .

مقدار دهی : این قسمت تنها یکبار در ابتدار اجرای حلقه اجرا می گردد و اغلب برای مقدار دهی ایندکس حلقه استفاده می شود .

شرط : تا زمانی که این شرط درست باشد , دستور زیر حلقه for اجرا خواهد شد .

گام حلقه : برای افزایش یا کاهش مقدار ایندکس حلقه به کار می رود .

لازم است بدانیم که هر کدام از این قسمت ها اختیاری می باشند و در صورت نیاز میتوانیم آن ها را ننویسیم و جای آن ها را خالی بگذاریم . ولی توجه داشته باشید که اگر شرط حلقه را ننویسید حلقه برنامه شما به حلقه بینهایت تبدیل خواهد شد .

به مثال for توجه کنید در این مثال اعداد زوج بین 0 تا 9 چاپ شده اند .

```php
<?php
for($a=0;$a<=9;$a+=2)
{
    echo $a."<br>";
}
?>
```

خروجی

```
0
2
4
6
8
```

دستور foreach

این دستور در PHP نسخه 4 ارائه شد و برای کار با آرایه ها استفاده می شود . بنابراین اگر شما دستور foreach را با متغییر هایی از نوع دیگری به کار ببرید یک خطا ایجاد خواهد شد . به طور کلی به دو روش از این دستور می توان استفاده کرد .

در روش اول یک ورودی از جنس آرایه به حلقه می دهیم به مثال foreach توجه کنید . در این مثال یک آرایه ایجاد نموده ایم و آن را به حلقه foreach داده ایم . برای دادن آرایه به حلقه ابتدا نام آرایه را می نویسیم و سپس از کلمه کلیدی as استفاده می کنیم و سپس یک متغییر برای ذخیره مقادیر آرایه می نویسیم که مقادیر آرایه از اول تا انتها درون این متغییر ذخیره می شود .

```php
<?php
$n=array(1,2,3,4,5,6);
foreach($n as $u)
{
    echo $u.' , ';
}
?>
```

خروجی

```
1 , 2 , 3 , 4 , 5 , 6 ,
```

در روش دوم دستور foreach یک ورودی از جنس آرایه دریافت می کند که این آرایه از جنس آرایه انجمنی است .

به مثال foreach2 توجه کنید .

```php
<?php
$age=array("peter"=>"35","amir"=>"25","mohsen"=>"19");
foreach($age as $key=>$n)
{
    echo $key.' = '.$n."<br>";
}
?>
```

خروجی

peter = 35
amir = 25
mohsen = 19

دستور Break

دستور Break برای خاتمه دادن به اجرای حلقه های for , foreach , while , do while و دستورات switch به کار می رود. به مثال break توجه کنید .

```php
<?php
$a=array("one","two","three","stop","four");
foreach($a as $key)
{
    if($key=="stop")
    break;
    echo $key."<br>";
}
?>
```

خروجی

one
two
three

دستور Continue

این دستور در حلقه های تکرار به کار می رود و زمانی که اجرا می شود PHP از بقیه دستورات حلقه صرف نظر می کند و کنترل برنامه را به ابتدای حلقه منتقل می کند . لازم به ذکر است که از این دستور در دستور switch هم می توان استفاده کرد . به مثال continue توجه کنید .

```php
<?php
for($a=0;$a<=10;$a++)
{
    if($a%2==0)
    continue;
    echo $a.' , ';
}
?>
```

خروجی

1 , 3 , 5 , 7 , 9 ,

فصل هشتم

نحوه ایجاد توابع

یک تابع بلاکی از کد ها است که یکبار تعریف می شوند و در قسمت های مختلف برنامه ممکن است به دفعات متعدد فراخوانی و اجرا گردد . در PHP توابع با دستور العمل function تعریف می شوند . به شکل توابع در دقت کنید .

```
<?php
function نام تابع (پارامتر دوم,پارامتر اول)
{
    بدنه تابع
}
?>
```

پارامتر ها با (,) از هم جدا می شوند . توابع می توانند بدون ورودی باشند . برای استفاده از تابع اگر درون تابع از دستور return که جلوتر با آن آشنا می شوید استفاده شده باشد باید نام تابع را نوشته و یا آن را چاپ نمود یا درون متغییری ذخیره نمایید به مثال function توجه کنید .

```
<?php
function c()
{
     echo 'Welcome';
}
c();
?>
```

خروجی

Welcome

توابع می توانند ورودی داشته باشند . به مثال function2 توجه کنید .

```
<?php
function c($name)
{
     echo $name.' , ';
}
c('ali');
c('mohsen');
c('amir');
?>
```

خروجی

ali , mohsen , amir ,

ورودی توابع می توانند مقدار اولیه داشته باشند به مثال function3 توجه کنید .

```php
<?php
function c($n=10)
{
     echo $n."<br>";
}
c('568');
c('69');
c();
?>
```

خروجی

568
69
10

نکته : به طور کلی یک تابع ممکن است مقداری را بر نگرداند . در این حالت مسلما در بدنه تابع از دستور return استفاده نخواهد شد , در اغلب موارد این نوع از توابع نتایج به دست آورده را چاپ می کند .

بعضی از توابع از دستور return استفاده می کنند که برای بازگشت مقدار تابع است به مثال returnfunction توجه کنید .

```php
<?php
function c($n)
{
    return $n."<br>";
}
echo c('568');
echo c('69');
?>
```

خروجی

568
69

محدوده متغییر

متغییر هایی که در داخل یک تابع تعریف می شوند متغییر محلی هستند و فقط در داخل آن تابع قابل دسترس هستند . متغییر هایی که در بیرون تابع تعریف می شوند به صورت سراسری هستند و در داخل تابع قابل استفاده نیستند همانطور که با متغییر global آشنا شدید به کمک آن می توان از متغییر سراسری در داخل تابع استفاده کرد .

توابع بازگشتی

زمانی که یک تابع خودش را فراخوانی کند تابع به تابع بازگشتی تبدیل می شود . بعضی اوقات مسئله ایی که حل می نمائیم به طور ذاتی حالت بازگشتی دارد و برای حل مسئله لازم است مسئله که به واحد های کوچکتری شکسته شده و یک الگوریتم بر روی تمامی آنها اعمال می شود و این مسئله به دفعات متعدد تکرار می گردد . استفاده از توابع بازگشتی در ریاضیات متداول می باشد . محاسبه فاکتوریل یک نمونه از مسائل بازگشتی است .در مثال functionbazgashti یک تابع بازگشتی نوشته شده است که این تابع فاکتوریل عدد داده شده را محاسبه می کند به مثال functionbazgashti توجه کنید .

```php
<?php
function fac($n){
    if($n>0)
    {
        return $n*fac($n-1);
    }
    return 1;
}
echo fac(4);
echo "<br>";
echo fac(5);
?>
```

خروجی

```
24
120
```

فصل نهم

آرایه های فوق سراسری

متغییر های از پیش تعریف شده آرایه های فوق سراسری هم نامیده می شوند زیرا بدون توجه به محدوده و بدون استفاده از کلمه کلیدی global در همه جای برنامه قابل دسترس می باشند . متغییر های از پیش تعریف شده موجود در زبان PHP را در زیر مشاهده می کنید که در ادامه کاربرد موارد مهم آن ها را توضیح می دهیم .

- $GLOBALS
- $_SERVER
- $_GET
- $_POST
- $_COOKIE
- $_FILES
- $_ENV
- $_REQUEST
- $_SESSION

متغییر $GLOBALS

تمامی متغییر سراسری موجود در یک برنامه PHP در این آرایه موجود می باشد . البته بیشتر متغییرهای موجود در این آرایه در سایر آرایه های فوق سراسری نیز موجود دارند . در قسمت های قبل نحوه استفاده از این متغییر را دیده اید .

متغییر $_SERVER

تمامی داده هایی که سرویس دهنده در یک پیام پاسخ HTTP به سرویس گیرنده ارسال می کند در این آرایه قرار می گیرند . این اطلاعات عبارتند از :

- نام اسکریپت در حال اجرا
- نام سرویس دهنده
- نسخه HTTP

- آدرس IP راه دور
- .....

در مثال زیر چند از این اطلاعات مهم آورده شده است به مثال server$ توجه کنید .

```php
<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['SERVER_ADDR'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>
```

خروجی

```
/10.php
localhost
localhost:8088
127.0.0.1
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.63 Safari/537.36
/10.php
```

متغییر $_REQUEST

این آرایه شامل عناصر آرایه های $_GET و $_POST و $_COOKIE می باشد . در مثال زیر توسط یک فرم ورودی را از کاربر میگرید و با استفاده از آرایه $_REQUEST مقدار آن را چاپ می کند . به مثال request$ توجه کنید .

```
<form method="post" action="">
Name: <input type="text" name="name">
<br />
<input type="submit">
</form>
<?php
if(isset($_REQUEST['name']))
{
    $name = $_REQUEST['name'];
    echo $name;
}
?>
```

خروجی

در مثال فوق یک فرم برای دریافت ورودی از کاربر ایجاد شده است که از متد POST استفاده می کند و با زدن دکمه Submit مقدار دریافتی کاربر به همین صفحه ارسال می شود و اما در قسمت PHP هم ابتدا عدم وجود پارامتر ارسالی name ابتدا چک می شود و اگر این متغییر به این صفحه ارسال شده باشد آن را چاپ می کند .

متغییر $_POST

یکی از روش های ارسال داده های فرم به اسکریپت سرویس دهنده می باشد . زمانی که یک فرم داده های خود را با متد POST به مقصد به مقصد ارسال می نماید آنگاه این داده ها در اسکریپت مقصد با آرایه $_POST قابل دسترس خواهند بود . وقتی که فرم داده های خود را با این روش ارسال می کند داده ها به صورت رمز نگاری شده ارسال می شوند و در URL نیز نشان داده نمی شوند و از امنیت بالایی نسبت به متد دیگر ارسال داده $_GET دارد . به مثال $post توجه کنید در این مثال ابتدا با تابع isset چک میکنیم که آیا متغییر age به این صفحه با متد Post ارسال شده است یا خیر اگر ارسال شده باشد این تابع مقدار true را بر میگرداند در غیر این صورت مقدار false را بر میگرداند و دستورات داخل if اجرا نمی شود . با زدن دکمه Submit هر مقداری که در کادر نوشته شده باشد چاپ می شود .

```
<form method="post" action="">
Age : <input type="text" name="age">
<br />
<input type="submit">
</form>
<?php
if(isset($_POST['age']))
{
    $age = $_POST['age'];
    echo $age;
}
?>
```

خروجی

Age :
Submit

متغییر $_GET

یکی دیگر از روش های ارسال داده های فرم به اسکریپت سرویس دهنده می باشد . زمانی که یک فرم داده های خود را با متد GET به مقصد ارسال می نماید آنگاه این داده ها در اسکریپت مقصد با آرایه $_GET قابل دسترس خواهند بود . وقتی که فرم داده های خود را با این روش ارسال می کند داده ها از طریق URL ارسال می شوند و امنیت کم دارد ولی سرعت آن نسبت به متد post بیشتر می باشد . به مثال $get توجه کنید با زدن دکمه Submit می بینید که هر چه داخل کادر نوشته شود از طریق Url به خوده صفحه ارسال می شود و چاپ می شود .

```
<form method="get" action="">
Name : <input type="text" name="n">
<br />
<input type="submit">
</form>
<?php
if(isset($_GET['n']))
{
    $name = $_GET['n'];
    echo $name;
}
?>
```

خروجی

Name :
Submit

متغییر $_COOKIE

این آرایه شامل تمامی کوکی هایی است که توسط مرورگر وب به سرویس دهنده وب ارسال شده است . در اسکریپت مقصد برای خواندن کوکی ها از این آرایه استفاده می شود . کوکی ها داده های کوچکی هستند که سرویس دهنده آن ها را با کمک مرورگر وب بر روی سیستم کاربران می نویسد تا کاربران را در مراجعات بعدی تشخیص دهد . برای نوشتن کوکی ها بر روی سیستم کاربران تابع setcookie() به کار می رود . امروزه به دلیل امنیت کم کوکی از آن خیلی کم استفاده می شود .

مقادیر پیشفرض دستور setcookie را در جدول زیر مشاهده می کنید .

مقادیر پیش فرض پارامترها	
نام پارامتر	مقدار پیش فرض
path	/ (تمامی دایرکتوری های سرویس دهنده)
domain	دامنه سرویس سرویس دهنده ای که کوکی را ایجاد کرده است
Expire information	تا زمانی که سرویس گیرنده وب بسته شود
secure	(غیر فعال شده) disabled

در مثال زیر خط اول ساختار دستور setcookie نشان داده شده است و در خط بعدی یک کوکی به اسم name و مقدار آن نیز مقدار متغییر $name است و زمان انقضا آن زمان سیستم به اضافه 3600 ثانیه است یعنی یک ساعت .

```
<?php
setcookie(زمان انقضا,مقدار کوکی,اسم کوکی);
setcookie("name","$name",time()+3600);
?>
```

به مثال $cookie توجه کنید . در این مثال در قسمت html یک فرم برای دریافت نام کاربر طراحی شده است که با متد get ورودی کاربر را به همین صفحه ارسال می کند . در قسمت PHP ابتدا با تابع isset

چک میکنیم که آیا ورودی ایی به این صفحه ارسال شده یا خیر . این تابع یک ورودی میگیرد اگر متغییری که به این ورودی می دهیم قبلا ایجاد شده باشد مقدار true را بر میگرداند در غیر این صورت مقدار false را بر میگرداند . اگر ورودی ارسال شده است آن را در کوکی به مدت انقضا یک دقیقه ذخیره می کنیم و اگر ورودی ارسال نشده باشد چک میکنیم که آیا کوکی کاربر که مدت انقضا آن یک دقیقه است وجود دارد یا خیر اگر وجود داشت آن را چاپ می کنیم .

```
<form action="" method="get">
Name :
<input type="text" name="name" />
<input type="submit" value="Save" />
</form>
<?php
if(isset($_GET['name']))
{
    $name=$_GET['name'];
    setcookie("name","$name",time()+60);
    echo "Welcome $name"."<br>"." Please Reopen Page";
}
elseif(isset($_COOKIE['name']))
{
    echo $_COOKIE['name']."<br>"." Is Read Cookie !";
}
?>
```

متغییر $_FILES

شامل فایل هایی است که کاربران به سرویس دهنده آپلود (ارسال) می کند .

متغییر $_ENV

شامل اطلاعاتی درباره سرویس دهنده و PHP می باشد . اطلاعاتی همانند نام کامپیوتر , سیستم عامل , درایو سیستم در این آرایه قرار می گیرند .

متغییر $_SESSION

شامل تمامی متغییر هایی است که هم اکنون به عنوان متغییر جلسه ثبت شده اند . جلسه یا session یک از روش هایی است که سرویس دهنده وب برای تشخیص و اهراز هویت کاربران در مراجعات مختلف به کار می رود . برای هر جلسه می توان به تعداد دلخواه متغییر های را تعریف کرد , این متغییر ها جزء

آرایه فوق سراسری $_SESSION می شوند و در جلسه قابل استفاده می باشند . جلسه با دستور session_start() شروع می شود . به مثال $session توجه کنید .

در این مثال فرمی برای ورود کاربر طراحی شده است و اما در قسمت PHP آن ابتدا جلسه را شروع می کنیم ابتدا چک میکنیم که آیا اطلاعات به این صفحه ارسال شده است یا خیر اگر ارسال نشده باشد چک میکنیم که آیا جلسه از دفعه قبل وجود دارد یا خیر اگر وجود داشته باشد مقدار آن را چاپ میکنیم ولی اگر اطلاعات ارسال شده بود یک جلسه را شروع میکنیم و یک متغییر جلسه می سازیم و اسم کاربر را درون آن قرار می دهیم و آدرس url مرورگر را به دوباره به خود این صفحه قرار می دهیم و سپس لینکی برای پایان جلسه و خروج کاربر قرار می دهیم که اگر کاربر روی لینک کلیک کند یک متغییر به نام action را با مقدار logout به همین صفحه با متد get ارسال می کند و در if آخر چک میکنیم که آیا کاربر روی این لینک کلیک کرده است یا خیر اگر کلیک کرده باشد جلسه را پایان میدهیم و دوباره مرورگر را به این صفحه ارجاع می دهیم . این مثال درست همانند صفحه login کاربران می باشد .

```
<form method="get" action="">
Name :
<input type="text" name="name" />
<input type="submit" value="Save!" />
</form>
<?php
session_start();
if(isset($_GET['name']))
{
    $name=$_GET['name'];
    $_SESSION['fname']=$name;
    header("location:10-5.php");
}
elseif(isset($_SESSION['fname']))
{
    echo "welcome : ".$_SESSION['fname']."<br>";
?>
<a href="?action=logout">loguot</a>
<?php
}
if(isset($_GET['action']) && $_GET['action']=='logout')
{
    session_destroy();
    header("location:10-5.php");
}
?>
```

دستور include

از این دستور برای پیوست کردن محتویات یک فایل PHP درون یک فایل دیگر استفاده می شود. در این حالت سرور PHP در هنگام اجرای صفحه محتویات فایل اضافه شده راخوانده و آن را با صفحه اول ترکیب می کند سپس خروجی را در قالب یک صفحه نمایش می دهد . در تصویر فوق ابتدا در خط اول نحوه استفاده از دستور include را نشان می دهد و در خط دوم مقدار فایل menu.php به این صفحه پیوست شده است .

```
<?php
include('آدرس و نام فایل');
include('menu.php');
?>
```

دستور require

این دستور همانند دستور include عمل می کند و برای پیوست کردن فایل به کار می رود اما این دو دستور با هم تفاوت هایی دارند . تفاوت بین دستور include و require این است که اگر فایل فرخوانی شده وجود نداشته باشد و یا پرمیشن لازم برای خواندن یا اجرا را نداشته باشد دستور include فقط یه warning بر میگرداند و برنامه را ادامه می دهد اما دستور require یک error بر میگرداند و اجرا برنامه را متوقف می کند .

این تفاوت از نظر امنیتی مهم است فرض کنید برای محیط ادمین فایل check.php را فراخوانی می کنید و در این فایل صحت کاربر صورت می گیرد و اگر کاربر درست باشد ادامه برنامه اجرا می شود و گر نه به صفحه قبل ارجا می شود حال اگر این فایل را با دستور Include پیوست کرده باشیم یک warning رخ می دهد و برنامه به محیط امین می رود ولی اگر با دستور require فراخوانی شود اجرا برنامه متوقف می شود .

```
<?php
require('آدرس و نام فایل');
require('menu.php');
?>
```

دستور include_once() و require_once()

تصور کنید در فایل index.php فایل x و y پیوست شده اند و درون فایل x فایل y پیوست شده است در این صورت شما فایل y را دوبار پیوست کرده اید . اینکار اگر روال اجرا کد ها را خراب نکند باعث پردازش بیش از حد می شود در این مواقع بهتر است از دستور include_once و یا دستور require_once استفاده کنیم این دو دستور کمک می کند که فایل را یکبار پیوست کنیم اگر فایلی را قبلا یکبار فراخوانی کرده ایم و به هر دلیلی دوباره آن را فراخوانی کنیم دفعه دوم , سوم و.. فراخوانی صورت نمیگیرد وطبیعتا اگر قبلا فراخوانی نشده بود آن را فرخوانی می کند . فرق این دو دستور هم مانند این دستور ها بدون کلمه once است .

```php
<?php
require_once('آدرس و نام فایل');
require_once('menu.php');
//
include_once('آدرس و نام فایل');
include_once('menu.php');
?>
```

فصل دهم

PHP و کار با فایل ها

PHP قابلیت خواندن و نوشتن فایل های سرویس دهنده را دارد . بنا بر این می توان برنامه های کاربردی متعددی ر انوشت تا از فایل سیستم سرویس دهنده استفاده کرده و یا فایل هایی را بر روی سرویس دهنده ایجاد یا تغییر دهند . PHP بر روی لینوکس به عنوان همان کاربر مالک سرویس دهنده آپاچی و بر روی ویندوز به عنوان کاربر میهمان اجرا می شود . به طور کلی PHP قابلیت با هر نوع فایلی را دارد اما به طور معمول ما با فایل های متنی کار خواهیم کرد . سیستم های عامل ویندوز و لینوکس تفاوت های متعددی در کار با فایل ها دارند . یکی از این تفاوت ها به نحوه مشخص کردن مسیر ها مربوط می شود . به طور مثال برای مشخص کردن مسیر یک فایل در لینوکس و سیستم عامل های همانند یونیکس از کاراکتر / استفاده می شود . در قسمت زیر نحوه استفاده از مسیر در تابع fopen آمده است :

```
/home/dan/data/data.txt
```

در سیستم عامل ویندوز نیز همانند مثال زیر از کاراکتر \ برای مسیر ها استفاده می شود :

```
c:\mydocs\data.txt
```

البته در صورتی که بخواهید مسیر های ویندوز را در توابع مختلف PHP به کار برید باید به جای یک کاراکتر \ از دو کاراکتر \\ استفاده کنید به طور مثال اگر مسیر c:\mydocs\data.txt را در یک تابع PHP به کار برید باید این مسیر را به صورت زیر تغییر دهید : c:\\mydocs\\data\\data.txt

تابع fopen

این تابع برای باز کردن یک فایل به کار می رود و handle مرتبط با فایل باز شده را بر می گرداند . این تابع سه آرگومنت زیر را دریافت می کند :

- نام فایل
- مد
- آرگومنت اختیاری

در صورتی که فایل به درستی باز شود مقدار بازگشتی یک مقدار صحیح مثبت خواهد بود اگر در باز کردن فایل مشکلی به وجود آید آنگاه این مقدار صفر خواهد شد . بنا بر این بعد از باز کردن یک فایل بهتر است بررسی کنیم که آیا فایل به درستی باز شده است یا نه . به کد زیر توجه کنید .

```
<?php
$fil=fopen("./data.txt","r");
if(!$fil) die("cannot open the file");
?>
```

آرگومنت اول آدرس فایلی است که می خواهیم آن را باز کنیم . این مسیر می تواند به طور نسبی یا به طور مطلق ذکر می شود . به طور مثال اگر دایرکتوری سرویس دهنده وب ما d:\in\wwwroot باشدآنگاه دستور fopen(“./data.txt”,”w”) یک فایل جدید بنام data را در دایرکتوری وب سایت یعنی d:\in\wwwroot ایجاد خواهد نمود این مسیر نسبی می باشد ولی اگر آدرس فایل را به صورت d:\in\wwwroot وارد کنیم آدرس مطلق نامیده می شود و یک فایل در آن مسیر ایجاد خواهد نمود .

آرگومنت دوم تابع fopen	
مقدار	توضیحات
r	فایل را برای خواندن باز می کند و نشانگر موقعیت جاری فایل را در ابتدای فایل قرار می دهد
r+	فایل را برای خواندن و نوشتن باز می کند و نشانگر موقعیت جاری فایل را در ابتدای فایل قرار می دهد
w	فایل را برای نوشتن باز می کند و تمامی محتوای فایل را از بین می برد . اگر فایل قبلا و جود نداشته باشد آنگاه آن را ایجاد می کند
w+	فایل را برای خواندن و نوشتن باز می کند و تمامی محتوای فایل را از بین می برد . اگر فایل قبلا و جود نداشته باشد آنگاه آن را ایجاد می کند
a	فایل را باز می کند تا اطلاعاتی را به انتهای آن بیفزاییم . محتوای فایل باز شده را از بین نمی برد و اگر فایل قبلا وجود نداشته باشد آنگاه آن را ایجاد می کند
a+	فایل را برای خواندن و افزودن اطلاعات به انتهای آن باز می کند . محتوای فایل را از بین نمی برد و اگر فایل قبلا وجود نداشته باشد آنگاه آن را ایجاد می کند

تابع fclose

بعد از اینکه پردازش ها و عملیات مورد نظر را بر روی فایل باز شده انجام دادیم باید فایل را ببندیم به همین منظور باید از تابع fclose استفاده کرد . این تابع فایل باز شده را می بندد در صورتی که این عملیات را با موفقعیت انجام دهد مقدار true را بر میگرداند در غیر این صورت مقدار false را بر می گرداند .

```php
<?php
$file=fopen("somefile.txt","w");
fclose($file);
?>
```

بدست آوردن اطلاعات فایل

فایل ها علاوه بر اطلاعات که نگهداری می کنند اطلاعاتی را نیز درباره ی خودشان دارا می باشد . نام , اندازه , زمان تغییر و ..... از جمله اطلاعات یک فایل می باشند . در PHP تابع stat() ما را قادر می سازد درباره یک فایل اطلاعاتی را بدست آوریم .

عناصر آرایه بازگشتی تابع stat()

ایندکس	نام	اطلاعات
0	dev	شماره device
1	ino	شماره incode
2	mode	مد حافظه incode
3	nlink	تعداد لینک ها
4	uid	شناسه کاربر مالک
5	gid	شناسه گروه مالک
6	rdev	نوع وسیله
7	size	اندازه بر حسب بایت
8	atime	زمان آخرین دستیابی

9	mtime	زمان آخرین تغییر
10	ctime	زمان آخرین تغییر
11	blksize	اندازه بلاک فایل سیستم
12	blocks	تعداد بلاک های اختصاص یافته

به مثال stat توجه کنید .

```php
<?php
echo "<pre>";
print_r(stat("12.php"));
echo "</pre>";
?>
```

خروجی

```
Array
(
    [0] => 2
    [1] => 0
    [2] => 33206
    [3] => 1
    [4] => 0
    [5] => 0
    [6] => 2
    [7] => 390
    [8] => 1389964018
    [9] => 1389964018
    [10] => 1389949099
    [11] => -1
    [12] => -1
    [dev] => 2
    [ino] => 0
    [mode] => 33206
    [nlink] => 1
    [uid] => 0
    [gid] => 0
    [rdev] => 2
    [size] => 390
    [atime] => 1389964018
    [mtime] => 1389964018
    [ctime] => 1389949099
    [blksize] => -1
    [blocks] => -1
)
```

تابع fread()

این تابع برای خواندن یک رشته از یک فایل به کار می رود . تابع fread() یک فایل و یک مقدار صحیح را به عنوان آرگومنت دریافت می کند سپس به اندازه مشخص از فایل مورد نظر خوانده و آن را بر می گرداند . به مثال fread توجه کنید . در این مثال فایل data.txt برای خواندن باز می شود و از اول آن به میزان 8 کاراکتر خوانده می شود و در متغییر $data ذخیره می شود .

```
<?php
$file=fopen("data.txt","r");
$data=fread($file,8);
echo $data;
?>
```

خروجی

user:moh

تابع fwrite()

از این تابع برای نوشتن اطلاعات بر روی یک فایل استفاده می شود . تابع fwrite همانند تابع fread() دو آرگومان می گیرد .این تابع آرگومنت دوم را در فایل مشخص شده توسط آرگومنت اولی می نویسد و سپس تعداد کاراکتر های نوشته شده را بر میگرداند و در صورتی که این تابع با خطایی مواجه شود مقدار 1– را بر میگرداند .

به مثال fwrite توجه کنید .

```
<?php
$name="Mohsen Rajabi";
$file=fopen("user.txt","w");
echo fwrite($file,$name);
?>
```

خروجی

13

این تابع همچنان ورودی سوم هم دریافت می کند که میزان حداکثر نوشتن در فایل است به شکل زیر نگاه کنید عدد 4 به معنی این است که در فایل باز شده رشته Mohs نوشته می شود .

```
$file=fopen("user.txt","w",4);
```

Upload کردن فایل

سایت های مختلف اینترنتی همانند یاهو امکان آپلود فایل ها را فراهم می آوردند . در این سایت ها کاربر می تواند فایل های مورد نظر را از سیستم خود به سرویس دهنده ارسال نماید . PHP نیز قابلیت مدیریت عملیات آپلود فایل ها را دارا می باشد . البته زمانی که شما به کاربران امکان آپلود فایل ها را می دهید ممکن است مشکلات امنیتی برای سرویس دهنده شما به وجود آید لذا به هنگام فعال سازی این قابلیت مسائل امنیتی را نیز به طور کامل در نظر بگیرید .

به مثال فوق توجه کنید . در این مثال ابتدا یک فرم برای انتخاب فایلی که قرار است آپلود شود ایجاد می کنیم که این فرم حتما باید با متد post ارسال شود و خاصیت enctype آن هم باید روی multipart/form-data تنظیم شده باشد . ابتدا یک ورودی برای انتخاب فایل در فرم قرار می دهیم و یک دکمه برای ارسال هم قرار ی دهیم و action فرم را هم روی فایل uploader.php می گذاریم . در فایل uploader.php ابتدا بررسی می کنیم که آیا فایلی از فرم ارسال شده است یا خیر اگر ارسال نشده باشد دوباره url مرورگر را به صفحه فرم بر میگردانیم ولی اگر اطلاعات ارسال شده باشد دوباره با متغییر سراسری $_FILES بررسی میکنیم که آیا error وجود دارد یا خیر اگر error وجود نداشته باشد با استفاده از تابع move_uploaded_file که دو ورودی می گیرد که پارامتر اول مسیر فایل موقت آپلود شده است و پارامتر دوم مسیری است که فایل باید آپلود شود .

مقدار فایل formupload.html را در تصویر فوق مشاهده می کنید .

```
<form action="uploader.php" method="post" enctype="multipart/form-data">
<input type="file" name="myfile" />
<br /><hr />
<input type="submit" value="Upload!" />
</form>
```

مقدار فایل uploader.php را در تصویر فوق مشاهده می کنید .

```php
<?php
if(!isset($_POST['myfile']))
{
    if($_FILES['myfile']['error']==0)
    {
        $name=$_FILES['myfile']['name'];
        $address=$_FILES['myfile']['tmp_name'];
        move_uploaded_file($address,"c:/".$name);
        echo 'Uploading Complete';
    }
    else
    {
        echo 'Uploading File Error';
    }

}
else
{
    header("location:formupload.html");
}


?>
```

مقداری آرایه فوق سراسری $_FILES را در جدول زیر مشاهده می کنید .

عناصر آرایه فوق سراسری $_FILES

عنصر	توضیحات
$_FILES['userfile']['name']	نام فایل آپلود شده
$_FILES['userfile']['type']	نوع فایل ارسال شده به طور مثال image/gif
$_FILES['userfile']['size']	اندازه فایل آپلود شده
$_FILES['userfile']['tmp_name']	نام فایل موقت دارای فایل آپلود شده
$_FILES['userfile']['error']	کد خطا ایجاد شده هنگام آپلود فایل

ارسال email با PHP

برای ارسال پیام های متنی ساده از دستور ()mail استفاده می شود . این دستور برای ارسال ایمیل به سیستم پست الکترونیکی محلی متکی می باشد . در تصویر فوق نحوه استفاده از این تابع آمده است .

```
mail($to,$subject,$message,$mailheader);
```

هر یک از ورودی های این تابع در جدول زیر آمده است .

پارامتر های تابع mail

پارامتر	توضیحات
$to	آدرس مقصد . اگر پیامی را بخواهید به چند نفر ارسال کنید باید آدرس ها را با کاراکتر (,) از هم جدا کنید
$subject	موضوع پیام
$message	بدنه یا خود پیام
$mailheader	این رشته به انتهای هدر پست الکترونیکی افزوده می شود . خطوط جداگانه در این رشته را می توان با \r\n ایجاد کرد .

در صورتی که ایمیل به درستی به سیستم پست الکترونیکی محلی تحویل شده باشد این تابع مقدار true را بر میگرداند ولی این مسئله به این معنی نمی باشد که ایمیل به مقصد نهایی تحویل داده شده باشد .

به مثال mail توجه کنید .

```php
<?php
$to="m.kabir8895@yahoo.com";
$subject="Persian3Da";
$user='Mohsen Rajabi';
$pass='565610';
$message="Hello , Welcome to our service . To access our site ,
Username=$user
Password=$pass
If you should have problems with our service ,
Please contact <mailto:m.kabir8895@yahoo.com>.";
$mailheader="to visite our site now , click here :
http://persian3da.ir/";
$checkmail=mail($to,$subject,$message,$mailheader);
if($checkmail)
echo "Send Email Complete";
else
echo "Send Email Error";
?>
```

فصل یازدهم

برنامه نویسی شئ گرا

قابلیت شئ گرایی در PHP نسخه 3 به بعد ایجاد شده است . روش برنامه نویسی که در جلسات قبل مشاهده کردید برنامه نویسی رویه ایی نام دارد . در این مدل برنامه نویسی یک برنامه کاربردی از یکسری توابع یا رویه ها تشکیل می شود و در حین اجرای برنامه توابع مورد نظر با داده های لازم فراخوانی و اجرا می گردند . در این جلسه می خواهیم تا متد متفاوتی در برنامه نویسی به نام برنامه نویسی شئ گرا را معرفی کنیم .

در برنامه نویسی شئ گرا با کمک اشیاء می توانیم اشیاء موجود در دنیای واقعی همانند چیز های مختلف , فرآیند ها و ایده ها را مدل سازی کنیم . در این مدل برنامه نویسی برنامه از یکسری شئ تشکیل می شوند که هر کدام در حقیقت نقش یکی از اشیاء موجود در دنیای واقعی را بازی می کند همچنین هر شئ عملیات خاصی را انجام می دهد و سرویس های خاصی را به سایر اشیاء می تواند ارائه دهد ولی به طور کلی جزوئیات نحوه انجام عملیات داخل هر شئ برای اشیاء دیگر پنهان می باشد .

Oop (object oriented programming) برای سادگی برنامه نویسان به وجود آمده است با استفاده از oop شما می توانید مسائل بزرگ را تبدیل به چند مسئله کوچک نمایید که نسبت به مسئله اصلی راحت تر حل می شوند .

اهداف اصلی oop را می توان به طور خلاصه چنین بر شمرد :

- قابلیت استفاده مجدد : یک شئ موجودیتی است که می تواند خصوصیات و رفتار ها را در بر داشته و یا با اشیاء دیگر مرتبط باشد . یک شئ اغلب برای رفع مجموعه مشخصی از مشکلات تولید می شود هرگاه در سایر پروژه ها نیاز به رفع مشکلات مشابهی باشد می توان از همان شئ در آن پروژه استفاده کرد .
- اصلاح مجدد : اگر نیاز به اصلاح پروژه های خود داشته باشید , oop به شما حداکثر سود را می رساند زیرا تمامی اشیاء موجود در برنامه , موجودیت های کوچکی هستند که شامل خصوصیات و رفتار های خاص خود می باشند . بنابر این تغییر و اصلاح آن ها به مراتب ساده تر است .
- قابلیت گسترش : با استفاده از oop میتوانید اشیاء خود را بازنویسی کرده و ویژگی ها ی خاصی را به آن ها اضافه کنید همزمان با این کار می توانید سازگاری با نسخه های قبلی را نیز حفظ نمایید . برای

این کار کد قبلی شما به صورت پایه تعریف و امکانات جدید به آن اضافه شده و آن را گسترش می دهد. بدین ترتیب اشیاء جدید تمامی خصوصیات و شئ والد را از آن به ارث می برند و سپس ویژگی های جدید را به آن اضافه می نمایند . این عمل اصطلاحا وراثت نام دارد و یکی از ویژگی های بسیار مهم oop به شمار می رود .

- قابلیت نگهداری : کد شئ گرا راحت تر نگهداری می شود برای مثال وقتی که یک برنامه نویس آنرا توسعه می دهد , بازنویسی و یا اشکال زدایی می کند می توان به راحتی ساختار کدنویسی داخلی آن را کشف کرد و کد را هر زمان به روز رسانی نمود .
- کارآیی : برنامه نویسی شئ گرا در حقیقت برای کارآیی بهتر و راحتی فرآیند توسعه نرم افزار به وجود آمده است .ابتدا مسئله ر ابه مجموعه ایی از مسائل کوچک تبدیل می کنید و سپس راه حل مسائل کوچک را می یابد و طبیعتا مسئله بزرگ و اصلی بطور خودکار حل خواهد شد .

تعاریف مهم در شئ گرایی

کلاس

یک کلاس الگویی برای ایجاد اشیاء است و ساختار اشیاء را مشخص کرده و متد ها و خاصیت های یک شئ را تعیین می نماید . کلاس چیزی به جزء قطعه ایی از کد با تعدادی خصوصیات (فیلد–field) و

رفتار ها (متد–method) نیست . خصوصیات داده های اشیاء هستند و به صورت متغییر هائی در داخل کلاس تعریف می شوند . یک خاصیت ممکن است یه مقدار , یک آرایه و حتی یک شئ باشد .

متد ها در حقیقت توابع یا عملیاتی هستند که در داخل کلاس یک شئ تعرف می گردد .

اشیاء

الگوی یک شئ است . در برنامه نویسی وقتی کلاس تعریف می شود این کلاس به تنهایی هیچ قابلیت اجرایی ندارد و قتی از کلاس شئ ایجاد می کنیم . آن وقت می توان از آن شئ در برنامه استفاده نمود و تمامی ویژگی هایی که در کلاس مربوطه تعریف شده است (اعم از خصوصیات,رفتارها و...) بر روی شئ مربوطه قابل دسترسی خواهند بود . در oop هیچ شئی بدون کلاس وجود ندارد و به عبارت دیگر اشیاء از روی کلاس ها ساخته می شوند و تا زمانی که کلاس نباشد شئ نیز وجود نخواهد داشت .

خاصیت

یک خاصیت یا فیلد در حقیقت متغییری از کلاس است که مستقیما درون خود کلاس (و نه درون متد های داخل کلاس) تعریف میشود . این عناصر توسط تمامی توابع (متد ها) در تمامی بخش های کلاس قابل دسترسی هستند .

متد

متد ها توابعی هستند که درون یک کلاس تعریف می شوند و رفتار های اشیاء ایجاد شده از کلاس را تعریف می کند .

کپسوله سازی

این اصطلاح به معنی مکانیزمی است که توسط آن کد درون کلاس (متدها) و داده های موجود در آن (فیلدها) به هم متصل می شوند و هر دو مورد (فیلد و متد) از تداخل خارجی و استفاده نامناسب مصون می مانند . در حقیقت به عمل محافظت از داده ها و متد ها از طریق یک سیستم واحد (کلاس) کپسوله سازی می گوند . مزیت اصلی کپسوله سازی تضمین صحت اطلاعات و انجام عملیات است .

وراثت

فرآیند مشتق شدن یک کلاس از کلاس دیگر و گسترش امکانات تعریف شده آن , وراثت نام دارد. وقتی یک کلاس را از کلاس دیگری مشتق می شود کلاس مشتق شده (فرعی) تمامی رفتار ها و خصوصیات کلاس والد (اصلی) را به ارث می برد . سپس کلاس فرعی می تواند متد ها و فیلد های دلخواه خود را اضافه کرده یا تغییر دهد .

چند ریختی

اگر یک کلاس از کلاس دیگیری مشتق شده باشد می تواند متد های آن را بازنویسی کند . بدین ترتیب اگر یک شئ از این کلاس مشتق شده ایجاد شود , این امکان وجود دارد که با فراخوانی یک متد خاص رفتاری متفاوت با زمانی که همان متد را از شئ والد ایجاد شده است صدا می زنیم , اجرا گردد به این

عمل , چند ریختی می گوییم که در آن یک شئ بسته به آن که از کلاس والد یا مشتق شده ایجاد شود , با فراخوانی یک متد خاص می تواند به اشکال مختلفی رفتار کند .

امتزاج

این اصطلاح بیانگر وابستگی کلاس ها به یکدیگر است . برای مثال ممکن است یک کلاس حاوی یک یا چند فیلد باشد که این فیلد ها در حقیقت اشیائی از کلاس های دیگر باشند . بدین ترتیب , اشیاء یک کلاس بدون وجود کلاس دیگر قادر به کار کردن نیستند . هر چقدر امتزاج یک کلاس کمتر باشد قابلیت استفاده مجدد کلاس مجبور در سایر پروژه ها افزایش خواهد یافت و برعکس .

کلاس برتر (والد)

در مفاهیم وراثت , کلاسی که سایر کلاس ها از آن مشتق می شوند و امکانات ان را توسعه می دهند , کلاس برتر یا والد (parent) نام دارد .

کلاس فرعی (فرزند)

به کلاسی که از یک کلاس دیگر مشتق شده باشد و خصوصیات و رفتار های آن را به ارث می برد , کلاس فرعی یا فرزند (child) می گویند .

ایجاد کلاس

برای ایجاد کلاس باید ابتدا کلمه کلیدی class و سپس نام آن کلاس را بنویسید و سپس بلاک باز کرده و خاصیت ها و متد های آن کلاس را نوشته و بلاک را می بندیم .

فیلد ها درون کلاس با کامه $this مورد استفاده قرار می گیرند .

به مثال class توجه کنید .

```php
<?php
class human{
private $name;
private $age;
private $family;
    function set($nam,$sen,$subnam)
    {
        $this->name=$nam;
        $this->family=$subnam;
        $this->age=$sen;
    }
    function get()
    {
        echo "Your Name : ".$this->name."<br>";
        echo "Your Subname : ".$this->family."<br>";
        echo "Your Age : ".$this->age."<hr>";
    }
}
$Mohsen=new human();
$Mohsen->set('Mohsen','19','Rajabi');
$Mohsen->get();
$Ali=new human();
$Ali->set('Ali','20','Astaneh');
$Ali->get();
?>
```

خروجی

Your Name : Mohsen
Your Subname : Rajabi
Your Age : 19

Your Name : Ali
Your Subname : Astaneh
Your Age : 20

در مثال فوق کلاسی به اسم human ساخته شده است که درون آن سه فیلد خصوصی تعریف شده است . ابتدا یک متد برای دریافت ورودی از کاربر ایجاد شده است با استفاده از $this و علامت -> (کاراکتر – یا تفریق و سپس علامت بزرگتر >) به فیلد ها و متد های موجود در کلاس می توان دسترسی پیدا کرد و با استفاده از $this فیلد های داخل کلاس را پر کرده است و متد بعدی برای چاپ خروجی است و خروجی را چاپ می کند .در آخر که نوشتن کلاس تمام می شود با استفاده از کلمه کلیدی new می

توان از کلاس موجود شئ ساخت دو شئ از کلاس ساخته می شود و مقادیر لازم برای متد اولی به شئ داده می شود سپس متد دومی کلاس فراخوانی می شود .

متد سازنده

در PHP متد سازنده یک کلاس با نام __construct مشخص می شود این متد یک استثنا است که در همه کلاس ها وجود دارد. اغلب برنامه نویسان از این متد برای عملیاتی همانند مقدار دهی اولیه خاصیت های شئ استفاده کنند . هر گاه بخواهیم یک شئ جدید از کلاس ایجاد کنیم , این متد بطور خودکار فراخوانی می شود . بنا بر این اگر می خواهیم کار هایی را در زمان آماده سازی و ایجاد شئ انجام دهیم , دستورات مربوطه را در این متد قرار می دهیم که به آن سازنده (constructor) می گوییم . یک متد سازنده می تواند آرگومنت هائی را نیز دریافت کرده و بر اساس آن عملیات مقدار دهی اولیه را انجام دهد . برای ارسال آرگومنت به متد سازنده باید روش زیر را به کار برد :

```
$ds=new adder(آرگومنت اول,آرگومنت دوم);
```

در تصویر فوق یک شئ از کلاس adder می سازد که درون کلاس متد سازنده ایی با دو ورودی وجود دارد که ورودی های آن به این گونه ارسال شده است .

نکته : متد سازنده یک شئ نمی تواند مقداری را بر گرداند .

متد تخریب کننده

عملگرد توابع تخریب کننده بر عکس توابع سازنده است و به هنگام از بین رفتن اشیاء فراخوانی می شوند . البته به دلیل اینکه PHP خودش تمامی منابع را بعد از استفاده رها می نماید , توابع تخریب کننده از اهمیت کمتری نسبت به توابع سازنده بر خوردار است . توابع تخریب کننده در شرایط زیر فراخوانی می شود :

- هنگامی که تمامی رجوع ها به یک شئ به اتمام برسد .
- هنگامی که اجرا اسکریپت به اتمام برسد .

برای ایجاد یک تخریب کننده باید متدی را به نام __destruct() را در کلاس تعریف کنید .

به مثال destruct توجه کنید .

```php
<?php
class human{
private $name;
private $age;
private $family;
    function __destruct()
    {
        echo "Class in being destroyed\n";
    }
    function set($nam,$sen,$subnam)
    {
        $this->name=$nam;
        $this->family=$subnam;
        $this->age=$sen;
    }
    function get()
    {
        echo "Your Name : ".$this->name."<br>";
        echo "Your Subname : ".$this->family."<br>";
        echo "Your Age : ".$this->age."<hr>";
    }
}
$mohsen=new human();
$mohsen=NULL;
?>
```

خروجی

Class in being destroyed

روش های دستیابی

یکی از مزایا برنامه نویسی شئ گرا استفاده از امکان کپسوله سازی است . برنامه نویسان با کمک این قابلیت می توانند دسترسی به خاصیت ها و متد های اشیاء را محدود نمایند . بیشتر زبان های برنامه نویسی شئ گرا مانند PHP , java , C++ سه روش دستیابی public , private , protected را برای محدود کردن دسترسی به خاصیت ها و متد ها موجود در کلاس مورد استفاده قرار می دهند . به هنگام تعریف هر یک از متغییر ها و متد های موجود در کلاس برای محدود کردن دسترسی به آن ها باید یکی از سه روش دستیابی public , private , protected را به کار ببریم . در صورتی که برای متغییر یا متد ها روش دستیابی در نظر گرفته نشود , روش دستیابی آن publice در نظر گرفته می شود .

Private (خصوصی) : فیلد ها و متد هایی که به صورت خصوصی تعریف می شوند , اجازه دسترسی از بیرون کلاس را ندارند . البته هر متدی درون کلاس می تواند به آن ها دسترسی داشته باشد و در واقع فقط از بیرون کلاس (از طریق اشیاء ایجاد شده از کلاس) قابل دسترس نیستند .

به مثال private توجه کنید .

```php
<?php
class human{
private $name;
private function s(){
    echo 'welcome';
}
}
$mohsen=new human();
$mohsen->s();
?>
```

خروجی

Fatal error: Call to private method human::s() from context '' in C:\wamp\www\15-2.php on line 18

Call Stack

#	Time	Memory	Function	Location
1	0.0002	249336	{main}( )	..\15-2.php:0

در مثال فوق همان گونه که مشاهده می کنید چون متد s() از نوع خصوصی است و در بیرون کلاس قابل استفاده نیست بخاطر این است که این مثال در زمان اجرا error داده است .

Public (عمومی)

هر عنصری از کلاس (اعم از فیلد یا متد) که به صورت عمومی تعریف می شود , به راحتی توسط اشیاء ایجاد شده از کلاس (بیرون کلاس) قابل دسترسی است .

Protected (محافظ شده)

اگر یک عنصر به صورت محافظت شده تعریف شده باشد , فقط می توانید از آن در خود کلاس استفاده کنید و از طریق اشیاء (بیرون کلاس) قابل دسترس نیست . تفاوت محافظت شده یا عمومی این است در

کلاس های فرعی که از کلاس اصلی مشتق می شوند نیز توانایی دسترسی به عناصر محاظت شده آن را خواهد داشت .

شاید این سوال پیش بیاید که چرا باید یک عنصر کلاس را به صورت خصوصی تعریف کنیم؟؟؟ جواب این سوال ساده است برای امنیت این کار انجام می شود , اگر عنصر مربوطه خصوصی نباشد , امکان تغییر آن به هر مقداری از طریق اشیاء ایجاد شده از کلاس وجود دارد . برای مثال فرض کنید کلاسی نوشته می شود که فهرست برندگان قرعه کشی را از پایگاه داده استخراج کرده و به کاربر نشان می دهد طبیعتا اگر این فهرست به صورت عمومی تعریف شده باشد هر شخصی می تواند یک شئ از کلاس ایجاد کند و اسم خود را به آرایه موجود در آن لیست اضافه کند در حالی که اگر این فیلد خصوصی باشد چنین امکانی وجود ندارد و می توان یک متد عمومی ایجاد کرد تا به فیلدخصوصی دسترسی داشته باشد و آن را چاپ کند .

فیلد ها و متد های ایستا (static)

کلمه کلیدی static در برنامه نویسی شئ گرا اهمیت بسیار زیادی دارد . فیلد ها و متد های ایستا نقش اساسی درالگوی طراحی برنامه ایفا می کنند , برنامه نویس می تواند با استفاده از فیلد ها و متد های ایستاتیک بصورت مستقیم و بدون ایجاد هر گونه شئ از کلاس مربوطه , دسترسی پیدا کند . عناصر ایستا مشابه یک عضو سراسری برای کلاس عمل می کنند و تمام اشیاء ایجاد شده از آن کلاس می توانند به آن ها دسترسی پیدا کنند . به علاوه فیلد های ایستا همیشه آخرین وضعیت (مقدار) خود را حفظ می کنند . در مثال فوق ابتدا کلاس human ایجاد شده و درون آن یک فیلد و متد ایستا ایجاد شده است . در داخل متد های کلاس برای دسترسی به خاصیت ایستا باید از کلمه self:: استفاده کرد. و همچنین برای استفاده از متد های ایستا درون کلاس از این کلمه کلیدی استفاده می شود اما در بیرون کلاس برای دسترسی به خاصیت های ایستا باید اینگونه عمل کرد : ::نام کلاس .

به مثال staticfunction توجه کنید .

```php
<?php
class human{
static public $name;
static public function s(){
    echo 'Welcome'."<br>";
    self::$name='mohsen';
    echo self::$name."<br>";
}
}
human::s();
echo human::$name;
?>
```

خروجی

Welcome
mohsen
mohsen

Self:: و parent::

PHP از دو نام کلاس رزرو شده به نام self:: و parent:: استفاده می نماید . self:: برای دسترسی به خاصیت ها و متد های استاتیک و همچنین ثابت ها برنامه به کار می رود .parent:: برای دسترسی به اعضا کلاس والد استفاده می شود . برای ایجاد یک کلاس مشتق شده از کلمه کلیدی extends استفاده می شود در ادامه به بررسی کامل آن می پردازیم . به مثال self&parent توجه کنید .

```php
<?php
class human{
static $name;
    static public function s(){
        self::$name='mohsen';
    }
}
class humantwo extends human{
const i=100;
    function sd(){
        parent::s();
        echo parent::$name."<br>";
        echo self::i;
    }
}
$a=new humantwo();
$a->sd();
?>
```

خروجی

```
mohsen
100
```

فیلد های ایستا مقدار مشترکی در تمامی اشیاء دارند و می توان از آن ها برای اشتراک گذاری مقادیر بین اشیاء یک کلاس استفاده نمود . عناصر ایستا برنامه نویسی شئ گرا را بسیار شبیه برنامه نویسی سنتی و رویه گرا می کنند , بدن ساخت اشیاء می توان مستقیما هر تابعی را صدا زده و به هر فیلدی دسترسی پیدا نمود . به همین علت باید از عناصر ایستا با دقت استفاده نمود .

ثابت ها در کلاس

برای تعریف ثابت در داخل کلاس باید از کلمه کلیدی const استفاده کرد . ثابت های تعریف شده درون کلاس با ثابت های عمومی تفاوت دارند و از ثابت های تعریف شده درون کلاس فقط می توان در داخل کلاس یا از طریق اشیاء ایجاد شده از کلاس استفاده نمود . ثابت ها در کلاس با استفاده از کلمه کلیدی self:: قابل دسترس هستند .

به مثال const توجه کنید .

```php
<?php
class human{
const i=10;
    public function s(){
        for($r=0;$r<=self::i;$r++)
        {
            echo $r."\t";
        }
    }
}
$a=new human();
$a->s();
?>
```

خروجی

```
0 1 2 3 4 5 6 7 8 9 10
```

وراثت

یکی از بهترین ویژگی های برنامه نویسی شئ گرایی وراثت است که می توانید یک کلاس را توسعه داده و کلاسی کاملا جدید بسازید . کلاس مشتق شده (فرزند) می تواند تمامی قابلیت های کلاس پایه (والد) را داشته باشد یا در صورت نیاز , برخی از آن ها را بازنویسی نماید . همچنین می تواند قابلیت های کاملا جدیدی را معرفی کند که در کلاس پایه وجود ندارد . برای ایجاد یک کلاس مشتق شده یا فرزند باید از کلمه کلیدی extends استفاده کنید . به مثال extends توجه کنید . در این مثال ابتدا یک کلاس به نام person تعریف شده است که دو فیلد سراسری دارد و یک متد که مقادیر آن فیلد ها رو چاپ می کند و سپس کلاس دیگری به نام student تعریف شده است که از کلاس اولی مشتق شده است که دارای یک فیلد سراسری است و یک متد که مقدار آن فیلد را چاپ می کند . در آخر یک شئ از کلاس student ساخته شده است و مقدار فیلد های سراسری را مقدار دهی می کند ومتد های آن را اجرا می کند .

```php
<?php
class person{
public $name;
public $family;
    function printt()
    {
        echo "Name : ".$this->name."<br>";
        echo "Family : ".$this->family."<br>";
    }
}
class student extends person{
public $studentid;
    function get()
    {
        echo "ID : ".$this->studentid."<br>";
    }
}
$a=new student();
$a->name="ali";
$a->family="rezaei";
$a->studentid="156";
$a->get();
$a->printt();
?>
```

خروجی

ID : 156
Name : ali
Family : rezaei

به طور کلی یک کلاس تنها یک کلاس پایه یا والد می تواند داشته باشد ولی یک کلاس والد می تواند چند کلاس فرزند داشته باشد . اگر بخواهیم تا از کلاس خاصی هیچ کلاس فرزندی ایجاد نشود باید از کلمه کلیدی final استفاده شود به مثال final توجه کنید .

```php
<?php
final class baseclass{
public function test ()
{
    echo "base class test";
}
}
class s extends baseclass{
}
$a=new s();
?>
```

خروجی

Fatal error: Class s may not inherit from final class (baseclass) in C:\wamp\www\15-7.php on line 18

بازنویسی متد ها (چند ریختی)

در صورتی که یک متد هم در کلاس والد و هم در کلاس فرزند تعریف شده باشد آنگاه اشیائی که از کلاس فرزند ایجاد می شوند , از متد تعریف شده در کلاس فرزند استفاده خواهند نمود همچنین اشیائی که از کلاس والد این متد را صدا می زنند از متد داخل کلاس والد استفاده خواهند نمود . به این مسئله چند ریختی می گویند یعنی فرخوانی یک متد بر روی اشیاء مختلف , اجرا های متفاوتی را در پی خواهند داشت . به مثال chandrikhti توجه کنید .

```php
<?php
class person{
    public function test($name){
        echo "My name is $name"."<br>";
    }
}
class two extends person{
    public function test($age){
        echo "Age : $age"."\n";
    }
}
$two=new person();
$two->test("Mohsen");
$one=new two();
$one->test("19");
?>
```

خروجی

My name is Mohsen
Age : 19

جلوگیری از بازنویسی متد ها

شاید گاهی اوقات در برنامه لازم باشد یک متد به نحوی نوشته شود که امکا بازنویسی آن در کلاس مشتق شده وجود نداشته باشد به بیان دیگر تمامی کلاسهای مشتق شده از کلاس پایه نیز آن را به همان شکل مورد استفاده قرار دهند . برای اینکار همانند کلاس ها باید از کلمه کلیدی final استفاده کرد به مثال finalfunction توجه کنید .

```php
<?php
class person{
    final public function test($name){
        echo "My name is $name"."<br>";
    }
}
class two extends person{
    public function test($age){
        echo "Age : $age"."\n";
    }
}
$two=new person();
$two->test("Mohsen");
$one=new two();
$one->test("19");
?>
```

خروجی

این مثال همان مثال قبلی است با این تفاوت که از کلمه کلیدی final در متد کلاس والد استفاده شده است بخاطر همین این برنامه خطا می دهد .

در داخل کلاس مشتق شده می توان به متد های کلاس والد دسترسی پیدا به همین منظور باید از دستور parent:: استفاده کرد . به مثال parentfunction توجه کنید .

```php
<?php
class person{
    function display(){
        echo "Simpeclass"."<br>";
    }
}
class human extends person{
    function display(){
        parent::display();
        echo "Extends Class"."<br>";
    }
}
$a=new human();
$a->display();
?>
```

خروجی

Simpeclass
Extends Class

تابع instanceof

در پروژه های بزرگ توجه به این نکته که آیا شئ مورد نظر از کلاس مد نظر ما ساخته شده است یا خیر. برای این کار باید از تابع instanceof استفاده کرد این تابع همچنین برای کلاس هایی که از یک کلاس دیگر ارث بری کرده اند نیز استفاده می شود . اگر شئ از کلاس مورد نظر ایجاد شده باشد مقدار true را بر میگرداند در غیر این صورت مقدار false را بر میگرداند . به مثال instanceof توجه کنید . در این مثال در دو شرط مقدار تابع برابر true می شود .

```php
<?php
class person{
    function display(){
        echo "class persona"."<br>";
    }
}
class human extends person{
    function display(){
        echo "class extends"."<br>";
    }
}
$a=new human();
if($a instanceof person)
{
    echo 'a instance of person'."<br>";
}
if($a instanceof human)
{
    echo 'a instance of human';
}
?>
```

خروجی

a instance of person
a instance of human

توابع سازنده و کلاس های فرزند

اگر یک کلاس فرزند دارای متد سازنده نباشد , به هنگام ایجاد شئ از کلاس فرزند متد سازنده کلاس والد به طور خودکار فراخوانی خواهد شد . اگر کلاس فرزند خود دارای یک متد سازنده باشد دیگر متد سازنده والد به طور خودکار فراخوانی نخواهد شد . به مثال constructparent توجه کنید .

```php
<?php
class test {
    function __construct ()
    {
        echo 'welcome';
    }
}
class exten extends test{

}
$a=new exten();
?>
```

خروجی

welcome

رابط ها (Interface)

یک رابط (واسط) متد هایی را مشخص می کند که یک کلاس باید آن ها را پیاده سازی کند . البته رابط ها تنها مشخصات متد مورد نظر را بیان می کنند و درباره ی کد های داخل متد هیچ توضیحی نمی دهد . واسط ها با کلمه کلیدی interface و همانند کلاس ها ایجاد می شوند ولی در واسط ها کد مربوط به هیچ متدی پیاده سازی نمی شود . حال اگر کلاسی بخواهد این رابط را پیاده سازی کند , باید تمامی متد های مشخص شده در آن را با همان ساختار , بازنویسی نماید . برای اینکه مشخص کنیم یک کلاس , قرار است یک رابط را پیاده سازی نماید باید از کلمه کلیدی implements استفاده شود این کلمه باید بعد از اسم کلاس و سپس نام رابط استفاده شود . به مثال interface توجه کنید . در این مثال اگر نام هر یک از متد های داخل کلاس و یا ساختار آن ها تغییر یابد برنامه با خطا مواجه خواهد شد .

```php
<?php
interface test{
    public function input($a,$b);
    public function display();
}
class zarb implements test
{
    private $one;
    private $two;
    private $sum;
    public function input($a,$b)
    {
        $this->one=$a;
        $this->two=$b;
    }
    public function display()
    {
        $this->sum=$this->one+$this->two;
        echo 'Sum '.$this->one.' And '.$this->two.' : '.$this->sum;
    }
}
$a=new zarb();
$a->input('46','54');
$a->display();
?>
```

خروجی

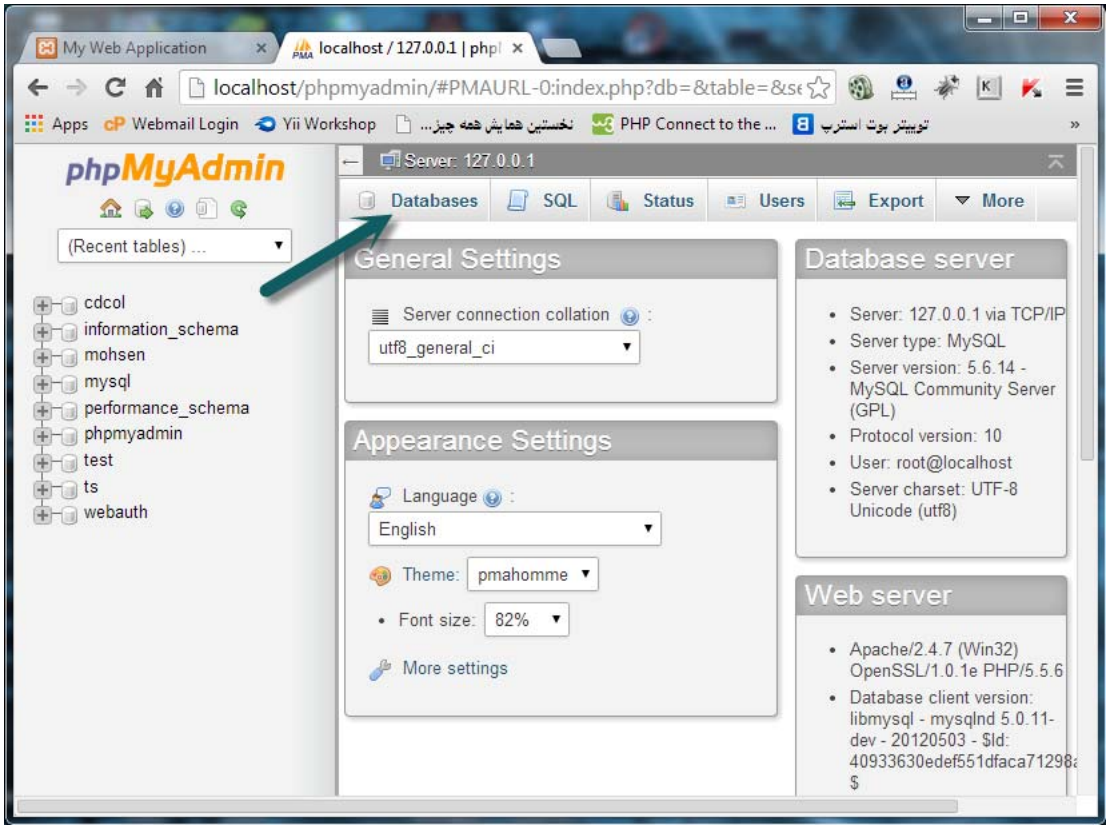
```
Sum 46 And 54 : 100
```

تمامی متدهایی که در یک واسط تعریف می شوند باید به صورت publice باشند . کلاس ها ممکن است بیش از یک واسط داشته باشند در این صورت نام واسط ها را با کاراکتر کاما (,) از همدیگر جدا میکنیم.

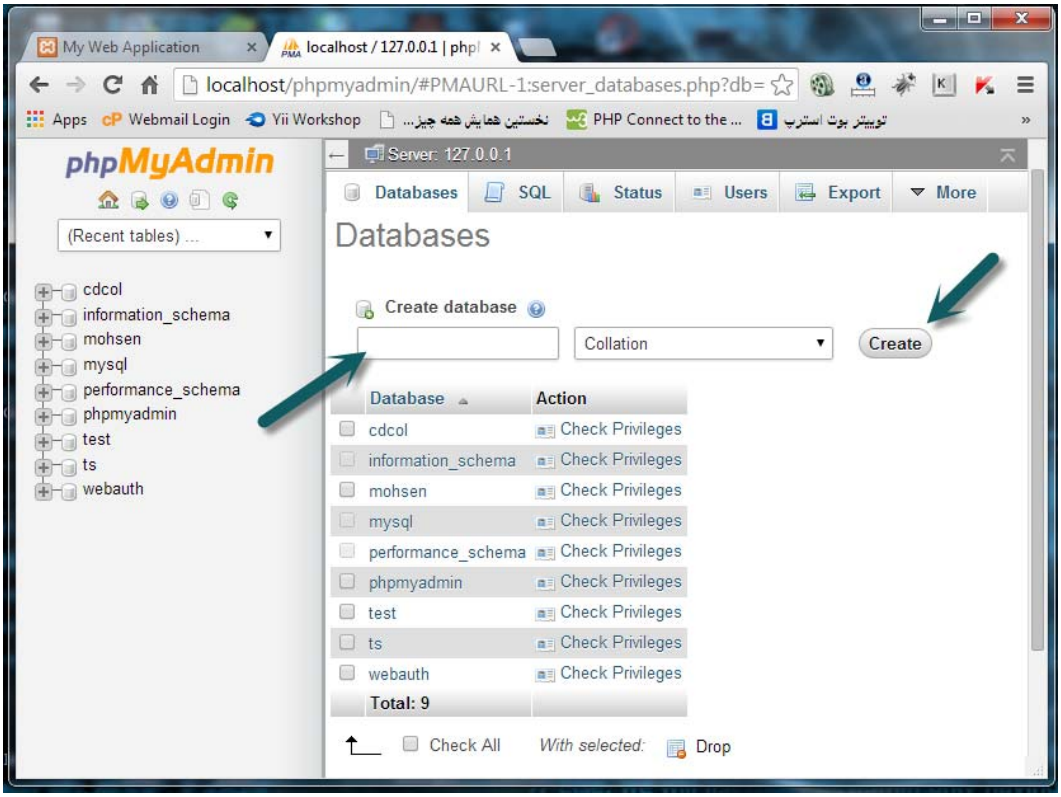
فصل دوازدهم

ارتباط با پایگاه داده

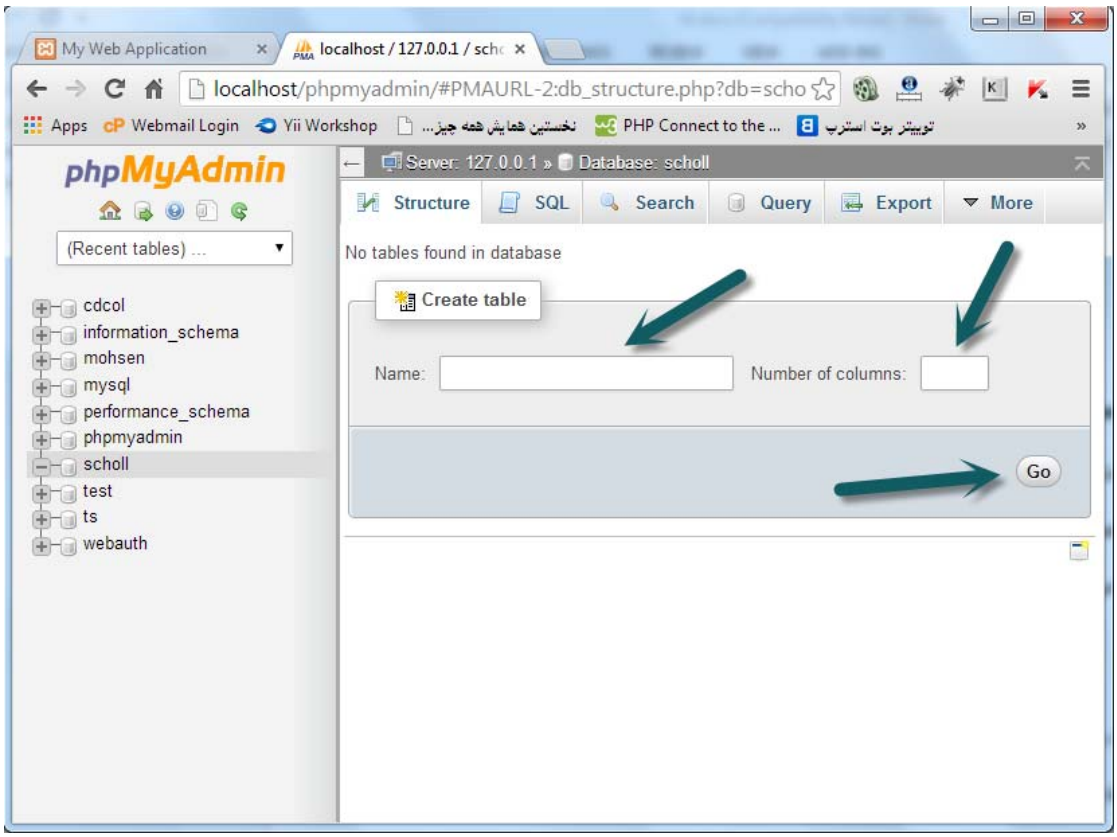
برای اتصال و کار با پایگاه داده Mysql از زبان و دستورات PHP استفاده می شود . از پایگاه داده برای ذخیره اطلاعات استفاده می شود . قبل از اینکه بتوانید به اطلاعات یک پایگاه داده دسترسی داشته باشید و آن ها را ویرایش نمایید . باید یک اتصال با ارتباط یا دیتابیس برقرار شود . برای این کار توابعی مختلفی وجود دارد همه این روش ها از نظر قدرت و امکانات همانند هم هستند . ابتدا phpmyadmin را باز می نماییم . و طبق تصویر گزینه databases را انتخاب می کنیم .



بعد از انتخاب گزینه بالا صفحه پایین باز می شود که در جای خالی اول نام پایگاه داده مورده نظر را می نویسیم و روی دکمه create کلیک می کنیم .



بعد در سمت راست نام پایگاه داده را انتخاب می کنیم و صفحه پایین باز می شود که نام جدول را در جای خالی اول و تعداد فیلد ها را در جدول دوم می نویسیم و روی دکمه Go کلیک می کنیم و نوع فیلد ها را تعریف می کنیم .



برای اتصال و کار با پایگاه داده Mysql از زبان و دستورات PHP استفاده می شود . از پایگاه داده برای ذخیره اطلاعات استفاده می شود . قبل از اینکه بتوانید به اطلاعات یک پایگاه داده دسترسی داشته باشید و آن ها را ویرایش نمایید . باید یک اتصال با ارتباط یا دیتابیس برقرار شود . برای این کار توابعی مختلفی وجود دارد همه این روش ها از نظر قدرت و امکانات همانند هم هستند . برای ایجاد ارتباط با پایگاه داده از کلاس mysqli استفاده می کنیم . این تابع 4 ورودی از کاربر می گیرد . که در آن ها را در زیر مشاهده می کنید :

- پارامتر اول : نام سرور mysql ی است که دیتابیس شما روی آن قرار دارد و شما میخواهید به آن متصل شوید .
- پامتر دوم : نام کاربری است که در سرور پایگاه داده تعریف شده .
- پارامتر سوم : کلمه عبور مرتبط با نام کاربری است .
- پارامتر چهارم : نام پایگاه داده مورد نظر ما از بین همه پایگاه داده های موجود در سرور است .

خروجی این تابع یک connection به دیتابیس مورد نظر است . وقتی که ارتباط برقرار شده باشد می تواند با استفاده از متد query از کلاس mysqli عملیات خود را در پایگاه داده انجام داد .

دستور Insert into

این دستور برای درج رکورد در جدول مورد نظر ما به کار می رود . شکل کلی این دستور به صورت زیر می باشد .

```
"Insert into نام جدول
('نام فیلد در جدول','نام فیلد در جدول')
values
('مقدار درجی','مقدار درجی')
"
```

به مثال check توجه کنید .

فایل check.php

```
<?php
if(!empty($_POST['name']) && !empty($_POST['family'])){
    $name=$_POST['name'];
    $family=$_POST['family'];
    $coo=@mysqli_connect("localhost","root","","school");
    if(!mysqli_connect_error()){
        mysqli_query($coo,"insert into user (name,family) values ('".$name."','".$family."')");
        mysqli_close($coo);
        header("location:check.html");
    }
    else
    {
        echo 'error connect';
    }
}
else
{
    header("location:check.html");
}
?>
```

فایل check.html

```
<form action="check.php" method="post">
Name :
<input type="text" name="name" />
Family :
<input type="text" name="family" />
<br />
<input type="submit" value="Register" />
</form>
```

با اجرا مثال فوق ابتدا یک فرم برای گرفتن ورودی از کاربر ایجاد شده است که با متد post ورودی ها رو به check.php ارسال می کند . در فایل check.php در خط اول ابتدا بررسی شده است که آیا اطلاعات به این صفحه ارسال شده است یا خیر اگر ارسال نشده باشد با تابع header در زبان PHP مرورگر دوباره به صفحه فرم بر میگردد اگر اطلاعات ارسال شده باشد ابتدا اطلاعات دریافتی را در متغییری ذخیره شده است و در خط چهارم از تابع mysqli برای اتصال استفاده شده است که سرور ما localhost و یوزر ما root بدون پسورد است و اسم پایگاه داده ما school می باشد که در کنار تابع mysqli از کاراکتر @ استفاده شده است که به این معنی است که اگر این تابع نتواند به پایگاه داده مورده نظر متصل شود هیچ error برای کاربر چاپ نمی شود و ما خودمان می خواهیم این خطا را مدیریت کنیم . در شرط بعدی ابتدا بررسی شده است که آیا اتصال درست و کامل انجام شده است و خطایی ندارد اگر خطایی داشت پیغام error connect برای کاربر نمایش داده می شود اگر خطایی نداشت با استفاده از mysqli_query کوئری مورد نظرمان را می نویسیم که کوئری درج در پایگاه داده می باشد و در آخر با استفاده از تابع mysqli_clise ارتباط را می بندیم و مرور گر را به صفحه فرم می فرستیم .

خواندن از پایگاه داده

در مثال قبل نحوه درج در پایگاه داه را آموختید . هر گاه در پایگاه داده نیاز باشد تا اطلاعات خاصی را از یک یا چند جدول استخراج کرده و آن ها را در خروجی نمایش دهید از دستور select استفاده می شود . به مثال select.php توجه کنید .

```php
<?php
$coo=@mysqli_connect("localhost","root","","school");
if(!mysqli_connect_error()){
    $query="select * from user";
    $table=mysqli_query($coo,$query);
    echo "<table border='1'>
          <tr>
          <th>ID</th>
          <th>Name</th>
          <th>Family</th>
          </tr>";
    while($rows=mysqli_fetch_assoc($table)) {
        echo "<tr>";
        echo "<th>".$rows['id']."</th>";
        echo "<th>".$rows['name']."</th>";
        echo "<th>".$rows['family']."</th>";
        echo "</tr>";
    }
    echo "</table>";
    mysqli_close($coo);
}
else
{
    echo 'error connect';
}
?>
```

خروجی

ID	Name	Family
5	Ali	Rezaei
6	Mohsen	Rajabi

در این مثال ابتدا با استفاده از تابع mysqli به پایگاه داده مورده نظر متصل شده ایم و همانند مثال قبل مدیریت خطا را با استفاده از دستور @ به دست گرفته ایم . ابتدا با دستر شرطی بررسی می کنیم که آیا ارتباط خطا دارد یا خیر اگر دارد پیغامی برای کاربر چاپ می کنیم در غیر این صورت کوئری مد نظر خود را می نویسیم که دستور select می باشد و با استفاده از متد query این تابع مقدار بازگشتی را در متغییری ذخیر ه می کنیم . با دستور echo دستور ساخت جدول را می نویسیم . برای چاپ خروجی روش های متداولی وجود دارد در اینجا با استفاده از حلقه و تابع mysqli_fetch_assoc اینکار انجام شده است.

این تابع یک ورودی می گیرد که همان مقدار بازگشتی تابع mysqli_query می باشد و خروجی این تابع یک آرایه می باشد . با استفاده از حلقه این آرایه پیمایش می شود تا به آخر برسد و مقدار false را تولید کن و از حلقه خارج شود بنا براین کل آرایه چاپ می شود و در آخر ارتباط را با پایگاه داده را قطع می کنیم .

دستور where

اما برای اینکه ما بتوانیم در کوئری های خود از عبارات شرطی هم استفاده کنیم از عبارت where استفاده می کنیم برای مثال ما میخواهم در پایگاه داده یوزر ها شخص خاصی را جستجو کنیم . به مثال 16 توجه کنید .

فایل 16.html

```
<form action="16.php" method="post">
Name :
<input type="text" name="name" />
Family :
<input type="text" name="family" />
<br />
<input type="submit" value="Search" />
</form>
```

فایل 16.php

```
<?php
if( !empty($_POST['name']) && !empty($_POST['family']) ){
    $name=$_POST['name'];
    $family=$_POST['family'];
    $coo=@mysqli_connect("localhost","root","","school");
    if(!mysqli_connect_error()){
        $rows=mysqli_query($coo,"select * from user where name='".$name."' and family='".$family."'");
        if(mysqli_num_rows($rows)>=1){
            echo 'Yes Found';
        }
        else
        {
            echo 'Not Found';
        }
        mysqli_close($coo);
    }
    else
    {
        echo 'error connect';
    }
}
else
{
    header("location:16.html");
}
?>
```

در این مثال نام و فامیلی برای جستجو در پایگاه داده از کاربر گرفته می شود و به صفحه 16.php ارسال می شود در این صفحه ابتدا چک میکنیم که آیا پارامتر نام و فامیلی به این صفحه ارسال شده است یا خیر اگر ارسال نشده باشد یعنی این آدرس دستی وارد شده است برای همین مرورگر را به صفحه وارد کردن فرم میفرستم اگر پارامتر ها ارسال شده باشد با استفاده از تابع mysqli به پایگاه داده مورد نظر متصل می شویم و از کاراکتر @ برای جلو گیری ایجاد خطا استفاده می کنیم و خودمان مدیریت خطا را به عهده می گیریم . دستور شرطی مینویسم برای چک کردن درست بودن اتصال به پایگاه داده می نویسم اگر ارتباط برقرار بود کوئری مد نظر را می نویسم و با استفاده از دستور where شرط مورد نظرما را می نویسیم این شرط بررسی می کند که نام و فامیلی دریافتی در جدول user وجود دارد یا خیر . تابع mysqli_num_rows تعداد رکورد برگشتی از کوئری مد نظرمان را بر میگیرداند این دستور یک ورودی می گیرد که همان خروجی دستور msqli_query می باشد . بنا براین اگه دستور mysqli_num_rows برابر یک باشد یعنی یک رکورد در پایگاه داده موجود است که این به این معنی است که جستجو مورد نظر در پایگاه داده موجود است بخاطر همین پیغام Yes Found برای کاربر نمایش داده می شود و در غیر این صورت پیغام Not Found به کاربر نمایش دادهد می شود .

دستور order by

دستور select اطلاعات تمامی فیلد های تعیین شده برای آن را بر اساس فیلد اول مرتب کرده و نمایش می دهد . امام ممکن است شما بخواهید اطلاعات را بر اساس مقدار یک فیلد دیگر غیر از فیلد اول بصورت همزمان مرتب نمایید برای این منظور بایستی پس از دستور select نام فیلد یا دو فیلدی که میخواهید اطلاعات خروجی بر اساس آن مرتب شوند را به ترتیب در مقابل عبارت order by تعیین کنید.

نکته : این دستور به صورت پیش فرض اطلاعات را به صورت صعودی (کوچک به بزرگ) مرتب می کند برای اینکه از نزولی (بزرگ به کوچک) مرتب کند پس از تایپ فیلد ها از عبارت DESC استفاده می کنیم. به مثال orderby توجه کنید . در این مثال با استفاده از این دستور مقدار خروجی بر اساس نام به صورت نزولی مرتب شده است .

```php
<?php
$coo=@mysqli_connect("localhost","root","","school");
if(!mysqli_connect_error()){
    $query="select * from user order by name";
    $table=mysqli_query($coo,$query);
    echo "<table border='3'>
          <tr>
          <th>ID</th>
          <th>Name</th>
          <th>Family</th>
          </tr>";
    while($rows=mysqli_fetch_assoc($table)) {
        echo "<tr>";
        echo "<th>".$rows['id']."</th>";
        echo "<th>".$rows['name']."</th>";
        echo "<th>".$rows['family']."</th>";
        echo "</tr>";
    }
    echo "</table>";
    mysqli_close($coo);
}
else
{
    echo 'error connect';
}
?>
```

خروجی

ID	Name	Family
5	Ali	Rezaei
7	ali	moradi
10	ali	rajaii
8	amir	aramiyan
6	Mohsen	Rajabi
9	sina	astaneh

به مثال orderby2 توجه کنید . در این مثال id به صورت نزولی در خروجی چاپ شده است .

```php
<?php
$coo=@mysqli_connect("localhost","root","","school");
if(!mysqli_connect_error()){
    $query="select * from user order by ID DESC";
    $table=mysqli_query($coo,$query);
    echo "<table border='3'>
          <tr>
          <th>ID</th>
          <th>Name</th>
          <th>Family</th>
          </tr>";
    while($rows=mysqli_fetch_assoc($table)) {
        echo "<tr>";
        echo "<th>".$rows['id']."</th>";
        echo "<th>".$rows['name']."</th>";
        echo "<th>".$rows['family']."</th>";
        echo "</tr>";
    }
    echo "</table>";
    mysqli_close($coo);
}
else
{
    echo 'error connect';
}
?>
```

خروجی

ID	Name	Family
10	ali	rajaii
9	sina	astaneh
8	amir	aramiyan
7	ali	moradi
6	Mohsen	Rajabi
5	Ali	Rezaei

دستور Update

این دستور برای تغییر مجود در یک فیلد با مقدار جدید است . شکل کلی این دستور به شکل زیر است به تصویر فوق دقت کنید .

```
Update  نام جدول
Set  نام فیلد = مقدار جدید
Where  نام فیلد = مقدار قبلی
```

در این تصویر اگر خواستیم مقدار چند فیلد را تغییر دهیم در جلو دستور set از کاما استفاده می کنیم.

به مثال 3-16 تووجه کنید . مقدار جدول ما به صورت زیر می باشد .

id	name	family
5	Ali	Rezaei
6	Mohsen	Rajabi
7	ali	moradi
8	amir	aramiyan
9	sina	astaneh
10	ali	rajaii

حال به مثال update توجه کنید .

```php
<?php
$co=@mysqli_connect("localhost","root","","school");
if(!mysqli_connect_error()){
    $q="Update user SET name='Mohsen' , family='Ahmadi' Where id='5'";
    mysqli_query($co,$q);
    mysqli_close($co);
}
else
{
    echo 'error connect';
}
?>
```

با اجرا مثال بالا جدول ما تغییر می کند به تصویر فوق توجه کنید تا تغییر را مشاهده کنید .

id	name	family
5	Mohsen	Ahmadi
6	Mohsen	Rajabi
7	ali	moradi
8	amir	aramiyan
9	sina	astaneh
10	ali	rajaii

دستور Delete

از این دستور برای حذف اطلاعات یک رکورد استفاده می شود . شکل کلی این دستور به صورت زیر است.

```
Delete from نام جدول where شرط
```

به مثال delete توجه کنید در این مثال از پایگاه داده مورده نظر ما رکوردی که id آن برار 5 باشد از جدول حذف می شود .

```php
<?php
$co=@mysqli_connect("localhost","root","","school");
if(!mysqli_connect_error()){
    $q="delete from user where id='5'";
    mysqli_query($co,$q);
    mysqli_close($co);
}
else
{
    echo 'error connect';
}
?>
```

از این دستور برای حذف کل اطلاعات هم استفاده می شود برای حذف کل اطلاعات به مثال delete2 توجه کنید . در این مثال تمام رکورد های جدول user پاک می شود . برای اینکار دو روش وجود دارد .

```
Delete From نام جدول
یا
Delete * From نام جدول
```

به مثال delete2 توجه کنید .

```php
<?php
$co=@mysqli_connect("localhost","root","","school");
if(!mysqli_connect_error()){
    $q="delete from user";
    mysqli_query($co,$q);
    mysqli_close($co);
}
else
{
    echo 'error connect';
}
?>
```

فصل سیزدهم

Ajax در jQuery

Ajax هنر تبادل داده ها با سرور و بروز رسانی بخش هایی از یک صفحه وب بدون بارگذاری مجدد صفحه است .

Ajax مخفف کلمات Asynchronous JavaScript And XML است . Ajax داده ها را در پس زمینه بارگذاری می کند و آن ها را در صفحه وب بدون بارگذاری مجدد کل صفحه نمایس می دهد . متد های زیادی برای استفاده از Ajax در jQuery فراهم شده است و یکی از این روش ها POST و Getرا می توانید از سرور راه دور درخواست کنید .

متد post

متد ()$.post با استفاده از روش POST داده ای را از سرور درخواست می کند . شکل کلی این دستور به صورت زیر می باشد .

```
$.post(Url,Data,CallBack);
```

- Url : الزامی است , Url فایل در خواست شده است .
- Data : اختیاری است , می تواند شامل تعدادی متغییر و مقادیر آن ها باشد که همراه درخواست به سرور ارسال می شود که این متغییر ها با کاما (,) از هم جدا می شوند و همه آن ها با هم درون {} قرار می گیرند .
- Callback : اختیاری است , تابعی است که بعد از اجرا کامل متد ()$.post اجرا می شود .

به مثال زیر توجه کنید . در این مثال یک ورودی برای دریافت از کاربر ایجاد شده است که خاصیت onkeyup آن تابعی را اجرا می کند . id این ورودی برابر f داده شده است و سپس تگ div ایجاد شده است که id آن برابر sss قرار داده شده . ابتدا Jquery را در صفحه وارد می کنیم و سپس یک تابع جاوا می نویسیم که در این تابع از تابع $.post استفاده می کنیم آرگومنت اولی این تابع را آدرس صفحه aja.php است که در آن فقط دستور echo برای چاپ مقدار دریافتی نوشته شده است آرگومنت دومی آن ما متغییری به نام name را ایجاد کرده و مقدار آن را با کمک Id ورودی برابر مقدار ورودی قرا ر داده ایم سپس آرگومنت سوم یک تابع نوشته شده است که یک ورودی دارد که این ورودی همان مقدار

برگشتی از صفحه aja.php است و با استفاده از id تگ div ما این data برگشتی را درون آن چاپ می کنیم . به مثالaja توجه کنید .

فایل aja.html

```
<script language="javascript" src="jquery.js"></script>
<script language="javascript">
function test(){
    $.post('aja.php',{name:document.getElementById("f").value},
    function(data){
        $('#sss').html(data);
    });
};
</script>
<input type="text" onkeyup="test();" id="f"/>
<br />
<div id="sss">
</div>
```

فایل aja.php

```
<?php
if(isset($_POST['name']))
{
    echo $_POST['name'];
}
else
{
    header("location:aja.html");
}
?>
```

در فایل aja.php ابتدا چک میکنیم که آیا ورودی به این صفحه ارسال شده است یا خیر اگر ارسال نشده باشد یعنی کاربر این آدرس را دستی وارد کرده و با استفاده از تابع header() صفحه مرورگر را به صفحه ورود اطلاعات ارجاع می دهیم .

به مثال 17 توجه کنید , در این مثال همانند مثال قبل تابعی با جاوا نوشته شده است که در آن از تابع $.post() استفاده شده است و مقدار ورودی را به صفحه 17.php ارسال می کند .در این صفحه بعد از اینکه چک می شود که آیا ورودی به این صفحه ارسال شده است به پایگاه داده متصل شده و با استفاده از دستور like کوئری مد نظرمان را می نویسیم که در این کوئری هر اسمی که دارای حروفی که این صفحه دریافت می کند باشد نشان می دهد و با استفاده از حلقه آن را چاپ می کند .

فایل 17.html

```
<script language="javascript" src="jquery.js"></script>
<script language="javascript">
function send(){
    $.post('17.php',{name:document.getElementById("f").value},
    function(data){
        $('#s').html(data);
    });
};
</script>
<input type="text" onkeyup="send();" id="f"/>
<br />
<div id="s">
</div>
```

فایل 17.php

```
<?php
if(isset($_POST['name']))
{
    $name=$_POST['name'];
    $coo=@mysqli_connect("localhost","root","","school");
    if(!mysqli_connect_error()){
        $query="select * from user where name like '%{$name}%'";
        $table=mysqli_query($coo,$query);
        while($rows=mysqli_fetch_assoc($table)) {
            echo $rows['name']."<br>";
        }
        mysqli_close($coo);
    }
    else
    {
        echo 'error connect';
    }
}
else
{
    header("location:17.html");
}
?>
```

به مثال 17-1p1.php و 17-1p2.php توجه کنید . در این مثال در صفحه 17-1p1.php ابتدا به پایگاه داده متصل شده و مقدار آن را در یک combo box قرا داده است که مانند مثال قبل از id برای دسترسی به مقادیر این combo box استفاده شده است و تابع java نوشته شده است که از تابع $.post() برای دسترسی به صفحه 17-1p2.php متصل شده و مقادیر را از آن صفحه به صورت ajax به خوده این صفحه میفرستد .

فایل 17p1.php

```
    <script language="javascript" src="jquery.js"></script>
    <script language="javascript">
    function send(){
        $.post('17-1p2.php'
            ,{
                name:document.getElementById("dbl").value
            },
            function(data){
            $('#s').html(data);
            });

    };
    </script>
<?php
    $coo=@mysqli_connect("localhost","root","","school");
    if(!mysqli_connect_error()){
        $query="select * from student";
        $table=mysqli_query($coo,$query);
        echo "<select id='dbl' onChange='send();'>";
    while($rows=mysqli_fetch_assoc($table)) {
    echo "<option value='".$rows['name']."'>";
    echo $rows['name'];
    echo "</option>";
    }
    echo "</select>";
    }
    else
    {
        echo 'error connect';
    }
?>
<br>
<br><br>
<br><br>
<br><br>
<div id="s">
</div>
```

فایل 17-1p2.php

```php
<?php
if(isset($_POST['name']))
{
    $name=$_POST['name'];
    $coo=@mysqli_connect("localhost","root","","school");
    if(!mysqli_connect_error()){
        $query="select * from student where name='".$name."'";
        $table=mysqli_query($coo,$query);
        while($rows=mysqli_fetch_assoc($table)) {
            echo "ID : ".$rows['id']."<br>";
            echo "Name : ".$rows['name']."<br>";
            echo "Family : ".$rows['family']."<br>";
            echo "Moadel : ".$rows['moadel']."<br>";
            echo "Age : ".$rows['age']."<br>";
        }
        mysqli_close($coo);
    }
    else
    {
        echo 'error connect';
    }
}
else
{
    header("location:17-1p1.php");
}
?>
```

متد get

متد get داده ها را از طریق متد get از سرور درخواست می کند . شکل کلی استفاده از این دستور به صورت زیر می باشد .

```
$.get(URL,callback);
```

پارامتر های ورودی این متغییر مانن متد post می باشد و می توان مانند متد post از آن استفاده نمود.

به مثال 17get توجه کنید .

فایل 17get.html

```
<script language="javascript" src="jquery.js"></script>
<script language="javascript">
function send(){
    $.get('17get.php',{name:document.getElementById("f").value},
    function(data){
        $('#s').html(data);
    });
};
</script>
<input type="text" onkeyup="send();" id="f"/>
<br />
<div id="s">
</div>
```

فایل 17get.php

```
<?php
if(isset($_GET['name']))
{
    $name=$_GET['name'];
    echo $name;
}
else
{
    header("location:17get.html");
}
?>
```

این مثال در صفحه 17get.html ابتدا ورودی را از کاربر دریافت می کنیم و به از طریق متد get به صفحه 17get.php ارسال می کنیم و در آن صفحه آن را دریافت و چاپ می کنیم .

فصل چهاردهم

در این فصل با آموخته های فصل های قبل اقدام به نوشتن چندین مثال نموده ایم ولی این بدان معنا نیست که در این فصل به دانش شما چیزی اضافه نشود و مطالب تکرار باشد .

به مثال زیر توجه کنید . مثال زیر یک مثال ساده از شئ گرایی می باشد که یک کلاس بسیار ساده از متصل شدن از پایگاه داده نوشته شده است که با ایجاد سئ از آن به پایگاه داده متصل شده و مقادیر را چاپ می کند .

فایل classdb.php

```php
<?php
class DB{
    private $server;
    private $pass;
    private $user;
    private $dbname;
    private $conection;
    private $result;
    function connect($server,$user,$pass,$dbname)
    {
        $this->server=$server;
        $this->user=$user;
        $this->pass=$pass;
        $this->server=$dbname;
        $this->conection=@mysqli_connect($server,$user,$pass,$dbname);
    }
    function query($query)
    {
        $this->result=mysqli_query($this->conection,$query);
        return $this->result;
    }
}
?>
```

فایل objectclassdb.php

```php
<?php
include 'classdb.php';
$a=new DB();
$a->connect('localhost','root','','school');
$b=$a->query("select * from student");
while($row=mysqli_fetch_assoc($b))
{
    echo $row['name']."<br>";
}
?>
```

خروجی

mohsen
sina
amir
Mehdi

به مثال زیر توجه کنید . در این مثال یک کلاس برای فرستادن email ساخته شده است . در این کلاس 4 فیلد خصوصی تعریف شده است برای استفاده در تابع ()mail و متد هایی برای مقدار دهی به این فیلد ها نوشته شده است . متد سازنده این کلاس فیلد sender را مقدار دهی می کند که همان نام فرستنده می باشد و فیلد recipients را به صورت آرایه تعریف می کند . متد addrecipient مقادیر ارسال شده را در آرایه recipients قرا می دهد . متد setsubject موضوع ایمیل را مقدار دهی می کند و متد setbody متن اصلی و بدنه ایمیل را مقدار دهی می کند . متد sendemail با استفاده از حلقه foreach تابع ()mail هر دفعه یک ایمیل به مقادیر آرایه recipients می فرستد و نتیجه آن را در متغییر result ذخیره می کند و در آخر چک میکنیم که آیا ایمیل به درستی ارسال شده است یا خیر و پیغامی برای کاربر نمایش می دهیم .

فایل classemail.php

```php
<?php
class emailer{
    private $sender;
    private $recipients;
    private $body;
    private $subject;
    function __construct($sender){
        $this->sender=$sender;
        $this->recipients=array();
    }
    function addrecipient($recipient){
        array_push($this->recipients,$recipient);
    }
    function setsubject($subject){
        $this->subject=$subject;
    }
    function setbody($body){
        $this->body=$body;
    }
    function sendemail(){
        foreach($this->recipients as $recipient){
            $result=mail($recipient,$this->subject,$this->body,"Form : $this->sender\r\n");
            if($result)
            {
                echo "Mail Successfully send to $recipient <br>";
            }
            else
            {
                echo "Mail Not Successfully";
            }
        }
    }
}
?>
```

فایل objectclassemail.php

```php
<?php
include 'classemail.php';
$email=new emailer("sina8895@yahoo.com");
$email->addrecipient("M.kabir8895@yahoo.com");
$email->setbody("Hi i am a test email");
$email->setsubject("Test");
$email->sendemail();
?>
```

به مثال زیر توجه کنید . در این مثال ابتدا یک کلاس برای مشخصات شخصی طراحی شده است و یک کلاس هم برای دانشجو طراحی شده است که از کلاس اولی مشتق شده است و یک آیدی و وضعیت دانشجو به کلاس دانشجو اضافه شده است و یک متد برای دریافت و چاپ مشخصات دانشجو نوشته

شده است که آیدی و وضعیت دانشجو و نام و ... را میگیرد و متد کلاس والد را اجرا می کند و مشخصه اضافه شده را چاپ می کند .

فایل classperson.php

```php
<?php
class person {
    private $name;
    private $family;
    private $age;
    function setperson($name,$family,$age){
        $this->name=$name;
        $this->family=$family;
        $this->age=$age;
    }
    function get(){
        echo 'Name : '.$this->name."<br>".
        'Family : '.$this->family."<br>".
        'Age : '.$this->age."<br>";
    }
}
class student extends person{
    private $id;
    private $status;
    function setstudent($name,$family,$age,$id,$status){
        $this->id=$id;
        $this->status=$status;
        parent::setperson($name,$family,$age);
        parent::get();
        echo 'ID : '.$this->id."<br>".
        'Status : '.$this->status."<br>";
    }
}
$a=new student();
$a->setstudent("Mohsen","Rajabi","19","12256","Rozane");
?>
```

ساخت تصویر امنیتی

فرض کنید یک فرم در سایت خود برای ارتباط با ما قرار داده اید اگر شما در این فرم تصویر امنیتی قرار ندهید حال یک نفر با نوشتن برنامه ای کوچک و ساده در بازه ی زمانی خیلی کم و به دفعات فرم شما را با یک مقدار جدید به سرور ارسال می کند و بعد از مدت کم پایگاه داده شما پر میشه و هنگ میکنه و پیام های دریافتی ایمیلتون هم پر میشه برای جلو گیری از این کار از تصاویر امنیتی استفاده می کنیم . به مثال 19 توجه کنید در این مثال یک فایل index.php که صفحه اصلی است وجو دارد و

در این صفحه ابتداsession اجرا شده است . در داخل $text حروفی را می نویسیم که میخواهیم در رمز عبور باشند . متغییر cod را ابتدا خالی قرار می دهیم بعد حلقه ایی را می نویسیم که 6 بار تکرار می شود و در خط اول یک عدد تصادفی از 0 تا تعداد حروف text به ما می دهد و در خط بعد بوسیله substr مقدار شروع را انتخاب زیر رشته رو همون عدد رندوم قرار دادیم و طولش هم یک گذاشتیم و درون متغییر cod ریختیم با اجرا حلقه 6 حرف تصادفی در cod قرار می گیرد و بعد از حلقه مقدار cod را برای امنیت بیشتر در session قرار می دهیم . وسپس با استفاده از تگ img یک لینک به تصویر مون که در صفحه securitycod.php هست قرار می دهیم . و یک ورودی برای تایپ رمز برای کاربر قرار می دهیم و با استفاده از ajax این مقادیر به صفحه securitycod.php ارسال می کنیم و همراه آن یک متغییر status برای کنترل برنامه میفرستیم و به آن یک مقدار می دهیم . اما در صفحه scuritycod.php ابتدا جلسه را اجرا می کنیم و چون قرار است خروجی تصویر باشد ما هدر صفحه رو jpg قرار دادیم و مقدار cod رااز جلسه دریافت و درون متغییر cod میریزیم . و یکسری متغییر برای ویژگی های تصویر تعریف کرده ایم مثله اندازه فونت , طول عکس ,ارتفاع عکس . سپس با دستور شرطی بررسی کرده ایم که آیا متغییر status خالی است یا خیر اگر خالی نباشد به این معنا است که کاربر چیزی تایپ است و تصویر قبلا ایجاد شده است اگر خالی باشد یعنی چیزی تایپ کرده و تصویر قبلا ساخته نشده است و باید تصویر را بسازیم .ابتدا با متد imagecreate با دادن طول و عرض یک تصویر می سازد و اون را درون متغییر image قرار می دهیم و سپس با متد imagecolorallocate تصویرمان را با رنگ خاکستری پر میکنیم . سپس رنگ نوشته هم با همان متد به رنگ مشکی میگیریم و درون متغییر قرار می دهیم .

حالا با استفاده از متد imagettftext متن کد رو روی صفحه تصویرمون بنویسیم :

- پارامتر اول : متغییر تصویر رو دریافت می کند .
- پارامتر دوم : سایز فونت رو دریافت میکنه .
- پارامتر سوم : زاویه چرخش متن رو تو عکسی دریافت میکنه .
- پارامتر چهارم : فاصله x رو دریافت میکنه , فاصله نوشته از سمت چپ تصویر .
- پارامتر پنجم : فاصله y رو دریافت میکنه , فاصله نوشته از سمت بالا تصویر .
- پارامتر ششم : رنگ متن رو که تولید کرده ایم رو دریافت میکنه .
- پارامتر هفتم : یه فونت رو دریافت میکنه که با اون متن رو مینویسه که آدرس فونت باید مستقیم باشه .
- پارامتر هشتم : متنی رو که روی تصویر چاپ میشه را دریافت میکنه .

در آخر هم با استفاده از متد imagejpeg که تصور رو ورودی میگیره تصویر رو تولید می کنیم .

فایل index.php

```
<script language="javascript" src="jquery.js"></script>
<script language="javascript">
function test(){
    $.post('SecurityCode.php',{user:document.getElementById("f").value,
status:2},
    function(data){
        $('#sss').html(data);
    });
};
</script>
<?Php
session_start();
$text='QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm0123456789';
$code='';
for($i=1;$i<=6;$i++)
{
$start=rand(0,strlen($text));
$code.=substr($text,$start,1);
}
$_SESSION['code']=$code;
?>
<img src="SecurityCode.php"><br />

<input type="text" onkeyup="test();" id="f" maxlength="6" size="10px"/>
<div id="sss">
</div>
```

فایل securitycod.php

```php
<?php
session_start();
header('Content-type: image/jpeg');
$code=$_SESSION['code'];
$font_size=18;
$image_width=100;
$image_height=40;
if(empty($_POST['status'])){
    $image= imagecreate($image_width,$image_height);
    imagecolorallocate($image,200,200,200);
    $text_color= imagecolorallocate($image,0,0,0);
    imagettftext($image, $font_size,0,11,28, $text_color,'calibri.ttf', $code);
    imagejpeg($image);
}
else {
  $user=$_POST['user'];
  if($user==$code)
  {
      echo 'Successfully Cod ! ';
  }
}
?>
```

ورود اطلاعات با Ajax

ابتدا یک فایل به اسم insertp1.php برای ورود اطلاعات ایجاد می کنیم . در این صفحه ابتدا یک فرم با دو ورودی و یک دکمه ایجاد می کنیم که خاصیت onclick دکمه یک تابع جاوا را اجرا می کند . این تابع جاوا اطلاعات را به صورت Ajax برای صفحه insertp2.php ارسال می کند که متغییر های name و family را با مقادیر داخل فرم پر می کند و اطلاعات برگشتی را درون div با آیدی mes چاپ می کند . اما در صفحه insertp2.php ابتدا چک میکنیم که آیا متغییر name و family هر دو دارای مقدار می باشند یا خیر با استفاده از تابع empty این کار را انجام می دهیم که اگر این دو متغییر مقدارشان خالی باشد مقدار true ر بر میگرداند و اگر مقدار دهی شده باشند مقدار false را بر میگرداند . اگر مقداری نداشته باشند پیغامی مبنی بر اینکه اطلاعات را وارد کنید چاپ می شود و اگر هر دو متغییر پر باشند دو متغییر را ابتدا فاصله های کناری آن ها را حذف کرده و همه حروف آن ها را به حروف کوچک تبدیل می کنیم و سپس با استفاده از تابع mysqli به پایگاه داده مورد نظر متصل می شویم و در خط بعد چک می کنیم که آیا این ارتباط به درستی برقرار شده است یا خیر اگر این ارتباط خطا داده باشد پیغام خطایی برای کاربر نشان می دهیم در غیر این صورت یک کوئری برای بررسی اینکه که کاربر اطلاعات تکراری وارد کرده باشد یا خیر می نویسیم و با دستور شرطی بررسی می کنیم که آیا در این کوئری رکوردی پیدا شده است

یا خیر اگر پیدا شده باشد یعنی کاربر کلمات تکراری وارد کرده است و پیغامی مبنی بر تکراری بودن برای کاربر نمایش می دهیم در غیر این صورت کوئری برای درج این نام و فامیلی می نویسیم و در آخر ارتباط با پایگاه داده را قطع می کنیم . و در خط بعد مقدار table را بررسی می کنیم که اگر این مقدار 1 باشد یعنی با موفقیت ثبت شده است و پیغامی مبنی بر درج موفقیت اطلاعات چاپ می کنیم و در غیر این صورت یعنی اطلاعات با موفقیت ثبت نشده است و پیغامی مبنی بر درج نا موفقیت اطلاعات نمایش می دهیم .

فایل insertp1.php

```
<script language="javascript" src="jquery.js"></script>
<script language="javascript">
function send(){
    $.post('insertp2.php',{name:document.getElementById("n").value,
                           family:document.getElementById("f").value},
    function(data){
        $('#mes').html(data);
    });
};
</script>
<form>
<input type="text" name="name" id="n"/>
<input type="text" name="family" id="f"/>
<input type="button" value="Insert !" onclick="send();" />
</form>
<br />
<div id="mes">
</div>
```

فایل insertp2.php

```php
<?php
if(!empty($_POST['name']) && !empty($_POST['family'])){
    $name=strtolower(trim($_POST['name']));
    $family=strtolower(trim($_POST['family']));
    $coo=@mysqli_connect("localhost","root","","student");
    if(!mysqli_connect_error()){
        $query1="select * from user where name='".$name."' and family='".$family."'";
        $table1=mysqli_query($coo,$query1);
        if(mysqli_num_rows($table1)==0){
            $query="insert into user (name,family) values ('".$name."','".$family."')";
            $table=mysqli_query($coo,$query);
            mysqli_close($coo);
            if($table)
            {
                echo 'نام و فامیلی با موفقیت ثبت شده';
            }
            else
            {
                echo 'خطا در ثبت نام و فامیلی';
            }
        }
        else
        {
            echo 'نام و فامیلی وارد شده تکراری می باشد';
        }
    }
    else
    {
        echo 'خطا ر برقراری اتصال به پایگاه داده';
    }
}
else
{
    echo 'نام/ فامیلی را وارد کنید';
}
?>
```

خواندن اطلاعات با Ajax

ابتدا یک فایل به اسم combop1.php ایجاد می کنیم که دارای یک combo box است که کاربر می تواند دو انتخاب داشته باشد که یکی DESC است که اطلاعات را به صورت نزولی چاپ می کند و دیگر ASC است که اطلاعات را به صورت صعودی چاپ می کند و این اطلاعات به صورت Ajax از یک صفحه دیگر می خواند و درون یک ناحیه متن (textarea) چاپ می کند . در ابتدا این فایل یک تابع جاوا نوشته شده است که متغییر status را با مقدار کمبوباکس پر می کند و به فایل combop2.php ارسال می کند

و نتیجه آن را در ناحیه متن نمایش می دهد . در فایل combop2.php ابتدا بررسی می شود که آیا متغیر status از طریق متد post ارسال شده است یا خیر اگر ارسال نشده باشد دوباره صفحه مرورگر را به صفحه ابتدا میفرستیم ولی اگر ارسال شده باشد ابتدا آن را درون متغییری ذخیره می کنیم و با استفاده از mysqli به پایگاه داده مد نظر متصل می شویم و کوئری برای دریافت داده ها می نویسیم که نحوه مرتب سازی آن را از کمبوباکس دریافت می کنیم و با استفاده از حلقه مقدار سن را چاپ می کنیم و در آخر ارتباط را قطع می کینم .

فایل combop1.php

```
<script language="javascript" src="jquery.js"></script>
<script language="javascript">
function send(){
    $.post('combop2.php'
        ,{
            status:document.getElementById("db").value
        },
        function(data){
        $('#sss').html(data);
        });
};
</script>
<select id="db" onChange="send();">
<option value="DESC">
DESC
</option>
<option value="ASC">
ASC
</option>
</select>
<br>
<textarea id="sss" style="width:150px;height:350px;margin:0px 70px;">
</textarea>
```

فایل combop2.php

```php
<?php
if(isset($_POST['status']))
{
    $status=$_POST['status'];
    $coo=@mysqli_connect("localhost","root","","age");
    if(!mysqli_connect_error()){
        $query="select * from sen ORDER BY age $status";
        $table=mysqli_query($coo,$query);
        while($rows=mysqli_fetch_assoc($table)) {
            echo "Age : ".$rows['age']."\n";
        }
        mysqli_close($coo);
    }
    else{
        echo 'error connect';
    }
}
else{
    header("location:combop1.php.php");
}
?>
```

ماشین حساب ساده

ابتدا یک فایل با نام cal.php ایجاد می کنیم که دو تا ورودی از کاربر می گیرد و چهار تا دکمه دارد که که برای چهار عمل اصلی می باشند که با زدن هر کدام از دکمه ها مقدار آن ها که با عملگر های خودشان پر شده است را به تابع جاوا می فرستد این تابع سه پارامتر را به صورت Ajax به صفحه cal2.php ارسال می کند و نتیجه آن را در برچسب نمایش می دهد . این سه پارامتری که به این تابع ارسال می کند دو ورودی کاربر می باشد و یکی نوع عملیاتی است که کاربر روی دکمه مورده نظر کلیک کرده است . اما در صفحه cal2.php ابتدا چک میکنیم که آیا داده ها به این صفحه ارسال شده است یا خیر اگه ارسال شده است آن ها را در متغییر ی ذخیره می کنیم . با استفاده از دستور شرطی switch نوع عملیات را میگیریم و بر اساس آن نوع عملیات را انتخاب می کنیم و در نهایت خروجی را چاپ می کنیم .

فایل cal.php

```
<script language="javascript" src="jquery.js"></script>
<script language="javascript">
function send(t){
    $.post('cal2.php',{a:document.getElementById("adadeaval").value,
        b:document.getElementById("adadedovom").value,
        c:t},
        function(data){
        $('#s').html(data);
        });
};
</script>
<input type="text" id="adadeaval" maxlength="6" size="10"/>
<input type="text" id="adadedovom" maxlength="6" size="10"/>
<label id="s">
=
</label>
<br />
<input type="button" value="+" onclick="send(this.value);"/>
<input type="button" value="-" onclick="send(this.value);"/>
<input type="button" value="*" onclick="send(this.value);"/>
<input type="button" value="/" onclick="send(this.value);"/>
```

فایل cal2.php

```php
<?php
if(!empty($_POST['a']) && !empty($_POST['b']) && !empty($_POST['c']))
{
    $a = $_POST['a'];
    $b = $_POST['b'];
    $c = $_POST['c'];
    switch($c)
    {
        case '+' :
        $sum = $a+$b;
        echo ' = '.$sum;
        break;
        case '-' :
        $sum = $a-$b;
        echo ' = '.$sum;
        break;
        case '*' :
        $sum = $a*$b;
        echo ' = '.$sum;
        break;
        case '/' :
        $sum = $a/$b;
        echo ' = '.$sum;
        break;
    }

}
else
{
    echo 'لطفا اعداد را وارد کنید';
}
?>
```

جدول ضرب

با استفاده از حلقه های تکرار متداخل میخواهیم یک جدول ضرب 10*10 ایجاد کنیم . ابتدا یک حلقه می نویسیم که از 1 تا 10 شمارنده دارد و درون آن هم یک حلقه دیگر می نویسیم که 1 تا 10 شمارنده دارد و سپس حاصل ضرب شمارنده های این دو حلقه را چاپ می کنیم . حلقه های متداخل به این صورت عمل می کنند که با یک حرکت حلقه بالایی حلقه درونی از ابتدا تا انتها انجام می شود . به مثال zarb.php دقت کنید.

```
<table border="1" cellpadding="7" cellspacing="3">
<?php
for($i=1;$i<=10;$i++)
{
    echo "<tr>";
    for($t=1;$t<=10;$t++)
    {
        echo "<th>".$t*$i."</th>";
    }

    echo "</tr>"."</th>";
}
?>
</table>
```

خروجی

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

حال این مثال را به صورت Ajax می نویسیم و مقدار جدول ضرب را از کاربر دریافت می کنیم . در اینجا ابتدا فایل zarbajax.php را ایجاد می کنیم که در این فایل دو ورودی از کاربر دریافت می کنیم و این دو ورودی را به صورت Ajax به صفحه zarbajax2.php ارسال می کنیم . در این فایل ابتدا بررسی می کنیم که آیا به این صفحه ورودی ارسال شده است یا خیر . داده ها را درون متغییر ذخیره می کنیم و مقدار حلقه های تکرار را تا مقدار ورودی دریافتی کاربر می نویسیم .

فایل zarbajax.php

```
<script language="javascript" src="jquery.js"></script>
<script language="javascript">
function send(){
    $.post('zarbajax2.php',{a:document.getElementById("a").value,
        b:document.getElementById("b").value},
        function(data){
        $('#s').html(data);
        });
};
</script>
<input type="text" id="a" size="2" onkeyup="send();" maxlength="2"/>
*
<input type="text" id="b" size="2" onkeyup="send();" maxlength="2"/>
<div id="s">
</div>
```

فایل zarbajax2.php

```
<table border="1" cellpadding="5" cellspacing="2">
<?php
if(!empty($_POST['a']) && !empty($_POST['b'])){
    $a=$_POST['a'];
    $b=$_POST['b'];
    for($i=1;$i<=$a;$i++)
    {
        echo "<tr>";
        for($t=1;$t<=$b;$t++)
        {
            echo "<th>".$t*$i."</th>";
        }
        echo "</tr>"."</th>";
    }
}
else
{
    echo 'لطفا عدد وارد کنید';
}
?>
</table>
```

معکوس کردن رشته

ابتدا یک صفحه برای دریافت کلمه از کاربر به نام reversp1.php ایجاد میکنیم و یک فرم طراحی میکنیم که یک ورودی و یک دکمه دارد و داده کاربر را به صفحه reversp2.php ارسال می کند . اما در صفحه reversp2.php ابتدا داده را دریافت میکنیم و یک متغییر به نام r از نوع آرایه تعیین می کنیم . و یک

تابع می نویسیم که کلمه را معکوس کند . این تابع ابتدا یک ورودی می گیرد و تعداد حروف ورودی را درون یک متغییر ذخیره می کنیم . یک حلقه می نویسیم که از تعداد حروف تا 0 شمارنده دارد و هر دفعه یکی از آن کم می شود . درون این حلقه با استفاده از تابع array_push() مقداری را به آخر آرایه r ذخیره می کنیم ما ورودی اول این تابع را نام آرایه می گذاریم و ورودی دوم را با استفاده از تابع substr رشته را میگیریم و از حرف شماره i شروع میکنیم و یکی از آن را بر می داریم بدین ترتیب با اجر اشدن حلقه از آخر رشته دریافتی شروع می کند و حروف آن را دونه دونه درون آرایه چاپ می کند . در آخر هم آریه را با استفاده از حلقه foreach چاپ می کنیم .

فایل reversp1.php

```
<form action="reversp2.php" method="post">
<input type="text" name="name" id="name" />
<input type="submit" value="Convert" />
</form>
```

فایل reversp2.php

```
<?php
$name=$_POST['name'];
$r=array();
function revers($nam)
{
    $t=strlen($nam);
    for($i=$t;$i>=0;$i--)
    {
        array_push($GLOBALS['r'],substr($nam,$i,1));
    }
    foreach($GLOBALS['r'] as $k)
    {
        echo $k;
    }
}
revers($name);
?>
```

حال این برنامه را با استفاده از Ajax می نویسیم . ابتدا یک فایل به اسم reversajaxp1.php ایجاد میکنیم که محتویات آن مانند فایل reversp1.php می باشد با این تفاوت که از تابع post برای Ajax استفاده شده است و ورودی کاربر را به صفحه reversajaxp2.php ارسال می کند . محتویات فایل reverajaxp2.php دقیقا مانند محتویات فایل reversp2.php می باشد .

فایل reversajaxp1.php

```
<script language="javascript" src="jquery.js"></script>
<script language="javascript">
function send(){
    $.post('reversajaxp2.php',{name:document.getElementById("name").value},
    function(data){
        $('#s').html(data);
    });
};
</script>
<input type="text" name="name" id="name" onkeyup="send();" size="50" />
<br /><br />
<div id="s">
</div>
```

فایل reversajaxp2.php

```
<?php
if(isset($_POST['name'])){

    $name=$_POST['name'];
    $r=array();
    function revers($nam)
    {
        $t=strlen($nam);
        for($i=$t;$i>=0;$i--)
        {
            array_push($GLOBALS['r'],substr($nam,$i,1));
        }
        foreach($GLOBALS['r'] as $k)
        {
            echo $k;
        }
    }
revers($name);
}
else
{
    header("location:reversajaxp1.php");
}
?>
```

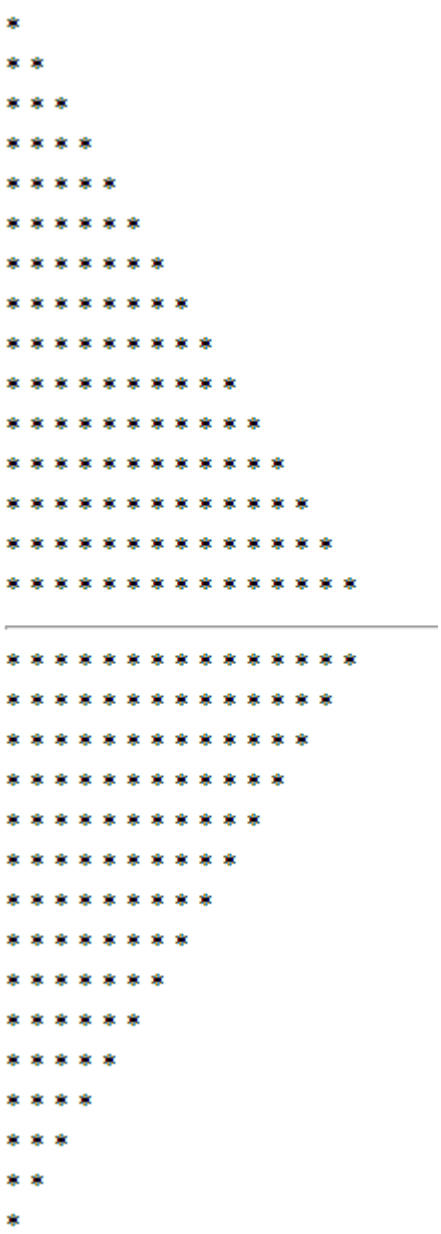
چند مثال از توابع

برنامه ave.php با استفاده از تابع و حلقه تکرار میانگین اعداد فرد بین 1 تا 500 را چاپ می کند .

فایل ave.php

```php
<?php
function av()
{
    $s=0;
    $ave=0;
    for($i=1;$i<=500;$i++)
    {
        if($i%2!=0)
        {
            $s+=$i;
        }
        $ave=$s/100;
    }
    return $ave;
}
echo av();
?>
```

با استفاده از حلقه های تکرار می خواهیم دو شکل زیر را طراحی کنیم . به تصویر این دو شکل نگاه کنید.



به برنامه setare.php توجه کنید .

```php
<?php
for($i=15;$i>=1;$i--)
{
    for($t=$i;$t<=15;$t++)
    {
        echo '*'."\t";
    }
    echo "<br>";
}
echo "<hr>";
for($i=1;$i<=15;$i++)
{
    for($t=$i;$t<=15;$t++)
    {
        echo '*'."\t";
    }
    echo "<br>";
}
?>
```

این برنامه با استفاده از دو حقله متداخل این اشکال را طراحی می کند .

رمزنگاری اطلاعات

برای ذخیره رمز کاربران در پایگاه داده بهتر است داده ها به صورت رمزنگاری در پایگاه داده ذخیره شود اگر رمز عبور کاربران به صورت رمزنگاری شده در دیتابیس ذخیره نشود وقتی هکری به دیتابیس سایت شما دسترسی پیدا کند تمامی رمز عبور ها را در اختیار دارد پس بهتر است داده ها را به صورت رمزنگاری شده در بیاید . برای اینکار PHP توابعی مختلفی دارد . MD5 یکی از محبوب ترین الگوریتم های رمزنگاری مورد استفاده توسط برنامه نویسان است . تابع md5 روش کاملا مطمئنی برای رمزنگاری نیست . اکثر کاربران تمایل دارند از رمز های عبور 5 یا 6 کاراکتری استفاده می کنند , رمز های عبور به این شکل که از پیچیدگی مناسبی بر خوردار نیستند و توسط یک حمله ساده آسیب پذیر خواهند بود . شکل صحیح استفاده از md5 به صورت زیر می باشد .

```php
<?php
echo md5(123456);
?>
```

برای اینکه این مشکل حل شود برنامه نویسان از دو روش استفاده می کنند , روش اول اضافه کردن یک عبارت نا مفهوم به پسورد ها قبل از رمزنگاری آن ها است . به تصویر زیر نگاه کنید .

```php
<?php
$salt='Super_Salt';
echo md5($password.$salt);
?>
```

روش دوم استفاده از الگوریتم های رمزنگاری طولانی تر (ولی کندتر) مانند sha1 , sha2 و ... است .

به تصویر زیر نگاه کنید .

```php
<?php
$userpass= sha1(md5($password));
?>
```

درج اطلاعات در فایل

ابتدا یک فایل به نام fileajax.php ایجاد می کنیم . که یک ورودی از کاربر میگیرد و این ورودی را به تابعی می دهد که این تابع این ورودی را به صورت Ajax به صفحه fileajax2.php ارسال می کند .اما در فایل fileajax2.php ابتدا بررسی میکینم که آیا ورودی به این صفحه ارسال شده است یا خیر اگر ارسال شده باشد یک فایل ایجاد می کنیم به اسم user.txt و مقدار ورودی را در این فایل می نویسیم و مقدار بازگشتی را که تعداد حروف نوشته شده است را چاپ می کنیم .

فایل fileajax.php

```php
<script language="javascript" src="jquery.js"></script>
<script language="javascript">
function send(){
    $.post('fileajax2.php',{in:document.getElementById("f").value},
    function(data){
        $('#s').html(data);
    });
};
</script>
<textarea type="text" onkeyup="send();" id="f" style="width:500px;height:250px">
</textarea>
<br /><br />
<div id="s">
</div>
```

فایل fileajax2.php

```php
<?php
if(!empty($_POST['in']))
{
    $input=$_POST['in'];
    $file=fopen("user.txt","w+");
    $c=fwrite($file,$input);
    echo 'تعداد کاراکتر نوشته شده : '.$c;
}
?>
```

درج روز هفته

ابتدا یک فایل به نامday1.php ایجاد می کنیم که روز هفته را از کاربر می گیرد و با استفاده از تابع Ajax آن را به صفحه day2.php می دهد و آن صفحه ورودی را به دستور switch می دهد و بر اساس شماره وارد شده توسط کاربر نام هفته را چاپ می کند .

فایل day1.php

```php
<script language="javascript" src="jquery.js"></script>
<script language="javascript">
function send(){
    $.post('arrayajax2.php',{n:document.getElementById("n").value},
    function(data){
        $('#s').html(data);
    });
};
</script>
<input type="text" name="n" id="n"/>
<input type="button" value="Insert !" onclick="send();" />
<br /><br />
<br /><br />
<div id="s">
</div>
```

فایل day2.php

```php
<?php
if(!empty($_POST['n'])){
    $num=$_POST['n'];
    switch($num)
    {
        case 1 :
        echo "شنبه";
        break;
        case 2 :
        echo "یکشنبه";
        break;
        case 3 :
        echo "دوشنبه";
        break;
        case 4 :
        echo "سه شنبه";
        break;
        case 5 :
        echo "چهارشنبه";
        break;
        case 6 :
        echo "پنج شنبه";
        break;
        case 7 :
        echo "جمعه";
        break;
        default;
        echo "اعداد بین 1 تا 7 وارد کنید";
    }

}
?>
```

محاسبه ی سن

ابتدا یک فایل به نام agetest1.php ایجاد میکنیم که سن کاربر را از ورودی بگیرد و آن را درون متغییر age قرار دهید و با تابع post آن را ب صورت ایجکس به صفحه agetest2.php بفرستد .اما در آن صفحه ابتدا بررسی میکنیم که آیا پارامتر age به این صفحه ارسال شده است یا خیر اگر ارسال شده باشد ابتدا متغییر day برای تعداد روز و متغییر week برای تعداد هفته و متغییر month برای تعداد ماه و متغییر h برای تعداد ساعت و متغییر m برای تعداد دقیقه و متغییر s برای تعداد ثانیه را در نظر میگیریم . سپس تابعی برای محاسبه می نویسیم که مقادیر بالا را محاسبه کند .

و در آخر پارامتر ورودی این صفحه را به تابع می دهیم . مقادیر را چاپ می کنیم .

فایل agetest1.php

```
<script language="javascript" src="jquery.js"></script>
<script language="javascript">
function send(){
    $.post('agetest2.php',{age:document.getElementById("n").value},
    function(data){
        $('#s').html(data);
    });
};
</script>
<input type="text" name="name" id="n" onkeyup="send();"/>
<br /><br />
<br /><br />
<br /><br />
<div id="s">
</div>
```

فایل agetest2.php

```
<?php
if(!empty($_POST['age'])){
    $y=($_POST['age']);
    $day;
    $week;
    $month;
    $h;
    $m;
    $s;
    function s($age)
    {
        $GLOBALS['day']=$age*365;
        $GLOBALS['week']=$GLOBALS['day']/7;
        $GLOBALS['month']=$GLOBALS['day']/31;
        $GLOBALS['h']=$GLOBALS['day']*24;
        $GLOBALS['m']=$GLOBALS['h']*60;
        $GLOBALS['s']=$GLOBALS['m']*60;
    }
    s($y);
    echo ' شما '.$s.' ثانیه و '.$m.' دقیقه و '.$h.' ساعت و '.
    $day.' روز و '.floor($week).' هفته و '.floor($month).' ماه سن دارید ';
}
?>
```

جستجو خطی

ابتدا یک فایل به نام searcharray.php ایجاد می کنیم که در این فایل یک آرایه موجود می باشد که دارای مقادیری می باشد این مقادیر می توانند از کاربر دریافت شده باشند یا از پایگاه داده دریافت شده باشد . با استفاده از تکنیک جستجو خطی می خواهم اسمی را درون این آرایه جستجو کنیم . ابتدا یک تابع می نویسیم که یک ورودی میگیرد و داخل این تابع متغییر s را برای بررسی قرار می دهیم و متغییری از جنس آرایه تعریف می کنیم . یک حلقه قرار می دهیم تا از اول آرایه تا آخر آرایه را جستجو کند برای این کار تعداد عناصر آرایه را با تابع count دریافت می کنیم . حلقه از 0 تا تعدا آرایه شمارنده دارد درون این حقله دستور شرطی می نویسیم تا آرایه با اندیس شمارنده را با مقدار ورودی تابع مقایسه کند اگر برابر بود به این معنی است که این کلمه درون آرایه وجود دارد و باید از حلقه خارج شد و مقدار s را برابر 1 میگذاریم بدین معنی که این مقدار پیدا شده است و در آخر دستور شرطی می نویسیم که آیا مقدار s مخالف 1 است یا خیر اگر مخالف باشد یعنی پیدا نشده و پیغامی مبنی بر پیدا نشدن برای کاربر چاپ می کنیم .

فایل searcharray.php

```php
<?php
$name=array('ali','mohsen','reza',
            'sina','mehdi','amir',
            'sajad','mohammad','javad');
function search($n)
{
    $s=0;
    $array=array();
    $array=$GLOBALS['name'];
    $t=count($array);
    for($i=0;$i<$t;$i++)
    {
        if($array[$i]==$n)
        {
            echo 'Yes Found in index number : '.$i;
            $s=1;
            break;
        }
    }
    if($s!=1) echo 'Not Fuond';
}
echo search('mohammad');
?>
```

خروجی

Yes Found in index number : 7

مرتب سازی آرایه

ابتدا یک فایل به نام sortarray.php ایجاد می کنیم . در این فایل ابتدا یک آرایه ایجاد می کنیم و مقادیر آن را به صورت دستی وارد می کنیم . یک تابع می نویسیم که مقدار آرایه را از بیرون میگیرد و تعداد مقادیر آن را با استفاده از تابع count درون متغییر t ذخیره می کنیم . دو حلقه برای مرتب کردن آرایه می نویسیم که از 0 شروع می شوند و تا تعداد عناصر آرایه منهای 1 ادامه پیدا می کند و مقادیر آرایه را با استفاده از دستور شرطی مرتب می کند در این دستور ابتدا بررسی می شود که آیا خانه بعدی آرایه بزرگتر از خانه کنونی آرایه است یا خیر اگر بزرگتر باشد مقدار کنونی را در متغییر ذخیره می کنیم تا از بین نرود و سپس مقدار خانه بعدی را در خانه کنونی می ریزیم و در آخر مقدار ذخیر ه شده که حاوی مقدار کنونی است را در خانه بعدی ذخیره می کنیم اینگونه مقدار آرایه مرتب می شود . در آخر هم با استفاده از حلقه مقادیر آرایه مرتب شده را چاپ می کنیم .

فایل sertarray.php

```php
<?php
$name=array(10,5,86,47,36,78,0,15,-8,888,165,11);
function sortt()
{
    $array=array();
    $array=$GLOBALS['name'];
    $t=count($array);
    for($i=0;$i<$t-1;$i++)
    {
        for($j=0;$j<$t-1;$j++)
        {
            if($array[$j+1]<$array[$j])
            {
                $temp=$array[$j];
                $array[$j]=$array[$j+1];
                $array[$j+1]=$temp;
            }
        }
    }
    foreach($array as $k)
    {
        echo $k."</br>";
    }
}
sortt();
?>
```

جستجو باینری

جستجو باینری از سرعت بسیا بالای نسبت به جستجو خطی می باشد و برای مقادیر زیاد بسیار بهینه می باشد . در الگوریتم باینری ابتدا مقادیر باید مرتب شده باشند این الگوریتم به این شکل عمل می کند که فرض کنید تعداد کل داده های ما برابر 200 عدد می باشد این الگوریتم ابتدا محل شروع جستجو و پایان آن رامشخص می کند و میانه را انتخاب می کند یعنی سراغ عنصر 100 می رود اگر این عنصر عنصر مورده نظر ما باشد با یک حرکت این عنصر را پیدا کرده است . اگر عنصر ما بزرگتر باشد الگوریتم شروع جستجو را از عنصر 101 شروع میکند چون عنصری که دنبال آن از میانه بزرگتر است حال میانه الگوریتم تغییر می کند و عنصر 150 میانه این الگوریتم می شود و اگر این عنصر همان عنصر باشد با دو حرکت جستجو به پایان می رسد و اگر دوباره عبارت مورده نظر بزرگتر باشد دوباره شروع الگوریتم از 151 می باشد و میانه الگوریتم به 175 تغییر می یابد . در واقع این الگوریتم به گونه ایی عمل می کند که عنصر وسط لیست انتخاب شده و با آرگومان جستجو مقایسه می شود تا تعیین شود از آن بزرگتر , کوچیکتر یا

مساوی است . اگر آرگومان از عنصر انتخاب شده بزرگتر باشد جستجو در نیمه بالایی و اگر کوچکتر باشد در نیمه پایینی لیست ادامه پیدا می کند .

فایل binerysearch.php

```php
<?php
$name=array(10,5,86,47,36,78,0,15,-8,888,165,11);
function sortt($nam)
{
    $array=array();
    $array=$nam;
    $t=count($array);
    for($i=0;$i<$t-1;$i++)
    {
        for($j=0;$j<$t-1;$j++)
        {
            if($array[$j+1]<$array[$j])
            {
                $temp=$array[$j];
                $array[$j]=$array[$j+1];
                $array[$j+1]=$temp;
            }
        }
    }
    return $array;
}
function binerysearch($a,$key){
    $arr=array();
    $arr=$a;
    $start=0;
    $end=(count($arr))-1;
    while($start<=$end){
        $mid=($start+$end)/2;
        if($arr[$mid]==$key){
            echo 'Found at : '.$key;
            break;
        }
        if($arr[$mid]>$key){
            $end=$mid-1;
        }
        elseif($arr[$mid]<$key){
            $start=$mid+1;
        }
    }
}
binerysearch(sortt($name),'165');
?>
```

اعداد زوج

ابتدا یک فایل به نام zoj.php ایجاد می کنیم و با استفاده ازحلقه اعداد زوج بین 0 تا 100 را چاپ میکند. شمارنده این حلقه 2 تایی می باشد .

فایل zoj.php

```php
<?php
$i=100;
do
{
    echo $i."<br>";
    $i-=2;
}while($i)
?>
```

حال برنامه ایی مینویسیم که اعداد که بر 5 بخش پذیر می باشند را چاپ می کند . با استفاده از شرط هر عددی که تقسم بر 5 می شود و باقیمانده آن 0 شود یعنی بر 5 بخش پذیر است بنابر این آن را چاپ می کنیم .

فایل baghimande5.php

```php
<?php
$i=100;
do
{
    if($i%5==0)
    echo $i."<br>";
    $i--;
}while($i)
?>
```

تابع میانگین

ابتدا یک فایل به نام average.php ایجاد می کنیم . درون این فایل یک تابع می نویسیم که یک آرایه از اعداد میگیرد و میانگین آن را چاپ می کند . ابتدا با استفاده از تابع count تعدا عناصر آرایه را محاسبه می کنیم و با استفاده از حلقه مقادیر آرایه را با هم جمع می کنیم و در متغییر ذخیره می کنیم در آخر مقدار کل را تقسیم بر تعداد می نماییم و چاپ می کنیم .

فایل average.php

```php
<?php
$n=array(14,12,11,18,17,19.75,15,14.5,20);
function av($a){
    $s=0;
    $t=count($a);
    foreach($a as $k){
        $s+=$k;
    }
    $s/=$t;
    echo $s;
}
av($n);
?>
```

ابتدا یک فایل با نام srtlen1.php ایجاد می کنیم . درون این فایل ابتدا یک ورودی برای دریافت کلمه از کاربر ایجاد می کنیم که مقدار ورودی کاربر را به تابع post می دهد که این تابع مقدار ورودی کاربر را با استفاده از Ajax به فایل srtlen2.php میفرستد . در این فایل ابتدا بررسی می کنیم که آیا پارامتر ورودی به این تابع ارسال شده است یا خیر اگر ارسال شده باشد آن را در متغییر ذخیره می کنیم و آن متغییر را به تابع strlen می دهیم که این تابع تعداد کاراکتر را بر میگیرداند و آن را چاپ می کنیم .

فایل strlen1.php

```php
<script language="javascript" src="jquery.js"></script>
<script language="javascript">
function send(){
    $.post('strlen2.php',{n:document.getElementById("f").value},
    function(data){
        $('#s').html(data);
    });
};
</script>
<input type="text" onkeyup="send();" id="f"/>
<br />
<div id="s">
</div>
```

فایل strlen2.php

```php
<?php
if(isset($_POST['n']))
{
    $name=$_POST['n'];
    echo ' تعداد کاراکتر  '.strlen($name);
}
else
{
    header("location:strlen1.php");
}
?>
```

جستجو درون رشته

با استفاده از تابع strpos() می توان یک آرایه را درون آرایه دیگر جستجو کرد . ابتدا یک فایل به نام searchajax.php ایجاد می کنیم که درون آن دو ورودی از کاربر دریافت می کنیم . ورودی اول کلمه است و ورودی دوم مقداری است که باید جستجو شود این دو مقدار را به صفحه searchajax2.php ارسال می کنیم . در این فایل ابتدا بررسی می کنیم که آیا ورودی به این صفحه ارسال شده است یا خیر اگر ارسال شده باشد این دو ورودی را در متغییر ذخیره می کنیم و به تابع strpos می دهیم که شماره کاراکتر را بر میگرداند .

فایل searchajax.php

```php
<script language="javascript" src="jquery.js"></script>
<script language="javascript">
function send(){
    $.post('searchajax2.php',{n:document.getElementById("f").value,
    k:document.getElementById("ff").value},
    function(data){
        $('#s').html(data);
    });
};
</script>
World :
<input type="text" id="f"/><br />
Keysearch
<input type="text" onkeyup="send();" id="ff"/>
<br />
<div id="s">
</div>
```

فایل searchajax2.php

```php
<?php
if(!empty($_POST['n']) && !empty($_POST['k']))
{
    $name=$_POST['n'];
    $key=$_POST['k'];
    echo ' در کاراکتر '.strpos($name,$key);
}
else
{
    header("location:searchajax.php");
}
?>
```

تکرار رشته

با استفاده از تابع str_repeat() می توان یک رشته را به میزان دلخواه تکرار کرد . ابتدا یک فایل به نام repeatajax1.php ایجاد می کنیم که دو ورودی از کاربر بگیرد اولی کلمه مورده نظر و دومی میزان تکرار آن می باشد و این دو متغییر را با استفاده از Ajax به صفحه repeatajax2.php میفرستد . در این صفحه ابتدا بررسی می کنیم که آیا دو پارامتر به این صفحه ارسال شده است یا خیر اگر ارسال شده باشد آن دو را درون متغییر ذخیره می کنیم و متغییر ها را به تابع str_repeat() می دهیم .

فایل repeatajax1.php

```php
<script language="javascript" src="jquery.js"></script>
<script language="javascript">
function send(){
    $.post('repeatajax2.php',{n:document.getElementById("f").value,
    k:document.getElementById("ff").value},
    function(data){
        $('#s').html(data);
    });
};
</script>
World :
<input type="text" id="f"/><br />
Repeat :
<input type="text" onkeyup="send();" id="ff"/>
<br />
<div id="s">
</div>
```

repeatajax2.php فایل

```php
<?php
if(!empty($_POST['n']) && !empty($_POST['k']))
{
    $n=$_POST['n'];
    $k=$_POST['k'];
    $n.="<br>";
    echo str_repeat($n,$k);
}
else
{
    header("location:repeatajax1.php.php");
}
?>
```

منابع

- کتاب مرجع کامل و مصور PHP 5 چاپ سوم . (مهندس مهرداد توانا و مهندس سعید هراتیان) انتشارات گروه مهندسی – پژوهشی ساحر
- زبان برنامه نویسی PHP 5 جلد اول . (محمد مصدری) انتشارات ناقوس
- زبان برنامه نویسی PHP 5 جلد دوم . (محمد مصدری) انتشارات ناقوس
- سایت www.w3schools.com
- سایت www.php.net
- سایت www.en.wikipedia.org

امیدوارم با خواندن این کتاب الکترونیکی به علم شما اضافه شده باشد .

محسن رجبی 1392/11/05

از طریق این ایمیل می توانید سوالات خود را با بنده در میان بگذارید .

m.kabir8895@yahoo.com