آموزش PHP

تالیف: مهندس افشین رفوآ
www.tahlildadeh.com

*

بسم الله الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتا بیس در ایران

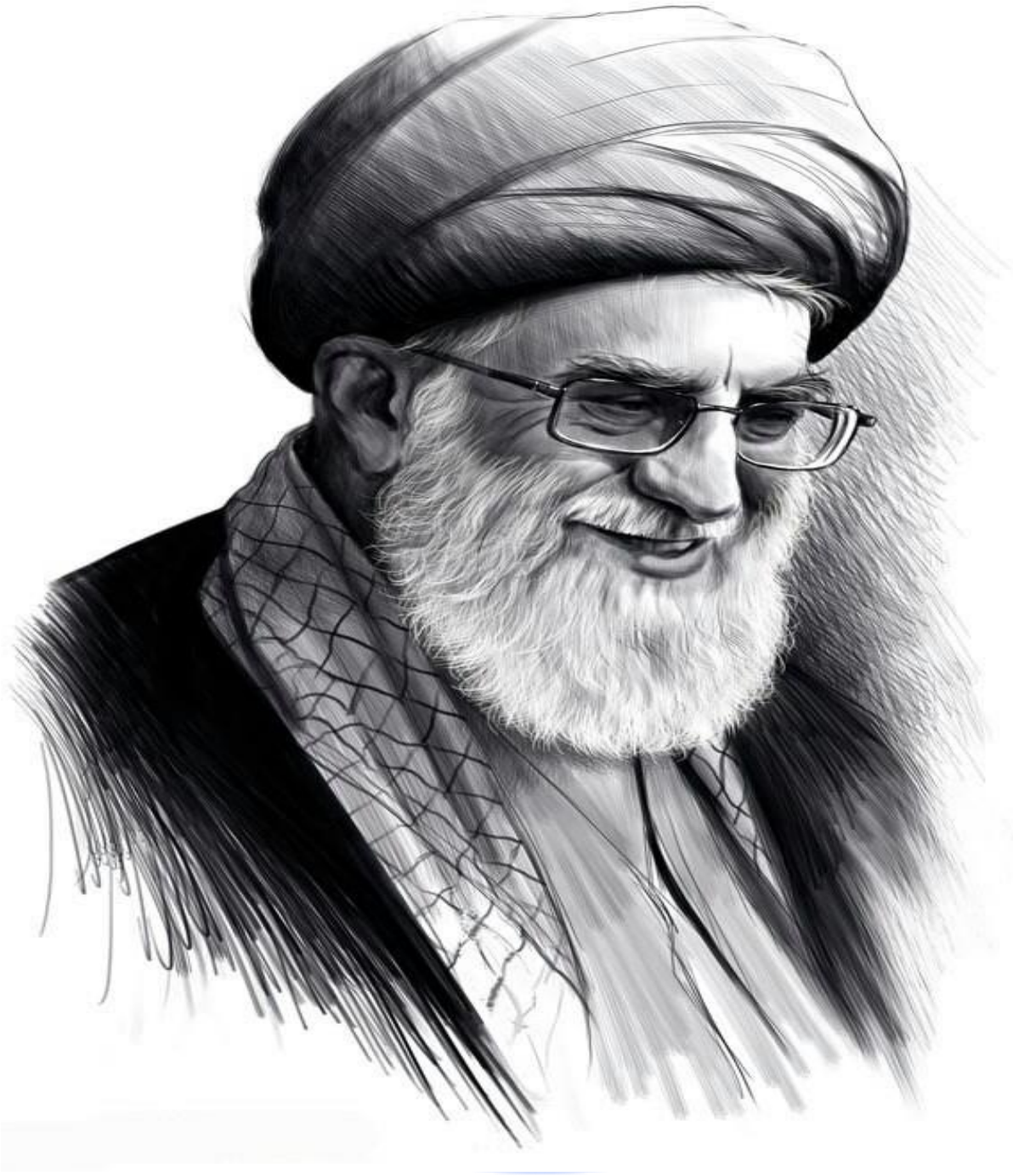
کتاب آموزشی برنامه نویسی اندروید در Android Studio

نویسنده : مهندس افشین رفوآ

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد7

88146323 - 88446780 - 88146330

*



تقدیم به نائب امام عصر، حضرت آیت الله خامنه ای که عصا زدنش ضرب آهنگ حیدری دارد

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد7

88146323 - 88446780 - 88146330

*

تقدیم به همهء جویندگان علم که توان و امکان شرکت

در کلاس های ما را ندارند

TAHLILGAR DADEHA INSTITUTE
آموزشگاه تحلیلگر داده ها

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد7

88146323 - 88446780 - 88146330

فهرست

بخش اول : آموزش مقدماتی PHP 10
معرفی زبان برنامه نویسی PHP 10
PHP چیست؟ 10
قبل از شروع آموزش PHP چه چیزهایی را بایستی بدانید؟ 10
یک فایل PHP چیست و چه کاربردی دارد؟ 10
PHP چه کارهایی می تواند انجام دهد؟ 11
چرا از PHP استفاده کنیم؟ 11
آموزش نصب زبانPHP 11
نصبPHP Parser 12
پیکربندی یا"configure Apache 12
پیکرندی فایلPHP.INI 12
پیکر بندیIIS WINDOWS 12
آموزش ساختار کدنویسیPHP 12
آموزش ساختار دستوری یا Syntax زبان: PHP 12
آموزش درجه توضیحات Comments در زبان: PHP 13
اهمیت بزرگ یا کوچک بودن حروف در: PHP 14
آموزش تعریف و استفاده از متغیرها در زبانPHP 14
قوانین نام گذاری متغیرها در: PHP 15
نمایش مقدار متغیرها در خروجی: PHP 15
PHPیک زبان برنامه نویسی بدون نیاز به تعریف نوع داده ای است 16
میدان دید (Scope) در متغیرهای: PHP 16
متغیرهای محلی Local و عمومی: Global 16
آموزش کاربرد واژه کلیدی Global در: PHP 17
آموزش استفاده از واژه کلیدی static درPHP 18
آموزش چاپ خروجی در زبانPHP 19
آموزش چاپ خروجی با دستورات echo و print در: PHP 19
آموزش کار با دستور echo در: PHP 19
آموزش کار با دستور Print در: PHP 20
آموزش انواع داده ای Data Tyes در: PHP 20
آشنایی با متغیر متنی String در: PHP 21
آشنایی با متغیرعددی Integer در: PHP 21

آشنایی با متغیر اعشاری یا Float در: PHP 22
آشنایی با متغیر درست یا غلط Boolean در: PHP 22
آشنایی با متغیر آرایه یا Array در: PHP 22
آشنایی با متغیر شی object در: PHP 22
آشنایی با متغیر تهی یا Null در: PHP 23
آشنایی با متغیر ارجاع به آدرس یا: Resource 24
آموزش کار با متن یا String در زبان: PHP 24
به دست آوردن طول یک رشته متن یاString 24
شمارش تعداد کلمات موجود در یک رشته متن یاString 24
برعکس کردن یک رشته متنی یا String در: PHP 24
آموزش جستجو برای یک کلمه خاص در یک رشته متنی String درPHP 25
آموزش جایگزین کردن یک کلمه درون متغیر متنی string در: PHP 25
مرجع کامل کار با متغیر متنی String درPHP 25
آموزش تعریف و کار با مقادیر ثابت یا Constant درPHP 25
آموزش تعریف و کار با مقادیر ثابت یا Constant درPHP 26
Constantها در PHP عمومی یا Global هستند: 27
آموزش کار با عملگرها یا operators در زبان: PHP 27
آموزش کار با ساختارهای شرطی if else درPHP 27
آموزش ساختار شرطیif ... else statement 29
آموزش ساختار شرطی: if ... else if ... else statement 29
آموزش کار با ساختار شرطی switch درPHP 30
آموزش کار با ساختار دستوری switch در: PHP 30
آموزش کار با حلقه های while loop درPHP 32
آموزش کار با حلقه while loop درPHP 32
آموزش کار با حلقه do…while درPHP 33
آموزش کار با تابع Function درPHP 34
آموزش تعریف تابع های دلخواه کاربر درPHP 34
آموزش ارسال پارامتر به تابع در زبانPHP 35
مقدار پیش فرض پارامترها در تابعPHP 36
آموزش مقدار برگشتی Return Value در تابعPHP 37
آموزش کار با آرایه یا Array در زبان: PHP 37
یک آرایه چیست؟ 37
آموزش تعریف آرایه Array درPHP 38

آموزش بدست آمدن طول آرایه با استفاده از تابع()Count 39
حرکت درون یک آرایه اندیس وار: 39
آموزش کار با حلقه های رابطه ای یا: Associative Arrays 39
حرکت درون آرایه های رابطه ای..... 40
آرایه های چند بعدیMultidimensionol Arrays 40
آموزش مرتب سازی آرایه ها (Sorting Array) درPHP 40
آموزش کار با تابع ()sort درPHP 41
آموزش کار با تابع ()rsort درPHP 41
آموزش کار با تابع ()asort درPHP 42
آموزش کار با تابع ()ksort درPHP 42
آموزش کار با تابع ()arsort درPHP 43
آموزش کار با تابع ()krsort در: PHP 43
آموزش کار با متغیر سراسری یا Global variables در: PHP 43
آموزش کار با متغیر سراسری $GLOBAL درPHP 44
آموزش کار با متغیر سراسری $SERVER درPHP 44
آموزش کار با متغیر سراسری $_REQUEST درPHP 47
آموزش کار با متغیر سراسری $_POST درPHP 48
آموزش کار با متغیر سراسری $_GET در: PHP 48
بخش دوم : آموزش مدیریت فرم (Form) درPHP 49
آموزش مدیریت فرم ها Form درPHP 49
مقایسه روش های GET و POST در ارسال اطلاعات: 50
چه زمانی از روش GET استفاده کنیم؟..... 51
چه زمانی از روش POST استفاده کنیم؟..... 51
آموزش اعتبارسنجی فرم ها در زبان: PHP 51
بررسی کادرهای متن فرم یاText Fields 52
بررسی دکمه های انتخابی یا: Radio Buttons 53
بررسی المنت فرم یاForm 53
نکته مهم در مورد امنیت فرم در: PHP 54
آموزش تعیین کادر های متن اجباری در فرم هایPHP 55
مثال عملی کادر متن اجباری درPHP 55
آموزش نمایش پیام های هشدار در فرم:PHP 56
آموزش اعتبارسنجی ایمیل و آدرس URL در فرم هایPHP 57
آموزش اعتبارسنجی مقدار Name در فرمPHP 57

آموزش اعتبارسنجی مقدار Email در فرمPHP 58
آموزش اعتبارسنجی آدرس صفحه URL درPHP 58
بررسی کد نهایی مثال و اعتبارسنجی کل اطلاعات: 58
آموزش کامل مثال عملی کار با فرم ها درPHP 60
آموزش نگهداری اطلاعات (values) در فرم هایPHP 60
مثال کامل کار با فرم های وب در: PHP 61
بخش سوم : آموزش دستورات پیشرفتهPHP 61
آموزش کار با تاریخ (Date) و ساعت (Time) درPHP 61
نکته آموزشی-نمایش اتوماتیک وار سال جهت کپی رایت 62
خواندن و نمایش ساده ساعت (time) به وسیله تابع()date 63
تعیین یک تاریخ با استفاده از تابع mktime درPHP 64
آموزش ایجاد یک تاریخ (Date) برحسب یک string با استفاده از تابع ()strtotime درPHP 64
مثال های عملی بیشتر کار با تابع ()date درPHP 65
مرجع کامل آموزش کار با Dateدر PHP 65
آموزش کار با تاریخ (Date) و ساعت (Time) درPHP 66
نکته آموزشی-نمایش اتوماتیک وار سال جهت کپی رایت 67
خواندن و نمایش ساده ساعت (time) به وسیله تابع()date 67
آموزش نحوه خواندن و نمایش ساعت محلی (time zone) درPHP 68
تعیین یک تاریخ با استفاده از تابع mktime درPHP 68
آموزش ایجاد یک تاریخ (Date) برحسب یک string با استفاده از تابع ()strtotime درPHP 69
مثال های عملی بیشتر کار با تابع ()date درPHP 70
آموزش کار با دستور include و require درPHP 70
مثال های عملی کار با دستور include درPHP 72
مقایسه دستورات require و include درPHP 73
آموزش مدیریت فایل ها در صفحاتPHP 74
آموزش دستکاری فایل ها درPHP 74
آموزش کار با تابع readfile درPHP 74
آموزش بازکردن' خواندن و بستن فایل درPHP 75
آموزش کار با تابع fopen درPHP 75
آموزش کار با فایل ()fread درPHP 76
آموزش کار با تابع ()fclose درPHP 77
آموزش کار با تابع ()fgets در PHP -خواندن یک خط از فایل 77
آموزش چک کردن انتهای فایل در PHP با تابع:()feof 78

آموزش خواندن یک کاراکتر در فایل PHP با()fgetc 78
آموزش ایجاد (crate) و نوشتن (write) در فایل هایPHP 78
آموزش ایجاد (create) فایل ها درPHP -تابع()fopen 79
آموزش اجازه دسترسی یا Peermission در فایل هایPHP 79
آموزش نوشتن (write)در یک فایل PHP با تابع:()fwrite 79
عملیات overwriting درPHP 80
آموزش ارسال فایل Upload در زبانPHP 80
مرحله اول تنظیم فایل"php.ini" 80
مرحله دوم-ایجاد فرم Html لازم جهت آپلودفایل 81
مرحله سوم-تعیین اسکریپت لازم جهت آپلود فایل 81
آموزش چک کردن این که فایل از قبل وجود داشته یا نه: 82
آموزش تعیین حجم مجاز فایل جهت آپلود: 83
آموزش تعیین نوع داده ای فایل جهت آپلود 83
کد کامل اسکریپت لازم جهت آپلود فایل 83
مرجع کامل کار با FileSystem درPHP 84
آموزش کار با کوکی Cookies در زبانPHP 85
آموزش ایجاد کوکی Cookie در زبانPHP 85
آموزش تغییر مقدار یک کوکی در زبانPHP 86
آموزش حذف یک کوکی درPHP 87
آموزش چک کردن فعال بودن کوکی در مرورگر کاربر 87
مرجع کامل HTTP درPHP 88
آموزش کار با Session در زبانPHP 88
مفهوم Session در PHP چیست؟ 88
آموزش نحوه کار شروع یک Session درPHP 89
آموزش خواندن مقادیر متغیر Session درPHP 89
تغییر یک متغیرPHP Session Variable 90
آموزش از بین بردن یکPHP Session 91
آموزش کار با Filters در زبانPHP 91
آموزش کار با تابع ()filter_var درPHP 92
آموزش نحوه پاکسازی یک String درPHP 92
آموزش چک کردن یک متغیر integer درPHP 93
آموزش چک کردن درستی یک IP با تابع()filter_var 93
آموزش اعتبارسنجی و پاکسازی یک Email با تابع()filter_var 94

آموزش اعتبارسنجی و پاکسازی یک URL با تابع: filter_var() 94
مرجع کامل کار با Filter ها درPHP 94
آموزش پیشرفته Filter کردن اطلاعات در زبانPHP 94
آموزش چک کردن مقدار یک عدد Integer در محدوده مورد نظر 95
آموزش اعتبارسنجی یک IPv6 با تابعfilter_var() 95
آموزش چک کردن Query String درURL 95
آموزش حذف کاراکترهای کد ASCII از یکString 95
مرجع کامل کار با Filter ها در زبانPHP 96
آموزش مدیریت خطا (Error Handling) درPHP 96
ایجاد یک مدیریت کننده خطا دلخواه(Custom Error Handling) 97
جدول درجه های خطا درPHP 98
آموزش تنظیم تابع مدیریت خطا درPHP 99
فعال کردن یک خطا یاerror trigger 100
آموزش ثبت خطا یا error logging درPHP 101
اموزش نحوه ارسال یک پیام خطا (Error Message) با ایمیل 101
آموزش کار با استثناء ها (Exception) در زبانPHP 102
یک استثناء یا Exception چیست؟ 102
آموزش نحوه ساده استفاده ازException 103
آموزش کار با ساختارهای دستوری Catch وthrow ، Try 104
آموزش ارسال مجدد خطا یاexception 108
آموزش تنظیم یک مدیریت کننده Exception سطح بالا 110
قوانین مربوط به exception در زبانPHP 110
بخش چهارم : آموزش پایگاه داده MySQL درPHP 111
آموزش کار با پایگاه داده MySQL درPHP 111
پایگاه داده MySQL چیست؟ 111
سیستم کارکرد پایگاه داده MySQL باPHP 112
آموزش جستجو در پایگاه داده یاDatabase Query 112
واقعیت هایی درباره پایگاه دادهMySQL 113
آموزش اتصال Connect به پایگاه دادهMySQL 113
کی و کجا از MySQL یا PDO استفاده کنیم؟ 113
ارائه مثال های عملی با ساختار دستوری PDO وMySQLi 114
آموزش نصبMySQLi 114
آموزش نصبPDO 114

آموزش باز کردن یا اتصال یا Connection بهMySQL 114
آموزش بستن یا Close اتصال یا Connection درMySQL 116
آموزش ایجاد یک پایگاه داده (Database) جدید درMySQL 117
آموزش ایجاد یک پایگاه داده MySQL جدید در روش PDO وMySQLi 117
آموزش ایجاد یک جدول (Table) جدید درMySQL 119
آموزش وارد کردن اطلاعات Data InsertدرMySQL 122
آموزش استخراج ID آخرین رکورد وارد شده در جدولMySQL 124
آموزش وارد کردن چندین رکورد همزمان درMySQL 126
آموزش استخراج اطلاعات (Select Data) از پایگاه دادهMySQL 128
آموزش خواندن (Select) اطلاعات درMySQLi 128
آموزش انتخاب اطلاعات (Select Data) با روش PDO و دستورات آمادهSQL 130
آموزش حذف اطلاعات (Data Delete) درMySQL 131
آموزش ویرایش اطلاعات (Update Data) درMySQL 133
آموزش محدود کردن اطلاعات (Limit Data) درMySQL 136
آموزش استفاده از دستورات آماده (Prepared) درSQL 137
آموزش کار با دستورات آماده SQL درMySQLi 138
مثال کار با دستورات آماده SQL درPDO 139
بخش پنجم: آموزش کاربرد XML درPHP 140
آموزش کار با زبان XML درPHP 140
XMLچیست؟ 140
یک مفسر زبان XML یا XML Parser چیست؟ 141
مفسر درختی XML یاTree-Based Parsers 141
مفسر مبتنی بر رویداد XML یاEvent-Based Parsers 141
آموزش کار با مفسر Parser SimpleXML در زبان PHP 142
نحوه نصب مفسر SimpleXML در PHP 142
آموزش خواندن XML از متن String در PHP 142
آموزش خواندن از فایل XML با مفسر: PHP SimpleXML 144
آموزش خواندن مقادیر گره ها با استفاده از SimpleXML 145
آموزش خواندن مقادیر گره ها (Node Values) باSimpleXML 145
آموزش خواندن مقادیر المنت های خاص باSimpleXML 145
آموزش خواندن مقادیر گره ها با استفاده از حلقه loop درPHP 146
آموزش خواندن مقدار خاصیت ها (Attribute) باSimpleXML 146
آموزش خواندن مقدار خاصیت ها (Attribute) با استفاده از حلقهloop 147

مثال های عملی بیشتر برای ابزارSimpleXML 147
آموزش کار با مفسر Expat XML درPHP 147
بررسی فایل XML مثال ها..... 148
مقداردهی اولیه XML Expat Parser درPHP 148
آموزش کار با مفسر XML DOM درPHP 150
آموزش نصب مفسر XML DOM درPHP 150
بررسی فایل XML مثال ها..... 150
آموزش خواندن و چاپ سند XML درPHP 151
حرکت درون فایل XML با استفاده از حلقهLoop 151
بخش ششم : آموزش کاربرد Ajax درPHP 152
Ajaxچیست؟..... 152
Ajaxچگونه کار می کنند؟..... 153
به کارگیری Ajax درgoogle Suggest 153
آموزش کار با Ajax درPHP 154
محتویات فایل PHP مثال"gethint.php" 155
واکشی اطلاعات از دیتابیس با فراخوانی توابعAJAX 156
آموزش AJAX – PHP و / XML واکشی اطلاعات از فایل XML با توابعAJAX 160
تابع ()showCD عملیات زیر را انجام می دهد 162
فایلPHP 162
آموزش – PHP پیاده سازی قابلیت جستجوی تعاملی و زنده با توابع(AJAX (AJAX Live search 163
پیاده سازی قابلیت جستجو تعاملی با توابعAJAX 163
صفحه HTML..... 165
فایلPHP 166
آموزش - PHP آموزش پیاده سازی خبر خوان RSS READER با AJAX درPHP 168
AJAX RSS Reader – RSS Readerای که داده ها را بدون بروز رسانی کل صفحه، در اپلیکیشن تحت وب بارگذاری می کند 168
فایلPHP 169
پیاده سازی قابلیت نظرسنجی با(AJAX (AJAX Poll 171
فایلPHP 172
فایل متنی 173

DADEHA INSTITUTE TAHLILGAR
آموزشگاه تحلیلگر داده‌ها

بخش اول : آموزش مقدماتی PHP

معرفی زبان برنامه نویسی PHP

PHP چیست؟

- PHP مخفف عبارت "PHP:Hypertext Preprecessor" است به معنای زبان پیش پردازنده مافوق متن.
- PHP یک زبان برنامه نویسی اسکریپتی پر کاربرد و اوپن سورس می باشد.
- اسکریپت های PHP بر روی سرور اجرا می شوند.
- زبان PHP ، رایگان است و شما می توانید آزادانه آن را دانلود و استفاده نمایید.

قبل از شروع آموزش PHP چه چیزهایی را بایستی بدانید؟

قبل از این شروع به مطالعه بخش آموزش PHP در سایت تحلیل داده نمایید، بایستی با مفاهیم و زبان های طراحی زیر آشنایی کافی داشته باشید. برای این منظور می توانید به بخش آموزش هر کدام از آن ها در سایت آموزشگاه تحلیل داده بروید :

- HTML زبان طراحی صفحات وب
- CSS روش قالب دهی اجزا صفحات وب
- JavaScript زبان اسکریپتی برنامه نویسی صفحات وب

یک فایل PHP چیست و چه کاربردی دارد؟

- فایل های PHP می توانند شامل متن، کدهای HTML یا CSS، اسکریپت های جاوا اسکریپت و یا کدهای PHP باشند.
- کدهای PHP بر روی سرور اجرا شده و نتایج حاصله جهت نمایش در مرورگر کاربر، به صورت HTML خالص ارسال می شوند.
- پسوند فایل های PHP به صورت php. می باشد.

راهنمایی : PHP یک زبان برنامه نویسی جذاب و پرطرفدار است. زبان PHP، به حدی قدرتمند است که می تواند هسته اصلی برنامه نویسی بزرگترین تیم وبلاگ دهی دنیا یعنی WordPress باشد.

زبان PHP، آن قدر دارای امکانات است که می تواند بزرگترین شبکه اجتماعی دنیا یعنی فیسبوک را اجرا کند.

از طرف دیگر، زبان PHP به حدی ساده است که می توان آن را به عنوان اولین زبان برنامه نویسی سمت سرور آموخت.

PHP چه کارهایی می تواند انجام دهد؟

- زبان PHP، می تواند محتویات دینامیک برای صفحات وب تولید کند.
- PHP می تواند فایل های موجود بر روی سرور را ایجاد کرده، باز کند، بخواند، بنویسد، پاک کرده و در نهایت ببندد.
- PHP می تواند اطلاعات ارسالی از طرف فرم های وب را جمع آوری و پردازش کند.
- PHP می تواند کوکی ها را بر روی سرور ارسال و دریافت کند.
- PHP می تواند اطلاعات موجود بر روی دیتابیس را اضافه کرده، حذف نموده و یا تغییر دهد.
- به وسیله زبان PHP، می توانید دسترسی کاربران را کنترل کنید.
- زبان PHP می تواند اطلاعات را رمزنگاری کند.

با استفا ده از زبان PHP، شما فقط محدود نیستید تا خروجی HTML تولید کنید. بلکه می توانید تصاویر، فایل های PDF و حتی فایل های تصویری فلش را نیز در خروجی نمایش دهید. علاوه بر این می توانید خروجی خود را به صورت متن یا XML نیز تولید کنید.

چرا از PHP استفاده کنیم؟

- PHP بر روی پلتفرم های مختلف مثل ویندوز، لینوکس، Unix و یا سیستم عامل Mac اجرا می شود.
- زبان PHP با اکثر سرورهای رایج امروزی مثل آپاچی، IIS و ... سازگار است.
- زبان PHP، طیف گسترده ای از پایگاه های داده را پشتیبانی می کند.
- PHP رایگان و آزاد بوده و به راحتی می توانید آن را از سرورهای PHP دانلود نمایید.
- یادگیری PHP بسیار ساده بوده و به صورت قدرتمند و موثر بر روی سرور اجرا می شود.

آموزش نصب زبانPHP

به منظور راه اندازی صفحات وب PHP سه جزو اساسی باید بر روی سیستم نصب شده باشد .
سرور وب _ سرور PHP تقریبا با تمام برنامه های وب سرور کار می کند از جمله Microsoft's Internet Information Server (IIS) اولی آنچه اغلب مواقع از ان استفاده می شود Apache Server است که همیشه آماده و رایگان در دسترس عموم قرار می گیرد. دیتابیس _ سرور PHP تقریبا با تمام برنامه های دیتابیس از جمله Oracle و Sybase کار می کند. تجزیه گر یا _PHP parser برای اینکه دستورهای اسکریپتی PHP پردازش شود یک parser باید نصب شده باشد تا خروجی HTML ایجاد شده را بتوان به مرورگر وب فرستاد. این راهنمای آموزشی شما را راهنمایی می کند چگونهPHP پارسر را روی سیستم نصب کنید.

نصبPHP Parser

قبل از اینکه ادامه دهید مهم است اطمینان کسب کنید محیط درست و مناسب PHP را نصب کرده اید تا بتوانید برنامه های وب خود را به کمک PHP طراحی کنید .

آدرس زیر را در آدرس باکس مرورگر خود وارد نمایید.

http://۱۲۷.۰.۰.۱/info.php

در صورتی که با این کار شما اطلاعات مربوط به نصب PHP را نشان دهد بدانید که وب سرور و PHP را با موفقیت نصب کرده اید.

پیکربندی یا"configure Apache

اگر در حال استفاده از از وب سرورapache هستید، این بخش چگونگی ِویرایش پیکرندی فایل های apacheرا تشریح می کند.

پیکرندی فایلPHP.INI

فایل پیکربندی PHP ویا ini ، php سریع ترین روش برای تغییر دادن عملکرد PHP هست.

پیکر بندیIIS WINDOWS

برای پیکربندی IIS بر روی سیستم عامل ویندوز خود می توانید از IIS راهنمای مرجع IIS که به همراه فرستاده شده استفاده کنید.

آموزش ساختار کدنویسیPHP

آموزش ساختار دستوری یا Syntax زبان: PHP

همانطور که قبلا اشاره کردیم، کدهای PHP بر روی سرور اشاره شده و سپس نتایج خروجی جهت نمایش بر روی کامپیوتر کاربر، به صورت تگ های خالص HTML ارسال می شود. یک کد PHP را می توانید در هر جای سند وب خود قرار دهید. اسکریپت هایPHP ، بایستی با عبارت پایان یابند .همانند کد مثال زیر:

```
<?php
<!--?php
// کدهای پی اچ پی را در این جا می نویسید
?-->
?>
```

پسوند پیش فرض فایل هایPHP ، به صورت php. می باشد. یک فایل PHP معمولا از تگ های HTML مقداری اسکریپت های PHP تشکیل می شود. در کد مثال عملی زیر، یک فایل ساده PHP را داریم که با استفاده از

یکی از توابع درون ساخته زبان PHP به نام"echo" ، می تواند متن عبارت "Hel l o worl d" را در خروجی چاپ کرده و نمایش دهد:

```
<?php
< !DOCTYPE html >
< html >
< body >

<h1>اولین صفحه پی اچ پی من</h1>

<!--?php
echo "Hello World!";
?-->

< /body >
< /html >
?>
```

نکته مهم:

عبارات دستوریPHP ، بایستی حتما با (;) سیمکالون خاتمه یابند.

آموزش درجه توضیحات Comments در زبان: PHP

یک توضیح یا Comment در کدهایPHP ، یک یا چند خط متن است که توسط کامپایلر خوانده نشده و اجرا هم نمی شود. از کامنت ها معمولا برای درجه توضیحات مربوط به کدها برای سایر برنامه نویسان، غیر فعال کردن یک بخش دستورات به صورت موقت و یا مشخص کردن بخش های مختلف کد استفاده می شود. به طور کلی توضیحات در موارد زیر کاربرد دارند:

- به سایر خوانندگان و برنامه نویسان اجازه می دهد بفهمند شما چه کاری انجام می دهید.
- به خودتان در یادآوری و فهمیدن بخش های مخالف کد کمک می کند. برخی موانع ممکن است پس از یک مدت طولانی به کدهای نوشته شده خود بازگردید، در اینجاست که بایستی بفهمید هر بخش چه کاری انجام می دهد.

در زبانPHP ، چندین روش برای درجه Comment در کدها وجود دارد که همه را در مثال عملی زیر نشان داده ایم:

```
<?php
<!--?php
// توضیح تک خط

# شکل دیگر توضیح تک خط

/*
نحوه درج توضیح
چند
```

```
خطی
*/

// شما همچنین می توانید از توضیحات برای عدم اجرای یک بخش از دستور استفاده کنید
$x = 5 /* + 15 */ + 5;
echo $x;
?-->
?>
```

اهمیت بزرگ یا کوچک بودن حروف در: PHP

در زبانPHP ، تمامی کلیدواژه های اصلی مثلel se ، if ، while، echo، کلاس ها، توابع و ... به حروف بزرگ یا کوچک حساس نیستند. به عبارت دیگر Case Senetive نمی باشند.

در کد مثال زیر، هر دستور echoo یک کار را انجام داده و نوشتن آن ها بدون اشکال است:

```
<?php
<!--?php
ECHO "Hello World!<br-->";
echo "Hello World!<br>";
EcHo "Hello World!<br>";
?>
?>
```

اما نام متغیرها یا variable به حروف بزرگ و کوچک حساس هستند .

در کد مثال زیر، فقط دستور خط اول مقدار متغیر $col or را نمایش خواهد داد. زیر سایر متغیرهای$Col oR و$coLORR به عنوان متغیرهای دیگری محسوب خواهند شد.

آموزش تعریف و استفاده از متغیرها در زبانPHP

همانطور که می دانید، متغیرها، فضاهایی در حافظه برای نگهداری اطلاعات هستند.

در زبانPHPP ، یک متغیر با کاراکتر $ شروع شده و پس از آن بایستی نام متغیر را تعیین کرد. مثال های عملی زیر، نحوه انجام کار را نشان می دهند.

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

پس از اجرا شدن قطعه کد مثال فوق، متغیر $txt مقدار متنی یا Hello world! را خواهد داشت، متغیر $x مقدار ۵ و متغیر $y نیز شامل مقدار ۱۰.۵ می شود.

نکته کاربردی:

در هنگام تعریف متغیرهای متنی در زبانPHP ، بایستی مقدار آن ها را بین دو " " قرار دهید.

نکته کاربردی:

بر خلاف سایر زبان های برنامه نویسی رایج، زبان PHP دستور خاصی برای تعریف متغیرها ندارد. هر متغیر، به محض مقداردهی شدن، ایجاد شده و قابل استفاده خواهد بود.

قوانین نام گذاری متغیرها در: PHP

یک متغیر درPHP ، می تواند یکم نام بسیار کوتاه مثل x یا y و یا یک نام بلند و بامفهوم مثلage ، username و ... داشته باشد. در نام گذاری متغیرهایPHP ، بایستی نکات زیر را رعایت نمایید:

- نام یک متغیر همواره با علامت $ شروع شده و پس از آن نام متنی تعیین می شود مثل.$age
- نام یک متغیر بایستی با یک حرف یا علامت (_) شروع شود.
- نام یک متغیر نمی تواند با عدد شروع شود.
- نام متغیرها در زبانPHP ، فقط می تواند شامل حروف انگلیسی، اعداد و (_) باشد و استفاده از سایر کاراکترها، غیر مجاز است.
- نام متغیر در زبان PHP به حروف بزرگ و کوچک حساس بوده و یا Case Sensetive هستند. بنابراین متغیرهای $age و $Age یا $AGE با هم متفاوت خواهند بود.

نمایش مقدار متغیرها در خروجی: PHP

در زبانPHP ، معمولا می توان با استفاده از دستورecho ، مقدار یک متغیر را بر روی خروجی نمایش داد. کد مثال عملی زیر، نحوه نمایش مقدار یک متغیر متنی را نشان می دهد:

```
<?php
<!--?php
$txt = "W3Schools.com";
echo "I love $txt!";
?-->
?>
```

همچنین، کد مثال زیر نیز، خروجی همانند مثال قبل تولید می کند:

```
<?php
<!--?php
$txt = "W3Schools.com";
echo "I love " . $txt . "!";
?-->
?>
```

راهنمایی:

در درس بعدی، به آموزش کامل دستور echo در PHP و چاپ اطلاعات در خروجی خواهیم پرداخت.

PHP یک زبان برنامه نویسی بدون نیاز به تعریف نوع داده ای است

اگر به کد مثال فوق دقت نمایید، متوجه می شوید که در هنگام تعریف یک متغیرPHP ، مجبور نیستیم نوع داده ای آن را به برنامه اعلام کنیم.

خود زبانPHPP ، به صورت اتوماتیک، متغیر مورد نظر را به نوع داده ای متناسب با مقدار داده شده، تبدیل کرده و ذخیره می کند. اگر مقدار متنی برای یک متغیر تعیین کرده باشید، PHPآن را از نوع متنی و در صورت تعیین عدد، متغیر را از نوع عددی مشخص می کند.

اما در سایر زبان های برنامه نویسی مثلC ، ++CCیا جاوا، برنامه نویس بایستی قبل از مقداردهی و استفاده از یک متغیر، حتما نوع داده ای آن را برای کامپایلر مشخص کند.

میدان دید (Scope) در متغیرهای: PHP

در زبانPHP ، یک متغیر را می توانید در هر بخشی از اسکریپت که بخواهید، تعریف کنید. میدان دید یا Scopeیک متغیر، محدوده ای را مشخص می کند که در آن می توانید به متغیر مورد نظر دسترسی داشته و آن را تغییر داده یا استفاده نمود.

در زبانPHP ، سه نوع میدان دید یا Scopee برای متغیرها به شرح زیر وجود دارد:

- ocal یا محلی
- gl obalیا سراسری
- Stat i cیا ثابت

در ادامه به تشریح هر یک از ۳ حالت میدان دید متغیرها در PHP خواهیم پرداخت.

متغیرهای محلی Local و عمومی: Global

اگر یک متغیر، خارج از محدوده یک f unct i on تعریف شود، همانند متغیر x در کد مثال زیر، دارای میدان دید عمومی یا GLOBAL SCOPE بوده و فقط در خارج آن تابع قابل استفاده است.

در کد مثال زیر، به دلیل تعریف متغیر X درخارج از تابع myTestt ، به کار بردن آن در داخل تابع باعث بروز خطا می شود:

```
<?php
<!--?php
$x = 5; // global scope

function myTest() {
// در این تابع باعث بروز خطا می شود x استفاده از
echo "<p-->Variable x inside function is: $x<p></p>";
}
myTest();
```

```
echo "<p>Variable x outside function is: $x</p>";
?>
?>
```

اگر یک متغیر، درونی یک تابع یا function تعریف شود، مثل متغیر x در مثال زیر، دارای میدان دید محلی یا LOCAL SCOPE بوده و فقط درون آن تابع قابل استفاده می باشد؛ در کد مثال زیر، به کار بردن متغیر X در خارج از تابع myTest() باعث بروز خطا می شود:

```
<?php
<!--?php
function myTest() {
$x = 5; // local scope
echo "<p-->Variable x inside function is: $x<p></p>";
}
myTest();

// در خارج از این تابع باعث بروز خطا می شود x استفاده از
echo "<p>Variable x outside function is: $x</p>";
?>
?>
```

نکته:

شما می توانید متغیرهای محلی Local Variables با نام های یکسان در توابع مختلف داشته باشید. زیرا یک متغیر محلی یا Local فقط درون تابع خود شناسایی شده و قابل استفاده است.

آموزش کاربرد واژه کلیدی Global در: PHP

از واژه کلیدی global در PHP ، برای فراهم نمودن امکان دسترسی به یک متغیر global یا سراسری، درون یک تابع استفاده می شود. کد مثال زیر، نحوه استفاده از واژه کلیدی global را به صورت عملی نشان داده است:

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

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

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

از طرف دیگر، PHPتمای متغیرهای سراسری یا gl obal برنامه را در یک آرایه به نام $GLOBALS[i ndex] نگهداری می کند. در این آرایه i ndex در واقع نام متغیرها می باشد. از این آرایه می توان برای دسترسی به متغیرهای سراسری برنامه، خواندن و یا تغییر مقدار آن ها درون توابع استفاده کرد. مثال زیر نحوه استفاده از این آرایه را در عمل نشان می دهد:

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

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

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

آموزش استفاده از واژه کلیدی static در PHP

به طور معمول، پس از این که یک تابع به طور کامل اجرا شده و عملیات آن پایان یابد، کلیه متغیرهای درون تابع حذف می شوند. اما گاهی اوقات ممکن است بخواهیم یک متغیر محلی l ocal پس از اتمام تابع آن، پاک نشود. برای این منظور بایستی از واژه کلیدی static استفاده نمود. به این صورت که در هنگام تعریف متغیر مورد نظر، کلمه static را همانند کد مثال زیر، قبل از نام متغیر قرار می دهیم:

```
<?php
<!--?php
function myTest() {
static $x = 0;
echo $x;
$x++;
}

myTest();
myTest();
myTest();
?-->
?>
```

در کد مثال فوق، هر بار که تابع myTest() فراخوانی شود، متغیر x دارای مقداری است که در آخرین بار اجرای تابع آن، به متغیر نسبت داده شده است.

نکته:

متغیر x در کد مثال فوق، همچنان یک متغیر محلی یا l ocal برای تابع فوق محسوب شده و در سایر تابع ها قابل استفاده نیست.

آموزش چاپ خروجی در زبانPHP

آموزش چاپ خروجی با دستورات echo و print در: PHP

در زبان PHP ، دو راه اصلی برای چاپ مقدار در خروجی و نمایش به کاربر وجود دارد : دستورات Print و echo.

در مثال های این درس، از توابع echo و Print برای چاپ خروجی بر روی صفحه استفاده خواهیم کرد. تابع های echo و print تقریبا عملکرد یکسانی داشته و از هر دو برای چاپ و نمایش خروجی بر روی صفحه استفاده می شود.

تفاوت اندکی بین توابع echo و Print به صورت زیر وجود دارد:

تابع echo دارای مقدار برگشتی یا return value نیست، در حالی که تابع print عدد ۱۱ را به عنوان خروجی بر می گرداند که آن را می توان در سایر دستورات استفاده کرد. از طرف دیگر تابع echo می تواند یک یا چند پارامتر را به عنوان ورودی دریافت کند (گرچه در عمل استفاده از آن بسیار کم است) ولی تابع Print نمی تواند پارامتر ورودی داشته باشد. در نهایت تابع echo در هنگام اجرا، کمی سریع تر از تابع Print است.

آموزش کار با دستور echo در: PHP

دستور echo را هم با پرانتز و هم بدون پرانتز می توان به کار برد. به صورت echo یا.()echo

مثال نمایش متن : در کد مثال زیر، نحوه نمایش متن یا Text به وسیله دستور echoo را نشان داده ایم. توجه داشته باشید که متن مورد نظر، می تواند حاوی تگ های HTML نیز باشد:

```
<?php
<!--?php
echo "<h2-->PHP is Fun!";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";
?>
?>
```

مثال نمایش متغیرها : Variables در مثال عملی زیر، نحوه نمایش متن (Text) به همراه متغیر variable را با استفاده از دستور echo نشان داده ایم:

```
<?php
<!--?php
$txt1 = "Learn PHP";
$txt2 = "TahlilDadeh.com";
$x = 5;
$y = 4;
```

```
echo "<h2-->$txt1";
echo "Study PHP at $txt2<br>";
echo $x + $y;
?>
?>
```

آموزش کار با دستور Print در: PHP

دستور Print را نیز هم می توان با پرانتز یا بدون پرانتز به کار برد. به صورت Print یا.() Print
مثال نمایش متن : کد زیر نحوه نمایش متن (Text) در خروجی را به وسیله دستور printt نشان می دهد.
توجه داشته باشید که متن می تواند شامل تگ های HTML نیز باشد:

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

مثال نمایش متغیر : کد زیر نحوه نمایش مقدار متغیرها variables به وسیله دستور Print را در خروجی نشان می دهد:

```
<?php
<!--?php
$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;

print "<h2-->$txt1";
print "Study PHP at $txt2<br>";
print $x + $y;
?>
?>
```

آموزش انواع داده ای Data Tyes در: PHP

متغیرها در زبانphp ، همانند سایر زبان های برنامه نویسی دیگر، دارای انواع مختلفی هستند که هر کدام کاربرد خاص خود را دارند. لیست زیر انواع داده ای مختلف زبان PHP را نشان داده است:

- متغیر متنی یاstring
- متغییر عددی معمولی یاinteger
- متغیر عددی اعشاری یا Float که به آن double هم می گویند
- متغیر درست یا غلط یاBoolean
- متغیر آرایه ای یاArray

- متغیر شی گونه یا object
- متغیر تهی یا Null
- متغیر ارجاع به آدرس یاResource

آشنایی با متغیر متنی String در: PHP

یک متغیر متنی یاString ، عبارت است از مجموعه از کاراکترهای حرفی پشت سر هم مثل

عبارت."Hello World!"

متغیرهای متنی String بایستی درون یک جفت یا " " تعریف شوند. همانند کد مثال زیر:

```
<?php
<!--?php
$x = "Hello world!";
$y = 'Hello world!';

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

آشنایی با متغیرعددی Integer در: PHP

یک متغیر عددی ساده یاinteger ، عددی غیر اعشاری در محدوده ۲.۱۴۷.۶۴۸- تا ۲.۱۴۷.۶۴۷ است. در تعیین متغیرهای عددی integer بایستی نکات زیر را رعایت نمایید:

- یک متغیر عددی integer حداقل بایستی دارای یک رقم عددی باشد.
- متغیرهای عددی integer نمی توانند اعشار داشته باشند.
- یک متغیر عددی integer می تواند مثبت یا منفی باشد.

در مثال عملی زیر، متغیر x یک integer است. تابع var_dump() هم مقدار و هم نوع متغیر را بر می گرداند:

```
<?php
<!--?php
$x = 5985;
var_dump($x);
?-->
?>
```

آشنایی با متغیر اعشاری یا Float در: PHP

یک متغیر اعشاری یا float شامل یک بخش عددی به همراه یک قسمت اعشاری یا نمایی (کسری) می باشد.

در کد مثال زیر، متغیر x به عنوان یک متغیر اعشاری float تعریف و مقداردهی شده است. سپس به کمک تابع var_dump() ، نوع و مقدار متغیر را در خروجی نشان داده ایم:

```
<?php
<!--?php
$x = 10.365;
var_dump($x);
?-->
?>
```

آشنایی با متغیر درست یا غلط Boolean در: PHP

یک متغیر Boolean می تواند مقدار درست true یا غلط false را داشته باشد، همانند کد مثال زیر:

```
<?php
$x = true;
$y = false;
?>
```

متغیرهای Boolean معمولا در عبارت های شرطی مورد استفاده قرار می گیرند. در درس های آینده به آموزش نحوه کار با دستورات شرطی خواهیم پرداخت.

آشنایی با متغیر آرایه یا Array در: PHP

یک آرایه یا Array ، مجموعه ای از دو یا چند متغیر هم نوع است که درون یک متغیر واحد تعریف می شوند.

در کد مثال عملی زیر، نحوه تعریف و مقداردهی متغیر آرایه $Car را نشان داده و سپس به وسیله تابع var_dumpp()، مقدار و نوع آن را در خروجی چاپ کرده ایم:

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

در یک درس، به تشریح کامل نحوه کار با متغیرهای آرایه خواهیم پرداخت.

آشنایی با متغیر شی object در: PHP

یک متغیر شی یا object ، نوع داده ای است که هم مقدار داده متغیر و هم اطلاعات لازم جهت پردازش داده های آن را نگهداری می کند.

در زبانPHP ، جهت استفاده از متغیر شی یا obj ect بایستی آن را کامل و صریح تعریف نمود. برای این منظور ما ابتدا بایستی یک کلاس cl ass برای شی تعریف کنیم. یک کلاس با واژه کلیدی calss تعیین شده و ساختاری است که می تواند خواص (pr operti es) و متدهای (met hods) مربوط به متغیر obj ect را در خود نگهداری کند.

مثال عملی زیر، نحوه تعریف کلاس Car و سپس ایجاد یک متغیر یا obj ect از روی آن کلاس به نام $herbieeرا نشان می دهد. پس از تعریف متغیر شی، می توان از آن در سطح کد برنامه استفاده کرد:

```
<?php
<!--?php
class Car {
function Car() {
$this--->model = "VW";
}
}

// ایجاد یک شی
$herbie = new Car();

// نمایش خواص شی
echo $herbie->model;
?>
?>
```

در درس آموزش شی یا object به بررسی کامل نحوه کار با اشیا در PHP خواهیم پرداخت.

آشنایی با متغیر تهی یا Null در: PHP

نوع داده ای Nul l یک متغیر خاصی است که فقط می تواند یک مقدار داشته باشد، خالی، تهی یا. Nul l یک متغیر از نوع داده ای Nul l l ، متغیری است که مقداری به آن نسبت داده نشده است.

راهنمایی:

اگر یک متغیر را بدون مقدار تعریف کنید، به صورت خودکار از نوع Nul l تعیین خواهد شد.

همچنین با نسبت دادن مقدار Nul l به یک متغیر، می توان آن را خالی کرد. همانند کد مثال زیر:

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

آشنایی با متغیر ارجاع به آدرس یا: Resource

نوع داده ای ویژهResource ، درواقع یک نوع داده ای مجزا نیست. این نوع متغیر، درواقع نگهدارنده یک ارجاع یا آدرس به یک تابع، منبع و یا متغیر دیگر در کدهای PHP می باشد.
یک نمونه راجع از استفاده از نوع داده ایResource ، فراخوانی پایگاه داده یا database call می باشد.
ما در این درس، از آنجا که متغیر Resourcee یک مبحث کاملا تخصصی است، به شرح توضیحات بیشتر نپرداخته و آن را به بخش های بعدی موکول می کنیم.

آموزش کار با متن یا String در زبان: PHP

یکString ، مجموعه ای پشت سر هم از کاراکترها مثل عبارت "Hello World!" است. در این درس به بررسی مهم ترین و پرکاربردترین توابع کار با متن ها یا string در PHP خواهیم پرداخت.

به دست آوردن طول یک رشته متن یا String

تابع strlen() درPHP ، طول یا به عبارت دیگر تعداد کاراکترهای موجود در یک متغیر رشته ای یا String را بر می گرداند.
کد زیر طول رشته "Hello World!" که برابر ۱۲۲ است را بر می گرداند:

```
<?php
<!--?php
echo strlen("Hello world!"); // outputs 12
?-->
?>
```

شمارش تعداد کلمات موجود در یک رشته متن یاString

به وسیلع تابع str_word_count() در زبانPHP ، می توانید تعداد کلمات موجود در یک متغیر متنی String شمرده و برگردانید. کد زیر نحوه انجام این کار را نشان می دهد که خروجی آن عدد ۲ است:

```
<?php
<!--?php
echo str_word_count("Hello world!"); // outputs 2
?-->
?>
```

برعکس کردن یک رشته متنی یا String در: PHP

تابع strrev() در زبانPHP ، می تواند یک متغیر متنی یا String را برعکس کرده و در خروجی نمایش دهد.
کد زیر نحوه انجام این کار را نشان می دهد.

```
<?php
```

```
<!--?php
echo strrev("Hello world!"); // outputs !dlrow olleH
?-->
?>
```

خروجی کد مثال فوق مقدار !dlrow olleH خواهد بود.

آموزش جستجو برای یک کلمه خاص در یک رشته متنی String در PHP

تابع()Strpos ، به دنبال یک کلمه یا Text در یک متغیر متنی String می پردازد.

اگر مورد یکسانی با گزینه در حال جستجو، پیدا شود، برنامه شماره اولین کاراکتر کلمه را بر می گرداند. اگر هم کلمه مورد نظر در رشته ی متنی نباشد، مقدار FALSE بازگشت داده می شود.

به وسیله کد مثال زیر، به دنبال کلمه "World" در متغیر متنی "Hello Worldd!" پرداخته ایم:

خروجی کد مثال فوق عدد ۶ بوده که شماره اولین کاراکتر کلمه "World" در متن است.

نکته:

شماره اندیس اولین کاراکتر در یک متغیر متنی، ۰ است نه یک.

آموزش جایگزین کردن یک کلمه درون متغیر متنی string در: PHP

به وسیله تابع ()str_rapl ace در PHP ، می توانید یک سری از کاراکترها را با کاراکترهای جدیدی در یک رشته متنیString ، جایگزین کنید. در کد مثال عملی زیر، کلمه "worl d" را با کلمه "Dol l y" عوض کرده ایم :

```
<?php
<!--?php
echo str_replace("world", "Dolly", "Hello world!"); // outputs Hello Dolly!
?-->
?>
```

مرجع کامل کار با متغیر متنی String درPHP

برای دریافت اطلاعات راجع به لیست کامل توابع کار با متن ها یا String درPHP ، به بخش مرجع String در سایت ما بروید.

آموزش تعریف و کار با مقادیر ثابت یا Constant در PHP

Constantها یا مقادیر ثابت، همانند متغیرها یا vari abl es هستند با این تفاوت که پس از تعریف و مقداردهی، نمی توان مقدار آن را تغییر داده و یا خالی کرد.

یک Constant یک شناسه (یا نام) برای یک مقدار ثابت در برنامه بوده که در خلال اجرای اسکریپت، مقدار

آن نمی تواند تغییر کند. نام یک Constantبایستی با یک حرف یا علامت (_) شروع شود و بر خلاف تعریف متغیرها، نیازی به استفاده از $ در ابتدای نام نیست.

نکته مهم:

بر خلاف متغیرها، مقدارهای ثابت یا Constant به صورت خودکار و پیش فرض، در کل کد برنامه عمومی یا Globalهستند.

آموزش تعریف و کار با مقادیر ثابت یا Constant درPHP

نحوه تعریف یک مقدار ثابت یا: Constant

برای تعریف یک Constant در PHP از ساختار کلی زیر استفاده می شود:

```
<?php
define(name, value, case-insensitive)
?>
```

- name : نام مورد نظر جهت Constant را تعیین می کند.
- value : مقدار مورد نظر برای Constant را مشخص می کند.
- Case : تعیین می کند که آیا نام Constant به حروف بزرگ و کوچک حساس باشد یا خیر. به صورت پیش فرض false است.

مثال عملی زیر، نحوه تعریف و استفاده از یک مقدار ثابت یا Constant را در PHP نشان می دهد:

```
<?php
<!--?php
define("GREETING", "Welcome to W3Schools.com!");
echo GREETING;
?-->
?>
```

در کد مثال زیر، نیز یک Constant با نام حساس به حروف بزرگ یا کوچک (case sensitive) ایجاد کرده ایم:

```
<?php
<!--?php
define("GREETING", "Welcome to W3Schools.com!", true);
echo greeting;
?-->
?>
```

Constant ها در PHP عمومی یا Global هستند:

در زبان PHP به صورت پیش فرض، مقادیر ثابت یا Constant ها، در سراسر کد برنامه عمومی یا Global هستند. به این معنا که آن ها را می توان توسط تمامی توابع و کلاس ها برنامه فراخوانی و استفاده نمود .

در کد مثال زیر، نشان داده ایم چگونه می توان یک Constant را درون یک تابع تعریف نموده و از آن در خارج تابع استفاده کرد:

```
<?php
<!--?php
define("GREETING", "Welcome to W3Schools.com!");

function myTest() {
echo GREETING;
}

myTest();
?-->
?>
```

آموزش کار با عملگرها یا operators در زبان: PHP

از عملگرها یا operators در زبانPHP ، برای انجام عملیات های ریاضی یا مقایسه ای بر روی متغیرها و مقادیر استفاده می شود. در زبانPHP ، عملگرها به گروه های زیر تقسیم می شوند که در ادامه به تشریح کامل هر یک از آن ها خواهیم پرداخت:

- عملگرهای حسابی یاArithmetic
- عملگرهای انتصابی یاAssignment Operators
- عملگرهای مقایسه ای یاComparison Operators
- عملگرهای کاهنده یا افزایندهIncrement/Decrement
- عملگرهای متنی یاString Operators

آموزش کار با ساختارهای شرطی if else در PHP

از ساختارهای شرطی، برای انجام عملیات های متفاوت بر حسب شرایط مختلف، استفاده می شود. برای مثال اگر مثلا عدد a بزرگتر از ۵ بود، یک مقدار در خروجی چاپ شده و در صورت کوچکتر بودن آن از ۵، مقدار دیگری نمایش داده شود.

همانطور که گفتیم، شما می توانید از دستورات شرطی برای انجام کارهای خاصی، در صورت درست بودن یک

مقدار یا عدم اجرای آن ها، استفاده کنید.

در زبان PHPP دارای ساختارهای شرطی مختلفی هستیم که در لیست زیر معرفی شده اند:

- **ساختار شرطی : if statement** این ساختار شرطی در صورتی که شرط دستور i f درست باشد، کارهای تعیین شده برای آن را اجرا کرده و در غیر این صورت، کدی را اجرا نمی کند.
- **ساختار شرطی : if ... else statement** این ساختار شرطی در صورت درست بودن شرط دستور i f یک سری دستورات و در صورت درست نبودن آن، دستورات بخش el see را اجرا می کند.
- **ساختار : if ... else if ... else statement** در این حالت می توان ساختارهای شرطی چندگانه تعریف کرده که در صورت درست بودن شرط هر یک از بخش ها، دستورات آن بخش را اجرا می کند.
- **ساختار شرطی : switch** با استفاده از ساختار شرطیswi t ch ، می توان بر حسب مقادیر مختلف یک متغیر، دستورات خاصی را برای اجرا، تعیین کرد.

آموزش کار با ساختار شرطی: i f

ساختار شرطی i f ، در صورت درست بودن شرط تعیین شده، دستورات بدنه i f را اجرا می کند. ساختار کلی این دستور به شکل زیر است:

```
۱  if (شرط){
۲  کدهایی که در صورت درست بودن شرط اجرا می شوند
۳  }
```

در مثال عملی زیر، در صورتی که مقدار ساعت (HOUR) کمتر از ۲۰ باشد، عبارت "Have a good day!" را در خروجی نمایش می دهد:

```
<?php
<!--?php
$t = date("H");

if ($t < "20") {
echo "Have a good day!";
}
?-->
?>
```

آموزش ساختار شرطی if ... else statement

این ساختار شرطی، در صورت درست بودن شرط دستور if ، یک سری دستورات و در صورت درست نبودن شرط آن، دستورات بخش else را اجرا می کند. شکل کلی نوشتن یک ساختار شرطی if ... else به صورت زیر است:

```
<?php
if ( شرط ) {
کدهایی که در صورت درست بودن شرط اجرا می شوند
}
else
{
کدهایی که در صورت درست نبودن شرط اجرا می شوند
}
?>
```

مثال عملی : در کد مثال زیر، اگر مقدار ساعت (HOUR) کمتر از ۲۰ باشد، برنامه عبارت "Have a good day!" و در غیر اینصورت عبارت "Have a good ni ght" را در خروجی نشان می دهد:

```
<?php
<!--?php
$t = date("H");

if ($t < "20") {
echo "Have a good day!";
} else {
echo "Have a good night!";
}
?-->
?>
```

آموزش ساختار شرطی: if ... else if ... else statement

در ساختار شرطیif ... else if ... else ، کد های مختلف برای اجرا در بیش از ۲ حالت مختلف تعیین می شود. برنامه از دستور if اول شروع به چک کردن شرط ها می کند، شرط هر if یا else if درست باشد، دستور آن بخش اجرا شده و مابقی بدون اجرا رها می شود. اما اگر شرط هیچ کدام از بخش های if یا else if تعیین شده، درست نبود، در نهایت دستور بخش else اجرا می شود .

ساختار کلی تعریف یک دستور شرطی if ... else if ... else see به صورت زیر است:

```
<?php
if ( شرط 1) {
کدهایی که در صورت درست بودن شرط اجرا می شوند
}
elseif ( شرط 2 )
{
کدهایی که در صورت درست نبودن شرط 1و درست بودن شرط 2اجرا می شوند
}
```

```
else
{
کدهایی که در صورت درست هیچ یک از شرط های 1و 2اجرا می شوند
}
?>
```

کد مثال عملی زیر، اگر ساعت کمتر از ۱۰ باشد، عبارت "good morning" را در خروجی نمایش می دهد. اما اگر ساعت کمتر از ۲۰ باشد، عبارت "good day" و در غیر این صورت عبارت "good night" را نشان خواهد داد:

```
<?php
<!--?php
$t = date("H");

if ($t < "10") {
echo " good morning!";
} elseif ($t < "20") {
echo " good day!";
} else {
echo "good night!";
}
?-->
?>
```

آموزش کار با ساختار شرطی switch در PHP

ساختار شرطی switch را به طور مفصل در درس بعدی آموزش خواهیم داد.

آموزش کار با ساختار دستوری switch در: PHP

از ساختار دستوریswitch ، برای اجرای دستورات مختلف بر حسب مقادیر متفاوت یک متغیر یا حالت استفاده می شود.

به عبارت دیگر، از دستور switch می توان برای اجرای یکی از بلوک های دستور متفاوت از بین چندین بلوک دستور تعیین شده، بر حسب مقدار یک مقدار متغیر استفاده کرد.

ساختار کلی نوشتن یک دستور switch به صورت زیر است:

```
<?php
switch (n) {
case label1:
code to be executed if n=label1;
break;
case label2:
code to be executed if n=label2;
break;
case label3:
code to be executed if n=label3;
break;
...
default:
code to be executed if n is different from all labels;
```

```
}
?>
```

نحوه کار ساختار دستوری switch به صورت زیر است:

در ابتدای ساختار، یک عبارت یا expression داریم که در کد فوق با حرف n مشخص شده است. این عبارت معمولا یک متغیر بوده و در ابتدای اجرای ساختار یک بار سنجیده شده و مقدار آن در حافظه قرار می گیرد. سپس مقدار عبارت با مقدار هر یک از Case های ساختار مقایسه می شود. اگر مقدار عبارت با هر یک از مقادیر Caseها یکسان بود، دستورات آن Case اجرا می شود. از دستور break در پایان هرCase ، برای جلوگیری از ادامه اجرای سایر Case های ساختار استفاده می شود. اگر از break استفاده نکنیم، دستورات Case های بعد از Case ای که مقدار آن با متغیر ساختار برابر بوده است، نیز اجرا خواهد شد که معمولا نتیجه مورد نظر ما نیست.

در صورتی که مقدار عبارت n با مقدار هیچ یک از Case های ساختار switch برابر نباشد، دستورات بخش defaultاجرا خواهد شد.

مثال

در مثال زیر یک متغیر به نام $favcolor تعیین کرده ایم که مقدار اولیه red را خواهد داشت. سپس مقدار این متغیر با مقدارهای تعیین شده در هر Case مقایسه شده و دستور Case اول با مقدار red اجرا می شود. به علت به کار بردن عبارت break در پایان تمامی دستورات Case ها، دستورات سایر Case ها اجرا نخواهد شد.

```
<?php
<!--?php
$favcolor = "red";
switch ($favcolor) {
case "red":
echo "Your favorite color is red!";
break;
case "blue":
echo "Your favorite color is blue!";
break;
case "green":
echo "Your favorite color is green!";
break;
default:
echo "Your favorite color is neither red, blue, nor green!";
}
?-->
?>
```

آموزش کار با حلقه های while loop درPHP

حلقه while loop تا زمانی که شرط تعیین شده برای آن، درست باشد، مجموعه دستورات بدنه خود را به صورت پشت سر هم تکرار می کند.

در بعضی مواقع می خواهید یک دستور یا دستورات خاصی، برای چندین بار به صورت پشت سر هم اجرا شوند. به جای این که یک کد یکسان را چندین بار در یک اسکریپت بنویسید، می توانید آن را یک بار در یک حلقه نوشته و سپس با مکانیزم های کنترلی پیش بینی شده، تعداد دفعات اجرای آن را مشخص کنید.

در زبانPHP ، ۴ نوع حلقه تکرار داریم که عبارتند از:

- حلقه : while حلقهwhile ، مجموعه دستورات خود را تا زمانی که شرط تعیین شده برای آن درست باشد، اجرا می کند.
- حلقه : do... while حلقه do...while ، یک بار دستورات تعیین شده برای آن را اجرا کرده و سپس تا زمانی که شرط تعیین شده برای آن درست باشد، به اجرای کدها ادامه می دهد.
- حلقه : for حلقهfor ، مجموعه دستورات بدنه خود را به تعداد تعیین شده، تکرار می کند.
- حلقه : foreach حلقهforeach ، معمولا برای کار با ساختار مجموعه ای مثل آرایه ها استفاده شده و مجموعه دستورات خود را به اعضای هر یک از اعضای آرایه، یک بار اجرا می کند.

آموزش کار با حلقه while loop در PHP

حلقه while loop تا زمانی که شرط تعیین شده برای آن درست باشد، مجموعه دستورات خود را اجرا می کند.

ساختار کلی نوشتن یک حلقه while به صورت زیر است:

```
<?php
while (condition is true) {
    code to be executed;
}
?>
```

مثال

در مثال عملی زیر، برنامه ابتدا مقدار متغیر $x را برابر ۱ قرار می دهد. سپس تا زمانی که مقدار متغیر $x از ۵ کمتر یا مساوی آن باشد، دستورات حلقه while پشت سر هم اجرا می شود. به وسیله دستور++x ، هر بار که دستورات حلقه اجرا می شود، مقدار متغیر $x یک واحد افزایش می یابد:

```
<?php
<!--?php
$x = 1;
while($x <= 5) {
echo "The number is: $x -->br<";
$x++;
}
?>
?>
```

آموزش کار با حلقه do...while در PHP

حلقه do..while ، همیشه یک بار دستورات بدنه خود را اجرا کرده و سپس شرط تعیین شده برای آن را بررسی می کند. تا زمانی که این شرط درست باشد، مجددا دستورات حلقه اجرا می شود.

ساختار دستوری نوشتن یک حلقه do...while به صورت زیر است:

```
<?php
do {
code to be executed;
} while (condition is true);
?>
```

مثال

در کد مثال عملی زیر، ابتدا برنامه مقدار متغیر $x را برابر یک قرار می دهد سپس حلقه یک بار دستورات خود را بدون توجه به درستی یا عدم درستی شرط اجرا می کند و مقدار x را نیز یک واحد افزایش ی دهد. سپس شرط حلقه) این که $x از ۵ کوچکتر یا مساوی آن است (چک شده و تا زمانی که شرط فوق درست باشد، اجرای دستورات ادامه می یابد:

```
<?php
<!--?php
$x = 1;
do {
echo "The number is: $x < br -->";
$x++;
} while ($x <= 5);
?>
?>
```

نکته:

توجه داتشه باشید که شرط حلقه do…whi l e در انتهای حلقه و پس از اجرای کامل دستورات آن چک می شود. بنابراین حتی اگر شرط حلقه do…whi l e از ابتدا هم نادرست باشد، دستورات آن حداقل یک بار اجرا خواهند شد.

مثال

در کد مثال عملی زیر، برنامه ابتدا مقدار متغیر $x را بر روی ۶ تنظیم کرده و یک بار دستورات حلقه را اجرا می کند. سپس در انتهای اجرای حلقه، شرط آن را چک کرده و به دلیل غلط بودن شرط، از تکرار دستورات حلقه خودداری می شود:

```
<?php
<!--?php
$x = 6;
do {
echo "The number is: $x <br-->";
$x++;
} while ($x <= 5);
?>
?>
```

حلقه های for و foreach را به صورت جداگانه در درس بعدی آموزش خواهیم داد.

آموزش کار با تابع Function در PHP

قدرت اصلی زبان برنامه نویسی PHP ناشی از تابع ها یا function های آن می باشد. زبان PHP دارای بیش از ۱۰۰۰ تابع پیش ساخته برای انجام امور مختلف برنامه نویسی است. در کنار تابع های از پیش تعریف شده زبانPHP ، کاربر می تواند تابع های دلخواه خود را نیز تعریف کند. یک تابع، مجموعه یا بلوکی از دستورات است که می توان آن را چندین بار در سطح برنامه استفاده کرد. به عبارت دیگر، در یک تابع مجموعه کدهایی را تعریف کرده و هر زمان که نیاز داشتید، آن کدها اجرا شوند، فقط با استفاده از نام تابع، آن را فراخوانی کرده و مجموعه کدها را اجرا می کنید.

نکته:

توجه داشته باشید که توابع به محض لود شدن صفحات اجرا نخواهد شد، بلکه بایستی آن را فراخوانی یا در اصطلاح Call نمایید.

آموزش تعریف تابع های دلخواه کاربر در PHP

با استفاده از واژه کلیدی function و با ساختار زیر می توانید یک تابع دلخواه را در زبان PHP تعریف کنید.

```
<?php
function functionName() {
code to be executed;
}
?>
```

نکته:

نام یک تابع بایستی با یک حرف یا علامت (_) شروع شده و استفاده از اعداد در ابتدای نام ها مجاز نیست.

راهنمایی

نامی را برای تابع خود انتخاب کنید که منعکس کننده عملکرد آن بوده و بعدا بتوانید آن را راحت تر در کد تشخیص دهید.

۲ نکته:

نام تابع ها در زبانPHP ، حساس به حروف بزرگ یا کوچک (Case-sensetive) نیست.

مثال عملی : در کد مثال عملی زیر، یک تابع با نام writeMsg() ایجاد کرده ایم. کاراکتر ({) یا براکت باز ابتدای محدوده کدهای تابع و کاراکتر ({) یا براکت بسته، انتهای کدهای تابع را مشخص می کند. تابع writeMsg()عبارت "Hello world" را در خروجی چاپ کرده و برای اجرای آن کافی است تا با نوشتن نام تابع در کد، فراخوانی نشود:

```
<?php
<!--?php
function writeMsg() {
 echo "Hello world!";
}

writeMsg(); // call the function
?-->
?>
```

آموزش ارسال پارامتر به تابع در زبان PHP

می توانید اطلاعات مورد نظر خود را از طریق پارامترها یا Argument به تابع ارسال کنید. یک پارامتر یا Argumentکاملا شبیه متغیر عمل نمی کند.
آرگومان را بایستی در پرانتز جلوی نام تابع تعریف کنید. هر تعداد آرگومان یا پارامتر که نیاز داشته باشید، می توانید برای تابع تعیین نمایید، فقط بایستی آن ها را با کاراکتر کاما (،) از هم تفکیک نمایید.

مثال عملی : کد مثال عملی زیر دارای یک تابع تک آرگومان به نام $fname می باشد. هنگامی که تابع familyName()فراخوانی یا Call می شود، به وسیله آرگومان $fanme یک نام مثل Jani را به تابع ارسال کنیم که در بدنه دستورات آن استفاده خواهد شد. سپس تابع چندین نام و نام خانوادگی را با نام یکسان (Jani) و فامیلی های متفاوت در خروجی چاپ می کند:

```
<?php
<!--?php
function familyName($fname) {
echo "$fname Refsnes.<br-->";
}

familyName("Jani");
familyName("Hege");
familyName("Stale");
familyName("Kai Jim");
familyName("Borge");
?>
?>
```

مثال عملی ۲ : در کد مثال عملی زیر نیز یک تابع با دو پارامتر به نام های $year و $f name تعریف شده است. تا بع نام هر شخص با سال تولد آن را در خروجی چاپ می کند.

```
<?php
<!--?php
function familyName($fname, $year) {
echo "$fname Refsnes. Born in $year <br-->";
}

familyName("Hege", "1975");
familyName("Stale", "1978");
familyName("Kai Jim", "1983");
?>
?>
```

مقدار پیش فرض پارامترها در تابع PHP

کد مثال زیر، نحوه استفاده از یک تابع با مقدار پیش فرض برای پارامترهای آن را نشان می دهد. اگر تابع set Hei ght () را بدون آرگومان فراخوانی کنید (مثل فراخوانی دوم تابع در مثال)، پارامتر مقدار پیش فرض یا def aul t val ue تعیین شده برای آن را استفاده خواهد کرد:

```
<?php
<!--?php
function setHeight($minheight = 50) {
echo "The height is : $minheight <br-->";
}

setHeight(350);
setHeight(); // will use the default value of 50
setHeight(135);
setHeight(80);
?>
?>
```

آموزش مقدار برگشتی Return Value در تابع PHP

هر تابع درPHP ، می تواند پس از اجرای کامل کدهای خود مقداری را به عنوان مقدار بازگشتی یا Return Valueبه برنامه بازگرداند.

برای این منظور بایستی از دستور return در انتهای کد تابع استفاده کرد. همانند کد مثال عملی زیر که تابع sum()در انتها، مقدار متغیر $z را به عنوان مقدار بازگشتی بر می گرداند:

```
<?php
<!--?php
function sum($x, $y) {
$z = $x + $y;
return $z;
}

echo "5 + 10 = " . sum(5, 10) . "<br-->";
echo "7 + 13 = " . sum(7, 13) . "<br>";
echo "2 + 4 = " . sum(2, 4);
?>
?>
```

آموزش کار با آرایه یا Array در زبان: PHP

یک آرایه یا Array می تواند چندین مقدار را درون یک متغیر واحد نگهداری کند. همانند کد مثال زیر که در آن یک متغیر آرایه ای به نام $car تعریف نموده و سه مدل خود رو به عنوان اعضای آن مشخص شده اند. سپس به کمک اندیس یا شناسنامه هر عضو، نام آن را در خروجی نشان داده ایم:

```
<?php
<!--?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?-->
?>
```

یک آرایه چیست؟

یک آرایه یاArray ، نوع خاصی از متغیرهاست که می تواند در یک زمان واجد چندین مقدار را در خود جای دهد.

اگر شما یک لیستی از آیتم ها (مثل لیست نام ماشین ها) داشته باشید، نگهداری آن ها در متغیرهای تکی به صورت زیر خواهد بود

```
<?php
$cars1 = "Volvo";
$cars2 = "BMW";
```

```
$cars3 = "Toyota";
?>
```

حال اگر بخواهید بین نام ماشین ها حرکت کرده و نام یک خودرو خاص را پیدا کنید، بدتر این که به جای ۳ ماشین، ۳۰۰ ماشین در لیست خود داشته باشید؟!
راه حل، ایجاد یک آرایه یا Array است. یک آرایه می تواند تعداد زیادی مقدار یا val ue را با یک نام مشابه در خود ذخیره کند. سپس می توانید به مقدار هر متغیر، با استفاده از شماره اندیس (index) آن دسترسی داشته باشید.

آموزش تعریف آرایه Array در PHP

در زبان PHP از تابع ()array برای تعریف یک آرایه به صورت زیر استفاده می شود:

```
array();
```

در زبانPHP ، ۳ نوع آرایه Array داریم:

- آرایه اندیس دار : I ndexed Arrays آرایه ای که در آن هر عضو یا یک اندیس i ndex مشخص می شود.
- آرایه رابطه ای یا : Associative Arrays در این نوع آرایه هر کلید دارای یک نام است.
- آرایه های چند بعدی : (Multidimensianal Arrays) در این نوع آرایه دو یا چند آرایه به صورت تو در تو و چند بعدی تعریف شده است.

آموزش کار با آرایه های اندیس وار : Indexed Arrays دو راه برای تعریف آرایه های اندیس وار یا Indexed Arraysوجود دارد:

اندیس یا i ndex را می توان به صورت خودکار به هر عضو آرایه نسبت داد. در این حالت اندیس آرایه از صفر شروع می شود، همانند کد زیر:

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

یا می توانید اندیس یا i ndex را به صورت دستی در هنگام تعریف اعضای آرایه تعیین کنید. همانند کد زیر:

```
<?php
$cars[0] = "Volvo";
$cars[1] = "BMW";
$cars[2] = "Toyota";
?>
```

مثال عملی:

در کد مثال زیر یک آرایه به نام $cars را تعریف نموده و سه عضو برایش تعیین کرده ایم. سپس برنامه یک عبارت متنی جلوی مقدار هر عضو آرایه را در خروجی نشان می دهد:

```
<?php
<!--?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?-->
?>
```

آموزش بدست آمدن طول آرایه با استفاده از تابع Count()

به وسیله تابع Count () در PHP می توان طول یک آرایه یا به عبارت دیگر تعداد اعضای آن را به دست آورد. در مثال عملی زیر کد انجام کار را نشان داده ایم:

```
<?php
<!--?php
$cars = array("Volvo", "BMW", "Toyota");
echo count($cars);
?-->
?>
```

حرکت درون یک آرایه اندیس وار:

به وسیله یک حلقه مثل for loop می توان به راحتی درون یک آرایه حرکت نموده و به مقایسه اعضای آن دسترسی داشت.

در کد مثال عملی زیر، یک متغیر به نام $cars را تعریف کرده و سپس تعداد اعضای آن را شمرده و در متغیر $arrlength ریخته ایم. سپس از این متغیر به عنوان شمارنده حلقه استفاده نموده و با حرکت درون آن، مقدار تمامی اعضای حلقه را در خروجی چاپ کرده ایم:

```
<?php
<!--?php
$cars = array("Volvo", "BMW", "Toyota");
$arrlength = count($cars);

for($x = 0; $x < $arrlength; $x++) {
echo $cars[$x];
echo "<br-->";
}
?>
?>
```

آموزش کار با حلقه های رابطه ای یا: Associative Arrays

آرایه های رابطه ای یا Associative Arrays آرایه هایی هستند که در آن هر عضو یک نام منحصر به فرد و یک مقدار دارد. دو راه کلی برای تعریف آرایه های رابطه ای به شکل های زیر وجود دارد:

```
<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
?>
```

یا

```
<?php
$age['Peter'] = "35";
$age['Ben'] = "37";
$age['Joe'] = "43";
?>
```

سپس می توان این آرایه های رابطه ای را در یک اسکریپت به صورت زیر به کار برد:

```
<?php
<!--?php
$age = array("Peter"-->"35", "Ben"=>"37", "Joe"=>"43");
echo "Peter is " . $age['Peter'] . " years old.";
?>
?>
```

حرکت درون آرایه های رابطه ای

برای حرکت درون یک آرایه رابطه ای و دسترسی به اعضای مختلف آن، بایستی از یک حلقه foreach همانند

مثال زیر استفاده کنید:

```
<?php
<!--?php
$age = array("Peter"-->"35", "Ben"=>"37", "Joe"=>"43");

foreach($age as $x => $x_value) {
echo "Key=" . $x . ", Value=" . $x_value;
echo "<br>";
}
?>
?>
```

آرایه های چند بعدی Multidimensionol Arrays

آرایه های چند بعدی را در یک درس مجزا به طور کامل آموزش خواهیم داد.

آموزش مرتب سازی آرایه ها (Sorting Array) درPHP

المنت های یک آرایه را می توان بر حسب حروف الفبا یا اعداد، به ترتیب نزولی یا صعودی مرتب کرد.

در این درس قصد داریم تا نحوه کار با توابع مرتب کننده اعضای آرایه ها در PHP را به شما آموزش دهیم،

این تابع ها عبارتند از:

- تابع() Sort ، اعضای آرایه را به ترتیب صعودی مرتب می کند.
- تابع() rsort ، اعضای آرایه را به ترتیب نزولی مرتب می کند.

- تابع assort() ، آرایه های رابطه ای (associative arrays) را در جهت صعودی بر حسب مقدار یا valueآن ها مرتب می کند.
- تابع ksort() ، آرایه های رابطه ای یا associative arrays را در جهت صعودی بر حسب مقدار کلید یا key آن ها مرتب می کند.
- تابع arsort() ، آرایه های رابطه ای را در جهت نزولی بر حسب مقدار یا value آن ها مرتب می کند.
- تابع krsort() ، آرایه های رابطه ای را در جهت نزولی و بر حسب مقدار کلید یا key آن ها مرتب می کند.

آموزش کار با تابع sort() در PHP

در کد مثال عملی زیر، اعضای آرایه $Cars را به وسیله تابع sort() بر حسب حروف الفبا و در جهت صعودی مرتب کرده ایم:

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

مثال عملی ۲:

در کد مثال عملی زیر نیز، اعضای آرایه $numbers را بر حسب عدد در جهت صعودی مرتب کرده ایم:

```
<?php
<!--?php
$numbers = array(4, 6, 2, 22, 11);
sort($numbers);
?-->
?>
```

آموزش کار با تابع rsort() در PHP

مثال عملی:

در مثال زیر با استفاده از تابع rsort() ، اعضای آرایه $Cars را بر حسب حروف الفبا و به صورت نزولی مرتب کرده ایم:

```
<?php
<!--?php
$cars = array("Volvo", "BMW", "Toyota");
rsort($cars);
```

```
?-->
?>
```

مثال ۲:

در کد مثال عملی زیر نیز با استفاده از تابع rsort() ، اعضای آرایه $cars را بر حسب اعداد و به صورت نزولی یا decending مرتب کرده ایم:

```
<?php
<!--?php
$numbers = array(4, 6, 2, 22, 11);
rsort($numbers);
?-->
?>
```

آموزش کار با تابع asort() در PHP

همانطور که در مقدمه هم اشاره کردیم، تابع assort اعضای یک آرایه رابطه ای را بر حسب مقدار یا value آن ها به صورت صعودی مرتب می کند.

در کد مثال عملی زیر، اعضای آرایه رابطه ای $age را با استفاده از تابع assort() به صورت صعودی و بر حسب مقدار یا value آن ها مرتب کرده ایم:

```
<?php
<!--?php
$age = array("Peter"-->"35", "Ben"=>"37", "Joe"=>"43");
asort($age);
?>
?>
```

آموزش کار با تابع ksort() در PHP

تابع ksort() ، اعضای یک آرایه رابطه ای یا associative array را بر حسب مقدار کلید key آن ها به صورت صعودی مرتب می کند. در کد مثال عملی زیر، اعضای آرایه رابطه ای $age را بر حسب مقدار کلید یا key آن ها به صورت صعودی مرتب کرده ایم:

```
<?php
<!--?php
$age = array("Peter"-->"35", "Ben"=>"37", "Joe"=>"43");
ksort($age);
?>
?>
```

آموزش کار با تابع arsort() در PHP

تابعassort() ، اعضای یک آرایه رابطه ای را بر حسب مقدار یا value آن ها به صورت نزولی مرتب می کند.

در کد مثال عملی زیر، اعضای آرایه $age را با استفاده از تابع arsort() و بر حسب مقدار یا value آن ها به صورت نزولی مرتب کرده ایم:

```
<?php
<!--?php
$age = array("Peter"-->"35", "Ben"=>"37", "Joe"=>"43");
arsort($age);
?>
?>
```

آموزش کار با تابع krsort() در: PHP

تابع krsort() می تواند اعضای یک آرایه رابطه ای را بر حسب کلید یا age آن ها به صورت نزولی مرتب کند.

در کد مثال زیر، اعضای آرایه $age را با استفاده از تابع jrsort() و به صورت نزولی بر حسب مقدار کلید یا keyآن ها مرتب کرده ایم:

```
<?php
<!--?php
$age = array("Peter"-->"35", "Ben"=>"37", "Joe"=>"43");
krsort($age);
?>
?>
```

آموزش کار با متغیر سراسری یا Global variables در: PHP

متغیرهای سراسری ویژه یا Superglobals variables از زبان PHP 4.1.0 به بعد معرفی شده و شامل متغیرهای درونی ساخته ای می شوند که برای تمام پروژه و کد قابل دسترس و تغییر هستند. به عبارت دیگر تمامی تابع ها، کلاس ها و متدهای پروژه به متغیرهای سراسری ویژه یا Superglobal variablesدسترسی دارند. برای این منظور هم نیاز به نوشتن کد خاص یا انجام کاری نخواهید داشت.

متغیرهای سراسری ویژه در PHP عبارتند از:

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

- $_COOKIE
- $_SESSION

در این درس به آموزش برخی از متغیرهای سراسری ویژه Superglobal variables خواهیم پرداخت و مابقی آن ها را در قسمت های دیگر تشریح می کنیم.

آموزش کار با متغیر سراسری $GLOBAL در PHP

متغیر $GLOBAL یک متغیر سراسری ویژه یا Supergl obal vari abl e است که برای دسترسی و مدیریت متغیرهای سراسری gl obal vari abl e در هر کجای کد برنامه) PHP حتی درون تابع ها یا متدها) استفاده می شود. زبانPHP ، کلیه متغیرهای سراسری gl obal vari abl e را در یک آرایه با نام $GLOBAL [index] نگهداری می کند که index نام متغیر در سطح برنامه است.

مثال عملی:

در کد مثال عملی زیر نحوه استفاده از متغیر سراسری ویژه superglobal variable $GLOBALS را در کد PHPنشان داده ایم:

```
<?php
<!--?php
$x = 75;
$y = 25;

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

addition();
echo $z;
?-->
?>
```

در کد مثال فوق، از آنجایی که متغیر Z توسط آرایه $GLOBAL تعریف شده است، می توان از خارج از تابع نیز به آن دسترسی داشت.

آموزش کار با متغیر سراسری $SERVER در PHP

متغیر سراسری $SERVER یک متغیر سراسری ویژه یا Superglobal Variable است که اطلاعات مربوط به هدرها(headers) ، مسیرها (paths) و مکان اسکریپت ها را در خود نگهداری می کند.

در کد مثال عملی زیر، نحوه استفاده از برخی المنت های متغیر $SERVER را آموزش داده ایم:

```
<!--?php
echo $_SERVER['PHP_SELF'];
```

```
echo "<br-->";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>
```

لیست زیر مهم ترین المنت های متغیر $SERVER را که می توانید از آن ها استفاده کنید، به همراه توضیح نشان داده است:

- $ SERVER [PHP-SELF]این المنت نام فایلی که هم اکنون اسکریپت آن در حال اجراست را بر می گرداند.
- $_SERVER['GATEWAY_INTERFACE']این المنت ورژن Common Gate way Interface (CGI)که هم اکنون سرور در حال استفاده از آن است را برمی گرداند.
- $_SERVER['SERVER_ADDR']این المنت آدرس هاست سرور جاری را بر می گرداند.
- $_SERVER['SERVER_NAME']این المنت نام هاست سرور جاری برای مثال (www.tahlildadeh.com)را بر می گرداند.
- $_SERVER['SERVER_SOFTWARE']این المنت مقدار متنی کد شناسایی سرور یا server identhfication stringرا مثل Apache/2.2.24 را بر می گرداند.
- $_SERVER['SERVER_PROTOCOL']این المنت ورژن و نام پروتکل اطلاعات سرور یا information protocolرا بر می گرداند، مثل.HTTP/1.1
- $_SERVER['REQUEST_METHOD']این المنت روش درخواست صفحه جاری یا Request modeرا بر می گرداند مثل.Post
- $_SERVER['REQUEST_TIME']این المنت timestawp شروع درخواست اجرای صفحه را بر می گرداند، برای مثال 1377687496.

- $_SERVER['QUERY_STRING']این المنت در صورتی که صفحه از query string استفاده کرده باشد، مقدار آن را بر می گرداند.
- $_SERVER['HTTP_ACCEPT']این المنت مقدار Accept header مربوط به درخواست یا requestجاری را بر می گرداند.
- $_SERVER['HTTP_ACCEPT_CHARSET']این المنت مقدار Accept_Charset header مربوط به درخواست جاری را بر می گرداند.
- $_SERVER['HTTPS']این المنت مقدار آدرس کامل URL صفحه جاری را بر می گرداند (البته چندان کاربرد ندارد زیرا بر اثر هاست ها از آن پشتیبانی نمی کنند).
- $_SERVER['HTTP_REFERER']این المنت مقدار secure HTTP protocol ای که در سرور در حال استفاده است را بر می گرداند.
- $_SERVER['REMOTE_ADDR']این المنت آدرس IP address سروری که کاربر در حال مشاهده صفحه توسط آن است را بر می گرداند.
- $_SERVER['REMOTE_HOST']این المنت نام هاست سروری که (Host server) کاربر به وسیله آن در حال مشاهده صفحه است را بر می گرداند.
- $_SERVER['REMOTE_PORT']این المنت مقدار port ای که کامپیوتر کاربر برای ارتباط با وب سرور از آن استفاده می کند را بر می گرداند.
- $_SERVER['SCRIPT_FILENAME']این المنت آدرس کامل Pathname که اسکریپت جاری بر روی آن در حال اجراست را بر می گرداند.
- $_SERVER['SERVER_PORT']این المنت شماره port ای که وب سرور برای ارتباط با کلانیت از آن استفاده می کند را بر می گرداند مثل 80.
- $_SERVER['SERVER_SIGNATURE']این المنت ورژن سرور (server version) و نام هاست مجازی (virtual host name) ای که به صفحات تولید شده توسط سرور اضافه شده اند را بر می گرداند.

- $_SERVER['PATH_TRANSLATED']این المنت مقدار آدرس مبتنی بر فایل سیستم file system based pathمربوط به اسکریپت جاری را بر می گرداند.
- $_SERVER['SCRIPT_NAME']'این المنت آدرس یا path اسکریپت جاری را بر می گرداند.
- $_SERVER['SCRIPT_URI']این المنت مقدار URI مربوط به صفحه جاری را بر می گرداند.
- $_SERVER['SERVER_ADMIN']این المنت مقدار value داده شده به SERVER_ADMINdirective در فایل تنظیمات برنامه یا Configuration file را بر می گرداند) اگر هاست شما به صورت مجازی Virtual hostاجرا می شود، این مقدار value تعیین شده برای سرور مجازی خواهد بود مثل some@tahlildadeh.com)

آموزش کار با متغیر سراسری $_REQUEST در PHP

از متغیر سراسری $_REQUEST درPHP ، برای جمع آوری اطلاعات پس از Submit یک فرم HTML استفاده می شود.

در کد مثال عملی زیر یک FORM همراه با یک کادر ورودی input field و یک دکمه submit button نشان داده شده است. هنگامی که کاربر با کلیک بر روی دکمه "submit" اطلاعات فرم را ارسال یا در اصطلاح submit می کند، این data به آدرس فایلی که در خاصیت action تگ تعیین شده است، ارسال می شود. در این مثال، ما اطلاعات فرم را برای پردازش به خود صفحه جاری ارسال کرده ایم. اگر شما می خواهید این اطلاعات به صفحه یا آدرس دیگری برود، بایستی آدرس مورد نظر خود را به جای فایل جاری در خاصیت actionبنویسید.

سپس می توانیم از متغیر سراسری ویژه $_REQUEST برای جمع آوری اطلاعات یا value کادر متنی input filedاستفاده کنیم:

```
Name: <input type="text" name="fname">
<input type="submit">


<!--?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
// collect value of input field
$name = $_REQUEST['fname'];
if (empty($name)) {
echo "Name is empty";
} else {
echo $name;
}
```

```
}
?-->
```

آموزش کار با متغیر سراسری $_POST در PHP

از متغیر سراسری $_POST برای جمع آوری اطلاعات یک فرم HTML پس از ارسال یا در اصطلاح submit آن به روش method="POST" استفاده می شود. همچنین از متغیر $_POST می توان برای ارسال متغیرها یا variables نیز استفاده کرد.

مثال:

در کد مثال عملی زیر، یک form با یک کادر ورود اطلاعات (input field) و یک دکمه ارسال (submit button) داریم. هنگامی که کاربر بر روی دکمه "submit" کلیک می کند، اطلاعات فرم به فایل تعیین شده در خاصیت action تگ

ارسال می شود. در این مثال، برای پردازش سریع اطلاعات، صفحه را به خودش ارسال کرده ایم. اما اگر شما تمایل دارید اطلاعات را به صفحه PHP دیگری بفرستید، کافی است آدرس و نام آن را در خاصیت action جایگزین کنید. در پایان کد، از متغیر سراسری $_POST برای جمع آورس مقادیر (value) کادر متن فرم استفاده می کنیم:

```
<?php
Name: <input type="text" name="fname">
<input type="submit">


<!--?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
// collect value of input field
$name = $_POST['fname'];
if (empty($name)) {
echo "Name is empty";
} else {
echo $name;
}
}
?-->
?>
```

آموزش کار با متغیر سراسری $_GET در: PHP

از متغیر سراسری $_GET نیز می توان برای جمع آوری اطلاعات یک فرم پس از Submit و ارسال آن به مقصد استفاده کرد. این متغیر همچنین می تواند اطلاعات فرستاده شده توسط URL را نیز جمع آورس کند. در نظر بگیرید که یک صفحه HTML حاوی یک لینک با چند پارامتر مثل کد زیر داشته باشیم:

```
<?php
<a href="test_get.php?subject=PHP&web=W3schools.com">Test $GET</a>
```

```
?>
```

هنگامی که کاربر بر روی لینک "Test $GET" کلیک می کند، پارامترهای "subject" و "web" به صفحه "test get.php" ارسال می شوند. سپس در صفحه "test_get.php" می توانید به این مقادیر یا values از طریق متغیر $_GET دسترسی داشته باشید.

کد مثال زیر، اسکریپت مربوط به صفحه "test-get.php" را نشان می دهد:

```
<?php
<!--?php
echo "Study " . $_GET['subject'] . " at " . $_GET['web'];
?-->
?>
```

نکته:

درباره متغیرهای $_POST و $_GET در بخش آموزش فرم های PHP بیشتر صحبت خواهیم کرد.

بخش دوم : آموزش مدیریت فرم (Form) درPHP

آموزش مدیریت فرم ها Form در PHP

متغیرهای سراسری ویژه superglobal $_POST variables و $_GET برای جمع آوری اطلاعات فرم ها در زبان PHP به کار می روند.

کد مثال عملی زیر، یک فرم ساده HTML Form با دو کادر ورود اطلاعات (input fields) و یک دکمه ارسال اطلاعات (submit button) را نشان می دهد. به ساختار کلی آن نگاهی می اندازیم:

```
<?php
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
?>
```

هنگامی که کاربر اطلاعات لازم را در کادرهای متنی فرم فوق وارد نموده و دکمه "submit" را برای ارسال اطلاعات کلیک می کند با اطلاعات فرم به فایل "welcome.php" که در خاصیت action تگ فرم

Top of Form

تعیین شده است، ارسال می شوند. از روش HTTP POST برای ارسال اطلاعات در مثال فوق استفاده شده است.

برای نمایش اطلاعات ارسال شده، می توانید به صورت زیر از دستور echo استفاده نمایید. کد فایل "welcome.php" به صورت زیر می باشد:

```
<?php
Welcome <!--?php echo $_POST["name"]; ?--><br>
Your email address is: <!--?php echo $_POST["email"]; ?-->
```

```
?>
```

و خروجی صفحه فوق هم به صورت زیر خواهد بود:

```
<?php
Welcome John
Your email address is john.doe@example.com
?>
```

با به کار بردن روش HTTP GET برای ارسال اطلاعات، همانند کد مثال زیر، نتیجه مشابهی حاصل خواهد شد:

```
<?php
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
?>
```

البته کد فایل "wel come.php" در حالت دوم به صورت زیر است:

```
<?php
Welcome <!--?php echo $_GET["name"]; ?--><br>
Your email address is: <!--?php echo $_GET["email"]; ?-->
?>
```

کد هر دو مثال فوق، نسبتا ساده هستند. اما در اصل، مهم ترین چیز را فراموش کرده ایم! شما بایستی اطلاعات فرم را قبل از ارسال تا اعتبارسنجی یا val i date کنید تا از ارسال کدهای مخرب به سرور جلوگیری نمایید.

راهنمایی مهم

در هنگام پردازش اطلاعات فرم های PHP به امنیت یا security اطلاعات بسیار دقت کنید.

در مثال های این درس، هیچ اعتبار سنجی یا val i dati on ی بر روی اطلاعات صورت نمی گیرد. این کدها فقط برای نشان دادن نحوه ارسال و دریافت اطلاعات فرم ها طراحی شده اند. اما در درس های بعدی، نحوه اعتبار سنجی و رعایت ا صول امنیتی را در هنگام پردازش فرم های PHP آموزش خواهیم داد. اعتبارسنجی یا val i dati on صحیح اطلاعات برای محافظت سایت شما از رخنه و نفوذ هکرها و اسپمرها نکته ای حیاتی است.

مقایسه روش های GET و POST در ارسال اطلاعات:

هر دو روش GET و POST در هنگام ارسال اطلاعات یک آرایه به صورت

(array key=>val ue,key۲=>val ue۲,key۳=>val ue۳)

ایجاد می کنند. این آرایه جفت های مقدار/نام یا key/val ue را نگهداری می کنند که در هر دو، کلیدها یا keysنام کنترل های موجود در فرم و مقادیر یا val ues اطلاعات وارد شده توسط کاربر در آن کنترل ها می باشند.

هر دو روش GET و POST از متغیرهای سراسری $_GET و $_POST برای ارسال نگهداری اطلاعات استفاده می کنند. به عبارت دیگر، این متغیرها supergl obal vari abl es بوده و به این معناست که همواره و از هر جای کد اسکریپت قابل دسترس هستند. شما می توانید به مقادیر این متغیرها در هر تابع، کلاس یا متدی بدون نیاز به انجام کار خاصی و بدون توجه به scope یا میدان دید در کل کد صفحه دسترسی داشته باشید.

آرایه متغیر $_GET از طریق آدرس یا URL صفحه به مقصد ارسال می شوند، در حالی که آرایه متغیر $_POST به روش POST و در پشت پرده اسکریپت انتقال می یابد.

چه زمانی از روش GET استفاده کنیم؟

اطلاعاتی که به و سیله متد GET ار سال می شوند، برای همگان قابل رویت هستند) زیرا نام متغیرها و مقدار آن ها یا val ues در آدرس مرورگر URL صفحه نشان داده می شود .(همچنین روش GET محدود شدیدی برای حجم اطلاعات ارسالی دارد. حداکثر اطلاعات قابل ارسال توسط این متد ۲۰۰۰ کاراکتر است. روش GET در برخی مواقع کاربرد مناسب دارد، مثل زمانی که بخواهید کاربر به راحتی بتواند صفحه را Bool mark کند، زیرا می توانید مقادیر مورد نظر خود را همراه با URL ارسال کنید. همچنین این روش برای ارسال اطلاعات غیر حساس به حروف بزرگ یا کوچک مناسب است.

نکته مهم:

اطلاعات مهم و حیاتی مثل رمز عبور، نام کاربری و ... را نبایستی با روش GET ارسال کرد.

چه زمانی از روش POST استفاده کنیم؟

اطلاعاتی که به وسیله روش POST ارسال می شوند) به دلیل این که مقادیر جفت های نام/مقدار به بدنه درخواست HTTP فرم الحاق می شوند(، در پشت پرده صفحه انتقال یافته و از دید سایرین مخفی هستند. همچنین محدودیتی در حجم اطلاعاتی به وسیله متد) POST بر خلاف متد (GET وجود ندارد. از طرف دیگر، متد POST از قابلیت های پیشرفته ای مثل امکان ارسال چند تیکه اطلاعات یا multi –port bai nary در هنگام آپلود فایل ها به سرور پشتیبانی می کند. ولی به دلیل این که مقدار متغیرهای اطلاعات در آدرس یا URL صفحه قرار نمی گیرند، روش POST برای Boomark صفحات مناسب نیست.

راهنمایی:

اکثر برنامه نویسان از روش POST برای ارسال اطلاعات فرم ها استفاده می کنند.

در درس بعدی، به آموزش روش اعتبارسنجی و ارسال امن اطلاعات فرم های وب در PHP خواهیم پرداخت.

آموزش اعتبارسنجی فرم ها در زبان: PHP

در این درس و درس بعدی قصد داریم تا نحوه اعتبارسنجی و کنترل اطلاعات وارد شده به وسیله کاربر در فرم های وب را توسط PHP آموزش دهیم.

راهنمایی:

در هنگام پردازش اطلاعات فرم های وب به بحث امنیت اطلاعات یا security بسیار جدی فکر کنید.

در این درس با تفکر کنترل و اعتبارسنجی اطلاعات ورودی توسط کاربر در فرم های وب کار خواهیم کرد.

اعتبارسنجی درست اطلاعات وارد شده در فرم ها، بسیار مهم بوده و شما را از حمله هکرها و اسپمرها در امان نگه می دارد.

فرم HTML ای که در این درس ها از آن استفاده خواهیم کرد، به صورت زیر بوده و شامل چندین فیلد ورود اطلاعات مختلف می باشد. این فیلدها عبارتند از کادرهای متنی (text fields) اختیاری و اجباری، و کدهای

انتخابی رادیویی (radio buttons) و یک دکمه ارسال اطلاعات (submit button) که خروجی آن بر روی صفحه به صورت زیر است:

Name: *

E-mail: *

Website:

Comment:

Gender: Female Male *

Submit

قوانین مربوط به اعتبارسنجی هر یک از کنترل های فرم فوق به صورت زیر است:

- فیلد Name : این فیلد از نوع اجباری یا Requi red بوده و کاربر بایستی حتما آن را پر کند. علاوه بر این کاربر در این فیلد مجاز است فقط کاراکترهای حروفی و فاصله وارد نماید.
- فیلدE-mail :این کادر اجباری بوده و بایستی شامل یک آدرس معتبر ایمیل با فرمت صحیح (با کاراکترهای @ و ۰) باشد.
- فیلد: website :این کادر از نوع اختیاری (opti onal) بوده و کاربر می تواند آن را پر کرده یا خالی رها کند. این کادر متن هم بایستی شامل یک URL با فرمت صحیح باشد.
- فیلد Comment :این فیلد نیز اختیاری بوده و به صورت کادر متن چند خطی (MnHi –l i ne i nput) یا t ext a r e a می باشد.
- فیلد Gender :این فیلد از نوع دکمه های رادیویی یا Radi o But t on بوده و کاربر بایستی حتما یکی از دو مورد را انتخاب کند، به عبارت دیگر اجباری است.

ابتدا بیایید نگاهی به سورس کد خاص HTML فرم بیندازیم.

بررسی کادرهای متن فرم یا Text Fields

کادر متن هایname، emailو website از نوع فیلد متن معمولی یا t ext i nput بوده و کادر متن Commentبه صورت چند خطی یا textarea می باشد. کد HTML این کنترل ها به صورت زیر است:

```
<?php
Name: <input type="text" name="name">
E-mail: <input type="text" name="email">
Website: <input type="text" name="website">
```

```
Comment: <textarea name="comment" rows="5" cols="40"></textarea>
?>
```

بررسی دکمه های انتخابی یا: Radio Buttons

فیلد Gender از نوع دکمه های رادیویی بوده و کد HTML آن به صورت زیر است:

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

بررسی المنت فرم یا Form

کد HTML فرم ورود اطلاعات به صورت زیر است:

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

هنگامی که فرم ارسال یا در اصطلاح submit می شود، اطلاعات آن به وسیله روش POST با تعیین خاصیت (method = "POST")به سرور ارسال می شوند.

سوال:

متغیر سراسری ویژه $_SERVER["PHP_SELF] یک متغیر superglobal variable است که نام فایل اجراکننده اسکریپت جاری را بر می گرداند. بنابراین، متغیر $_SERVER["PHP_SELF] اطلاعات فرم را به خود صفحه جاری ارسال می کند تا این که به صفحه دیگری پرش نماید.در این حالت، کاربر خطاهای احتمالی یا error message های فرم را در همان صفحه ای که فرم قرار دارد، مشاهده خواهد کرد.

سوال:

تابع htmlspecialchars() function چیست و چه کاربردی دارد؟

تابع htmlspecialchars ، کاراکترهای ویژه را به موجودیت های HTML تبدیل می کند. این کار بدین معناست که کاراکترهایی مثل <و >به صورت $h; و $gt درخواهند آمد. این تبدیل از جمله های تزریق کد درون فرم ها یا injecting HTML جلوگیری کرده و مانع از تخریب صفحه می شوند.

نکته مهم در مورد امنیت فرم در: PHP

متغیر $_SERVER["PHP_SELF"] می تواند توسط هکرها استفاده شود.

اگر متغیر PHP_SELF در صفحه شما استفاده شود، کاربر می تواند یک اسلش یا (/) را در ابتدای یک کادر متن وارد نموده و سپس دستورات ؟؟؟؟؟؟؟ مخرب Cross site scripting یا XSS را جهت اجرا، وارد نماید.

آموزش : حملات Cross –site scripting یا XSS یک نوع آسیب پذیری امنیتی کامپیوتر است که به طور معمول در برنامه های تحت وب وجود دارد XSS .به حمله کننده ها این امکان را می دهد تا اسکریت های کلانیت ساید را به درون صفحات وب ای که توسط سایر کاربران نیز دیده می شود، تزریق کنند.

فرض کنید که تگ

در فایل وب "test_formphp" به صورت زیر باشد:

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

حال اگر کاربر آدرس معمولی صفحه را به صورت www.example.com/test_form.php را در URL وارد نماید، کد تگ به صورت زیر تبدیل خواهد شدBottom of Form:

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

کد حالت مناسب و مورد نظر ماست.

حال اگر کاربر، آدرس زیر را در URL صفحه وارد نماید:

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

در این حالت، کد فوق به صورت زیر تبدیل خواهد شد:

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

آموزش تعیین کادر های متن اجباری در فرم هایPHP

در این درس قصد داریم تا نحوه اجباری کردن ورود اطلاعات در کادر های متن در فرم های PHP و همچنین نحوه صدور پیام هشدار مناسب در صورت بروز خطا را به آن ها آموزش دهیم. فرض کنید یک فرم HTML در صفحه PHP خود دارید که می خواهد کاربر حتما در کادرهای متن نام کاربری و آدرس ایمیل مقداری وارد کرده و نتواند آن ها را خالی کند. به این نوع کادرهای متن در اصطلاح Requierd Fieldsگفته و در این درس با نحوه کار آن ها آشنا خواهید شد.

مثال عملی کادر متن اجباری درPHP

همان طور که در جدول قوانین مربوط به اعتبارسنجی کنترل های فرم مثال ها در درس قبلی مشاهده کردید کادرهای متن "name" ، "E-mail" و "Gender" کادرهای متن اجباری بودند. این field ها را نمی توان خالی کرد و حتما بایستی برای submitو ارسال فرم مقدار مناسب در آن ها قرار گیرد. جدول قوانین مربوط به اعتبارسنجی کادرهای متن مثال به صوت زیر است:

- Nameاین کادر متن اجباری (Required) بوده و فقط می تواند شامل کاراکتر و فاصله باشد.
- mailاین کادر متن نیز اجباری بوده و بایتی شامل یک ایمیل با فرمت مناسب) با کاراکتر a و@ (باشد.
- این کادر متن اختیاری website: یا optional بوده و درصورت وارد کردن اطلاعات بایستی حاوی یک URL صحیح باشد.
- commentاین نیز کادر متن اختیاری بوده و به صورت چندخطی (Multi-line)به صورت textarea می باشد.
- Genderاین کنترل نیز اجباری بوده و کاربر بایستی یکی از دو گزینه را انتخاب کند

در درس اول کلیه کادرهای متن فرم اختیاری یا بودندoptional

اما در مثال این درس ما چند متغیر جدید به نام های

"$emailErr"

"$genderErr"

"$websiteErr"

را به کد اضافه کردیم که پیام های هشدار مربوط به فیلدهای متن (error message) مربوط به فیلدهای متن اجباری را در خود نگهداری خواهد کرد.

ما همچنین یک دستور شرطی if را بر متغیر $_POST اضافه کرده ایم.این دستور if با کمک تابع empty() در PHP چک می کند آیا مقدار متغیر $_POST مورد نظر خالی است یا خیر.اگر خالی باشد یک پیام خطا در هر

متغیر مرتبط با فیلد ذخیره شده واگر هم خالی نباشد اطلاعات فرم را با کمک تابع test_input به مقصد ارسال می کند:

```
<!--?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
  if (empty($_POST["name"])) {
    $nameErr = "Name is required";
  } else {
    $name = test_input($_POST["name"]);
  }

  if (empty($_POST["email"])) {
    $emailErr = "Email is required";
  } else {
    $email = test_input($_POST["email"]);
  }

  if (empty($_POST["website"])) {
    $website = "";
  } else {
    $website = test_input($_POST["website"]);
  }

  if (empty($_POST["comment"])) {
    $comment = "";
  } else {
    $comment = test_input($_POST["comment"]);
  }

  if (empty($_POST["gender"])) {
    $genderErr = "Gender is required";
  } else {
    $gender = test_input($_POST["gender"]);
  }
}
?-->
```

آموزش نمایش پیام های هشدار در فرم:PHP

در مرحله بعدی یک اسکریپت را پس از هر کنترل کادر متن اجباری اضافه می کنیم تا در صورتی که کاربر بخواهد مقدار آن را خالی کند یک پیام هشدار مناسب تولید کند. نحوه انجام کار در کد زیر نشان داده شده است:

```
<?php
">

Name: <input type="text" name="name">
<span class="error">* <!--?php echo $nameErr;?--></span>
<br><br>
```

```
E-mail:
<input type="text" name="email">
<span class="error">* <!--?php echo $emailErr;?--></span>
<br><br>
Website:
<input type="text" name="website">
<span class="error"><!--?php echo $websiteErr;?--></span>
<br><br>
Comment: <textarea name="comment" rows="5" cols="40"></textarea>
<br><br>
Gender:
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male
<span class="error">* <!--?php echo $genderErr;?--></span>
<br><br>
<input type="submit" name="submit" value="Submit">
?>
```

در مرحله بعدی قصد داریم تا اطلاعات وارد شده کاربر را بر اساس مقدار درست و موردنظرمان اعتبرسنجی کنیم. برای مثال آیا کاربر در کادر متن "name"فقط کاراکترهای حرفی و فاصله وارد کرده و یا از کاراکترهای غیر مجاز مثل عدد هم استفاده کرده یا خیر.

یا در مثالی دیگر آیا کاربر یک آدرس ایمیل با فرمت صحیح در کادر متن" E-mail" وارد کرده یا خیر) آدرسی که دارای کاراکترهای a و ۰ با فرمت صحیح باش(

همچنین آیا کاربر در کادر متن "website " یا آدرس URL با فرمت درست وارد کرد یا خیر.

در درس بعدی به آموزش نحوه اعتبارسنجی اطلاعات وارد شده توسط کاربر در فرم های PHP خواهیم پرداخت.

آموزش اعتبارسنجی ایمیل و آدرس URL در فرم های PHP

دراین درس قصد داریم تا نحوه اعتبارسنجی مقادیر وارد شده توسط کاربر در کنترل های متنName ، Email وآدرس URL را آموزش دهیم.

آموزش اعتبارسنجی مقدار Name در فرم PHP

کد زیر، یک روش ساده جهت چک کردن این که آیا مقدار وارد شده برای Name فقط حاوی حروف (letters) و فاصله خالی (whitespace) است یا خیر را نشان می دهد. اگر مقدار وارد شده درست نباشد، یک پیام خطا در متغیر خطا مربوط به کادر متن ذخیره می شود:

```
<?php
$name = test_input($_POST["name"]);
if (!preg_match("/^[a-zA-Z ]*$/",$name)) {
  $nameErr = "Only letters and white space allowed";
```

```
}
?>
```

نکته:

تابع ()Prey_match مقدار متن وارد شده را با الگوی تعیین شده برای آن مقایسه کرده واگر pattern یا الگو رعایت شده باشد، مقدار true و در غیراین صورت مقدار false را بر می گرداند.

آموزش اعتبارسنجی مقدار Email در فرمPHP

راحت ترین و امن ترین راه برای چک کردن کردن این که آیا فرمت ایمیل وارد شده توسط کاربر دارای فرمت درست است یا خیر، استفاده از تابع **filter_var()** درPHP است.

در کد مثال زیر، اگرفرمت ایمیل وارد شده توسط کاربر درست نباشد، یک پیام خطا در متغیری خطا مربوط به **Email**ذخیره می شود:

```
<?php
$email = test_input($_POST["email"]);
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
$emailErr = "Invalid email format";
}
?>
```

آموزش اعتبارسنجی آدرس صفحه URL در PHP

کد زیر، یک روش ساده جهت چک کردن فرمت آدرس یا URL وارد شده توسط کاربر را نشان می دهد) الگو یا **Regulare expression** به کاررفته در این مثال، اجازه استفاده از اسلش / رابه کاربر می دهد .(اگر آدرس صفحه یا URL وارد شده توسط کاربر، دارای فرمت صحیح نباشد، یک پیام خطا درمتغیر مربوطه ذخیره خواهد شد:

```
<?php
$website = test_input($_POST["website"]);
if (!preg_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-
9+&@#\/%?=~_|!:,.;]*[-a-z0-9+&@#\/%=~_|]/i",$website)) {
$websiteErr = "Invalid URL";
}
?>
```

بررسی کد نهایی مثال و اعتبارسنجی کل اطلاعات:

در نهایت، کد اسکریپت مثال این درس بایستی به صورت زیر باشد:

```
<?php
<!--?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
// check if name only contains letters and whitespace
        if (!preg_match("/^[a-zA-Z ]*$/",$name)) {
$nameErr = "Only letters and white space allowed";
        }
    }

    if (empty($_POST["email"])) {
        $emailErr = "Email is required";
    } else {
        $email = test_input($_POST["email"]);
// check if e-mail address is well-formed
        if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
$emailErr = "Invalid email format";
        }
    }

    if (empty($_POST["website"])) {
        $website = "";
    } else {
        $website = test_input($_POST["website"]);
// check if URL address syntax is valid (this regular expression also
allows dashes in the URL)
        if (!preg_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-
9+&@#\/%?=~_|!:,.;]*[-a-z0-9+&@#\/%=~_|]/i",$website)) {
$websiteErr = "Invalid URL";
        }
    }

    if (empty($_POST["comment"])) {
        $comment = "";
    } else {
        $comment = test_input($_POST["comment"]);
    }

    if (empty($_POST["gender"])) {
        $genderErr = "Gender is required";
    } else {
        $gender = test_input($_POST["gender"]);
    }
}
?-->
?>
```

در درس بعدی، به آموزش نحوه جلوگیری از خالی شدن کادرهای متن فرم در هنگام ارسال اطلاعات یا submit form خواهیم پرداخت.

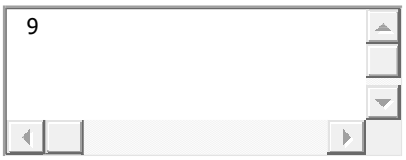
آموزش کامل مثال عملی کار با فرم ها در PHP

در این درس به مرور مجدد و کلی مثال کار با فرم در PHP که در چند درس گذشته با آن کار کردیم، خواهیم پرداخت. همچنین به آموزش نحوه نگه داشتن اطلاعات در کادرهای ورودی فرم (input fields) در هنگام کلیک کاربر بر روی دکمه ارسال یا submit فرم می پردازیم.

آموزش نگهداری اطلاعات (values) در فرم های PHP

برای نمایش مقادیر یا values در کادرهای ورود اطلاعات (input fields) در زمانی که کاربر بر روی دکمه ارسال فرم یا submit کلیک کند، یک تیکه کد اسکریپت PHP را درون خاصیت values کادرهای متن name ، email و website اضافه می کنیم.

در فیلد توضیح یا Comment که به صورت کادر متن چندخطی textarea است، کد اسکریپت PHP را بین تگ باز و بسته قرار می دهیم.



کدهایی که به تگ های فوق اضافه کردیم، مقدار یا value متغیرهای $name ، $email، $website و $comment را نمایش خواهند داد.

همچنین برای نمایش این که کدام گزینه از دکمه های رادیویی Radio Button مربوط به جنسیت کاربر یا Gender انتخاب شده است، بایستی با خاصیت cheched این کنترل کار کنیم دقت کنید که در این نوع کنترل مقدار value آن ها را کنترل نمی کنیم)

نحوه نمایش اطلاعات متغیرها در فرم مثال PHP در کد زیر نشان داده شده است

:

```
Name: <input type="text" name="name" value="<?php echo $name;?>">

E-mail: <input type="text" name="email" value="<?php echo $email;?>">

Website: <input type="text" name="website" value="<?php echo $website;?>">

Comment: <textarea name="comment" rows="5" cols="40"><?php echo
$comment;?></textarea>
```

```
Gender:
<input type="radio" name="gender" <?php="" if="" (isset($gender)="" &&=""
$gender="="female")" echo="" "checked";?="">
value="female">Female
<input type="radio" name="gender" <?php="" if="" (isset($gender)="" &&=""
$gender="="male")" echo="" "checked";?="">
value="male">Male
```

مثال کامل کار با فرم های وب در: PHP

خروجی مثال کار با فرم ها در PHP به صورت زیر خواهد بود؛ برای مشاهده کارکرد فرم، اقدام به ورود اطلاعات نمایید:

Name: *

E-mail: *

Website:

Comment:

Gender: Female Male *

Submit

بخش سوم : آموزش دستورات پیشرفتهPHP

آموزش کار با تاریخ (Date) و ساعت (Time) درPHP

از تابع date() در زبان PHP برای قالب بندی و کار با تاریخ (Date) و یا ساعت (Time) استفاده می شود. درواقع تابع date() درPHP ، یک متغیرزمانی را به فرمتی قابل خواندن و استفاده تبدیل می کند. ساختار کلی تابع date() در PHP به صورت زیر است:

```
<?php
date(format,timestamp)
?>
```

- formپارامتر format قالب بندی کلی برچسب زمانی یا timestamp را تعیین کرده و اجباری (Requierd)است.
- timestampاین پارامتر برچسب زمانی (مقدار تاریخ و ساعت) را برای متغیر تعیین می کند. این پارامتر اختیاری (optional) بوده و مقدار پیش فرض آن تاریخ و ساعت جاری سیستم است.

نکته:

یک برچسب زمانی (timestamp) یک متغیر یا سلسله کاراکترهای پشت سر همی است که تاریخ و ساعت مورد نظر را در یک فرمت کلی نشان می دهد.

خواندن و نمایش ساده تاریخ به وسیله تابع: Date()

پارامتر اجباری format در تابع date() تعیین کننده قالب یا format نمایش تاریخ (Date) و ساعت (time) می باشد. در لیست زیر چندین کاراکتر پرکاربرد برای قالب بندی و نمایش زمان در تابع date() معرفی شده است:

- کارکتر : d این کاراکتر بیانگر تاریخ روز مورد نظر در ماه جاری (عددی بین ۰۱ تا ۳۱ است).
- کاراکتر mاین کاراکتر بیانگر ماه (month) جاری در تاریخ (عددی بین ۰ تا ۱۲)است.
- کاراکتر :Y این کاراکتر بیانگر عدد سال (year) در تاریخ جاری بوده که به صورت چهاررقمی است.
- کاراکترهای دیگری مثل "/" ، "." یا "-" را نیز می توان برای اضافه کردن قالب بندی مورد نظر بین اعداد تاریخ قرار داد.

مثال عملی : در کد مثال عملی زیر، تاریخ روز جاری را به صورت ۳ شکل مختلف نمایش داده ایم:

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

نکته آموزشی-نمایش اتوماتیک وار سال جهت کپی رایت

در اکثر وب سایت ها معمولا یک علامت کپی رایت و عدد سال جاری برای نمایش در کنار برند سایت استفاده می شود. می توانید با استفاده از کد ساده زیر در PHP و استفاده از تابع date() همواره عدد سال جاری را در سایت به روز کرده و نمایش دهید:

```
<?php
© 2010-<!--?php echo date("Y");?-->
?>
```

خواندن و نمایش ساده ساعت (time) به وسیله تابع date()

در لیست زیر به معرفی کاراکترهای رایجی که برای قالب بندی نمایش زمان در PHP می توانید استفاده کنید پرداخته ایم:

- کاراکتر h از این کاراکتر برای نمایش عدد ساعت جاری (عددی بین ۰۱ تا۱۲) استفاده می شود.
- کاراکتر m از این کاراکتر برای نمایش عدد دقیقه یا minute ساعت جاری (بین ۰۰ تا ۵۹) استفاده می شود.
- کاراکتر s از این کاراکتر برای نمایش عدد ثانیه یا seconds ساعت جاری(بین ۰۰ تا ۵۹) استفاده می شود.
- کاراکتر a این کاراکتر قبل از ظهر بودن ساعت (am) و بعد از ظهر بودن آن (pm) را در فرمت ۱۲ ساعتی تعیین می کند.

مثال عملی:

در کد مثال عملی زیر، مقدار ساعت جاری را در فرمت دلخواه نمایش داده ایم:

```
<?php
<!--?php
echo "The time is " . date("h:i:sa");
?-->
?>
```

نکته مهم:

توجه داشته باشید که تابع date() تاریخ و ساعت جاری سرور را نشان می دهد نه کامپیوتر کلاینت کاربر.

آموزش نحوه خواندن و نمایش ساعت محلی (time zone) در PHP

اگر مقدار ساعت یا تاریخی که از سرور دریافت می کنید با ساعت منطقه زمانی شما یکسان نیست به دلیل این است که سرور شما در یک محدوده زمانی (time zone) متفاوت با مکان شما قرار دارد (مثلا سرور در آلمان یا آمریکاست)

بنابراین اگر می خواهید زمان (time) صحیح را برحسب محل خودتان به کاربر نمایش دهید بایستی از timezone در تابع date() استفاده کنید.

```
<?php
<!--?php
date_default_timezone_set("America/New_York");
echo "The time is " . date("h:i:sa");
?-->
```

مثال عملی : مثال عمل:در کد مثال عملی زیر، محدوده زمانی یا timezone را به مقدار "Iran/tehrn" تغیر داده ایم تا ساعت صحیح نشان داده شود:

```
<?php
<!--?php
date_default_timezone_set("Iran/tehrn");
```

```
echo "The time is " . date("h:i:sa");
?-->
?>
```

تعیین یک تاریخ با استفاده از تابع mktime درPHP

همانطور که در بخش ابتدای این درس بیان کردیم پارامتر دلخواه stamp time در تابع date() یک برچسب زمانی خاص را تعیین می کند.اگر مقدار خاصی برای پارامتر timestamp تعیین نشودسیستم تاریخ و زمان جاری سیستم را مورد استفاده قرار می دهد)همانطور که در کد مثال عملی زیر نشان داده شده است(تابع mktime برچسب زمانی Unix را برای تاریخ بر می گرداند.برچسب زمانی Unix تعداد ثانیه های سپری شده از دوره زمانی Unix یا تاریخ)January/۱۹۷۰ ۰۰:۰۰:۰۰ GMT(و تاریخ تعیین شده را بر می گرداند.همانند کد مثال زیر:

ساختار کلی تابع mktime به صورت زیر است:

```
<?php
mktime(hour,minute,second,month,day,year)
?>
```

مثال عملی،کد مثال عملی زیر، یک زمان و تاریخ را برحسب پارامترهای تعیین شده برای تابع mktime ایجاد می کند:

```
<?php
<!--?php
$d=mktime(11, 14, 54, 8, 12, 2014);
echo "Created date is " . date("Y-m-d h:i:sa", $d);
?-->
?>
```

آموزش ایجاد یک تاریخ (Date) برحسب یک string با استفاده از تابع strtotime()درPHP

از تابع strtotime در PHP برای تبدیل یک رشته متنی قابل خواندن برای انسان یا string ه یک زمان Unixاستفاده می شود.

ساختار کلی استفاده از تابع strtotime به صورت زیر است:

```
<?php
strtotime(time,now)
?>
```

مثال عملی: در کد مثل عملی زیر، با استفاده از تابع strtotime() یک تاریخ و ساعت را بر حسب string داده شده تولید کرده ایم:

```
<?php
<!--?php
$d=strtotime("10:30pm April 15 2014");
echo "Created date is " . date("Y-m-d h:i:sa", $d);
```

```
?-->
?>
```

زبان PHP در تبدیل رشته یا string به زمان بسیار باهوش است بنابراین می توانید آن را با مقادیر مختلف مثل کد عملی مثال زیر به کار ببرید:

```
<?php
<!--?php
$d=strtotime("tomorrow");
echo date("Y-m-d h:i:sa", $d) . "<br-->";

$d=strtotime("next Saturday");
echo date("Y-m-d h:i:sa", $d) . "<br>";

$d=strtotime("+3 Months");
echo date("Y-m-d h:i:sa", $d) . "<br>";
?>
```

اما تابع strtotime() چندان مطمئن نیست بنابراین حتما مقدار متنی وارد شده برای آن را چک نماید

مثال های عملی بیشتر کار با تابع date() درPHP

مثال عملی: کد مثال عملی زیر، تاریخ مربوط به شنبه آتی را نشان می دهد:

```
<?php
<pre id="codes" class="brush: php;"><!--?php
$startdate = strtotime("Saturday");
$enddate = strtotime("+6 weeks", $startdate);

while ($startdate < $enddate) {
echo date("M d", $startdate) . "<br-->";
$startdate = strtotime("+1 week", $startdate);
}
?>
```

مثال: کد مثال عملی زیر نیز تعداد روزهای باقی مانده تا تاریخ چهارم جولای را نشان می دهد:

```
<?php
$d1=strtotime("July 04");
$d2=ceil(($d1-time())/60/60/24);
echo "There are " . $d2 ." days until 4th of July.";
?-->
?>
```

مرجع کامل آموزش کار با Date درPHP

برای دریافت اطلاعات کامل برای کار با تاریخ Date در PHP به مرجع آموزش زمان در بخش آموزش PHP بروید. مرجع آموزش کار با زمان در PHP شامل توضیح کامل تابع های پرکاربرد کار با زمان به همراه مثال عملی است.

آموزش کار با تاریخ (Date) و ساعت (Time) درPHP

از تابع date() در زبان PHP برای قالب بندی و کار با تاریخ (Date) و یا ساعت (Time) استفاده می شود. درواقع تابع date() درPHP ، یک متغیرزمانی را به فرمتی قابل خواندن و استفاده تبدیل می کند. ساختار کلی تابع date() در PHP به صورت زیر است:

```
<?php
date(format,timestamp)
?>
```

- formپارامتر format قالب بندی کلی برچسب زمانی یا timestamp را تعیین کرده و اجباری (Requierd)است.
- timestampاین پارامتر برچسب زمانی (مقدار تاریخ و ساعت) را برای متغیر تعیین می کند. این پارامتر اختیاری (optional) بوده و مقدار پیش فرض آن تاریخ و ساعت جاری سیستم است.

نکته:

یک برچسب زمانی (timestamp) یک متغیر یا سلسله کاراکترهای پشت سر همی است که تاریخ و ساعت مورد نظر را در یک فرمت کلی نشان می دهد.

خواندن و نمایش ساده تاریخ به وسیله تابع: Date()

پارامتر اجباری format در تابع date() تعیین کننده قالب یا format نمایش تاریخ (Date) و ساعت (time)می باشد. در لیست زیر چندین کاراکتر پرکاربرد برای قالب بندی و نمایش زمان در تابع date() معرفی شده است:

- کارکتر d این کاراکتر بیانگر تاریخ روز مورد نظر در ماه جاری (عددی بین 01 تا 31 است).
- کاراکتر mاین کاراکتر بیانگر ماه (month) جاری در تاریخ (عددی بین ۰ تا ۱۲)است.
- کاراکتر Y این کاراکتر بیانگر عدد سال (year) در تاریخ جاری بوده که به صورت چهاررقمی است.
- کاراکترهای دیگری مثل "/" ، "." یا "–" را نیز می توان برای اضافه کردن قالب بندی مورد نظر بین اعداد تاریخ قرار داد.

مثال عملی : در کد مثال عملی زیر، تاریخ روز جاری را به صورت ۳ شکل مختلف نمایش داده ایم:

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

نکته آموزشی-نمایش اتوماتیک وار سال جهت کپی رایت

در اکثر وب سایت ها معمولا یک علامت کپی رایت و عدد سال جاری برای نمایش در کنار برند سایت استفاده می شود. می توانید با استفاده از کد ساده زیر در PHP و استفاده از تابع date() همواره عدد سال جاری را در سایت به روز کرده و نمایش دهید:

```
<?php
© 2010-<!--?php echo date("Y");?-->
?>
```

خواندن و نمایش ساده ساعت (time) به وسیله تابع date()

در لیست زیر به معرفی کاراکترهای رایجی که برای قالب بندی نمایش زمان در PHP می توانید استفاده کنید پرداخته ایم:

- کاراکتر hاز این کاراکتر برای نمایش عدد ساعت جاری (عددی بین ۰۱ تا۱۲) استفاده می شود.
- کاراکتر m از این کاراکتر برای نمایش عدد دقیقه یا minute ساعت جاری (بین ۰۰ تا ۵۹) استفاده می شود.
- کاراکترsاز این کاراکتر برای نمایش عدد ثانیه یا seconds ساعت جاری(بین ۰۰ تا ۵۹) استفاده می شود.
- کاراکتر aاین کاراکتر قبل از ظهر بودن ساعت (am) و بعد از ظهر بودن آن (pm) را در فرمت ۱۲ ساعتی تعیین می کند.

مثال عملی: در کد مثال عملی زیر، مقدار ساعت جاری را در فرمت دلخواه نمایش داده ایم:

```
<?php
<!--?php
echo "The time is " . date("h:i:sa");
?-->
?>
```

نکته مهم:

توجه داشته باشید که تابع date() تاریخ و ساعت جاری سرور را نشان می دهد نه کامپیوتر کلاینت کاربر.

آموزش نحوه خواندن و نمایش ساعت محلی (time zone) در PHP

اگر مقدار ساعت یا تاریخی که از سرور دریافت می کنید با ساعت منطقه زمانی شما یکسان نیست به دلیل این است که سرور شما در یک محدوده زمانی (time zone) متفاوت با مکان شما قرار دارد (مثلا سرور در آلمان یا آمریکاست)

بنابراین اگر می خواهید زمان (time) صحیح را برحسب محل خودتان به کاربر نمایش دهید بایستی از timezoneدر تابع date() استفاده کنید.

```
<?php
<!--?php
date_default_timezone_set("America/New_York");
echo "The time is " . date("h:i:sa");
?-->
?>
```

مثال عمل:در کد مثال عملی زیر، محدوده زمانی یا timezone را به مقدار "Iran/tehrn" تغیر داده ایم تا ساعت صحیح نشان داده شود:

```
<?php
<!--?php
date_default_timezone_set("Iran/tehrn");
echo "The time is " . date("h:i:sa");
?-->
?>
```

تعیین یک تاریخ با استفاده از تابع mktime درPHP

همانطور که در بخش ابتدای این درس بیان کردیم پارامتر دلخواه stamp time در تابع date() یک برچسب زمانی خاص را تعیین می کند.اگر مقدار خاصی برای پارامتر timestamp تعیین نشودسیستم تاریخ و زمان جاری سیستم را مورد استفاده قرار می دهد همانطور که در کد مثال عملی زیر نشان داده شده است تابع mktime برچسب زمانی Unix را برای تاریخ بر می گرداند.برچسب زمانی Unix تعداد ثانیه های سپری شده از دوره زمانی Unix یا تاریخ January/1970 00:00:00 GMT و تاریخ تعیین شده را بر می گرداند.همانند کد مثال زیر:

ساختار کلی تابع mktime به صورت زیر است:

```
<?php
mktime(hour,minute,second,month,day,year)
?>
```

مثال عملی،کد مثال عملی زیر، یک زمان و تاریخ را برحسب پارامترهای تعیین شده برای تابع mktme ایجاد می کند:

```
<?php
<!--?php
$d=mktime(11, 14, 54, 8, 12, 2014);
echo "Created date is " . date("Y-m-d h:i:sa", $d);
?-->
```

آموزش ایجاد یک تاریخ (Date) برحسب یک string با استفاده از تابع strtotime()درPHP

از تابع strtotime در PHP برای تبدیل یک رشته متنی قابل خواندن برای انسان یا string ه یک زمان Unixاستفاده می شود.

ساختار کلی استفاده از تابع strtotime به صورت زیر است:

```
<?php
strtotime(time,now)
?
```

مثال عملی: در کد مثل عملی زیر، با استفاده از تابع strtotime() یک تاریخ و ساعت را بر حسب string داده شده تولید کرده ایم:

```
<?php
<!--?php
$d=strtotime("10:30pm April 15 2014");
echo "Created date is " . date("Y-m-d h:i:sa", $d);
?-->
?>
```

زبان PHP در تبدیل رشته یا string به زمان بسیار باهوش است بنابراین می توانید آن را با مقادیر مختلف مثل کد عملی مثال زیر به کار ببرید:

```
<?php
<!--?php
$d=strtotime("tomorrow");
echo date("Y-m-d h:i:sa", $d) . "<br-->";

$d=strtotime("next Saturday");
echo date("Y-m-d h:i:sa", $d) . "<br>";

$d=strtotime("+3 Months");
echo date("Y-m-d h:i:sa", $d) . "<br>";
?>
?>
```

اما تابع strtotime() چندان مطمئن نیست بنابراین حتما مقدار متنی وارد شده برای آن را چک نماید

مثال های عملی بیشتر کار با تابع date() درPHP

مثال عملی:کد مثال عملی زیر، تاریخ مربوط به شنبه آتی را نشان می دهد:

```
<!--?php
$startdate = strtotime("Saturday");
$enddate = strtotime("+6 weeks", $startdate);

while ($startdate < $enddate) {
echo date("M d", $startdate) . "<br-->";
$startdate = strtotime("+1 week", $startdate);
}
?>
```

مثال۲: در کد مثال عملی زیر نیز تعداد روزهای باقی مانده تا تاریخ چهارم جولای را نشان می دهد:

```
<?php
<!--?php
$d1=strtotime("July 04");
$d2=ceil(($d1-time())/60/60/24);
echo "There are " . $d2 ." days until 4th of July.";
?-->
?>
```

مرجع کامل آموزش کار با Dateدر PHP

برای دریافت اطلاعات کامل برای کار با تاریخ Dateدر PHP به مرجع آموزش زمان در بخش آموزش PHP بروید .مرجع آموزش کار با زمان در PHP شامل توضیح کامل تابع های پرکاربرد کار با زمان به همراه مثال عملی است.

آموزش کار با دستور include و require درPHP

دستور include و require کلیه متن،کد یا تگ های موجود در فایل تعیین شده برای آن را گرفته و در فایل استفاده کننده از دستور کپی می کند. از دستور های include و require برای تکرار یک دستور یا محتوی تکراری در درون چند صفحه استفاده می شودو عملکردی مانند قابلیت متدپیج درAsp.Net دارد . به عبارت دیگر دستور include زمانی بسیار کاربرد دارد که می خواهید محتویات یکسان HTML، PHP یا متنی را درون چندین صفحه یک وب سایت عینا تکرار کنید . این امکان وجود دارد که با استفاده از دستورات include یا require محتویات یک فایل PHP را درون یک فایل PHP دیگر وارد نمایید)قبل از این که سرور فایل مقصد را خوانده و اجرا کند (.

عملکرد دستورات include و require کاملا شبیه هم هستند و تنها تفاوت آن ها واکنش در هنگام بروز خطاست که به صورت زیر تقسیم می شوند:

دستور require در هنگام بروز خطا یک error کامل (fatal) بروز داده مثل (E_COMPAE_ERRO) و اجرای اسکریپت را متوقف می کند.

دستور include در هنگام بروز خطا یک هشدار یا warning مثل (E_WARNING) صادر کرده و به اجرای اسکریپت ادامه می دهد.

بنابراین اگر می خواهید اجرای صفحات همواره ادامه داشته و خروجی را به کاربر نشان دهد حتی اگر صفحه یا فایل دستور include گم شده یا قابل خواندن نباشد باید از دستور include استفاده کنید . از طرف دیگر در موار کار با فریم ورک ها، CMS ها و یا نرم افزارهای کامل تحت PHP از دستور require استفاده کنید تا همواره مطمئن شوید که فایل تعیین شده برای کپی در صفحات در روند اجرای برنامه قرار گیرد .

استفاده از دستور require امنیت برنامه PHP و تمامیت آن همواره حذف شود زیرا کاربر نمی تواند باحذف عمدی یا اتفاقی یک فایل کلیدی یا بخشی از کد، صفحات را اجرا کند . برای درک بهتر یک مثال می زنیم.فرض کنید بخش چک کردن هویت کاربر از طریق یک دستور require به تمامی صفحات کپی می شود. اگر از دستور include استفاده شود کاربر می تواند فایل مربوط به دستورات هویت را پاک کرده و صفحه را بدون آن اجرا نموده و مسئله هویت را دور بزند. اما در صورت استفاده از دستور require در صورت پاک شدن فایل کد هویت دیگر پروژه اجرا نخواهد شد و امنیت آن به خطر نمی افتد .

استفاده از دستورات include و require حجم زیادی از کار را کاهش می دهد به این معنا که می توانید بخش های زیادی مثل هدر، فوتر و منو را در فایل های مجزا تولید کرده و آن را در تمامی صفحات وب سایتتان فقط با یک دستور ساده استفاده کنید. از طرف دیگر برای به روزرسانی و opdate فقط کافی است فایل اصلی کد را تغییر داده و این تغییر در همه صفحات وب سایت درجا اعمال شده و آپدیت بسیار سریع خواهد بود. ساختار کلی استفاده از دستور include یا require به صورت زیر است:

```
<?php
include 'filename';
or
require 'filename';
```

```
?>
```

مثال های عملی کار با دستور include درPHP

فرض کنید یک فایل فوتر یکسان به نام "footer.php" برای صفحات خود طراحی کرده ایم که به صورت زیر است:

```
<!--?php
echo "<p-->
Copyright © 1999-" . date("Y") . " W3Schools.com<p></p>";
?>
```

برای اضافه کردن کد صفحه footer به صفحه مورد نظر از کدی مثل کد زیر استفاده می شود:

```
<?php
< html>
< body>
< h1>Welcome to my home page!
< p>Some text.<p></p>
< p>Some more text.<p></p>
<!--?php include 'footer.php';?-->
< /body>
< /html>
?>
```

مثال

فرض کنید کد منوی سایت را همانند کد مثال زیر در یک فایل به نام "menu.PHP" قرار داده ایم:

```
<!--?php
echo '<a href="/default.asp"-->
Home -
<a href="/html/default.asp">HTML Tutorial</a> -
<a href="/css/default.asp">CSS Tutorial</a> -
<a href="/js/default.asp">JavaScript Tutorial</a> -
<a href="default.asp">PHP Tutorial</a>';
?>
```

تمامی صفحات سایت بایستی از این منو استفاده کنند.در کد مثال زیر نحوه قرار دادن این منو در تمامی صفحات فایل با استفاده ز تگ را نشان داده ایم. بعدا می توانید با CSS استایل و ظاهر منو را نیز تغیر دهید:

مثال

فرض کنید یک فایل با نام "vars.PHP" با تعداد متغیر تعریف شده در آن به صورت زیر داریم:

```
<?php
<!--?php
$color='red';
$car='BMW';
?-->
?>
```

بنابراین اگر فایل "vars.php" رادر یک فایل دیگر include کنیم می توان از این متغیرها در فایل فراخوانی شده استفاده کرد به صورت کد مثال عملی زیر:

```
<?php
< html>
< body>
< h1>Welcome to my home page!
<!--?php include 'vars.php';
echo "I have a $color $car.";
?-->
< /body>
< /html>
?>
```

مقایسه دستورات require و include درPHP

همانطور که در ابتدای درس نیز اشاره کردیم، از دستور require نیز می توان برای قرار دادن کدهای یک فایل در یک فایل PHP دیگر استفاده کرد .

اما بین این دو دستور یک اختلاف عمده وجود دارد اگر یک فایل را با دستور include در یک فایل دیگر PHP وارد کرده و PHP نتواند فایل include شده را پیدا کند، اجرای اسکریپت بدون توجه به نقصان فایل ادامه خواهد داشت.همانند کد زیر:

```
<?php
< html>
< body>
< h1>Welcome to my home page!
<!--?php include 'noFileExists.php';
echo "I have a $color $car.";
?-->
< /body>
< /html>
?>
```

اما اگر در مثال فوق به جای دستور include از دستور require استفاده کنید در صورت فقدان فایل اضافه شده دستور echo به دلیل صدور خطای fatal یا نابود کننده از سوی فایل مقصد، اجرا نخواهد شد و عملیات فایل متوقف می شود. همانند کد زیر:

```
<?php
< html>
< body>
< h1>Welcome to my home page!
<!--?php require 'noFileExists.php';
echo "I have a $color $car.";
?-->
< /body>
< /html>
?>
```

نتیجه گیری:

از دستور require زمانی استفاده کنید که می خواهید فایل کپی شده حتما در فایل مقصد PHP وجود داشته و برنامه بدون آن اجرا نشود. اما از دستور include زمانی استفاده کنید که حتی فقدان فایل کپی شده اهمیتی نداشته و اجرای برنامه بدون آن، خطر خاصی نداشته باشد.

آموزش مدیریت فایل ها در صفحاتPHP

مدیریت فایل ها (File Handling) یک بخش بسیار مهم از نرم افزار تحت وب می باشد. در بسیاری از موارد نیاز دارید یک فایل را برای اهداف مختلف خوانده و پردازش کنید.

آموزش دستکاری فایل ها در PHP

زبان PHP دارای توابع مختلفی برای اموری مانندایجاد، خواندن، آپلود کردن و ویرایش فایل ها می باشد.

راهنمایی:

در هنگام دستکاری و کار با فایل ها بسیار مراقب باشید!

در صورت اشتباه در هنگام کار با فایل ها می توانید آسیب زیادی به سیستم وارد نمایید. برخی خطاها ی احتمالی مثل ویرایش یک فایل اشتباه، پر کردن فضای هاست سایت با اطلاعات به درد نخور و یا حذف فایل های ضروری به صورت اتفاقی می تواند کل سرور را از کار بیاندازد.

آموزش کار با تابع readfile درPHP

تابع read file() یک فایل را خوانده و آن را در حافظه موقت buffer قرار می دهد، برای مثال فرض کنید یک فایل متنی به نام "web dictionary.text" همانند فایل زیر را بر روی سرور داریم:

```
<?php
AJAX = Asynchronous JavaScript and XML
CSS = Cascading Style Sheets
HTML = Hyper Text Markup Language
PHP = PHP Hypertext Preprocessor
SQL = Structured Query Language
SVG = Scalable Vector Graphics
XML = EXtensible Markup Language
?>
```

کد عملی برای خواندن یک فایل در PHP و قرار دادن آن در حافظه موقت به وسیله read file() به صورت زیر است.این تابع، پس از اتمام خواندن فایل تعداد بایت های حجم فایل را در صورت موفقیت در پردازش کامل، به عنوان خروجی بر می گرداند:

```
<?php
<!--?php
echo readfile("webdictionary.txt");
?-->
?>
```

تابع read file() در زمانی که بخواهید یک فایل را باز کرده و محتویات آن را بخوانید کاربرد دارد. در درس های بعدی، به آموزش مباحث بیشتری راجع به کار با فایل ها خواهیم پرداخت.

آموزش بازکردن, خواندن و بستن فایل درPHP

در این درس قصد داریم تا نحوه بازکردن (open) ، خواندن (read) و بستن (close) فایل ها در PHP را آموزش دهیم.

آموزش کار با تابع fopen درPHP

راه بهتر برای باز کردن فایل ها در PHP استفاده از تابع fopen() به جای تابع read file() است زیرا تابع fopenامکانات بیشتری را در اختیارتان قرار می دهد.

در مثال های آموزشی این درس از فایل متنی "web.text" که محتویات آن به صورت زیر است استفاده خواهیم کرد:

```
<?php
AJAX = Asynchronous JavaScript and XML
CSS = Cascading Style Sheets
HTML = Hyper Text Markup Language
PHP = PHP Hypertext Preprocessor
SQL = Structured Query Language
SVG = Scalable Vector Graphics
XML = EXtensible Markup Language
?>
```

پارامتر اول در تابع fopen() نام فایلی که می خواهیم آن را باز کرده و پارامتر دوم تعیین کننده متد باز کردن فایل است. در کد مثال عملی زیر، اقدام به باز کردن فایل متنی "web.text" کرده ایم که در آن از متد "r" به معنای بازکردن به صورت فقط خواندنی یا read-only استفاده شده است.تابع fopen() اگر نتواند فایل مورد نظر را باز کند، یک پیام هشدار نیز صادر خواهد کرد:

```
<?php
```

```
<!--?php
$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
echo fread($myfile,filesize("webdictionary.txt"));
fclose($myfile);
?-->
?>
```

انواع متدهای ممکن برای باز کردن فایل ها با استفاده از تابع fopen() عبارتند از:

- rدراین حالت فایل به صورت فقط خواندنی (readonly) و از ابتدای آن خوانده می شود.
- wدر این حالت فایل در حال نوشتن (write) باز می شود.اگر فایل از قبل وجود داشته باشد محتویات آن پاک شده و محتویات جدید در آن نوشته می شود.اگر هم فایل وجود نداشته باشد یک فایل جدید ایجاد می شود.در این متد هم اشاره گر از ابتدای فایل شروع می کند.
- aدر این حالت نیز فایل برای نوشتن (write) باز شده با این تفاوت که محتویات قبلی فایل حفظ شده و محتویات جدید به انتهای آن اضافه می شود.درواقع اشاره گر از انتهای فایل شروع کرده و در صورت موجودنبودن فایل ، یک فایل جدید اضافه خواهد شد.
- xاین حالت یک فایل جدید را جهت نوشتن (write) باز می کند.درصورتی که فایل مورد نظر از قبل وجود داشته error داده و مقدار FALSE را بر می گرداند.
- r+در این حالت برنامه فایل را جهت خواندن و نوشتن (read/write) باز می کند. اشاره گر در ابتدای فایل قرار خواهد گرفت.
- w+در این حالت برنامه فایل را برای خواندن و نوشتن (read/wife) باز کرده و در صورتی که فایل موجود باشد محتویات آن را پاک نموده و در صورت عدم وجود فایل ، یک نمونه جدید ایجاد می کند. در این حالت نیز اشاره گر از ابتدای فایل شروع می کند.

a : در این حالت نیز فایل برای خواندن و نوشتن (read/write) باز شده و محتویات قبلی آن حفظ شده و محتویات جدید به انتهای فایل اضافه می شود. اشاره گر از انتهای فایل شروع خواهد کرد.

x : در این حالت یک فایل جدید برای خواندن و نوشتن (read/write) باز شده و در صورت وجود داشتن فایل یک error رخ داده و مقدار FALSE بر می گرداند.

آموزش کار با فایل fread() درPHP

تابع fread()محتویات یک فایل باز را می خواند.

پارامتر اول در تابع dread() نام فایل جهت خواندن و پارامتر دوم تعداد بایت هایی که بایستی از فایل خوانده شود را مشخص می کند.

کد مثال عملی زیر کلیه محتویات فایل "webdictionary.text" را تا انتها می خواند.توجه داشته باشید اگر مقداری برای پارامتر دوم تعیین نشود، برنامه به صورت پیش فرض کل فایل را می خواند.

```
<?php
```

```
fread($myfile,filesize("webdictionary.txt"));
?>
```

آموزش کار با تابع fclose() درPHP

تابع fclose() جهت بستن یک فایل open به کار می رود.

راهنمایی:

بهتر است همواره پس از این که کارتان با یک فایل باز شده در برنامه تمام شد آن را حتما ببندید زیرا یک فایل باز در حافظه سرور باقی مانده و با مصرف منابع آن باعث کند شدن سیستم می شود.

تابع fclose() دارای یک پارامتر متنی بوده که نام فایل مورد نظر جهت بستن را تعیین می کند. به وسیله کد مثال عملی زیر فایل باز شده "web dictionary.txt" را می بندیم:

```
<?php
<!--?php
$myfile = fopen("webdictionary.txt", "r");
// some code to be executed....
fclose($myfile);
?-->
?>
```

آموزش کار با تابع fgets() در PHP -خواندن یک خط از فایل

از تابع fgets() برای خواندن یک خط از فایل مورد نظر استفاده می شود. در کد مثال عملی زیر با استفاده از تابع fgets() اولین خط از فایل webdictionary.txt" را خوانده و در خروجی نمایش داده ایم:

```
<?php
<!--?php
$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
echo fgets($myfile);
fclose($myfile);
?-->
```

نکته:

پس از هر بار فراخوانی تابع fgets() اشاره گر برنامه به ابتدای خط بعدی در فایل می رود.

آموزش چک کردن انتهای فایل در PHP با تابع:feof()

تابع feof() چک می کند آیا برنامه به انتهای فایل end-of-life یا EOF رسیده یا خیر. تابع feof() معمولا برای جستجو در دیتایی که طول آن را نمی دانیم مناسب است.

در کد مثال عملی زیر، محتویات فایل را به صورت خط به خط خوانده و نمایش داده ایم تا به انتهای فایل برسیم:

```
<?php
<!--?php
$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
// Output one line until end-of-file
while(!feof($myfile)) {
  echo fgets($myfile) . "<br-->";
}
fclose($myfile);
?>
?>
```

آموزش خواندن یک کاراکتر در فایل PHP باfgetc()

از تابع fgetc() در PHP برای خواندن یک کاراکتر تنها در فایل استفاده می شود. در کد مثال عملی زیر، محتویات فایل "web dictionary.text " را به صورت کاراکتر به کاراکتر خوانده و در خروجی نمایش داده ایم تا به انتهای فایل برسیم:

```
<!--?php
$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
// Output one character until end-of-file
while(!feof($myfile)) {
  echo fgetc($myfile);
}
fclose($myfile);
?-->
```

نکته:

پس از خواندن یک کاراکتر توسط fgetc() در فایل، اشاره گر به ابتدای کاراکتر بعدی خواهد رفت.

آموزش ایجاد (crate) و نوشتن (write) در فایل هایPHP

در این درس قصد داریم تا نحوه ایجاد (create) و نوشتن (write) بر روی فایل ها در زبان PHP را آموزش دهیم.

آموزش ایجاد (create) فایل ها درPHP -تابعfopen()

از تابع fopen() در زبان PHP برای برای ایجاد یک فایل جدید در سرور استفاده می شود. ممکن است کمی گیج کننده باشد ولی در PHP همان تابعی که برای باز کردن فایل ها استفاده می شود برای ایجاد آن ها نیز کاربرد دارد.

اگر از تابع fopen() برای باز کردن فایلی که وجود ندارد استفاده کنید، PHPبه صورت خودکار آن فایل را ایجاد کرده و آن را برای عمل نوشتن (writing) یا اضافه کردن محتوی (appending) آماده می کند.

کد مثال عملی زیر، یک فایل جدید به نام "totfile.txt" را بر روی سرور ایجاد می کد. فایل جدید در همان پوشه ای که فایل اجرا کننده برنامه قرار دارد، ساخته می شود.

```
<?php
$myfile = fopen("testfile.txt", "w")
?>
```

آموزش اجازه دسترسی یا Peermission در فایل های PHP

اگر در اجرای کدی مثل کد مثال فوق با خطا یا error مواجه شدید، بایستی امکان دسترسی به فایل مورد نظر و امکان نوشتن در آن را بررسی کنید.

آموزش نوشتن (write)در یک فایل PHP با تابع:fwrite()

از تابع fwrite() در PHP برای نوشتن (write) در یک فایل استفاده می شود.

تابع fwrite() دارای دو پارامتر بوده که پارامتر اول نام فایل مورد نظر جهت نوشتن و پارامتر دوم مقدار متن (string) را تعیین می کند.

کد مثال عملی زیر، تعدای نام را درون فایل جدید "newfile.txt" می نوسید:

```
<!--?php
$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
$txt = "John Doe\n";
fwrite($myfile, $txt);
$txt = "Jane Doe\n";
fwrite($myfile, $txt);
fclose($myfile);
?-->
```

توجه داشته باشید که در کد فوق، ما دو بار درون فایل "newfile.txt" نوشته ایم. در هر بار نوشتن درون این فایل، متغیر متنی $txt را به تابع ارسال کرده ایم که در مرحله اول شامل نام "John Doe" و در مرحله دوم شامل نام "Jane Doe" می باشد. پس از اتمام عملیات نوشتن در فایل ، با استفاده از تابع fclose() آن

را بسته ایم.

اگر فایل "newfile.txt" را پس از اجرای کد مثال، باز کنید محتویات آن به صورت زیر خواهد بود:

```
<?php
John Doe
Jane Doe
?>
```

عملیات overwriting در PHP

اکنون که فایل "newfile.txt" شامل مقداری دیتا است می توان نشان داد اگر یک فایل موجود را برای نوشتن باز کنیم چه اتفاقی می افتد. کلیه اطلاعات موجود در فایل رونویسی یا overwrite می شود و PHP با یک فایل خالی جدید شروع خواهد کرد.

در کد مثال زیر، فایل موجود "newfile.txt" را باز کرده و مقداری اطلاعات در آن نوشته ایم:

```
<!--?php
$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
$txt = "Mickey Mouse\n";
fwrite($myfile, $txt);
$txt = "Minnie Mouse\n";
fwrite($myfile, $txt);
fclose($myfile);
?-->
```

اگر پس از اجرای کد فوق، فایل "newfile.txt" را باز کرده، خواهید دید که محتویات آن به صورت زیر تغییر می کند:

```
<?php
Mickey Mouse
Minnie Mouse
?>
```

آموزش ارسال فایل Upload در زبان PHP

آپلود فایل ها بر روی سرور با استفاده از زبان PHP بسیار ساده است.اما همواره با سادگی خطراتی هم همراه خواهد بود. بنابراین هنگام آپلود فایل ها بر سرور به دلیل امکان ارسال فایل های مخرب یاحجیم بسیار دقت کنید.

مرحله اول تنظیم فایل"php.ini"

در مرحله اول بایستی مطمئن باشید که سرور PHP شما جهت امکان دریافت فایل،تنظیم شده باشد. درون فایل "php.ini"به دنبال خاصیت file Upload گشته و همانند کد زیر مقدار آن را بر روی on قرار دهید:

```
<?php
```

```
file_uploads = On
?>
```

مرحله دوم-ایجاد فرم Html لازم جهت آپلودفایل

در مرحله دوم، بایستی یک فرم Html طراحی کنید که در آن کاربر بتواند یک فایل مثل عکس یا image را انتخاب کرده و آپلود کند. همانند کد فرم زیر:

```
<?php
< !DOCTYPE html>
< html>
< body>
< form action="upload.php" method="post" enctype="multipart/form-data">
Select image to upload:
<input type="file" name="fileToUpload" id="fileToUpload">
<input type="submit" value="Upload Image" name="submit">
< /form>
< /body>
< /html>
?>
```

قواعد زیر را بایستی در هنگام تعریف کد فرم Html رعایت کنید:

- حتما مطمئن شوید که خاصیت "method="Post تنظیم شده باشد.
- همچنین لازم است فرم حاوی خاصیت "enctype="multiparty/form-data باشد. این خاصیت تعیین کننده نوع محتویات قابل ارسال "count-type"" در هنگام ارسال یا submit فرم است.بدون تنظیم موارد فوق، آپلود فایل با مشکل مواجه خواهد شد.

موارد زیر را نیز در تنظیم فرم رعایت کنید:

- تعیین خاصیت " type="file در تگ باعث می شود تا این کنترل به شکل یک دکمه فرمان انتخاب فایل یا file-select control درآمده و یک دکمه Browse جهت پیدا کردن فایل مورد نظر را در کنار کنترل input نشان دهد.

فرم Html کد مثال فوق، اطلاعات فایل را به صفحه "upload.PHP" " که در مرحله بعدی ایجاد خواهیم کرد،می فرستد.

مرحله سوم-تعیین اسکریپت لازم جهت آپلود فایل

کد فایل "upload.PHP"که برای آپلود فایل به کار می رود به صورت زیر است.آن را مرور کنید. در انتها به توضیح نکات مهم آن پرداخته ایم:

```
<!--?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = pathinfo($target_file,PATHINFO_EXTENSION);
// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
$check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
if($check !== false) {
echo "File is an image - " . $check["mime"] . ".";
$uploadOk = 1;
} else {
echo "File is not an image.";
$uploadOk = 0;
}
}
?-->
```

توضیح موارد موجود در کد فوق به شرح زیر است:

- تعیین خاصیت = " $target_dir" "uploads/" پوشه ای که فایل آپلود شده بایستی در آن قرار بگیرد را نشان می دهد.
- خاصیت $target_file"" مسیر لازم جهت انتخاب فایل برای آپلود را مشخص می کند.
- خاصیت =1 $uploadOK در حال حاضر استفاده نمی شود و در مراحل بعدی به تشریح آن خواهیم پرداخت.
- خاصیت $ imageFileTypeتعیین کننده پسوند یا extension فایل جهت آپلود است.
- در مرحله آخر هم چک کرده ایم که آیا فایل image یک فایل واقعی عکس یا یک فایل جعلی است که در صورت جعلی بودن آن از انجام عملیات آپلود جلوگیری می شود.

نکته:

شما باید یک پوشه جدید به نام "uploads" را در مسیری که فایل upload.php"" قرار دارد ایجاد کنید. فایل های أپلود شده در این پوشه ذخیره خواهند شد.

آموزش چک کردن این که فایل از قبل وجود داشته یا نه:

در مراحل بعدی قصد داریم تا چندین مراحل محدودیت یا کنترل فایل را جهت آپلود فایل وضع کنیم. در بخش اول بایستی بررسی کنیم که آیا فایلی که می خواهیم آپلود کنیم از قبل وجود داشته یا نه. اگر فایل مورد نظر

از قبل وجود داشته باشد باید از عمل آپلود جلوگیری کرده و یک پیام هشدار را به صورت کد زیر به کاربر نمایش دهیم. همچنین با تنظیم خاصیت $ uploadOK=0 عملیات آپلود لغو خواهد شد:

```
<?php
/ Check if file already exists
if (file_exists($target_file)) {
echo "Sorry, file already exists.";
$uploadOk = 0;
}
?>
```

آموزش تعیین حجم مجاز فایل جهت آپلود:

فیلد آپلود فایل یا file input field در کد Htmlفوق به نام "file To Upload" می باشد. در بخش دوم بایستی حجم سایز جهت آپلود را چک کنیم.اگر حجم فایل بیشتر از ۵۰۰کیلوبایت باشد، از ارسال فایل جلوگیری کرده و یک پیام هشدار به کاربر نمایش خواهیم داد. به صورت زیر:

```
<?php
// Check file size
if ($_FILES["fileToUpload"]["size"] > 500000) {
echo "Sorry, your file is too large.";
$uploadOk = 0;
}
?>
```

با تعیین مقدار خاصیت $ uploadOK=0 عملیات ارسال فایل لغو خواهد شد.

آموزش تعیین نوع داده ای فایل جهت آپلود

کد زیر به کاربر امکان آپلود فایل هایی با پسوندJPEG، JPG ، PNGو GIF که پسوندهای رایج عکس می باشند را می دهد. اگر نوع فایل به غیر از انواع تعیین شده باشد، یک پیام هشدار به کاربر نشان داده شده و با تنظیم خاصیت $ uploadOK=0عملیات ارسال فایل لغو خواهد شد:

```
<?php
// Allow certain file formats
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType !=
"jpeg"
&& $imageFileType != "gif" ) {
echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
$uploadOk = 0;
}
?>
```

کد کامل اسکریپت لازم جهت آپلود فایل

در نهایت، کد اسکریپت لازم جهت آپلود فایل به صورت زیر درخواهد آمد:

```
<!--?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
```

```
$uploadOk = 1;
$imageFileType = pathinfo($target_file,PATHINFO_EXTENSION);
// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
$check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
if($check !== false) {
echo "File is an image - " . $check["mime"] . ".";
$uploadOk = 1;
} else {
echo "File is not an image.";
$uploadOk = 0;
}
}
// Check if file already exists
if (file_exists($target_file)) {
echo "Sorry, file already exists.";
$uploadOk = 0;
}
// Check file size
if ($_FILES["fileToUpload"]["size"] --> 500000) {
echo "Sorry, your file is too large.";
$uploadOk = 0;
}
// Allow certain file formats
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType !=
"jpeg"
&& $imageFileType != "gif" ) {
echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
$uploadOk = 0;
}
// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0) {
echo "Sorry, your file was not uploaded.";
// if everything is ok, try to upload file
} else {
if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"],
$target_file)) {
echo "The file ". basename( $_FILES["fileToUpload"]["name"]). " has
been uploaded.";
} else {
echo "Sorry, there was an error uploading your file.";
}
}
```

مرجع کامل کار با FileSystem در PHP

برای دریافت اطلاعات کامل درباره توابع کار با فایل ها به مرجع کامل آموزش PHPFileSystem در سایت تحویل داده بروید.

آموزش کار با کوکی Cookies در زبانPHP

از کوکی (Cookie) معمولا برای شناسایی کاربر بر روی وب استفاده می شود. یک کوکی (cookie) یک قطعه اطلاعات یا یک فایل اضافه است که سرور به کامپیوتر کاربر الحاق یا embed می کند. هر بار که کامپیوتر صفحه ای را طی یک درخواست یا request فراخوانی کند اطلاعات کوکی به همراه درخواست وی ارسال می شود. استفاده از زبان PHP هم می توانید برای کاربران کوکی ایجاد کرده و یا مقادیر کوکی را دریافت کنید. برای مثال، مثلا وقتی صفحه که صفحه ایمیل خود را باز می کنید ، مشاهده می کنید که از قبل بر روی سرور log inشده اید و اطلاعات حساب کاربری شما در مرورگر نمایش داده می شود. این اطلاعات کاربری از طریق کوکی موجود در صفحه هر بار که ایمیل خود را باز می کنید، به سرور ارسال می شود.

آموزش ایجاد کوکی Cookie در زبان PHP

کوکی در زبان PHP به وسیله تابع setcookie() ایجاد می شود. ساختار تعریف Cookie به صورت زیر است :

```
<?php
setcookie(name, value, expire, path, domain, secure, httponly);
?>
```

در ساختار فوق فقط تعیین پارامتر name ضروری بوده و بقیه پارامترها اختیاری هستند. پارامتر name تعیین کننده نام Cookie است که از آن برای فراخوانی و شناسایی کوکی در سطح برنامه استفاده می شود. مثال عملی : کد زیر یک کوکی به نام "user" با مقدار "John Doe" را ایجاد می کند. این کوکی بعد از 30 روز (30*86400) منقضی می شود. به کار بردن علامت (/) به این معناست که این کوکی در کل سطح برنامه یا سایت قابل دسترس خواهد بود (یا می توانید یک پوشه یا مسیر دیگری را تعیین کرده که کوکی فقط در آن بخش قابل دسترس باشد.)

سپس با استفاده از متغیر سراسری $_COOKIE مقدار کوکی User را خوانده ایم. همچنین از تابع isset() برای فهمیدن این که آیا کوکی تنظیم شده یا خیر استفاده کرده ایم:

```
<?php
<!--?php
$cookie_name = "user";
$cookie_value = "John Doe";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); //
86400 = 1 day
?-->
< html>
< body>
<!--?php
```

```
if(!isset($_COOKIE[$cookie_name])) {
echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
echo "Cookie '" . $cookie_name . "' is set!<br-->";
echo "Value is: " . $_COOKIE[$cookie_name];
}
?>
< /body>
< /html>
?>
```

نکته1:

تابع Setcookie() را بایستی قبل از تگ تعریف کنید.

نکته2:

مقدار یا value کوکی به صورت خودکار در زمان ارسال URLencoded می شود.یعنی کاراکترهای غیر مجاز آن به صورت قابل ارسال با استفاده از URL تبدیل می کنیم (و همچنین در هنگام دریافت آن URLdecoded می گردد. برای جلوگیری از رخ دادن URLencoding از تابع setrowcookie() به جای تابع فوق استفاده کنید.

آموزش تغییر مقدار یک کوکی در زبان PHP

برای تغییر مقدار یک کوکی، بایستی مجددا آن کوکی را با استفاده از تابع setcookie() تعریف کنید. به صورت کد زیر:

```
<?php
<!--?php
$cookie_name = "user";
$cookie_value = "Alex Porter";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
?-->
< html>
< body>
<!--?php
if(!isset($_COOKIE[$cookie_name])) {
echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
echo "Cookie '" . $cookie_name . "' is set!<br-->";
echo "Value is: " . $_COOKIE[$cookie_name];
}
?>
< /body>
< /html>
?>
```

آموزش حذف یک کوکی در PHP

برای حدف یک کوکی، بایستی با استفاده از تابع setcookie() مجددا آن را تنظیم کرده ولی این بار تاریخ انقضای کوکی را بر روی یک زمان ماقبل از زمان فعلی قرار دهید، همانند کد زیر:

```
<?php
<!--?php
// set the expiration date to one hour ago
setcookie("user", "", time() - 3600);
?-->
< html>
< body>
<!--?php
echo "Cookie 'user' is deleted.";
?-->
< /body>
< /html>
?>
```

آموزش چک کردن فعال بودن کوکی در مرورگر کاربر

مرورگرها و یا سایت های تحت وب این قابلیت را دارند که خواندن یا ایجاد کوکی ها را در آن ها غیر فعال کرد. بنابراین در برنامه هایی که استفاده از کوکی ضروری است، بایستی قبل از طراحی کد، چک کنیم آیا این امکان فعال است یا خیر.

در کد مثال عملی زیر، یک اسکریپت ساده ابتدا چک می کند که آیا کوکی ها فعال هستند یا خیر. در ابتدای کد تلاش شده تا با استفاده از تابع setcookie() یک کوکی به نام "test_cookie" ایجاد شود، با شمارش تعداد متغیر $_COOKIE می فهمیم آیا کوکی فعال است یا خیر:

```
<?php
<!--?php
setcookie("test_cookie", "test", time() + 3600, '/');
?-->
< html>
< body>
<!--?php
if(count($_COOKIE) --> 0) {
echo "Cookies are enabled.";
} else {
echo "Cookies are disabled.";
}
?>
< /body>
< /html>
?>
```

مرجع کامل HTTP در PHP

برای دریافت اطلاعات کامل درباره توابع کار با HTTP درPHP ، به بخش مرجع آموزش PHP HTTP در سایت تحلیل داده بروید.

آموزش کار با Session در زبان PHP

یک Session راهی برای نگهداری اطلاعات درون متغیر یا variable است که بتوان آن را در صفحات مختلف سایت استفاده کرد.

برخلاف کوکی یاCookie ، اطلاعات Session بر روی مرورگر کاربر و کامپیوتر وی ذخیره نشده و بر روی سرور سایت نگهداری می شوند.

مفهوم Session در PHP چیست؟

هنگامی که با یک نرم افزار یا application کار می کنید، آن را باز کرده، تغییراتی در آن اعمال نموده، در نهایت آن را می بندید. این عملکرد بسیار شبیه session است. در نرم افزار های تحت ویندوز، کامپیوتر می داند شما چه کسی هستید. کامپیوتر از شروع باز کردن برنامه تا پایان کارتان، درک کاملی از هویت شما دارد. اما بر روی اینترنت یک مشکل بزرگ وجود داشته و آن این است که سرور وب نمی داند شما چه کسی بوده و چه کاری دارید انجام می دهید، زیرا آدرس و پروتکل های HTTP بی ثبات و ناپایدار بوده و وضعیت شما را نگهداری نمی کنند.

متغیرهای Session با نگهداری اطلاعات کاربر در فضاهای مشخص مشکل فوق را حل کرده و این اطلاعات را در تمامی صفحات مورد استفاده کاربر در اختیار وب سرور قرار می دهند (اطلاعاتی مثل نام کاربری، رنگ مورد علاقه و ...).

بنابراین متغیرهای Session اطلاعات مربوط به یک کاربر خاص را نگهداری کرده و آن را در اختیار تمامی صفحات مورد استفاده وی قرار می دهند.

نکته:

اگر می خواهید اطلاعاتی را به صورت دایمی نگهداری کنید، بایستی آن ها را در پایگاه داده قرار دهید.

آموزش نحوه کار شروع یک Session در PHP

عملکرد یک Session توسط تابع ()session_start شروع می شود. متغیرهای session درون متغیر سراسری $_SESSION نگهداری می شوند. برای درک بهتر، بیایید یک مثال عملی را با ایجاد یک صفحه جدید PHP به نام "demo_session1.php" بررسی کنیم. در این صفحه ما یک session جدید را ایجاد کرده و مقدار تعدادی متغیر را برای آن تنظیم می کنیم، به صورت کد زیر:

```
<?php
// Start the session
session start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>
</body>
</html>
```

نکته مهم:

توجه داشته باشید که تابع ()session_start بایستی در بالای صفحه و قبل از هر تگ HTML ای تعریف شود.

آموزش خواندن مقادیر متغیر Session در PHP

در مرحله بعدی، یک صفحه PHP دیگر به نام "demo_session2.php" را ایجاد خواهیم کرد. در این صفحه جدید، به اطلاعات Session ای که در صفحه قبل ("demo_session1.php") ایجاد کردیم، دسترسی خواهیم داشت.

توجه داشته باشید که متغیرهای Session به صورت جداگانه به صفحه جدید ارسال نمی شوند. بلکه آن ها به یک باره توسط تابع ()session_start ای که در آغاز هر صفحه قرار داده ایم، خوانده می شوند.

همچنین توجه داشته باشید که تمامی متغیرهای session variables درون متغیر سراسری $_SESSION نگهدارای می شوند.

کد زیر، اطلاعات صفحه دوم طراحی شده (demo_session2.php) را نشان می دهد:

```
<?php
session_start();
?<
<!DOCTYPE html<
```

```
<html<
<body<
<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"] . ".<br<";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?<
</body<
</html<
```

راه دیگر برای نشان دادن متغیرهای Session مربوط به یک کاربر، استفاده از کد زیر است:

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
print_r($_SESSION);
?>
</body>
</html>
```

راهنمایی:

Sessionچطوری کار می کند و از کجا می داند من کی هستم؟

بسیاری از Sessionih ها یک کد مخصوص کاربر یا (user-key) را بر روی کامپیوتر وی تنظیم کرده که مشابه چیزی مثل +کد است. پس وقتی یک Session بر روی صفحه ای دیگر باز می شود، کامپیوتر را برای یافتن user-keyاسکن می کند. اگر کلید مشابهی پیدا کند به آن دسترسی پیدا کرده و از روی آن می فهمد کاربر کیست و تنظیمات مورد نظر وی را ارایه می دهد . اما اگر کلید مشابهی پیدا کند، یک Session جدید را بر روی سیستم شروع می کند.

تغییر یک متغیر PHP Session Variable

برای تغییر مقدار متغیر یکSession ، کافی است آن را همانند کد زیر رو نویسی یا overwrite کنید:

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
// to change a session variable, just overwrite it
$ SESSION["favcolor"] = "yellow";
print_r($_SESSION);
?>
</body>
```

```
</html>
```

آموزش از بین بردن یک PHP Session

برای از بین بردن و پاک کردن کلیه Session ها و متغیر های global session variables بر روی سیستم، همانند کد زیر از تابع های session_unset() و session_destroy() استفاده کنید:

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
// remove all session variables
session_unset();
// destroy the session
session_destroy();
?>
</body>
</html>
```

آموزش کار با Filters در زبان PHP

اعتبارسنجی اطلاعات یا (validating data) به معنای چک کردن صحت اطلاعات و این که آیا دیتای درست ارسال شده است یا خیر، می باشد.

اما پاک سازی یا (sanitizing data) به معنای حذف کاراکترهای غیر مجاز و اضافی از اطلاعات و ارسال دیتای خالص می باشد.

از Filter ها در زبان PHP برای اعتبارسنجی و پاک سازی اطلاعات ورودی استفاده می شود.

افزونه فیلتر در پی اچ پی یا PHP Filter Extension شامل تعداد زیادی تابع است که از آن ها می توان برای چک کردن اطلاعات ورودی کاربر استفاده کرد.این افزونه برای اعتبارسنجی راحت تر و سریع تر اطلاعات ورودی به کار می رود.

تابع filter_list() برای لیست کردن مواردی که افزونه PHP Filter جهت چک کردن اطلاعات پیشنهاد می دهد، استفاده می شود:

```
<table>
<tr>
<td>Filter Name</td>
<td>Filter ID</td>
</tr>
<?php
foreach (filter_list() as $id =>$filter) {
echo '<tr><td>' . $filter . '</td><td>' . filter_id($filter) .
'</td></tr>';
```

```
    }
  ?>
</table>
```

نرم افزارهای تحت وب، اطلاعات ورودی زیادی را مثل موارد زیر دریافت می کنند:

- اطلاعات ورودی توسط کاربر در فرم ها.
- کوکی ها یا.Cookies
- اطلاعات وب سرویس ها.
- متغیرهای سرور.
- نتایج حاصل از query ها در دیتابیس.

نکته مهم:

همواره بایستی اطلاعات ورودی را چک کنید!

اطلاعات اشتباه submit شده توسط فرم، می تواند باعث بروز مشکلات امنیتی شده و صفحه شما را از کار بیاندازد. با ساتفاده از PHP Filter ها می توانید مطمین شوید که نرم افزار تحت وب شما همواره اطلاعات درست دریفات می کند.

آموزش کار با تابع filter_var() در PHP

تابع Filter_var() می تواند اطلاعات ورودی را کنترل و پاکسازی کند.

تابع filter_var() یک متغیر تنها را با استفاده از یک فیلتر خاص کنترل می کند. این تابع دو پارامتر اصلی به صورت زیر دریافت می کند:

- نام متغیری که می خواهید اطلاعات آن را کنترل کند.
- نوع اطلاعاتی که می خواهید بر حسب آن نوع متغیر چک شود.

آموزش نحوه پاکسازی یک String در PHP

در کد مثال عملی زیر، با استفاده از تابع filter_var() کلیه تگ های HTML را از یک متغیر string حذف کرده ایم:

```
<?php
```

```
$str = "<h1>Hello World!</h1>";
$newstr = filter_var($str, FILTER_SANITIZE_STRING);
echo $newstr;
?>
```

آموزش چک کردن یک متغیر integer درPHP

در کد مثال عملی زیر، تابع filter_var() چک می کند که آیا متغیر $int یک متغیر عددی یا integer هست یا خیر.اگر متغیر $int یک متغیر integer باشد خروجی کد به صورت "Integer is Valid" و در غیر این صورت خروجی آن "Integer is not Valid" می باشد:

```
<!--?php
$int = 100;
if (!filter_var($int, FILTER_VALIDATE_INT) === false) {
echo("Integer is valid");
} else {
echo("Integer is not valid");
}
?-->
```

نکته مهم:

مشکل تابع filter_var() با عدد صفر

در کد مثال فوق، اگر مقدار متغیر $int صفر باشد، خروجی کد عبارت "Integer is not Valid" می باشد، در حالی که عدد صفر، یک متغیر است. برای حل این مشکل، از کد زیر استفاده کنید:

```
<?php
$int = 0;
if (filter_var($int, FILTER_VALIDATE_INT) === 0 || !filter_var($int,
FILTER_VALIDATE_INT) === false) {
echo("Integer is valid");
} else {
echo("Integer is not valid");
}
?>
```

آموزش چک کردن درستی یک IP با تابع filter_var()

کد مثال عملی زیر، از تابع filter_var() برای چک کردن این که آیا متغیر $ip یک معتبر است یا خیر استفاده می کند:

```
<?php
$ip = "127.0.0.1";
if (!filter_var($ip, FILTER_VALIDATE_IP) === false) {
echo("$ip is a valid IP address");
} else {
echo("$ip is not a valid IP address");
}
?>
```

آموزش اعتبارسنجی و پاکسازی یک Email با تابع filter_var()

در کد مثال عملی زیر، ابتدا با استفاده از تابع filter_var() کلیه کاراکترهای غیر مجاز و اضافی را از متغیر $email حذف کرده ایم. سپس چک کردیم آیا متغیر $email شامل یک آدرس ایمیل صحیح است یا خیر:

```
<?php
$email = "john.doe@example.com";
// Remove all illegal characters from email
$email = filter_var($email, FILTER_SANITIZE_EMAIL);
// Validate e-mail
if (!filter_var($email, FILTER_VALIDATE_EMAIL) === false) {
echo("$email is a valid email address");
} else {
echo("$email is not a valid email address");
}
?>
```

آموزش اعتبارسنجی و پاکسازی یک URL با تابع: filter_var()

در کد مثال عملی زیر، ابتدا با استفاده از تابع filter_var() کلیه کاراکترهای غیر مجاز و اضافی را از متغیر $url حذف کرده ایم. سپس چک کردیم آیا متغیر $url شامل یک آدرس URL صحیح است یا خیر:

```
<?php
$email = "john.doe@example.com";
// Remove all illegal characters from email
$email = filter_var($email, FILTER_SANITIZE_EMAIL);
// Validate e-mail
if (!filter_var($email, FILTER_VALIDATE_EMAIL) === false) {
echo("$email is a valid email address");
} else {
echo("$email is not a valid email address");
}
?>
```

مرجع کامل کار با Filter ها در PHP

برای دسترسی به مرجع کامل کار با Filter در زبان PHP به بخش آموزش کامل PHP Filter در سایت تحلیل داده بروید.

مرجع کامل آموزش PHP Filter حاوی توضیحات کامل و نکات کاربردی تابع های مختلف Filter کردن اطلاعات در زبان PHP است

آموزش پیشرفته Filter کردن اطلاعات در زبان PHP

در این درس به آموزش کارهای پیشرفته تر جهت Filter کردن اطلاعات در زبان PHP خواهیم پرداخت.

آموزش چک کردن مقدار یک عدد Integer در محدوده مورد نظر

در کد مثال زیر، به وسیله تابع filter_var() ابتدا چک کرده ایم ایا متغیر $int از نوع Integer هست یا خیر.

سپس بررسی کردیم آیا این عدد بین 1 تا 100 هست یا نه:

```
<?php
$int = 122;
$min = 1;
$max = 200;
if (filter_var($int, FILTER_VALIDATE_INT, array("options" =>
array("min_range"=>$min, "max_range"=>$max))) === false) {
echo("Variable value is not within the legal range");
} else {
echo("Variable value is within the legal range");
}
?>
```

آموزش اعتبارسنجی یک IPv6 با تابع filter_var()

در کد مثال عملی زیر، چک کرده ایم آیا متغیر $ip یک نوع IPv6 معتبر است یا خیر:

```
<?php
$ip = "2001:0db8:85a3:08d3:1319:8a2e:0370:7334";
if (!filter_var($ip, FILTER_VALIDATE_IP, FILTER_FLAG_IPV6) === false) {
echo("$ip is a valid IPv6 address");
} else {
echo("$ip is not a valid IPv6 address");
}
?>
```

آموزش چک کردن Query String در URL

در کد مثال عملی زیر، با استفاده از تابع filter_var() چک کرده ایم آیا متغیر $url یک URL دارای QueryStringاطلاعات اضافی ارسالی پس از آدرس صفحه هست یا خیر:

```
<?php
$url = "www.tahalildadeh.com";
if (!filter_var($url, FILTER_VALIDATE_URL, FILTER_FLAG_QUERY_REQUIRED)
=== false) {
echo("$url is a valid URL");
} else {
echo("$url is not a valid URL");
}
?>
```

آموزش حذف کاراکترهای کد ASCII از یک String

در کد مثال عملی زیر، از تابع filter_var() برای پاکسازی یک متغیر متنی String استفاده کرده ایم. این تابع هم تگ های HTML و هم کاراکترهای با کد ASCII بزرگتر از 127 از متغیر str 4حذف می کند:

```
<?php
$str = "<h1>Hello WorldÆ►Å!</h1>";
```

```
$newstr = filter_var($str, FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_HIGH);
echo $newstr;
?>
```

مرجع کامل کار با Filter ها در زبان PHP

برای دسترسی کامل به مرجع توابع filter در زبانPHP ، به بخش آموزش کامل PHP Filter در سایت تحلیل داده بروید.

مرجع کامل آموزشPHP Filter ، شامل توضیحات و مثال های کامل درباره تابع های مختلف Filter کردن اطلاعات در زبان PHP است.

آموزش مدیریت خطا (Error Handling) در PHP

سیستم مدیریت خطای پیش فرض یا default error handling در PHP بسیار ساده عمل می کند. این سیستم در هنگام بروز خطا، یک پیام هشدار حاوی نام فایل، شماره خط بروز خطا به همراه توصیفی از شرح خطا را به مرور گر ارسال می کند.

در هنگام طراحی اسکریپت ها و نرم افزارهای تحت وب، مدیریت خطا یا errorHandling بخش مهمی از کد می باشد. اگر کد شما نتواند error های احتمالی را چک کند، برنامه تان بسیار غیر حرفه ای به نظر رسیده و می تواند شما را با چالش های امنیتی رو به رو کند.

در این بخش آموزشPHP ، مهم ترین متدهای چک کردن خطا در زبان PHP را بررسی خواهیم کرد. این متدهای مدیریت خطا عبارتند از:

- دستورهای ساده با تابع."die()"
- errorهای خاص و بررسی تحریک کننده خطا.(error triggers)
- آموزش نحوه گزارش کردن خطاها یا.(error reporting)

آموزش روش پایه مدیریت خطا-PHP تابع die()

در کد مثال اول مدیریت خطا، به وسیله اسکریپت زیر برنامه قصد دارد تا یک فایل متنی (text file) را باز کند:

```
<!--?php
$file=fopen("welcome.txt","r");
?-->
```

اگر در کد فوق، فایل مورد نظر وجود نداشته باشد، احتمالا با خطاهایی همانند متن زیر مواجه خواهید شد:

```
<?php
Warning: fopen(welcome.txt) [function.fopen]: failed to open stream:
No such file or directory in C:\webfolder\test.php on line 2
?>
```

برای جلوگیری از نمایش پیغام خطایی مثل متن فوق به کاربر، کد را به صورت زیر تغییر داده ایم. در این کد جدید ابتدا برنامه وجود فایل را بررسی کرده و سپس در صورت موجود بودن آن، اقدام به باز کردن فایل می کند:

```
<!--?php
if(!file_exists("welcome.txt")) {
die("File not found");
} else {
$file=fopen("welcome.txt","r");
}
?-->
```

حال اگر فایل مورد نظر وجود نداشته باشد، برنامه پیغام خطای زیر را صادر می کند:

```
File not
1
```

کد روش دوم خطایابی، بسیار بهتر و تاثیرگذارتر از روش اول است. زیرا با استفاده از یک مکانیزم ساده خطایابی، قبل از بروز خطا، اسکریپت را از ادامه کار متوقف می کند. اما، همواره متوقف کردن اسکریپت از ادامه اجرا، روش خوبی برای مدیریت خطا نیست. در ادامه به گزینه های دیگری که زبان PHP برای مدیریت خطا در اختیارمان قرار داده است، خواهیم پرداخت.

ایجاد یک مدیریت کننده خطا دلخواه (Custom Error Handling)

ایجاد یک مدیریت کننده دلخواه خطا (Custom Error Handling) در زبانPHP ، کار نسبتا ساده ای است. می توانیم به راحتی یک تابع دلخواه PHP را ایجاد کنیم تا در هنگام بروز خطا، فراخوانی شود. تابع دلخواهی که برای مدیریت خطا ایجاد می کنیم، بایستی حداقل بتواند دو پارامتر خطا) درجه خطا (error level)و پیام خطا ((error message) را مدیریت کند و تا پارامتر اختیاری را دریافت کند (پارامترهای اختیاری مثل نام فایل، شماره خط کد بروز خطا و محتویات خطا).

ساختار کلی تعریف یک تابع مدیریت کننده دلخواه در PHP به صورت زیر است:

```
<?php
error_function(error_level,error_message,error_file,error_line,error_context)
?>
```

پارامترهای تابع مدیریت کننده خطا در PHP				
error_level	error_message	error_file	error_line	error_context
تعیین این پارامتر اجباری بوده و درجه گزارش خطا را برای error مورد نظر کاربر تعیین می کند. بایستی یک مقدار عددی باشد که انواع حالت های ان را در جدول بعدی توضیح داده ایم.	پارامتر اجباری. این پارامتر پیام خطا مربوط به error رخ داده را تعیین می کند.	پارامتر اختیاری. این پارامتر نام تابعی که خطا در آن رخ داده است را مشخص می کند.	پارامتر اختیاری. این پارامتر شماره و آدرس خط کدی که خطا در آن رخ داده را تعیین می کند.	پارامتر اختیاری. این پارامتر یک آرایه است که نام و مقدار کلیه متغیرهای برنامه را در هنگام بروز خطا ثبت و نگهداری می کند.

جدول درجه های خطا در PHP

در بخش قبل گفتیم که هر خطای رخ داده در برنامه های PHP شامل یک درجه یا level است که درجات آن را در لیست زیر مشاهده می کنید:

پارامترهای تابع مدیریت کننده خطا در PHP						
E_WARNING	E_NOTICE	E_USER_ERROR	E_USER_WARNING	E_USER_NOTICE	E_RECOVERABLE_ERROR	E_ALL
درجه خطاهای غیر مخرب در زمان اجرا است که باعث توقف اجرای اسکریپت نمی شوند.	این درجه شامل اخطارهای زمان اجرا می شود. اسکریپت ممکن است چیزی مشاهده کند که ممکن است خطا باشد، یا این که این اشکالات می تواند در روند عادی اجرای اسکریپت رخ دهد.	این درجه شامل خطاهای مخربی می شود که توسط کد کاربر رخ داده اند. این نوع خطاها شبیه یک E_ERROR هستند که توسط کاربر و با استفاده از تابع trigger_error() تنظیم شده اند.	این درجه شامل اخطارهای غیر مخربی می شود که توسط کد کاربر رخ می دهد. این نوع اخطارها مثل یک E_WARNING هستند که توسط کاربر و با استفاده از تابع trigger_error() تنظیم شده اند.	این درجه شامل اعلان های غیر مخربی می شود که توسط کد کاربر رخ می دهد. این نوع اعلان های مشکل توسط کاربر و با استفاده از تابع trigger_error() تنظیم شده اند.	این درجه خطا شامل خطاهای مخرب ولی قابل ردیابی است. این نوع error همانند یک E_ERROR است که توسط یک تابع مدیریت خطا (همانند Set_error_handler) شناسایی شده است.	این درجه از خطا شامل تمام error های رخ داده در سطح برنامه می شود.

اکنون بیایید یک تابعی ایجاد کنیم تا خطاهای موجود را مدیریت یا handle کند. کد تابع به صورت زیر است:

```
<?php
function customError($errno, $errstr) {
echo "<b>Error:</b> [$errno] $errstr<br>";
echo "Ending Script";
die();
}
?>
```

کد فوق یک تابع ساده مدیریت خطا (Error Handler) را نشان داده که در زمان فراخوانی شدن دو مقدار دریافت می کند. یک مقدار درجه خطا (error level) و دیگر پیام مرتبط با خطا . (error message) سپس تابع درجه و پیام خطا را در خروجی نشان داده و اجرای اسکریپت را متوقف می کند. اکنون که یک تابع مدیریت کننده خطا را طراحی کردیم، بایستی تعیین کنیم چه زمانی لازم است این تابع فراخوانی و اجرا (trigger) شود.

آموزش تنظیم تابع مدیریت خطا در PHP

تابع مدیریت خطای پیش فرض در PHP یک تابع پیش ساخته است که در سورس اصلی این زبان تعریف شده. در این درس قصد داریم تا تابعی که خودمان طراحی کردیم را به عنوان تابع پیش فرض مدیریت خطا در برنامه قرار دهیم.

این امکان نیز وجود دارد که یک تابع مدیریت خطا را به گونه ای تنظیم کنیم تا فقط برای برخی error ها اجرا شود، در این صورت می توان error های مختلف را به روش های گوناگون مدیریت کرد. اما در مثال عملی این درس، قصد داریم تابع خود را برای مدیریت تمامی خطاهای برنامه تنظیم کنیم:

```
<?php
set_error_handler("customError");
?>
```

از آنجایی که می خواهیم تابع مدیریت خطای دلخواه ما، تمامی error های احتمالی برنامه را مدیریت کند، تابع set_error_handler() فقط به یک پارامتر نیاز خواهد داشت و در پارامتر دوم می توان درجه خطا را تنظیم کرد.

مثال عملی:

در کد مثال عملی زیر، قصد داریم تا تابع مدیریت خطا را با تلاش برای چاپ مقدار یک متغیر که وجود ندارد، امتحان کنیم:

```
<!--?php
//error handler function
function customError($errno, $errstr) {
echo "<b-->Error: [$errno] $errstr";
}
//set error handler
set_error_handler("customError");
//trigger error
echo($test);
?>
```

خروجی کد مثال فوق به صورت زیر خواهد بود:

```
Error: [8] Undefined variable
1
```

فعال کردن یک خطا یا error trigger

در اسکریپت هایی که کاربر می تواند اطلاعاتی را درون کادرهای متن وارد کند، فعال کردن error ها در زمان ارسال مقادیر غیر مجاز بسیار کاربردی است. در زبان PHP این کار با استفاده از تابع trigger_error() انجام می شود.

مثال عملی:

در کد مثال عملی زیر، در صورتی که مقدار متغیر "test" بیشتر از یک باشد، خطا مورد نظر فعال می شود:

```
<!--?php
$test=2;
if ($test-->=1) {
trigger_error("Value must be 1 or below");
}
?>
```

خروجی کد مثال فوق به صورت زیر خواهد بود:

```
Notice: Value must be 1 or
in C:\webfolder\test.php on
1
2
```

یک error دلخواه را می توان در هر کجای اسکریپت که لازم باشد، فراخوانی کرده و پارامتر دوم در تابع تعیین کننده درجه خطا (error level) می باشد. انواع مختلف error های محتمل عبارتند از:

پارامترهای تابع مدیریت کننده خطا در PHP		
E_USER_ERROR	E_USER_WARNING	E_USER_NOTICE
این دسته شامل خطاهای مخربی می شوند که توسط کد کاربر تولید شده اند. در این نوع خطاها دیگر نمی توان برنامه را از نقطه بروز خطا باز گرداند و پردازش اسکریپت متوقف خواهد شد.	این نوع خطا شامل اخطارهایی می شود که توسط کد کاربر در زمان اجرا رخ داده؛ ولی باعث توقف اجرای اسکریپت نمی شوند.	این درجه خطا، درجه پیش فرض کلیه error های احتمالی بوده و درواقع یک اعلان است که در هنگام بروز خطا در کد در حال اجرا صادر می شود.

در این حالت اسکریپت ممکن است به موردی برخورد کرده باشد که از نظر آن خطا بوده، ولی امکان دارد در هنگام اجرای برنامه به صورت عادی رخ دهد.

مثال عملی:

در کد مثال عملی زیر، در صورتی که مقدار متغیر "test" از بیشتر باشد، یک E_USER_WARNING رخ خواهد داد. اگر E_USER_WARNING رخ دهد، ما از تابع مدیریت خطای مورد نظر خود استفاده کرده و اجرای اسکریپت را متوقف خواهیم کرد:

```
<!--?php
//error handler function
function customError($errno, $errstr) {
echo "<b-->Error: [$errno] $errstr<br>";
echo "Ending Script";
die();
}
//set error handler
set_error_handler("customError",E_USER_WARNING);
//trigger error
$test=2;
if ($test>=1) {
trigger_error("Value must be 1 or below",E_USER_WARNING);
}
?>
```

خروجی کد مثال فوق به صورت زیر خواهد بود:

```
1 Error: [512] Value must be 1 or
2 Ending
```

هم اکنون که با نحوه طراحی error های دلخواه در سطح برنامه و فراخوانی آن ها آشنا شدید، قصد داریم تا نحوه ثبت کردن خطا یا (error logging) را در PHP آموزش دهیم.

آموزش ثبت خطا یا error logging در PHP

به صورت پیش فرض و بر حسب این که مقدار error-log در فایل تنظیمات برنامه) فایل (php.ini چگونه تنظیم شده باشد، PHPدر هنگام بروز خطا، یک کد ثبت خطا یا error log را به سیستم خطاها یا به یک فایل خاصی ارسال می کند.

با استفاده از تابع ()error_log می توانید ثبت جزئیات خطاها (error log) را به یک فایل خاص یا یک سرور راه دور ارسال کنید. همچنین ارسال پیام های خطا به وسیله email نیز یک راه بسیار مطمئن جهت اطلاع از بروز خطا در سیستم است.

اموزش نحوه ارسال یک پیام خطا (Error Message) با ایمیل

در کد مثال عملی زیر، در صورتی که یک خطای خاص رخ دهد، پیام خطایی را به وسیله ایمیل به کاربر ارسال کرده و اجرای اسکریپت را متوقف کرده ایم:

```
<!--?php
//error handler function
```

```
function customError($errno, $errstr) {
echo "<b-->Error: [$errno] $errstr<br>";
echo "Webmaster has been notified";
error_log("Error: [$errno] $errstr",1,"someone@example.com","From: webmaster@example.com");
}
//set error handler
set_error_handler("customError",E_USER_WARNING);
//trigger error
$test=2;
if ($test>=1) {
trigger_error("Value must be 1 or below",E_USER_WARNING);
}
?>
```

خروجی کد مثال به صورت زیر است که توسط ایمیل به کاربر ارسال می شود:

```
1 Error: [512] Value must be 1 or
2 Webmaster has been no
```

در نهایت، متن ایمیل خطایی که کاربر دریافت می کند، به صورت زیر خواهد بود:

```
1 Error: [512] Value must be 1 or
```

البته ارسال خطا به وسیله ایمیل در موارد حیاتی ضروریت داشته و مابقی error ها را بایستی توسط سیستم ثبت کننده خطاهای PHP در فایل log ذخیره نمود.

آموزش کار با استثناء ها (Exception) در زبانPHP

استثناءها (Exception) در زبانPHP ، برای تغییر فرایند اولی برنامه یا اسکریپت، در زمانی که یک خطای خاصی رخ دهد، استفاده می شوند.

در این درس به آموزش کار با Exception و نحوه مدیریت خطا در برنامه خواهیم پرداخت .

یک استثناء یا Exception چیست؟

همزمان با ارائه نسخه جدیدPHP5 ، یک راه جدید شی گرا (object oriented)برای مدیریت خطاها یا errors برنامه معرفی شد.

مدیریت استثناءها (Exception handling) برای تغییر روند اجرای برنامه در زمانی که یک خطای خاص (exceptional)رخ دهد، استفاده می شود. این شرط یا خطای خاص را Exception می گویند.

در لیست زیر، مراحل عادی که در هنگام فعال شدن یک Exception رخ می دهد، به ترتیب بیان شده است :

- وضعیت جاری کد برنامه ذخیره می شود.

- اجرای کد برنامه به تابع دلخواه و تعیین شده ای که وظیفه مدیریت Exception را دارد، سوئیچ می کند.
- برحسب شرایط، مدیریت کننده کد برنامه یاhandler ، ممکن است اجرای کد را از نقطه ای که ذخیره کرده شروع نماید، یا این که اجرای کل برنامه را متوقف نموده و یا ادامه اجرای برنامه ای مشخص شده در کد از سر گیرد.

در این درس، متدهای مختلف مدیریت Exception را به شرح زیر بررسی خواهیم کرد :

- نحوه ساده استفاده از.Exception
- طراحی یک مدیریت کننده Exception Handler دلخواه.
- مدیریت خطاهای چندگانه یا.Multiple exception
- طراحی و تنظیم یک مدیریت کننده سطح بالا خطا یا.top level exception

نکته :

از Exception فقط بایستی در صورت بروز یک خطای خاص استفاده نموده و نمی توان از آن ها برای پرش از یک نقطه خاص در کد به نقطه دیگر استفاده کرد .

آموزش نحوه ساده استفاده از Exception

هنگامی که یک exception در برنامه رخ می دهد، کد بعد از آن اجرا نخواهد شد و PHP به جستجوی ساختار دستوری "catch" متناظر با آن خواهد پرداخت.

اگر Exception رخ داده در برنامه را نتوان ردگیری و دریافت کرد، یک خطای آسیب زننده (fatal error) به همراه یک پیام با مضمون "خطای یافت نشده" خواهد داد.

در کد زیر، یک exception را در برنامه ایجاد کرده ایم، بدون این که آن را ردگیری کنیم :

```
<!--?php
//create function with an exception
function checkNum($number) {
if($number-->1) {
throw new Exception("Value must be 1 or below");
}
```

```
    return true;
}

//trigger exception
checkNum(2);
?>
```

در صورت اجرای کد فوق، پیام خطای زیر صادر می شود:

```
<?php
Fatal error: Uncaught exception 'Exception'
with message 'Value must be 1 or below' in C:\webfolder\test.php:6
Stack trace: #0 C:\webfolder\test.php(12):
checkNum(28) #1 {main} thrown in C:\webfolder\test.php on line 6
?>
```

آموزش کار با ساختارهای دستوری Catch وthrow ، Try

برای جلوگیری از صدور پیام خطایی مثل کد فوق، بایستی یک ساختار لازم جهت مدیریت یک exception را ایجاد کنیم.

یک ساختار درست جهت مدیریت خطا یا exception بایستی شامل موارد زیر باشد :

- **بخش** : Try تابعی که از یک exception استفاده می کند، بایستی در بلوک کد "try" تعریف شود. اگر exception فعال نشود، فرآیند اجرای کد حالت نرمال خود را طی خواهد کرد. اما اگر یک exceptionفعال شود، در اصطلاح می گوییم که آن Exception صادر یا "trown" شده است.
- **بخش** : Throw بخش Throw مشخص کننده نحوه فعال سازی یک exception است. هر "throw" بایستی حداقل دارای یک بخش "catch" نیز باشد .
- **بخش** : Catch بلاک "Catch" استثناء یا exception رخ داده را دریافت نموده و یکی شی (object) حاوی اطلاعات مربوط به exception را ایجاد می کند .

حال بیایید exception مثال قبل با یک ساختار کامل و درست مجددا فعال کنیم :

```
<!--?php
//create function with an exception
function checkNum($number) {
if($number-->1) {
throw new Exception("Value must be 1 or below");
}
return true;
}

//trigger exception in a "try" block
try {
checkNum(2);
```

```
//If the exception is thrown, this text will not be shown
echo 'If you see this, the number is 1 or below';
}

//catch exception
catch(Exception $e) {
echo 'Message: ' .$e->getMessage();
}
?>
```

در صورت اجرای کد فوق، پیام هشدار زیر صادر می شود:

```
Message: Value must be 1 or below
```

توضیح کد مثال فوق:

همانطور که مشاهده کردید، کد مثال عملی فوق یک exceptionرا فعال کرده و آن را دریافت می کند. مراحل زیر در طی انجام کار رخ می دهند :

- تابع chechNum() ایجاد می شود. این تابع چک می کند آیا عدد یا number وارد شده، بزرگتر از 1 می باشد یا خیر. در صورت بزرگتر بودن از 1، خطا یا exception روی خواهد داد.
- تابع CheckNum() درون ساختار "try" فراخوانی می شود .
- exceptuionکد به همراه تابع CheckNum() صادر می شود.
- پیام خطا حاصل از فعال شدنexception ، به وسیله فراخوانی کد getMessage()<-$e از شی مربوط بهexception ، صادر می شود.

اما به هر حال، راه حل دور زدن قانون این که "هر throw بایستی یک ساختار Catch داشته باشد"، طراحی یک مدیریت کننده exception سطح بالاست که در پایان این درس به اموزش آن خواهیم پرداخت .

ایجاد یک کلاس دلخواه : Exception Class

برای ایجاد یک مدیریت کننده exception دلخواه، بایستی یک کلاس ویژه با توابعی که در زمان فعال شدن exceptionفراخوانی خواهند شد، را تعریف کنید. این کلاس بایستی یک زیرمجموعه یا فرزند از کلاس اصلی exception classدر PHP باشد.

Exception classدلخواهی که ایجاد کرده اید، خواص (properties)خود را از کلاس اصلی PHP

Exception Classبه ارث برده و شما می توانید توابع مورد نیاز خود را نیز بدان اضافه کنید.

در کد مثال عملی زیر، یک کلاس exception class را ایجاد کرده ایم :

```
<!--?php
class customException extends Exception {
public function errorMessage() {
//error message
$errorMsg = 'Error on line '.$this--->getLine().' in '.$this->getFile()
.': <b>'.$this->getMessage().'</b> is not a valid E-Mail address';
return $errorMsg;
}
}

$email = "someone@example...com";

try {
//check if
if(filter_var($email, FILTER_VALIDATE_EMAIL) === FALSE) {
//throw exception if email is not valid
throw new customException($email);
}
}

catch (customException $e) {
//display custom message
echo $e->errorMessage();
}
?>
```

کلاس جدید ایجاد شده، درواقع یکی کپی از کلاس exeption class قدیمی است که تابع errorMessage() را به آن اضافه کرده اید. از آنجایی که کلاس جدید یک کپی از کلاس اصلی و قدیمی exception class بوده و خواص و متدهای آن را به ارث برده است، می توانید در کلاس جدید از متدهای کلاس قدیمی مثل تابع های getline()، getFile()و getMessage() استفاده کنید.

توضیح مثال : کد مثال عملی این بخش، یک exception را فعال کرده و سپس با استفاده از یک کلاس exceptionآن را ردگیری می کند .مراحل انجام کار به صورت زیر است :

1. کلاس customException() به عنوان یک مدل توسعه یافته از کلاس قدیمی exception class ایجاد شده است. از آنجایی که این کلاس از کلاس قدیمی به ارث رفته است، تمامی خواص و متدهای آن کلاس را به ارث برده است.

2. تابع errorMessage() ایجاد می شود. این تابع در صورت وارد کردن یک ایمیل نادرست، پیام خطا صادر می کند.

3. متغیر $email با یک مقدار متنی (string) پر می شود که حاوی یک ایمیل نادرست است.
4. بلوک دستوری "try" اجرا شده و از آنجایی که ایمیل وارد شده درست نیست، exceptionتعیین شده روی می دهد.
5. بلوک دستوری"catch" ، خطا یا exception رخ داده را دریافت کرده و پیام خطا را نمایش می دهد.

آموزش کار با exception چندگانه :

این امکان وجود دارد که در یک اسکریپت، از چندین exceptionکه به آن Multiple Exception می گوید، برای چک کردن شرایط مختلف کد استفاده کنید. همچنین می توانید از چندین دستورif ... else ، ساختار switchو یا exception های چندگانه استفاده کنید. این خطا یا exceptuon ها می تواند از کلاس مختلف exception class استفاده کرده و پیام های هشدار متفاوتی صادر کنند .

مثال عملی :

در کد مثال عملی زیر، نحوه استفاده از exception چندگانه نشان داده شده است. کد را مرور کنید، در ادامه به تشریح مثال خواهیم پرداخت :

```
<!--?php
class customException extends Exception {
public function errorMessage() {
//error message
$errorMsg = 'Error on line '.$this--->getLine().' in '.$this->getFile()
.': <b>'.$this->getMessage().'</b> is not a valid E-Mail address';
return $errorMsg;
}
}

$email = "someone@example.com";

try {
//check if
if(filter_var($email, FILTER_VALIDATE_EMAIL) === FALSE) {
//throw exception if email is not valid
throw new customException($email);
}
//check for "example" in mail address
if(strpos($email, "example") !== FALSE) {
throw new Exception("$email is an example e-mail");
}
}
```

```
catch (customException $e) {
    echo $e->errorMessage();
}

catch(Exception $e) {
    echo $e->getMessage();
}
?>
```

توضیح مثال :

کد مثال عملی فوق، دو شرط یا Condition متفاوت را چک کرده و در صورتی که هر کدام از آن ها درست نباشند، exceptuionتعیین شده صادر می شود.

1. کلاس customException() به عنوان یک نسخه جدید از کلاس قدیمی exception ایجاد می شود. در این حالت، کلاس جدید تمامی خواص و متدهای کلاس قدیمی را به ارث می برد.
2. تابع errorMessage() ایجاد شده و این تابع در صورتی که ایمیل وارد شده نادرست باشد، یک پیام خطا را بر می گرداند.
3. متغیر $email با یک مقدار متنی (string) که حاوی یک ایمیل درست می باشد، پر شده ولی شامل کلمه "example" خواهد بود.
4. بلوک دستوری "try" اجرا شده و به دلیل درست بودن شرط اول کد، خطا یا exception روی نمی دهد .
5. شرط دوم کد به دلیل وجود کلمه "example" در متغیر ایمیل ($email) یک خطا یا exception صادر می کند.
6. بلوک دستوری"Catch" ، خطا یا exception صادر شده را دریافت کرده و یک پیام هشدار صادر می کند.

آموزش ارسال مجدد خطا یا exception

گاهی اوقات ممکن است در هنگام رخ دادن یکexception ، آن را به شیوه ای متفاوت از حالت استاندارد مدیریت کنید. این امکان وجود دارد که یک exception خاص را به وسیله یک بلوک کد "Catch" مجددا فعال یا ارسال نمایید.

یک اسکریپت بایستی خطاهای سیستم را از کاربر مخفی نگه دارد. خطاهای سیستمی برای کدنویس پروژه اهمیت زیادی دارد، ولی کاربران معمولی علاقه ای به این مباحث ندارند. جهت آسان تر کردن مسائل برای کاربر، می توانید exceptionرخ داده را با ارسال یک پیام مناسب و کاربر پسند، مجددا نمایش دهید .

```
<!--?php
class customException extends Exception {
public function errorMessage() {
//error message
$errorMsg = $this--->getMessage().' is not a valid E-Mail address.';
return $errorMsg;
}
}

$email = "someone@example.com";

try {
try {
//check for "example" in mail address
if(strpos($email, "example") !== FALSE) {
//throw exception if email is not valid
throw new Exception($email);
}
}
catch(Exception $e) {
//re-throw exception
throw new customException($email);
}
}

catch (customException $e) {
//display custom message
echo $e->errorMessage();
}
?>
```

توضیح مثال :

کدد مثال عملی فوق، چک می کند آیا کلمه "example" در آدرس ایمیل وجود دارد یا خیر. اگر وجود داشته باشد، خطا یا exception را مجددا ارسال (re-thrown) می کند، مراحل انجام کار به صورت زیر است:

1. کلاس customException() به عنوان یک نسخه جدید از روی کلاس exception ایجاد می شود، به همین دلیل این کلاس، کلیه خواص و متدهای کلاس اصلی را به ارث می برد.
2. تابع errorMessage() ایجاد می شود. این تابع در صورتی که آدرس ایمیل وارد شده، نادرست باشد یک پیام هشدار را بر می گرداند.

3. متغیر $email با یک متغیر متنی (string) که شامل کلمه "example" نیز خواهد بود، پر می شود.
4. خطا یا exception به دلیل وجود کلمه "example" در ایمیل، فعال خواهد شد.
5. بلوک دستوری "catch" خطا یا exception رخ داده را دریافت کرده و یک "customException" دریافت شده و یک پیام هشدار صادر می کند.

اگر خطا یا exception در بلوک "try" جاری آن گرفته نشود، برنامه به دنبال دریافت آن در بلوک "catch" و یا مراتب بالاتر خواهد گشت .

آموزش تنظیم یک مدیریت کننده Exception سطح بالا

تابع set_exception_handler() یک تابع تعیین شده توسط کاربر را به عنوان مدیریت کننده کلیه خطاها یا exceptionهای گرفته نشده و رصد نشده در برنامه، تعیین می کند. کد زیر نحوه انجام کار را نشان خواهد داد:

```
<!--?php
function myException($exception) {
echo "<b-->Exception: " . $exception->getMessage();
}

set_exception_handler('myException');

throw new Exception('Uncaught Exception occurred');
?>
```

خروجی کد مثال فوق به صورت زیر خواهد بود:

```
Exception: Uncaught Exception occurred
```

در کد مثال عملی فوق هیچ ساختار دستوری "catch" ای تعریف نشده است. به جای آن، در صورت بروز هر خطا، مدیریت کننده exception سطح بالا یا top level exception handler فعال خواهد شد. این تابع کلیه خطاهای گرفته نشده در برنامه را دریافت و مدیریت می کند .

قوانین مربوط به exception در زبان PHP

معمولا کدها را در ساختارهای "try" قرار می دهیم تا امکان ردگیری بالقوه exception را افزایش دهیم.

هر بلوک دستوری "block" یا "throw" بایستی حداقل یک بلوک کد "catch" متناظر داشته باشد.

از exception های چندگانه می توان برای گرفتن کلاس های مختلف eception های برنامه استفاده نمود.هر Exceptionرا می توان درون یک ساختار Catch متعلق به یک بلوکtry ، ارسال یا ارسال مجدد-re) thrown)کرد.

توجه :

اگر یک خطا یا exception را ارسال می کنید (throw) حتما بایستی آن را دریافت (cathch) کنید .

بخش چهارم : آموزش پایگاه داده MySQL درPHP

آموزش کار با پایگاه داده MySQL درPHP

با استفاده از زبان PHP ، می توانید به یک پایگاه داده متصل شده و اطلاعات را ذخیره و ویرایش کنید. پایگاه داده MySQL محبوب ترین پایگاه داده ای است که در سیستم برنامه نویسی PHP استفاده می شود .

پایگاه داده MySQL چیست؟

پایگاه داده MySQL یک سیستم پایگاه داده است که بر روی وب استفاده می شود.

MySQLیک سیستم پایگاه داده می باشد که بر روی سرور اجرا می شود.

MySQLیک database ایده آل برای پروژه های کوچک و بزرگ است.

پایگاه MySQL بسیار سریع، قابل انعطاف و ساده برای کارکردن است.

پایگاه داده MySQL از زبان استاندارد SQL برای کدنویسی استفاده می کند.

پایگاه داده SQL را می توان بر روی طیف وسیعی از پلتفرم ها اجرا نمود.

پایگاه داده SQL یک database این سورس بوده و دانلود و استفاده از آن رایگان می باشد.

پایگاه داده MySQL توسط شرکت اورالک طراحی، توزیع و پشتیبانی می شود.

اطلاعات پایگاه داده MySQL درون جدول (Table) ذخیره می شود. یک جدول یا Table مجموعه ای از داده های مرتبط به هم می باشد که از تعدادی سطر (row) و ستون (column) تشکیل شده است. معمولا پایگاه داده ها (Database) برای نگهداری اطلاعات دسته بندی شده مناسب هستند. برای مثال یک شرکت ممکن است شامل یک پایگاه داده با جدول های مثل لیست زیر باشد :

- Employees
- Products
- Customers
- Order

سیستم کارکرد پایگاه داده MySQL با PHP

کارکرد سیستم PHP و پایگاه داده MySQL به صورت cross-platformاست، به این معنی که PHP را می توان بر روی یک سرور مثل ویندوز اجرا کرد، در حالی که پایگاه داده MySQL بر روی یک سرور با سیستم عامل دیگری مثل Unix قرار داشته باشد. به عبارت دیگر می توان سرور PHP و MySQLرا با سیستم عامل های متفاوتی اجرا کرده و مشکلی رخ نمی دهد .

آموزش جستجو در پایگاه داده یا Database Query

به بیان ساده یک Query در پایگاه داده، یک سوال یا جستجو یا درخواست (request) می باشد. شما می توانید یک database را برای یافتن اطلاعات خاصی جستجو کرده) با استفاده از یک (Query و سپس نتایج را در یک یا چند رکورد دریافت نمایید.

به Query زیر که با زبان استاندارد SQL نوشته شده است، دقت کنید :

```
SELECT LastName FROM Employees
```

Queryفوق کلیه اطلاعات درون ستون "LastName" پایگاه داده "Employees" را انتخاب می کند. برای یادگیری زبانSQL ، به بخش آموزش پایگاه داده SQL در سایت تحلیل داده بروید .

آموزش نحوه دانلود پایگاه داده : MySQL

اگر دارای یک سرور PHP هستید که پایگاه داده MySQL بر روی آن نصب نشده است، می توانید سورس این پایگاه داده را از آدرس www.mysql.com دانلود و نصب نماید .

واقعیت هایی درباره پایگاه داده MySQL

پایگاه دادهMySQL ، database بالفعل و در حال استفاده برای سیستم های عظیم با حجم اطلاعات گسترده و تعداد کاربران زیاد مثل سایت های Facebook، twitter و یا ویکی پدیا است.
از طرف دیگر، پایگاه داده MySQL را می توان برای اجرای پایگاه داده های سبک و کم حجم نیز استفاده کرد.
با مراجعه به آدرس www.mysql.com/customers/ می توانید لیست کاملی از شرکت ها و وب سایت های استفاده کننده از پایگاه داده MySQL را مشاهده کنید .

آموزش اتصال Connect به پایگاه داده MySQL

نسخه های PHP5 و بالاتر برای اتصال به یک پایگاه داده MySQL از موارد زیر استفاده می کنند :

- MySQLi extension حرف i مخفف عبارت پیشرفته تر یا improved است
- اشیای داده ای PHP با PDO (PHP Data Objects).

نسخه های قدیمی تر PHP از MySQL extension برای اتصال به MySQL Daqtabase استفاده می کردند.
اما این افزونه از سال 2012 به بعد، منسوخ شده است .

کی و کجا از MySQL یا PDO استفاده کنیم؟

اگر بخواهیم خلاصه بگوییم کی و از کجا از MySQLi یا PDO استفاده کنیم، جواب اینه "هر جا و هرجور که دوست داشتید."!

اما هر دو MySQLi و PDO مزایای خاص خود را دارند.

PDO می تواند با 12 سیستم مختلف پایگاه داده کار کند، در حالی که MySQLi فقط می تواند با پایگاه داده MySQL کار کند.

بنابراین اگر بخواهید پایگاه داده پروژه خود عوض کنید، PDO کار را بسیار راحت تر خواهد کرد. در این حالت فقط نیاز دارید تا رشته ارتباطی (Connection String) را به همراه تعدادی از query ها تغییر دهید، اما در صورت استفاده از MySQL بایستی کل کد پروژه را از اول نوشته و query ها را نیز تغییر

دهید.

هر دو روش PDO و MySQLi روش برنامه نویسی شی گرا (oop) را ارائه داده، ولی استفاده از روش برنامه نویسی رویه ای در MySQLi نیز امکان پذیر است.

هر دو روش از دستورات امن و مناسب SQL یا Prepared Statment پشتیبانی می کنند. دستورات امن SQLسیستم را از حملات تزریق SQL injection محافظت کرده که برای امنیت نرم افزار تحت وب بسیار ضروری است.

ارائه مثال های عملی با ساختار دستوری PDO و MySQLi

در این درس و درس های آینده ما هر 3 روش کار با پایگاه داده MySQL را آموزش داده و مثال های عملی مرتبط با آن ها را ارائه می دهیم :

- روش MySQL به صورت شی گرا یا.Object-oriented
- روش MySQL به صورت رویه ای یا.Procedural
- روش.PDO

آموزش نصب MySQLi

در سیستم های ویندوز وLinux ، هنگامی که پکیج PHP5 MySQL را نصب می کنند، معمولا افزونه MySQL نیز به صورت خودکار نصب می شود.

برای دریافت اطلاعات بیشتر جهت نصب MySQL به آدرس php.net/manual/en/mysqli.installation.phpبروید .

آموزش نصب PDO

برای دریافت آموزش های نصب لازم جهت PDO به آدرس php.net/manual/en/pdo.installation.phpبروید .

آموزش باز کردن یا اتصال یا Connection به MySQL

قبل از این که بتوانیم به اطلاعات درون پایگاه داده MySQL دسترسی داشته باشیم، بایستی یک اتصال یا Connectionرا به سرور برقرار کنیم. کد زیر نحوه اتصال به MySQL را به صورت عملی نشان داده است :

کد مثال شی گرا MySQLi

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

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

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

راهنمایی :

به یک نکته مهم در زمینه کارکرد شی گرایی مثال بالا اشاره می کنیم .

خاصیت $Connect-error در نسخه های قبل از PHP5.2.9 و 5.3.0، دچار مشکل می شد. اگر می خواهید از سازگاری دستور را با نسخه های قبل از PHP5.3.0 مطمئن شوید، از کد اصلاح شده زیر به جای کد فوق استفاده کنید .

کد مثال MySQL رویه ای:

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

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

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

کد مثال PDO :

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

try {
$conn = new PDO("mysql:host=$servername;dbname=myDB", $username,
$password);
```

```
// set the PDO error mode to exception
$conn--->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
echo "Connected successfully";
}
catch(PDOException $e)
{
echo "Connection failed: " . $e->getMessage();
}
?>
```

نکته روش PDO

در کد مثال روش PDO مثال فوق، ما یک پایگاه داده به نام myDB را نیز تعیین کرده ایم. به این دلیل که PDOبرای کارکرد صحیح، نیاز به یک پایگاه داده معتبر دارد. اگر پایگاه داده ای برای آن تعیین نشود، یک خطا exceptionرخ می دهد .

نکته 2 :

یک فایده مهم استفاده از روش PDO این است که این روش دارای یک کلاس exception class ویژه برای مدیریت خطاهایی است که ممکن است در query های پایگاه داده رخ دهد. اگر یک خطا یا exception در بلوک try{} برنامه رخ دهد، اسکریپت اجرای کد را متوقف کرده و به صورت مستقیم به کد اولین بلوک Catch{} بعد از آن پرش می کند .

آموزش بستن یا Close اتصال یا Connection در MySQL

اتصال (Connection) به پایگاه داده، به صورت اتوماتیک پس از اجرای کدهای اسکریپت بسته می شود، اما برای بستن زودتر آن می توان از کدهای زیر استفاده کرد :

کد مثال شی گرا MySQLi

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

کد مثال MySQL رویه ای

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

کد مثال PDO

```
$conn = null;
```

آموزش ایجاد یک پایگاه داده (Database) جدید در MySQL

هر پایگاه داده (database) از یک یا چند جدول (Table) تشکیل شده است. برای ایجاد یک پایگاه داده جدید درMySQL ، شما بایستی مجوز لازم (مثل کاربر ادمین بودن) را داشته باشید .

آموزش ایجاد یک پایگاه داده MySQL جدید در روش PDO و MySQLi

از دستور CREATE DATABASE برای ایجاد یک پایگاه داده جدید در MySQL استفاده می شود. کد مثال عملی زیر یک پایگاه داده جدید به نام "MyDB" را ایجاد کرده است :

کد مثال (شی گرا) MySQLi

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn--->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . $conn->error;
}
$conn->close();
?>
```

نکته 1 :

هنگامی که یک پایگاه داده جدید را ایجاد می کنید، بایستی سه پارامتر اول لازم جهت شی mysqli object را تعیین کنید (پارامترهایServernameو Username ، Password)

نکته 2 :

اگر مجبور هستید از یک Port خاص در هنگام تعریف پایگاه داده استفاده کنید، یک رشته متنی خالی empty string را به لیست آرگومان های نام پایگاه داده اضافه کنید، مثل کد:

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

کد مثال MySQLi رویه ای

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = mysqli_connect($servername, $username, $password);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
// Create database
$sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql)) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . mysqli_error($conn);
}
mysqli_close($conn);
?-->
```

کد مثال PDO

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

نکته :

کد مثال عملی روشPDO ، یک پایگاه داده جدید به نام "myDBPDO" ایجاد می کند.

توجه :

مزیت مهم استفاده از روش PDO در ایجاد یک پایگاه داده جدید MySQL این است که روش PDO دارای یک کلاس exception class خاص جهت مدیریت خطاهای احتمالی رخ داده در دستورات database query است. اگر یک خطا یا exception جدید در ساختار try{} کد رخ دهد، اسکریپت ادامه اجرای کدها را متوقف کرده و به کد موجود در بخش Catch{} بعد از آن پرش می کند .

برای مثال، در کد فوق و در بخش Catch{} دستور SQL به همراه پیام خطا تولید شده را در خروجی نمایش داده ایم .

آموزش ایجاد یک جدول (Table) جدید درMySQL

یک جدول پایگاه داده یا database table دارای یک نام منحصر به فرد بوده و اطلاعات خود را در قالب چندین سطر (row) و ستون (column)نگهداری می کند.
از دستور CREATE TABLE در MySQL برای ایجاد یک جدول (table) جدید استفاده می شود.
در کد مثال های عملی این بخش، یک جدول جدید به نام "MyGuests"و با پنج ستون"id" ، "firstname"، "lastname"، "email"و "reg_date"ایجاد کرده ایم .

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

نکته های مربوط به دستور فوق

Data typeتعیین کننده نوع داده ای (مثل عدد، متن و ..) است که جدول در خود نگهداری می کند. برای دریافت اطلاعات بیشتر راجع به انواع داده ای قابل کاربرد در زبان PHP به بخش آموزش انواع داده ای در PHPبروید .

پس از تعیین نوع داده ای ستون (Column) در جدول، می توانید خواص (attribute) های دلخواه زیر را نیز برای هر ستون مشخص کنید .

عنوان جدول				
NOT NULL	DEFAULT value	UNSIGNED	AUTO INCREMENT	PRIMERY KEY
با تعیین این خاصیت، ستون مورد نظر حتما بایستی دارای مقدار بوده و مقدار تهی یا NULL پذیرفته نمی شود.	این خاصیت یک مقدار پیش فرض را برای ستون تعیین نکرده که در صورتی که کاربر مقداری را برای آن ستون وارد کند، مقدار پیش فرض به صورت خودکار در Column قرار می گیرد.	این خاصیت برای انواع عددی به کار رفته و کاربر را فقط مجاز به استفاده از اعداد مثبت و صفر می کند.	در این حالت MySQL مقدار ستون مورد نظر را به ازای وارد کردن یک آیتم جدید، به صورت اتوماتیک، یک واحد افزایش می دهد.	این خاصیت باعث تعیین ستون مورد نظر به عنوان کلید اصلی جدول (primary key) می شود. ستون کلید اصلی در هر جدول، بایستی برای هر رکورد دارای یک مقدار یکتا و غیر تکراری (unique) بوده و مقادیر تکراری در آن مجاز نیست. از این فیلد برای شناسایی رکورد در سطح برنامه استفاده شده و معمولا یک مقدار عددی به عنوان ID تعیین می شود که به صورت خودکار نیز افزایش می یابد. هر جدول بایستی دارای یک فیلد کلید اصلی باشد.

در کدهای مثال عملی زیر، نحوه تعریف یک جدول (table) جدید را در MySQL نشان می دهد:

کد مثال MySQLi شی گرا

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn--->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// sql to create table
$sql = "CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP
)";
if ($conn->query($sql) === TRUE) {
    echo "Table MyGuests created successfully";
} else {
    echo "Error creating table: " . $conn->error;
}
$conn->close();
?>
```

کد مثال MySQLi رویه ای

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
```

```
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
// sql to create table
$sql = "CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP
)";
if (mysqli_query($conn, $sql)) {
    echo "Table MyGuests created successfully";
} else {
    echo "Error creating table: " . mysqli_error($conn);
}
mysqli_close($conn);
?-->
```

کد مثال PDO

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";
try {
$conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
// set the PDO error mode to exception
$conn--->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
// sql to create table
$sql = "CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP
)";
// use exec() because no results are returned
$conn->exec($sql);
echo "Table MyGuests created successfully";
}
catch(PDOException $e)
{
echo $sql . "<br>" . $e->getMessage();
}
$conn = null;
?>
```

آموزش وارد کردن اطلاعات Data Insertدر MySQL

پس از این که یک پایگاه داده (database) و جدول (table)درون آن را ایجاد کردید، می توانید اقدام به وارد کردن اطلاعات (Insert Data)نمایید.

در لیست زیر، برخی از قوانین مهم در هنگام وارد کردن اطلاعات در MySQL را بیان کرده ایم :

- کد SQL یا (SQL query) را بایستی درون کدهای PHP قرار دهید .
- مقادیر متنی یا String در دستورات SQL query بایستی داخل دو " " قرار بگیرند.
- مقادیر عددی را بایستی بدون " " قرار دهید (به صورت ساده).
- کلمه کلیدی Null نیز نباید بین دو " " قرار گیرد.

از دستور INSERT INTO همانند کد زیر برای وارد کردن تعدادی رکورد جدید درون جدول MySQL Table استفاده می شود :

```
<?php
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
?>
```

برای آشنایی کامل با دستوراتSQL ، به بخش آموزش زبان SQL سایت تحلیل داده بروید.

در درس قبلی، یک جدول خالی به نام "MyGuests" را با چهار ستون"id" ، "firstname"، "lastname"و "reg data"ایجاد کردیم. حال می خواهیم این جدول را با اطلاعات مورد نظرمان پر کنیم .

نکته :

اگر یک ستون به صورت افزایش خودکار (AUTO_INCREMENT)مثل ستون ("id") تعریف شده باشد یا حاوی یک برچسب زمانی یا TIMESTAMP مثل ستون ("reg_data") باشد، نیازی نیست در دستور SQL مقداری جهت آن تعیین کنید. برنامه MySQL به صورت خودکار این نوع ستون ها را مقداردهی می کند .

مثال عملی : در کد مثال های عملی زیر نحوه وارد کردن اطلاعات جدید به جدول MySQL را با دستور INSERT INTOنشان داده ایم

کد مثال MySQLi شی گرا

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn--->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";
if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>
```

کد مثال MySQL رویه ای

```
<!--?php
$servername = " localhost";
$username="username" ;
$password="password" ;
$dbname="myDB" ; /-->/ Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";
if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}
mysqli_close($conn);
?>
```

کد مثال PDO

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";
try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
    // set the PDO error mode to exception
    $conn--->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";
// use exec() because no results are returned
$conn->exec($sql);
echo "New record created successfully";
}
catch(PDOException $e)
{
echo $sql . "<br>" . $e->getMessage();
}
$conn = null;
?>
```

آموزش استخراج ID آخرین رکورد وارد شده در جدول MySQL

اگر بر روی جدولی که دارای یک فیلد با قابلیت افزایش خودکار (AUTO_INCREMENT) است، دستور ورود اطلاعات (INERST) یا ویرایش اطلاعات (UPDATE) را اجرا کنیم، می توان همان لحظه ID آخرین رکورد وارد شده یا اصلاح شده را استخراج کرد .در کد مثال عملی زیر، در جدول"MyGuests" ، ستون id به صورت افزایش خودکار (AUTO_INCREMENT) تعریف شده است :

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

کد مثال عملی زیر همانند مثال درس قبلی است (وارد کردن اطلاعات با دستورات(INSERT INTO ، با این تفاوت که یک خط کد جدید را برای استخراج ID آخرین رکورد وارد شده، اضافه شده است. در پایان هم مقدار این ID را در خروجی چاپ کرده ایم :

کد مثال MySQLi شی گرا

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn--->connect_error) {
die("Connection failed: " . $conn->connect_error);
```

```
}
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";
if ($conn->query($sql) === TRUE) {
$last_id = $conn->insert_id;
echo "New record created successfully. Last inserted ID is: " . $last_id;
} else {
echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>
```

کد مثال MySQLi رویه ای:

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
die("Connection failed: " . mysqli_connect_error());
}
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";
if (mysqli_query($conn, $sql)) {
$last_id = mysqli_insert_id($conn);
echo "New record created successfully. Last inserted ID is: " . $last_id;
} else {
echo "Error: " . $sql . "<br-->" . mysqli_error($conn);
}
mysqli_close($conn);
```

کد مثال PDO

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";
try {
$conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
// set the PDO error mode to exception
$conn--->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";
// use exec() because no results are returned
$conn->exec($sql);
$last_id = $conn->lastInsertId();
echo "New record created successfully. Last inserted ID is: " . $last_id;
}
catch(PDOException $e)
{
echo $sql . "<br>" . $e->getMessage();
```

```
}
$conn = null;
?>
```

آموزش وارد کردن چندین رکورد همزمان در MySQL

برای وارد کردن اطلاعات چندین رکورد به صورت همزمان و یا اجرای چند دستور SQL با هم، بایستی از تابع mysql_multi_query() در MySQL استفاده کرد.

در کد مثال عملی زیر، 3 رکورد را در جدول "MyGuests" وارد کرده ایم :

کد مثال MySQLi شی گرا

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn--->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com');";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Mary', 'Moe', 'mary@example.com');";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Julie', 'Dooley', 'julie@example.com')";
if ($conn->multi_query($sql) === TRUE) {
echo "New records created successfully";
} else {
echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>
```

نکته :

هر دستور SQL را بایستی با کاراکتر از یکدیگر جدا نمود .

کد مثال MySQLi رویه ای

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
```

```
    die("Connection failed: " . mysqli_connect_error());
}
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
        VALUES ('John', 'Doe', 'john@example.com');";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
        VALUES ('Mary', 'Moe', 'mary@example.com');";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
        VALUES ('Julie', 'Dooley', 'julie@example.com')";
if (mysqli_multi_query($conn, $sql)) {
    echo "New records created successfully";
} else {
    echo "Error: " . $sql . "<br-->" . mysqli_error($conn);
}
mysqli_close($conn);
?>
```

اما اجرای مثال فوق در روش MySQLi PDO کمی متفاوت و به صورت زیر است:

کد مثال: MySQLi PDO

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";
try {
$conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
// set the PDO error mode to exception
$conn--->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
// begin the transaction
$conn->beginTransaction();
// our SQL statements
$conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')");
$conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Mary', 'Moe', 'mary@example.com')");
$conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Julie', 'Dooley', 'julie@example.com')");
// commit the transaction
$conn->commit();
echo "New records created successfully";
}
catch(PDOException $e)
{
// roll back the transaction if something failed
$conn->rollback();
echo "Error: " . $e->getMessage();
}
$conn = null;
?>
```

آموزش استخراج اطلاعات (Select Data) از پایگاه داده MySQL

از دستور SELECT برای استخراج اطلاعات مورد نظر از یک یا چند جدول استفاده می شود. شکل کلی استفاده از دستور SELECT در MySQL به صورت زیر است :

```
SELECT column_name(s) FROM table_name
```

اگر از کاراتر * به جای نام ستون استفاده کنید، دستور SELECT اطلاعات کلیه ستون های جدول را می خواند:

```
SELECT * FROM table_name
```

برای دریافت اطلاعات کامل درباره نحوه کار دستور SELECT به بخش آموزش دستور select در زبان sql سایت تحلیل داده بروید.

آموزش خواندن (Select) اطلاعات در MySQLi

کد مثال عملی زیر، اطلاعات ستون هایid ، firstnameو lastnameرا از جدول MyGuests انتخاب کرده و در خروجی نشان می دهد :

کد مثال MySQLi شی گرا:

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn--->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
// output data of each row
while($row = $result->fetch_assoc()) {
echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
$row["lastname"]. "<br>";
}
} else {
echo "0 results";
}
$conn->close();
?>
```

در ادامه به تشریح نکات کد مثال فوق پرداخته ایم :

در ابتدا یک SQL query را با استفاده از دستور select ایجاد کرده و اطلاعات فیلدهایid ، firstnameو lastnameرا از جدول MyGuestsخوانده ایم.

خط بعدی SQL query فوق را اجرا کرده و اطلاعات دریافت شده را در یک متغیر به $result ذخیره می کند.

در مرحله بعدی تابع ()num_rows چک می کند آیا اطلاعات یک یا تعداد بیشتری سطر (از صفر بیشتر) برگردانده شده است یا خیر. اگر بیشتر از صفر سطر یا رکورد برگردانده شده باشد، تابع ()fetch_assoc اطلاعات را در یک متغیر آرایه رابطه ای (associative array) قرار داده که می توانیم با استفاده از دستوراتی مثل حلقه یا loop به جستجو درون آن بپردازیم.

در مرحله آخر هم، حلقهloop ()while ، به جستجو درون آرایه پرداخته و اطلاعات خروجی شامل فیلدهایid ، firstnameو lastname را نشان می دهد.

کد مثال زیر، دستورات مثال قبل را با استفاده از روش عینا انجام داده و MySQLi رویه ای همان خروجی را بر می گرداند :

کد مثال MySQLi شی گرا:

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
die("Connection failed: " . mysqli_connect_error());
}
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) --> 0) {
// output data of each row
while($row = mysqli_fetch_assoc($result)) {
echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
$row["lastname"]. "<br>";
}
} else {
echo "0 results";
}
mysqli_close($conn);
?>
```

همچنین می توانید با طراحی کدی مثل کد زیر، اطلاعات استخراج شده را در یک جدول HTML نشان دهید :

کد مثال MySQLi رویه ای:

```
<!--?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn--->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
echo "";
// output data of each row
while($row = $result->fetch_assoc()) {
echo "";
}
echo
"<table><tbody><tr><th>ID</th><th>Name</th></tr><tr><td>".$row["id"]."</td><td>".$row["firstname"]." ".$row["lastname"]."</td></tr></tbody></table>";
} else {
echo "0 results";
}
$conn->close();
?>
```

آموزش انتخاب اطلاعات (Select Data) با روش PDO و دستورات آماده SQL

در کد مثال عملی زیر از روش MySQLi PDO و دستورات آماده (Prepared Statements) که در درس های بعدی به آموزش آن خواهیم پرداخت، استفاده کرده ایم.

کد این مثال اطلاعات فیلدهایid ، firstnameو lastname را از جدول MyGuests خوانده و در یک جدول HTMLنشان می دهد .

کد مثال MySQLi PDO

```
<?php
echo "<table style='border: solid 1px black;'>";
echo "<tr><th>Id</th><th>Firstname</th><th>Lastname</th></tr>";
class TableRows extends RecursiveIteratorIterator {
function __construct($it) {
parent::__construct($it, self::LEAVES_ONLY);
}
function current() {
return "<td style='width:150px;border:1px solid black;'>" .
parent::current(). "</td>";
}
```

```
function beginChildren() {
echo "<tr>";
}
function endChildren() {
echo "</tr>" . "\n";
}
}
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";
try {
$conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$stmt = $conn->prepare("SELECT id, firstname, lastname FROM MyGuests");
$stmt->execute();
// set the resulting array to associative
$result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
foreach(new TableRows(new RecursiveArrayIterator($stmt->fetchAll())) as
$k=>$v) {
echo $v;
}
}
catch(PDOException $e) {
echo "Error: " . $e->getMessage();
}
$conn = null;
echo "</table>";
?>
```

آموزش حذف اطلاعات (Data Delete) در MySQL

از دستور DELETE در SQL برای حذف اطلاعات یک یا چند رکورد از جدول پایگاه داده استفاده می شود. شکل کلی استفاده از دستور DELETEبه صورت زیر است :

```
DELETE FROM table_name
WHERE some_column = some_value
```

نکته مهم :

به کاربرد عبارت WERE در دستور فوق دقت کنید !

عبارت WHERE در دستور فوق تعیین می کند چه رکورد یا رکوردهایی بایستی حذف شوند. اگر در دستور DELETE از عبارت WHERE استفاده نکنید، کلیه اطلاعات جدول مورد نظر حذف خواهد شد.

برای دریافت اطلاعات کامل تر درباره دستور DELETE به بخش آموزش دستور DELETE در زبان SQL سایت تحلیل داده بروید.

به اطلاعات جدول MyGuests توجه کنید :

عنوان جدول				
reg_date	email	lastname	firstname	id
۱۴:۲۶:۱۵ ۲۰۱۴-۱۰-۲۲	john@example.com	Doe	John	۱
۱۰:۲۲:۳۰ ۲۰۱۴-۱۰-۲۳	mary@example.com	Moe	Mary	۲
۱۰:۴۸:۲۳ ۲۰۱۴-۱۰-۲۶	julie@example.com	Dooley	Julie	۳

در کد مثال عملی زیر، اطلاعات رکورد با id=3 را از جدول MyGuests حذف کرده ایم

کد مثال MySQLi شی گرا

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn-<connect_error) {
die("Connection failed: " . $conn-<connect_error);
}
// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";
if ($conn-<query($sql) === TRUE) {
echo "Record deleted successfully";
} else {
echo "Error deleting record: " . $conn-<error;
}
$conn-<close();
?>
```

کد مثال MySQLi رویه ای

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
die("Connection failed: " . mysqli_connect_error());
}
// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";
if (mysqli_query($conn, $sql)) {
```

```
echo "Record deleted successfully";
} else {
echo "Error deleting record: " . mysqli_error($conn);
}
mysqli_close($conn);
?gt
```

کد مثال MySQLi PDO

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";
try {
$conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
// set the PDO error mode to exception
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";
// use exec() because no results are returned
$conn->exec($sql);
echo "Record deleted successfully";
}
catch(PDOException $e)
{
echo $sql . "<br>" . $e->getMessage();
}
$conn = null;
?>
```

پس از اجرای دستور DELETE در مثال فوق، جدول MyGuests به صورت زیر تغییر می کند:

عنوان جدول				
reg_date	email	lastname	firstname	id
۱۴:۲۶:۱۵ ۲۰۱۴-۱۰-۲۲	john@example.com	Doe	John	۱
۱۰:۲۲:۳۰ ۲۰۱۴-۱۰-۲۳	mary@example.com	Moe	Mary	۲

آموزش ویرایش اطلاعات (Update Data) در MySQL

از دستور UPDATE برای ویرایش و به روز رسانی اطلاعات موجود در رکوردهای یک جدول استفاده می شود.

ساختار کلی استفاده از دستور UPDATEدر MySQL به صورت زیر است :

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

نکته مهم :

به کاربرد عبارت WHERE در دستور UPDATE دقت کنید !
عبارت WHERE تعیین می کند اطلاعات چه رکورد یا رکوردهایی از جدول ویرایش (Update) شوند، اگر از عبارت WHERE استفاده نکنید، اطلاعات کلیه رکوردهای جدول ویرایش خواهد شد .

اطلاعات جدول "MyGuests"				
id	firstname	lastname	email	reg_date
۱	John	Doe	john@example.com	۱۴:۲۶:۱۵ ۲۰۱۴-۱۰-۲۲
۲	Mary	Moe	mary@example.com	۱۰:۲۲:۳۰ ۲۰۱۴-۱۰-۲۳

برای دریافت اطلاعات کامل تر درباره دستور UPDATE به بخش آموزش دستور UPDATE در زبان SQL سایت تحلیل داده بروید .
برای درک بهتر نحوه کارکرد دستور UPDATE چند مثال عملی می زنیم. به اطلاعات جدول "MyGuests" دقت کنید :

کد مثال عملی زیر، اطلاعات رکورد با فیلد id=2 را در جدول ویرایش می کند:

کد مثال MySQLi شی گرا

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";
if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . $conn->error;
}
$conn->close();
?>
```

کد مثال MySQLi رویه ای

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

```
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";
if (mysqli_query($conn, $sql)) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . mysqli_error($conn);
}
mysqli_close($conn);
?>
```

کد مثال MySQLi PDO

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";
try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";
    // Prepare statement
    $stmt = $conn->prepare($sql);
    // execute the query
    $stmt->execute();
    // echo a message to say the UPDATE succeeded
    echo $stmt->rowCount() . " records UPDATED successfully";
}
catch(PDOException $e)
{
    echo $sql . "<br>" . $e->getMessage();
}
$conn = null;
?>
```

پس از اجرای کد فوق و UPDATE اطلاعات جدول، رکوردهای جدول MyGuests به صورت زیر تغییر می کنند:

پس از ویرایش "MyGuests" اطلاعات جدول				
id	firstname	lastname	email	reg_date
۱	John	Doe	john@example.com	۱۴:۲۶:۱۵ ۲۰۱۴-۱۰-۲۲
۲	Mary	Doe	mary@example.com	۱۰:۲۲:۳۰ ۲۰۱۴-۱۰-۲۳

آموزش محدود کردن اطلاعات (Limit Data) در MySQL

MySQLعبارت LIMIT را برای محدودکردن تعداد رکوردهای خوانده شده و یا تعیین شمار رکوردهای مورد نظر جهت خواندن، در اختیارمان قرار داده است.

عبارتLIMIT ، مدیریت چندین صفحه اطلاعات رکوردهای خوانده شده و امکان صفحه بندی اطلاعات در SQL را فراهم کرده است. این ابزار در هنگام کار با جدول های اطلاعاتی بزرگ، بسیار کاربرد دارد .

نکته :

خواندن و بازگرداندن حجم زیادی از رکوردها در یک دستور، می تواند کارایی سیستم را به شدت کاهش دهد .

فرض کنید می خواهیم اطلاعات رکوردهای 1 تا 30 را از جدول "Orders" استخراج کنیم .لازم SQL query جهت این کار به صورت زیر خواهد بود

```
$sql = "SELECT * FROM Orders LIMIT 30";
```

هنگامی که دستور SQL query فوق اجرا شود، اطلاعات 30 رکورد اول جدول را بر می گرداند.

حال اگر بخواهیم اطلاعات رکوردهای 16 تا 25 را بخواهیم، بایستی چه کار کنیم؟ MySQL یک راه حل برای مدیریت این مسئله در اختیارمان قرار داده است، استفاده از عبارت OFFSET .

SQL query زیر به برنامه می گوید اطلاعات فقط 10 رکورد جدول از شماره 16 به بعد (OFSET 15) را برگرداند

```
$sql = "SELECT * FROM Orders LIMIT 10 OFFSET 15";
```

همچنین می توانید یک ساختار دستوری کوتاه تر را همانند SQL query زیر اجرا کرده و نتیجه ای یکسان بگیرید:

```
$sql = "SELECT * FROM Orders LIMIT 15, 10";
```

توجه داشته باشید که در صورت استفاده از کاما، اعداد در SQL query فوق برعکس می شوند.

آموزش استفاده از دستورات آماده (Prepared) در SQL

دستورات آماده در SQL یا Preapared statement راه حلی بسیار مطمئن برای جلوگیری از حملات اسکریپتی SQL injection و طراحی سریع کدهای برنامه هستند. در این درس به آموزش کار با دستورات آماده یا Prepared statement در MySQL خواهیم پرداخت .

معرفی دستورات آماده (Prepared statements) **و پارامترهای متصل** : (Bound Parameters)

یک دستور آماده SQL یا Prepared statement قابلیتی است که به وسیله آن می توان یک دستور یکسان SQL یا دستوری مشابه آن را به صورت مکرر و با تاثیرگذاری بالا اجرا کرد.

دستورات آماده Prepared statement SQL به صورت کلی با روش زیر عمل می کنند:

1– آماده سازی اولیه (Prapare) دستور : یک الگو برای دستور SQL مورد نظر ایجاد شده و به پایگاه داده ارسال می شود. مقادیر مشخصی به صورت تعیین نشده در دستور باقی می مانند که به آن ها پارامتر گفته و با نماد ؟ در کد SQL جایگزین می شوند. برای مثال، الگویی همانند کد زیر برای یک دستور درج اطلاعات ایجاد می شود :

```
INSERT INTO MyGuests VALUES(?, ?, ?)
```

2– پایگاه داده الگوی دستوری SQL را خوانده و اجرا می کند. همچنین با استفاده از قابلیت بهینه سازی queryیا query optimization سریع ترین راه را برای اجرای آن مشخص می کند. در انتها نتایج حاصل از queryرا بدون اجرای نهایی در حافظه نگهداری می کند .

3– مرحله سوم اجرا : (Excute) در مرحله آخر، برنامه مقادیر (values) را به پارامترها ارسال کرده و پایگاه داده دستور SQL را به صورت کامل اجرا می کند. برنامه می تواند یک دستور واحد SQL را با مقادیر مختلف، هر چند بار که نیاز داشته باشد اجرا کرده و خروجی های مختلف تولید کند .در مقایسه با اجرای مستقیم دستوراتSQL ، استفاده از الگوهای آماده SQL دو مزیت عمده دارد :

دستورات آماده Prepared statement SQL زمان اجرای کدهای برنامه را به دلیل آماده سازی query قبل از اجرا، کاهش می دهند (حتی با وجود این که دستورات مهم است چندین بار اجرا شوند.)

استفاده از پارامترها، حجم اطلاعات ارسالی به سرور را بسیار کاهش می دهند، زیرا در هر بار اجرای query فقط کافی است مقادیر جدید پارامترها را ارسال کنید نه کل دستور query را.

دستورات اماده SQL روش بسیار موثری جهت مقابله با حملات اسکریپتی SQL injection هستند. زیرا مقادیر پارامترها بعدا و توسط یک پروتکل متفاوت به سرور ارسال شده و از حملات SQL جلوگیری می کنند.

آموزش کار با دستورات آماده SQL در MySQLi

کد مثال عملی زیر از دستورات آماده و پارامترهای متصل در MySQLi استفاده می کند :

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// prepare and bind
$stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email)
VALUES (?, ?, ?)");
$stmt->bind_param("sss", $firstname, $lastname, $email);
// set parameters and execute
$firstname = "John";
$lastname = "Doe";
$email = "john@example.com";
$stmt->execute();
$firstname = "Mary";
$lastname = "Moe";
$email = "mary@example.com";
$stmt->execute();
$firstname = "Julie";
$lastname = "Dooley";
$email = "julie@example.com";
$stmt->execute();
echo "New records created successfully";
$stmt->close();
$conn->close();
?>
```

در ادامه به توضیح برخی از کدهای مثال فوق می پردازیم:

```
"INSERT INTO MyGuests (firstname, lastname, email) VALUES (?, ?, ?)"
```

در دستور SQL فوق، از کاراکتر ؟ برای تعیین محل قرارگیری یک متغیر، استفاده کرده ایم. به عبارت دیگر به ازای هر ؟ در دستور فوق، یک متغیر رشته ای، عددی، اعشاری و ... قرار گیرد.

سپس به کد تابع bind_param() دقت کنید:

```
$stmt->bind_param("sss", $firstname, $lastname, $email);
```

تابع bind_param() پارامترها و مقادیرشان را به SQL query ارسال می کند. پارامتر "sss" نیز لیست انواع داده ای پارامترها را به ترتیب مشخص می منند. کاراکتر "s" در پارامتر فوق به mysal اعلام می کند که پارامتر مورد نظر از نوع رشته ای یا string است. کاراکتر این پارامتر می تواند مخفف یکی از مقادیر زیر باشد :

i :عددی یاinteger

d :اعشاری یاdouble

S : رشته ای یاstring

b : عددی بزرگ یاBinary large object

در کد فوق، بایستی نوع داده ای پارامتر را با یکی از کاراکترهای اختصاری تعیین کنید. همچنین با تعیین تنوع داده ای پارامتر، ریسک حملات اسکریپتی SQL injection را کاهش می دهید .

نکته :

اگر می خواهید مقادیر را از تابع خارجی مثل داده های ورودی کاربر ارسال کنید، بایستی آن ها را حتما اعتبارسنجی و پاکسازی کنید .

مثال کار با دستورات آماده SQL در PDO

کد مثال عملی زیر، از دستورات آماده Prepared SQL Statement در روش MySQLi PDO استفاده می کند

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

```
try {
$conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
// set the PDO error mode to exception
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
// prepare sql and bind parameters
$stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email)
VALUES (:firstname, :lastname, :email)");
$stmt->bindParam(':firstname', $firstname);
$stmt->bindParam(':lastname', $lastname);
$stmt->bindParam(':email', $email);
// insert a row
$firstname = "John";
$lastname = "Doe";
$email = "john@example.com";
$stmt->execute();
// insert another row
$firstname = "Mary";
$lastname = "Moe";
$email = "mary@example.com";
$stmt->execute();
// insert another row
$firstname = "Julie";
$lastname = "Dooley";
$email = "julie@example.com";
$stmt->execute();
echo "New records created successfully";
}
catch(PDOException $e)
{
echo "Error: " . $e->getMessage();
}
$conn = null;
?>
```

بخش پنجم: آموزش کاربرد XML درPHP

آموزش کار با زبان XML در PHP

XMLچیست؟

زبان XML یک روش برای قالب دهی و نگهداری اطاعات جهت اشتراک گذاری و استفاده در سطح وب است.

برخی از نرم افزارها و خدمات تحت وب مثل خوراک خوان های RSS و پادکست ها به زبان XML نوشته شده اند.

XMLساختاری شبیه HTML دارد، با این تفاوت که در آن می توانید تگ های مورد نظر خود را ایجاد

نمایید XML .برای طراحی و نگهداری اطلاعات بسیار راحت و کاربردی است.

برای دریافت اطلاعات بیشتر راجع به زبان XML به بخش آموزش زبان XML در سایت تحلیل داده بروید .

یک مفسر زبان XML یا XML Parser چیست؟

برای خواندن، ایجاد، به روزرسانی و تغییر یک سندXML ، به یک مفسر زبان XML یا XML Parser نیاز دارید.

در زبان PHP دو نوع مفسر یا XML Parser اصلی به شرح زیر داریم :

- مفسر درختی XML یا.Tree-Based Parsers
- مفسر مبتنی بر رویداد XML یا.Event-Based Parsers

مفسر درختی XML یا Tree-Based Parsers

مفسر درختی XML یا Tree-Based Parsers کل سند XML را خوانده و در حافظه قرار می دهد. سپس ساختار سند XML را به صورت درختی ترسیم کرده و کل اطلاعات آن را آنالیز می کند. این ؟؟؟؟؟؟؟؟ امکان دسترسی به اعضای هر درخت را مبتنی بر ؟؟؟؟؟؟؟ DOM فراهم می کند .مفسر درختی XML برای مدیریت اسناد ؟؟؟؟؟؟؟ کوچک مناسب بوده و در فایل های بسیار بزرگ XML کارآیی نداشته و می تواند عملکرد سیستم را مختل کند .از مدل های مفسر درختی XML می تواند به نمونه های زیر اشاره کرد :

- XMLساده یا.SimpleXML
- مدل.DOM

مفسر مبتنی بر رویداد XML یا Event-Based Parsers

مفسر مبتنی بر رویداد در XML یا Event-Based Parsers کل سند XML را در حافظه قرار نداده و به جای آن، هر یک از گره یا node های سند XML را تک تک خوانده و اجازه دسترسی به آن ها را می دهد. هنگامی که کار شما با یک گره یا node به پایان برسد، مفسر به گره بعدی رفته و اطلاعات گره قدیمی از حافظه چک می شوند.

این مدل مفسر برای کار با فایل های XML بسیار بزرگ مناسب بوده و اسناد XML را سریع تر لود می کند. همچنین حجم بسیار کمتری حافظه سیستم را اشغال خواهد کرد.

از نمونه های مفسر مبتنی بر رویئداد XML می توان به موارد زیر اشاره کرد :

- XML Reader
- XML Expat Parsers.

آموزش کار با مفسر Parser SimpleXML در زبان PHP

مفسر زبان XML با نام SimpleXML یک افزونه زبان PHP است که به ما امکان ویرایش و خواندن اطلاعات فایل های XML را می دهد.

مفسر SmpleXML یک مفسر درختی XML یا tree-based Parser است.

مفسر زبان XML نوع SimpleXML، یک راه ساده برای دریافت مقدار نام ((name، خواص ((attributeو محتوی متنی هر عنصر ((elementدر یک سند XML را به شرط دانستن ساختار یا قالب آن فایل، فراهم می کند.

مفسر SimpleXML یک سند XML را به ساختار داده ای ویژه ای تبدیل کرده که می توانید با آن همانند یک مجموعه از آرایه ها و اشیاء رفتار کنید.

در مقایسه با روش DOM یا مفسر Excapt Parser، مفسر SimpleXML، برای خواندن یک عنصر ((elementدر فایل های XML، به حجم کدنویسی کمتری نیاز دارد.

نحوه نصب مفسر SimpleXML در PHP

از نسخه PHP به بعد، توابع مربوط به مفسر SimpleXML بخشی از هسته اصلی زبان PHP شده و برای استفاده از این قابلیت و تابع های آن نیاز به نصب برنامه جداگانه ای ندارید.

آموزش خواندن XML از متن String در PHP

از تابع ()Simplexml_load_stringبرای خواندن اطلاعات XML از یک متغیر متنی String استفاده می شود.

فرض کنید که یک متغیر متنی string که حاوی اطلاعات XML به شرح زیر است را داریم :

```
$myXMLData =
"<?xml version='1.0' encoding='UTF-8' ?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>";
```

در کد مثال عملی زیر، نحوه خواندن اطلاعات XML از یک متغیر متنی String با استفاده از تابع simplexml_load_string()را آموزش داده ایم:

```
<?php
$myXMLData =
"<?xml version='1.0' encoding='UTF-8' ?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>";
$xml=simplexml_load_string($myXMLData) or die("Error: Cannot create object");
print_r($xml);
?>
```

خروجی کد مثال عملی فوق به صورت زیر خواهد بود:

```
<?php
SimpleXMLElement Object ( [to] => Tove [from] => Jani [heading] => Reminder
[body] => Don't forget me this weekend! )
?>
```

نکته : نحوه مدیریت خطا (Error Hardling) در کد مثال فوق

از توابع مربوط به شی libxml برای دریافت و ذخیره کلیه خطاهای رخ داده در هنگام خواندن سند XML استفاده نمایید. در کد مثال عملی زیر، سعی کرده ایم تا یک فایل ناقص XML را خوانده و سپس خطاهای رخ داده را نمایش داده ایم .

```
<?php
libxml_use_internal_errors(true);
$myXMLData =
"<?xml version='1.0' encoding='UTF-8' ?>

<document>
<user>John Doe</wronguser>
<email>john@example.com</wrongemail>
</document>";
$xml = simplexml_load_string($myXMLData);
if ($xml === false) {
echo "Failed loading XML: ";
foreach(libxml_get_errors() as $error) {
echo "<br>", $error->message;
}
} else {
print_r($xml);
}
?>
```

خروجی کد مثال فوق به صورت زیر خواهد بود:

```
<?php
Failed loading XML:
Opening and ending tag mismatch: user line 3 and wronguser
Opening and ending tag mismatch: email line 4 and wrongemail
?>
```

آموزش خواندن از فایل XML با مفسر: PHP SimpleXML

از تابع simplexml_load_file() در PHP برای خواندن اطلاعات از یک فایل XML استفاده می شود .فرض کنید که یک فایل XML به نام note.xml داریم که محتوی آن به صورت زیر است :

```
<?xml version="1.0" encoding="UTF-8" ?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

در کد مثال عملی زیر، نحوه خواندن فایل XML به وسیله تابع simple_load_file() را آموزش داده ایم:

```
<?php
$xml=simplexml_load_file("note.xml") or die("Error: Cannot create object");
print_r($xml);
?>
```

خروجی کد مثال فوق به صورت زیر خواهد بود:

```
<?php
SimpleXMLElement Object ( [to] => Tove [from] => Jani [heading] => Reminder
[body] => Don't forget me this weekend! )
?>
```

راهنمایی :

دردرس بعدی به آموزش نحوه دریافت یا خواندن مقادیر گره ها (node values) در یک فایل XML یا SimpleXMLخواهیم پرداخت .

مثال هایی بیشتر درباره مفسر SimpleXML

برای دریافت اطلاعات بیشتر درباره مفسر SimpleXML به بخش مرجع آموزش SimpleXML XML در سایت تحلیل داده بروید .

آموزش خواندن مقادیر گره ها با استفاده از SimpleXML

همانطور که در درس قبلی اشاره کردیم، SimpleXMLیک افزونه PHP است که امکان ویرایش، خواندن و نوشتن اطلاعات فایل های XML را در زبانPHP ، فراهم می کند .

آموزش خواندن مقادیر گره ها (Node Values) با SimpleXML

در کد مثال عملی زیر، نحوه خواندن مقادیر گره ها (Node Values) را با استفاده از ابزارSimpleXML ، آموزش داده ایم. در این مثال اطلاعات گره های موجود در فایل note.xml استخراج شده اند :

```
<?php
$xml=simplexml_load_file("note.xml") or die("Error: Cannot create object");
echo $xml->to . "<br>";
echo $xml->from . "<br>";
echo $xml->heading . "<br>";
echo $xml->body;
?>
```

خروجی کد مثال فوق به صورت زیر خواهد بود:

```
Tove
Jani
Reminder
Don't forget me this weekend!
```

آموزش خواندن مقادیر المنت های خاص با SimpleXML

فرض کنید فایل XML ای دیگری به نام books.xml داریم که اطلاعات آن به صورت زیر است :

```
<?xml version="1.0" encoding="utf-8" ?>
<bookstore>
<book category="COOKING">
<title lang="en">Everyday Italian</title>
<author>Giada De Laurentiis</author>
<year>2005</year>
<price>30.00</price>
</book>
<book category="CHILDREN">
<title lang="en">Harry Potter</title>
<author>J K. Rowling</author>
<year>2005</year>
<price>29.99</price>
</book>
<book category="WEB">
<title lang="en-us">XQuery Kick Start</title>
<author>James McGovern</author>
<year>2003</year>
<price>49.99</price>
</book>
<book category="WEB">
<title lang="en-us">Learning XML</title>
<author>Erik T. Ray</author>
```

```
<year>2003</year>
<price>39.95</price>
</book>
</bookstore>
```

با استفاده از کد مثال عملی زیر، مقادیر گره های عنصر <title>را در المنت <book>اول و دوم فایل books.xmlرا خوانده ایم:

```
<?php
$xml=simplexml_load_file("books.xml") or die("Error: Cannot create object");
echo $xml->book[0]->title . "<br>";
echo $xml->book[1]->title;
?>
```

خروجی کد مثال فوق به صورت زیر خواهد بود:

```
Everyday Italian
Harry Potter
```

آموزش خواندن مقادیر گره ها با استفاده از حلقه loop در PHP

در کد مثال عملی زیر، با استفاده از یک حلقه loop به درون کلیه المنت های <book>در فایل "books.xml"حرکت کرده و مقدار گره های المنت های <title>، <year>و <price>را خوانده ایم :

```
<?php
$xml=simplexml_load_file("books.xml") or die("Error: Cannot create object");
foreach($xml->children() as $books) {
echo $books->title . ", ";
echo $books->author . ", ";
echo $books->year . ", ";
echo $books->price . "<br>";
}
?
```

خروجی کد مثال فوق به صورت زیر خواهد بود:

```
<?php
Everyday Italian, Giada De Laurentiis, 2005, 30.00
Harry Potter, J K. Rowling, 2005, 29.99
XQuery Kick Start, James McGovern, 2003, 49.99
Learning XML, Erik T. Ray, 2003, 39.95
?>
```

آموزش خواندن مقدار خاصیت ها (Attribute) با SimpleXML

در کد مثال عملی زیر، مقدار خاصیت "Category" مربوط به عنصر اول <book>و مقدار خاصیت "lang" مربوط به عنصر <title>در دومین المنت <book>را با استفاده از SimpleXML خوانده ایم :

```
<?php
$xml=simplexml_load_file("books.xml") or die("Error: Cannot create object");
echo $xml->book[0]['category'] . "<br>";
echo $xml->book[1]->title['lang'];
?>
```

خروجی کد مثال فوق به صورت زیر خواهد بود:

```
COOKING
en
```

آموزش خواندن مقدار خاصیت ها (Attribute) با استفاده از حلقه loop

در کد مثال عملی زیر و با استفاده از یک حلقه <loop>، مقدار المنت های <title>را در فایل "books.xml"خوانده ایم :

```
<?php
$xml=simplexml_load_file("books.xml") or die("Error: Cannot create object");
foreach($xml->children() as $books) {
    echo $books->title['lang'];
    echo "<br>";
}
?>
```

خروجی کد مثال عملی فوق به صورت زیر است:

```
en
en
en-us
en-us
```

مثال های عملی بیشتر برای ابزار SimpleXML

برای دریافت اطلاعات بیشتر درباره تابع های مرتبط با ابزار SimpleXML SimpleXML به بخش مرجع آموزش کار با SimpleXML در سایت تحلیل داده بروید .

آموزش کار با مفسر Expat XML در PHP

مفسر درون ساختهXML Expat Parser ، امکان خواندن و پردازش اطلاعات فایل های XML را در زبان PHPفراهم می کند.

همانطور که قبلا اشاره کردیم، XML Expatیک مفسر مبتنی بر رویداد یا event-based Parser است .

به قطعه کد XML زیر دقت کنید :

```
<from>Jani</from>
```

یک مفسر مبتنی بر رویداد، قطعه کد XML فوق را به صورت یک مجموعه سه تایی از رویدادهای متوالی، گزارش می کند. به صورت زیر :

المنت آغازین یا Start element که "form" است.

شروع کننده بخش CDATA که دارای مقدار "Jani" می باشد.

المنت پایان دهنده Close element که باز هم "form" است.

مفسر زبان XML نوع Expat یکی از توابع درون ساخته زبان PHPبوده و برای استفاده از آن نیاز به نصب افزونه یا برنامه خاصی ندارید .

بررسی فایل XML مثال ها

از فایل XML به نام "note.xml" با محتویات زیر در مثال های این درس استفاده شده است :

```
<?xml version="1.0" encoding="UTF-8" ?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

مقداردهی اولیه XML Expat Parser در PHP

در کد زیر، ما مفسر XML Expat Parser را مقداردهی اولیه کرده و تعدادی مدیریت کننده رویداد (handler)را برای چندین رویداد XMLتعیین کرده ایم. در نهایت هم فایل XML را خوانده و پردازش کرده ایم :

```
<?php
// Initialize the XML parser
$parser=xml_parser_create();
// Function to use at the start of an element
function start($parser,$element_name,$element_attrs) {
switch($element_name) {
case "NOTE":
echo "-- Note --<br>";
break;
case "TO":
echo "To: ";
break;
case "FROM":
echo "From: ";
break;
case "HEADING":
echo "Heading: ";
```

```
                    break;
                case "BODY":
            echo "Message: ";
                    }
                    }
// Function to use at the end of an element
function stop($parser,$element_name) {
                echo "<br>";
                    }
// Function to use when finding character data
function char($parser,$data) {
                echo $data;
                    }
// Specify element handler
xml_set_element_handler($parser,"start","stop");
// Specify data handler
xml_set_character_data_handler($parser,"char");
// Open XML file
$fp=fopen("note.xml","r");
// Read data
while ($data=fread($fp,4096)) {
xml_parse($parser,$data,feof($fp)) or
die (sprintf("XML Error: %s at line %d",
xml_error_string(xml_get_error_code($parser)),
xml_get_current_line_number($parser)));
                    }
// Free the XML parser
xml_parser_free($parser);
?>
```

توضیح کد مثال فوق

با استفاده از تابع XML_Parser_Create() ، مفسر برنامه را مقداردهی اولیه کرده و راه اندازی نموده ایم.

در مرحله بعدی توابع مختلف را برای کار با مدیریت کننده های رویدادها یا event handlers تعیین نموده ایم.

با استفاده از تابع xml_set_element_handler() تعیین کرده ایم کدام تابع در زمانی که مفسر XML وارد تگ های ابتدایی و انتهایی سند می شود، اجرا شوند.

با استفاده از تابع xml_set_character_data_handler() تعیین کرده ایم کدام تابع در زمانی که مفسر XML وارد کاراکترهای داده ای می شود، اجرا شود.

با استفاده از تابع XML_Parse() فایل "note.xml" را خوانده و پردازش کرده ایم.

در موارد رخ دادن خطاها(errors) ، تابع XML_error_string()را اضافه کرده ایم تا خطای XML روی داده را به خطایی قابل خواندن و متنی تبدیل کند.

در پایان مثال هم، تابع XML_Parser-free() را برای رهاسازی حافظه ای که تابع XML_Parser_create() اشغال کرده است را فراخوانی کرده ایم.

مثال های عملی بیشتر مفسر XML Expat Parser

برای دریافت اطلاعات بیشتر درباره تابع های مفسر Expat Parser به بخش مرجع آموزش مفسر Expat Parserدر سایت تحلیل داده بروید .

آموزش کار با مفسر XML DOM در PHP

مفسر درون ساختهXML DOM ، امکان پردازش و خواندن فایل های XML را در PHP فراهم می کند. مفسر XML DOM یک مفسر درختی یا tree-basded parser است.

به قطعه کد XML زیر دقت کنید :

```
<?xml version="1.0" encoding="UTF-8" ?>
<from>Jani</from>
```

مفسرXML DOM ، کد XML فوق را به صورت یک ساختار درختی 3 سطحی می بیند :

- سطح 1 یا : Level1 سند فایل.XML
- سطح 2 یا : Level2 المنت اصلی یا Root element که المنت <form>است.
- سطح 3 یا : Level3 المنت متنی یا Text element که دارای مقدار "Jani" می باشد.

آموزش نصب مفسر XML DOM در PHP

تابع های مفسر XML DOM بخشی از هسته اصلی زبان PHP بوده و برای استفاده از آن، نیازی به نصب افزونه یا برنامه خاصی نیست .

بررسی فایل XML مثال ها

از فایل XML به نام "note.xml" با محتویات زیر در مثال های این درس استفاده شده است :

```
<?xml version="1.0" encoding="UTF-8" ?>
<note>
<to>Tove</to>
```

```
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

آموزش خواندن و چاپ سند XML در PHP

در این بخش قصد داریم تا مفسر XML DOM را مقداردهی اولیه کرده، فایل XML را خوانده و اطلاعات آن را در خروجی نشان دهیم. برای این منظور کد زیر را می نویسیم :

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

خروجی کد مثال فوق به صورت زیر است:

```
Tove Jani Reminder Don't forget me this weekend!
```

اگر در مرورگر سورس کد صفحه را بگیرید) گزینه("View Source" ، کد HTML زیر را مشاهده می کنید:

```
<?xml version="1.0" encoding="UTF-8" ?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

مثال عملی فوق، یک شی DOM (Document object Model) را ایجاد کرده و فایل "note.xml"را می خواند و درون آن قرار می دهد. سپس تابع()saveXML ، محتویات XML استخراج شده را در یک متغیر متنی stringقرار داده که می توان آن را در خروجی چاپ نمود.

حرکت درون فایل XML با استفاده از حلقه Loop

در کد مثال عملی زیر، مفسر XML DOM را مقداردهی اولیه کرده و فایل XML را خوانده ایم. سپس با استفاده از یک حلقه loop به درون عناصر المنت حرکت کرده ایم :

```
<?php
$xmlDoc = new DOMDocument();
$xmlDoc->load("note.xml");
$x = $xmlDoc->documentElement;
foreach ($x->childNodes AS $item) {
print $item->nodeName . " = " . $item->nodeValue . "<br>";
}
?>
```

خروجی کد فوق به صورت زیر خواهد بود:

```
<?php
```

```
#text =
to = Tove
#text =
from = Jani
#text =
heading = Reminder
#text =
body = Don't forget me this weekend!
#text =
?>
```

در مثال فوق، مشاهده می کنید که خالی بین هر یک از عنصرها وجود دارد. هنگامی که فایل XML تولید می شود، برخی مواقع فواصل سفید خالی بین گره ها به وجود می آید. مفسر XML DOMبا این عناصر همانند عنصرهای معمولی برخورد کرده و اگر از وجود آن ها اطلاع نداشته باشید، ممکن است برخی مواقع برنامه را دچار مشکل کنند.

بخش ششم : آموزش کاربرد Ajax درPHP

Ajaxچیست؟

Ajaxروشی جدید برای ویرایش بخش هایی از یک صفحه، بدون رفرش شدن و لود مجدد کل صفحه است. به عبارت دیگر، به وسیله Ajax فقط بخشی از صفحه که می خواهیم تغییر کند را ویرایش کرده و کل صفحه مجددا بارگذاری می شود.

Ajaxدر خلاصه به معنای استفاده غیر همزمان از جاوا اسکریپت و XML است.

Ajaxیک تکنیک جدید برای ایجاد و ساخت صفحات وب دینامیک می باشد.

Ajaxبه صفحات وب امکان می دهد تا بخش های مختلف خود را به صورت غیر همزمان و با رد و بدل حجمی از اطلاعات در پشت صفحه، به روز رسانی کنند. این کار به معنای این است که می توانید بخش های مختلفی از یک صفحه را بدون نیاز به لود کامل آن، به روز رسانی کنید.

صفحه های قدیمی و کلاسیک که از تکنولوژی Ajax استفاده نمی کنند، برای تغییر در هر بخشی از صفحه، مجبور هستند کل آن را مجددا لود کنند. این کار حجم درخواست های غیر ضروری و اطلاعات مبادله شده با سرور را بسیار افزایش داده و عملیات به روزرسانی صفحات را کند می کند.

بسیاری از سایت های بزرگ از تکنولوژی Ajax در سایت خود استفاده می کنند مثلyahoo ، google map، gmailو . ...

Ajaxچگونه کار می کنند؟

دیاگرام زیر نشان می دهد که Ajax چگونه کار می کند؟

Ajax با استانداردهای موجود ایجاد شده است:

Ajaxبر مبنای تکنولوژی های موجود وب ایجاد شده و درواقع ترکیبی از آن هاست :

از شی XML HTTP Request برای تبادل غیر همزمان اطلاعات با سرور استفاده می کنند.

از Javascript/DOM برای نمایش و دریافت اطلاعات استفاده می کند.

از CSS برای قالب دهی و نمایش اطلاعات بهره می گیرد.

از XML برای قالب بندی و انتقال اطلاعات استفاده می کند.

نکته :

تکنولوژی Ajax مستقل از نوع مرورگر و یا سیستم عامل عمل می کند .

به کارگیری Ajax در google Suggest

تکنولوژیAjax ، بیشتر در سال 2005 و به دنبال استفاده گوگل از آن در قابلیت google suggest معروف شد.

قابلیت google suggest یک امکان پویا در موتور جستجو گوگل است، به این صورت که وقتی شما به دنبال چیزی در گوگل جستجو کرده و نام آن را تایپ می کنید، با توجه به کاراکترهای وارد شده، گوگل لیستی از گزینه ها و یا نام های مرتبط با آن کلمه را به صورت دینامیک به شما نشان می دهد.

در بخش آموزش PHP سایت تحلیل داده، ما نحوه ویرایش بخش هایی از یک صفحه وب را به وسیلهAjax ، بدون نیاز به لود شدن کامل صفحه، آموزش می دهیم. کد مورد نظر برای انجام کار را نیز به زبان PHP خواهیم نوشت.

برای دریافت اطلاعات کامل تر، به بخش آموزش Ajax در سایت تحلیل داده بروید .

آموزش کار با Ajax در PHP

همانطور که در درس قبل اشاره کردیم، از Ajax برای ایجاد صفحات دینامیک و پویا در سایت های PHP استفاده می شود.

در کد مثال عملی زیر، نشان داده ایم چگونه یک صفحه وب می تواند در حالی که کاربر کاراکترهای مورد نظر خود را در کادر متن جستجو وارد می کند، در پشت صحنه با سرور تعامل داشته و اطلاعات خود را به روز کند

توضیح مثال :

در کد متال فوق، هنگامی که کاربر در کادر متن نوشته ای را وارد می کند، یک تابع به نام "show Hint()" اجرا می شود. این تابع با رویداد on key up فراخوانی می شود.

کد HTML صفحه مثال به صورت زیر است :

```
<?php
<html>
<head>
<script>
function showHint(str) {
if (str.length == 0) {
document.getElementById("txtHint").innerHTML = "";
return;
} else {
var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function () {
if (this.readyState == 4 && this.status == 200) {
document.getElementById("txtHint").innerHTML =
this.responseText;
}
};
xmlhttp.open("GET", "gethint.php?q=" + str, true);
xmlhttp.send();
}
}
</script>
</head>
<body>
<p><b>Start typing a name in the input field below:</b></p>
<form>
First name: <input type="text" onkeyup="showHint(this.value)">
</form>
<p>Suggestions: <span id="txtHint"></span></p>
</body>
</html>
?>
```

نکات مثال

در مرحله اول، برنامه چک می کند آیا کادر متن ورودی خالی است یا خیر) آیا مقدار str.length==0 است یا نه .(اگر مقدار آن خالی باشد، محتویات txt Hint placeholder را چک کرده و از تابع خارج می شود. اما اگر مقدار کادر متن خالی نباشد، کارهای زیر را به ترتیب انجام می دهد :

یک شی جدید XMLHttp Request را ایجاد می کند.

تابع لازم جهت اجرا را پس از این که پاسخ سرور آماده شد، تهیه می کند .

درخواست یا request را به یک فایل به نام "gethint.php" بر روی سرور ارسال می کند.

توجه داشته باشید که پارامتر q به آدرس (url) صفحه اضافه شده است.(gethint.php?q="+str)

متغیر str نیز مقدار کادر متن را در خود ذخیره می کند.

محتویات فایل PHP مثال "gethint.php"

فایل PHP یک آرایه از نام ها را چک می کند و مقادیری که با مقدار جستجو شده کاربر مرتبط هستند را به مرورگر باز می گرداند :

```
<?php
// Array with names
$a[] = "Anna";
$a[] = "Brittany";
$a[] = "Cinderella";
$a[] = "Diana";
$a[] = "Eva";
$a[] = "Fiona";
$a[] = "Gunda";
$a[] = "Hege";
$a[] = "Inga";
$a[] = "Johanna";
$a[] = "Kitty";
$a[] = "Linda";
$a[] = "Nina";
$a[] = "Ophelia";
$a[] = "Petunia";
$a[] = "Amanda";
$a[] = "Raquel";
$a[] = "Cindy";
$a[] = "Doris";
$a[] = "Eve";
$a[] = "Evita";
$a[] = "Sunniva";
$a[] = "Tove";
$a[] = "Unni";
$a[] = "Violet";
$a[] = "Liza";
$a[] = "Elizabeth";
```

```
$a[] = "Ellen";
$a[] = "Wenche";
$a[] = "Vicky";
// get the q parameter from URL
$q = $_REQUEST["q"];
$hint = "";
// lookup all hints from array if $q is different from ""
if ($q !== "") {
$q = strtolower($q);
$len=strlen($q);
foreach($a as $name) {
if (stristr($q, substr($name, 0, $len))) {
if ($hint === "") {
$hint = $name;
} else {
$hint .= ", $name";
}
}
}
}
// Output "no suggestion" if no hint was found or output correct values
echo $hint === "" ? "no suggestion" : $hint;
?>
```

واکشی اطلاعات از دیتابیس با فراخوانی توابع AJAX

مثال ذیل نشان می دهد چگونه یک صفحه ی وب (از اپلیکیشن تحت وب) می تواند با فراخوانی توابعAJAX ، داده از دیتابیس واکشی کند:

در این مثال کاربر اسم شخص مورد نظر را از لیست کشویی انتخاب کرده و اطلاعات مربوطه ی آن را بدون اینکه کل صفحه بروز رسانی شود، از دیتابیس می خواند و در زیر نمایش می دهد:

آموزشگاه تحلیلگر داده ها

Example

Peter Griffin

Firstname	Lastname	Age	Hometown	Job
Peter	Griffin	41	Quahog	Brewery

Example

Lois Griffin

Firstname	Lastname	Age	Hometown	Job
Lois	Griffin	40	Newport	Piano Teacher

Example

Joseph Swanson

Firstname	Lastname	Age	Hometown	Job
Joseph	Swanson	39	Quahog	Police Officer

Example

Glenn Quagmire

Firstname	Lastname	Age	Hometown	Job
Glenn	Quagmire	41	Quahog	Pilot

شرح مثال – دیتابیس MySQL

جدول دیتابیسی که اطلاعات مثال فوق از آن واکشی شده و به نمایش در می آید، مشابه زیر می باشد:

Job	Hometown	Age	LastName	FirstName	id
Brewery	Quahog	۴۱	Griffin	Peter	۱
Piano Teacher	Newport	۴۰	Griffin	Lois	۲
Police Officer	Quahog	۳۹	Swanson	Joseph	۳
Pilot	Quahog	۴۱	Quagmire	Glenn	۴

شرح مثال

در مثال فوق، زمانی که یک کاربر شخص مورد نظر را از لیست کشویی (drop-down list) انتخاب می کند، در واقع یک تابع به نام "showUser()" فراخوانی می شود.

این تابع را رخداد onchange فراخوانی می کند.

کد HTML مربوطه را در زیر مشاهده می کنید.

مثال:

```
<?php
<script>
function showUser(str) {
if (str == "") {
document.getElementById("txtHint").innerHTML = "";
return;
} else {
if (window.XMLHttpRequest) {
// code for IE7+, Firefox, Chrome, Opera, Safari
xmlhttp = new XMLHttpRequest();
} else {
// code for IE6, IE5
xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById("txtHint").innerHTML =
this.responseText;
}
};
xmlhttp.open("GET","getuser.php?q="+str,true);
xmlhttp.send();
}
}
</script>


<select name="users" onchange="showUser(this.value)">
<option value="">Select a person:</option>
<option value="1">Peter Griffin</option>
<option value="2">Lois Griffin</option>
<option value="3">Joseph Swanson</option>
<option value="4">Glenn Quagmire</option>
</select>

<br>
<div id="txtHint"><b>Person info will be listed here...</b></div>
?>
```

شرح کد:

ابتدا می بایست شخص مورد نظر را انتخاب نمایید .چنانچه شخصی انتخاب نشده باشد(str == "") ، در آن صورت محتوای txtHint را حذف و از تابع خارج می شود. اگر شخص دلخواه انتخاب شده باشد، آنگاه اقدامات زیر صورت می گیرد :

- یک آبجکت XMLHttpRequest ایجاد می شود.
- یک تابع تعریف می شود که به هنگام آماده شدن پاسخ سرویس دهنده(server response) ، فراخوانی شده و اجرا می شود.
- درخواست (request) را به فایل مستقر بر روی سرویس دهنده هدایت می شود .
- یک پارامتر (q) به URL به همراه محتوای لیست کشویی) اضافه می شود.

فایل PHP

صفحه ای از اپلیکیشن تحت وب مستقر در سرویس دهنده که کد JavaScript بالا آن را فراخوانی می کند، در واقع یک فایل PHP به نام "getuser.php"می باشد .

کد موجود در فایل "getuser.php" یک کوئری از دیتابیس MySQL می گیرد و نتیجه را در قالب یک جدول HTMLبرمی گرداند:

```
<?php
<style>
table {
width: 100%;
border-collapse: collapse;
}

table, td, th {
border: 1px solid black;
padding: 5px;
}

th {
text-align: left;
}
</style>


<!--?php
$q = intval($_GET['q']);
$con = mysqli_connect('localhost','peter','abc123','my_db');
if (!$con) {
die('Could not connect: ' . mysqli_error($con));
}
mysqli_select_db($con,"ajax_demo");
$sql="SELECT * FROM user WHERE id = '".$q."'";
$result = mysqli_query($con,$sql);
echo "<table-->

Firstname
Lastname
```

```
Age
Hometown
Job
";
while($row = mysqli_fetch_array($result)) {
echo "";
echo "" . $row['FirstName'] . "";
echo "" . $row['LastName'] . "";
echo "" . $row['Age'] . "";
echo "" . $row['Hometown'] . "";
echo "" . $row['Job'] . "";
echo "";
}
echo "";
mysqli_close($con);
?>
?>
```

شرح: زمانی که کوئری و دستور درخواست داده از JavaScript به فایل PHP ارسال می شود، اتفاقات زیر به ترتیب رخ می دهد:

1. PHPیک connection به سرور MySQL ایجاد می کند.
2. شخص مورد نظر و اطلاعات مربوطه ی آن یافت می شود.
3. یک جدول HTML ایجاد شده با داده های مورد نیاز پر می شود و متعاقبا به پارامتر مکان نگهدار "txtHint" (placeholder)ارسال می گردد.

آموزش AJAX – PHP و / XML واکشی اطلاعات از فایل XML با توابع AJAX

مثال XML و AJAX

مثال زیر نمایش می دهد یک صفحه ی وب چگونه اطلاعات لازم را از فایل XML با توابعAJAX ، بدون نیاز به بارگذاری مجدد کل صفحه، واکشی کرده و نشان می دهد:

Example

Bob Dylan

TITLE: Empire Burlesque
ARTIST: Bob Dylan
COUNTRY: USA
COMPANY: Columbia
PRICE: 10.90
YEAR: 1985

Example

Bee Gees

TITLE: One night only
ARTIST: Bee Gees
COUNTRY: UK
COMPANY: Polydor
PRICE: 10.90
YEAR: 1998

Example

Cat Stevens

TITLE: The very best of
ARTIST: Cat Stevens
COUNTRY: UK
COMPANY: Island
PRICE: 8.90
YEAR: 1990

شرح مثال – صفحه ی HTML

زمانی که کاربر یک CD را از لیست کشویی حاضر انتخاب می کند، تابعی به نام "showCD()" فراخوانی و اجرا می شود.

تابع مورد نظر به دنبال رخداد "onchange" صدا زده می شود:

```
<?php
<script>
function showCD(str) {
if (str==="") {
document.getElementById("txtHint").innerHTML="";
return;
}
if (window.XMLHttpRequest) {
// code for IE7+, Firefox, Chrome, Opera, Safari
xmlhttp=new XMLHttpRequest();
} else {  // code for IE6, IE5
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
```

```
xmlhttp.onreadystatechange=function() {
if (this.readyState==4 && this.status==200) {
document.getElementById("txtHint").innerHTML=this.responseText;
}
}
xmlhttp.open("GET","getcd.php?q="+str,true);
xmlhttp.send();
}
</script>


Select a CD:
<select name="cds" onchange="showCD(this.value)">
<option value="">Select a CD:</option>
<option value="Bob Dylan">Bob Dylan</option>
<option value="Bee Gees">Bee Gees</option>
<option value="Cat Stevens">Cat Stevens</option>
</select>
?>
```

تابع showCD() عملیات زیر را انجام می دهد

- بررسی می کند آیا آیتمی انتخاب شده یا خیر.
- یک آبجکت XMLHttpRequest ایجاد می کند.
- زمانی که پاسخ سرور آماده می شود، یک تابع اعلان می شود.
- درخواست را به فایل مستقر بر روی سرور هدایت می کند .
- پارامتر (q) به) URL به همراه محتوای لیست کشویی) ارسال می شود .

فایل PHP

صفحه ای که کد جاوا اسکریپت آن را فراخوانی می کند، در واقع یک فایل PHP به نام "getcd.php" از اپلیکیشن تحت وب می باشد .

اسکریپت PHP فایلXML ، "cd_catalog.xml"، را بارگذاری کرده، یک کوئری بر روی فایل XML اجرا می کند و در پایان نتیجه ی مورد نظر را در قالب HTML از سرور برمی گرداند:

```
<!--?php
$q=$_GET["q"];
$xmlDoc = new DOMDocument();
$xmlDoc--->load("cd_catalog.xml");
$x=$xmlDoc->getElementsByTagName('ARTIST');
for ($i=0; $i<=$x->length-1; $i++) {
//Process only element nodes
if ($x->item($i)->nodeType==1) {
```

```
    if ($x->item($i)->childNodes->item(0)->nodeValue == $q) {
                $y=($x->item($i)->parentNode);
                }
                }
                }
            $cd=($y->childNodes);
        for ($i=0;$i<$cd->length;$i++) {
            //Process only element nodes
        if ($cd->item($i)->nodeType==1) {
    echo("<b>" . $cd->item($i)->nodeName . ":</b> ");
  echo($cd->item($i)->childNodes->item(0)->nodeValue);
            echo("<br>");
            }
            }
?>
```

زمانی که کوئری CD از اسکریپت JavaScript به صفحه ی PHP ارسال می شود، اتفاقات زیر به ترتیب رخ می دهد:

- PHPیک آبجکت XML DOM ایجاد می کند .
- تمامی المان های که با اسم ارسالی از جاوا اسکپریت منطبق می باشد را پیدا می کند.
- اطلاعات آلبوم را به پارامتر جایگزین و مکان نگهدار "txtHint" ارسال می کند.

آموزش – PHP پیاده سازی قابلیت جستجوی تعاملی و زنده با توابع AJAX (AJAX Live search)

پیاده سازی قابلیت جستجو تعاملی با توابع AJAX

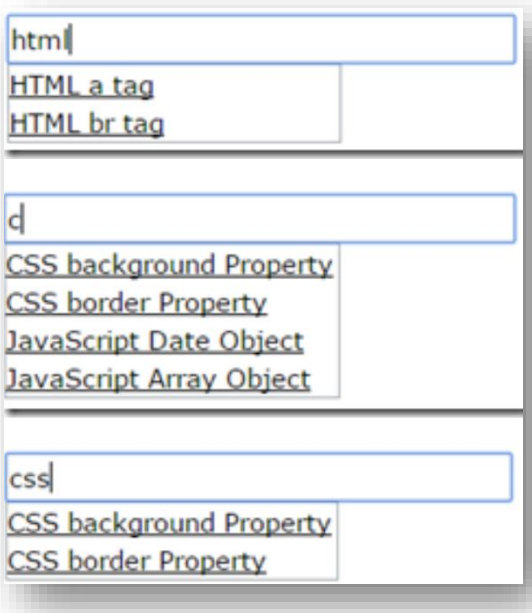
مثال زیر یک کادر با قابلیت جستجوی تعاملی و زنده را پیاده سازی می کند که در آن به هنگام درج واژگان مورد نیاز در کادر و بدون بارگذاری مجدد کل صفحه، کاربر بلافاصله نتایج دلخواه را مشاهده می کند.

قابلیت جستجوی تعاملی و زنده (Live search) در مقایسه با جستجو به روش قدیمی، از مزایای زیر برخوردار می باشد :

- همین که کاربر در حال تایپ واژگان جستجو در کادر می باشد، نتایج در لحظه برای وی نمایش داده می شوند.
- همین که کاربر تایپ را ادامه می دهد، نتایج جستجو محدود و محدودتر می شود .

- زمانی که نتایج بیش از حد محدود می شود، کاربر می تواند با حذف تعدادی از کاراکترها در کادر، موارد بیشتری را مشاهده کند.

نمونه ی زیر یک کادر با قابلیت جستجوی تعاملی و نمایش نتایج در لحظه را نمایش می دهد:



نتایجی که در مثال بالا مشاهده می کنید، داخل یک فایل XML قرار دارد که محتوای آن را در زیر مشاهده می کنید:

```
<?php
<pages>
<link>
<title>HTML a tag</title>
<url>https://www.w3schools.com/tags/tag_a.asp</url>

<link>
<title>HTML br tag</title>
<url>https://www.w3schools.com/tags/tag_br.asp</url>

<link>
<title>CSS background Property</title>
<url>
https://www.w3schools.com/cssref/css3_pr_background.asp
</url>

<link>
<title>CSS border Property</title>
```

```
<url>https://www.w3schools.com/cssref/pr_border.asp</url>

<link>
<title>JavaScript Date Object</title>
<url>https://www.w3schools.com/jsref/jsref_obj_date.asp</url>

<link>
<title>JavaScript Array Object</title>
<url>
https://www.w3schools.com/jsref/jsref_obj_array.asp
</url>

</pages>
?>
```

شرح مثال :

صفحه HTML

زمانی که کاربر یک کاراکتر را در فیلد ورودی بالا درج می کند، تابع "showResult()" فراخوانی و اجرا می شود. تابع ذکر شده به دنبال اتفاق افتادن رخداد) "onkeyup" رها شدن دکمه پس از درج کلمه توسط کاربر) فراخوانی می شود :

```
<?php
<script>
function showResult(str) {
if (str.length==0) {
document.getElementById("livesearch").innerHTML="";
document.getElementById("livesearch").style.border="0px";
return;
}
if (window.XMLHttpRequest) {
// code for IE7+, Firefox, Chrome, Opera, Safari
xmlhttp=new XMLHttpRequest();
} else {  // code for IE6, IE5
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.onreadystatechange=function() {
if (this.readyState==4 && this.status==200) {
document.getElementById("livesearch").innerHTML=this.responseText;
document.getElementById("livesearch").style.border="1px solid #A5ACB2";
}
}
xmlhttp.open("GET","livesearch.php?q="+str,true);
xmlhttp.send();
}
</script>



<input size="30" onkeyup="showResult(this.value)" type="text">
<div id="livesearch"></div>
?>
```

شرح کد :

زمانی که فیلد یا کادر جستجو تهی است (str.length==0)، تابع محتوای پارامتر مکان نگهدار livesearch (placeholder)را پاک کرده و از تابع خارج می شود .

اگر input field تهی نبود، تابع showResult() عملیات ذیل را به ترتیب انجام می دهد :

1. یک آبجکت XMLHttpRequest ایجاد می کند.
2. تابعی را ایجاد می کند که به هنگام آماده شدن پاسخ سرویس دهنده(server response) ، فراخوانی شده و اجرا می شود .
3. درخواست را به فایل مستقر در سرویس دهنده هدایت می کند .
4. یک پارامتر) (q) به همراه محتوای فیلد) به URL اضافه می کند .

فایل PHP

صفحه ی مورد نظر از اپلیکیشن تحت وب که کد JavaScript آن را فراخوانی می کند، یک فایل PHP به نام "livesearch.php"می باشد.

کد موجود در فایل "livesearch.php" به دنبال عنوان هایی) المان های (title که با عبارت جستجو (search string)منطبق هستند، داخل فایل XML می گردد و در صورت یافتن نتایج مورد نیاز آن ها را واکشی می کند:

```
<?php
load("links.xml");
$x=$xmlDoc->getElementsByTagName('link');
//get the q parameter from URL
$q=$_GET["q"];
//lookup all links from the xml file if length of q>0
if (strlen($q)>0) {
$hint="";
for($i=0; $i<($x->length); $i++) {
$y=$x->item($i)->getElementsByTagName('title');
$z=$x->item($i)->getElementsByTagName('url');
if ($y->item(0)->nodeType==1) {
//find a link matching the search text
if (stristr($y->item(0)->childNodes->item(0)->nodeValue,$q)) {
if ($hint=="") {
$hint="" .
$y->item(0)->childNodes->item(0)->nodeValue . "";
} else {
$hint=$hint . "
```

```
" .
$y->item(0)->childNodes->item(0)->nodeValue . "";
}
}
}
}
}
// Set output to "no suggestion" if no hint was found
// or to the correct values
if ($hint=="") {
$response="no suggestion";
} else {
$response=$hint;
}
//output the response
echo $response;
?>
?>
```

زمانی که اسکریپت یا کد مربوطه ی((strlen($q) > 0)) JavaScript ، متنی را به سرویس دهنده ارسال می کند، اتفاقات زیر به ترتیب رخ می دهند:

- محتوای یک فایل XML داخل آبجکت XML DOM در حافظه بارگذاری می شود.
- داخل تمامی المان های title حلقه زده (می چرخد) و تا (زمانی که) مورد منطبق با متن ارسال شده از کد JavaScript را پیدا کند.
- urlصحیح و title مناسب را در متغیر "$response"ذخیره می کند. زمانی که مورد منطبق بیش از یک آیتم باشد، در آن صورت تمامی موارد منطبق و یافت شده داخل متغیر مزبور قرار داده می شوند.
- اگر هیچ مورد منطبقی یافت و استخراج نشد، مقدار متغیر $response برابر رشته ی "no suggestion"تنظیم می شود .

آموزش - PHP آموزش پیاده سازی خبر خوان RSS READER با AJAX در PHP

AJAX RSS Reader – RSS Readerای که داده ها را بدون بروز رسانی کل صفحه، در اپلیکیشن تحت وب بارگذاری می کند

مثال زیر یک RSS Reader را نمایش می دهد که در آن خبرهای RSS با فراخوانی توابع ajax و بدون لود مجدد کل صفحه در اپلیکیشن بارگذاری می شود:

شرح مثال – فایلHTML

زمانی که کاربر یک آیتم RSS-feed را از لیست کشویی انتخاب می کند، تابعی به نام "showRSS()" فراخوانی می شود که خبر را از خبرخوان مربوطه خوانده در صفحه، بدون بارگذاری کل آن، بارگذاری می کند .

تابع ذکر شده به دنبال رخداد "onchange" اجرا می شود :

```
<?php
<script>
function showRSS(str) {
  if (str.length==0) {
    document.getElementById("rssOutput").innerHTML="";
    return;
  }
  if (window.XMLHttpRequest) {
```

```
    // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    } else {  // code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange=function() {
    if (this.readyState==4 && this.status==200) {
document.getElementById("rssOutput").innerHTML=this.responseText;
    }
    }
    xmlhttp.open("GET","getrss.php?q="+str,true);
    xmlhttp.send();
    }
</script>



<select onchange="showRSS(this.value)">
<option value="">Select an RSS-feed:</option>
<option value="Google">Google News</option>
<option value="NBC">NBC News</option>
</select>

<br>
<div id="rssOutput">RSS-feed will be listed here...</div>
?>
```

تابع showRSS() عملیات زیر را به ترتیب انجام می دهد :

- بررسی می کند آیا یک آیتم RSS-feed انتخاب شده است یا خیر.
- یک آبجکت XMLHttpRequest ایجاد می کند.
- یک تابع تعریف می کند و این تابع زمانی اجرا می شود که سرویس دهنده آماده ی پاسخ به درخواست ارسالی از کد جاوااسکریپت می باشد.
- درخواست ارسال شده به سرویس دهنده را به فایل مربوطه هدایت می کند.
- یک پارامتر (q) به URL با محتوای لیست کشویی) اضافه می کند.

فایلPHP

صفحه ای که بر روی سرویس دهنده قرار داشته و کد JavaScriptفوق آن را فراخوانی می کند، در واقع یک فایل PHP به نام "getrss.php"می باشد. محتوای فایل مزبور را در زیر مشاهده می کنید:

```
<!--?php
//get the q parameter from URL
$q=$_GET["q"];
```

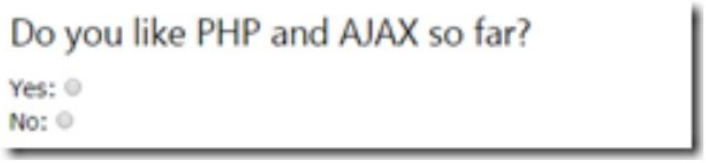
```
//find out which feed was selected
if($q=="Google") {
$xml=("http://news.google.com/news?ned=us&topic=h&output=rss");
} elseif($q=="NBC") {
$xml=("http://rss.msnbc.msn.com/id/3032091/device/rss/rss.xml");
}
$xmlDoc = new DOMDocument();
$xmlDoc--->load($xml);
//get elements from "<channel>"
$channel=$xmlDoc->getElementsByTagName('channel')->item(0);
$channel_title = $channel->getElementsByTagName('title')
->item(0)->childNodes->item(0)->nodeValue;
$channel_link = $channel->getElementsByTagName('link')
->item(0)->childNodes->item(0)->nodeValue;
$channel_desc = $channel->getElementsByTagName('description')
->item(0)->childNodes->item(0)->nodeValue;
//output elements from "<channel>"
echo("<p><a href="" . $channel_link
. ">" . $channel_title . "</a>");
echo("<br>");
echo($channel_desc . "</p>");
//get and output "<item>" elements
$x=$xmlDoc->getElementsByTagName('item');
for ($i=0; $i<=2; $i++) {
$item_title=$x->item($i)->getElementsByTagName('title')
->item(0)->childNodes->item(0)->nodeValue;
$item_link=$x->item($i)->getElementsByTagName('link')
->item(0)->childNodes->item(0)->nodeValue;
$item_desc=$x->item($i)->getElementsByTagName('description')
->item(0)->childNodes->item(0)->nodeValue;
echo ("<p><a href="" . $item_link
. ">" . $item_title . "</a>");
echo ("<br>");
echo ($item_desc . "</p>");
}
?>
</item></channel></channel>
```

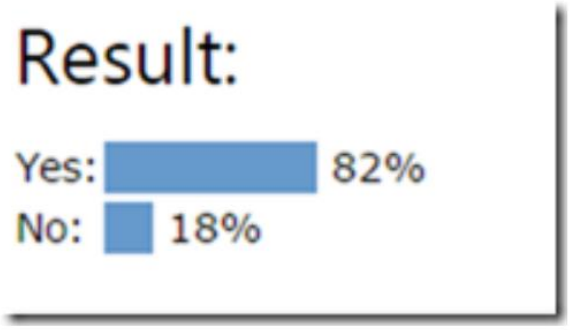
زمانی که درخواستی جهت واکشی خبر RSS از اسکریپت سمت کلاینت JavaScript به سرویس دهنده فرستاده می شود، عملیات زیر به ترتیب انجام می شوند :

- ابتدا بررسی می کند کدام آیتم انتخاب شده است.
- یک آبجکت XML DOM جدید ایجاد می کند.
- فایل RSS را در متغیر xml بارگذاری می کند.
- المان ها را از المان channel استخراج کرده و به عنوان خروجی پاس می دهد.
- المان ها را از المان item استخراج کرده و به عنوان خروجی بازمی گرداند.

پیاده سازی قابلیت نظرسنجی با(AJAX Poll) AJAX

مثال زیر یک قابلیت نظرسنجی ساده را نمایش می دهد که در آن نتایج بدون بارگذاری مجدد کل صفحه و در لحظه نمایش داده می شود. در واقع زمانی که کاربر یکی از دو گزینه را انتخاب می کند، برنامه ی زیر نتیجه را از سرویس دهنده خوانده و بلافاصله بدون بروز رسانی کل صفحه، در اپلیکیشن تحت وب حاضر به نمایش می گذارد.





شرح مثال – فایل HTML

زمانی که کاربر یکی از دو گزینه ی فوق را انتخاب می کند، تابعی به نام "getVote()" فراخوانی می گردد. این تابع به هنگام فعال و اجرا شدن رخداد"onclick" ، فراخوانی می شود :

```
<?php
<script>
function getVote(int) {
if (window.XMLHttpRequest) {
// code for IE7+, Firefox, Chrome, Opera, Safari
xmlhttp=new XMLHttpRequest();
} else {  // code for IE6, IE5
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.onreadystatechange=function() {
if (this.readyState==4 && this.status==200) {
document.getElementById("poll").innerHTML=this.responseText;
}
}
xmlhttp.open("GET","poll_vote.php?vote="+int,true);
xmlhttp.send();
}
</script>
```

```
<div id="poll">
<h3>Do you like PHP and AJAX so far?</h3>

Yes:
<input name="vote" value="0" onclick="getVote(this.value)" type="radio">
<br>No:
<input name="vote" value="1" onclick="getVote(this.value)" type="radio">

</div>
?>
```

تابع getVote() عملیات زیر را به ترتیب انجام می دهد :

- ابتدا یک آبجکت XMLHttpRequest ایجاد می کند .
- تابعی اعلان می کند که به هنگام آماده شدن پاسخ از سرویس دهنده(server response) ، فراخوانده و اجرا می شود.
- درخواست را به فایل مستقر بر روی سرویس دهنده هدایت می کند .
- یک پارامتر (vote) به URL الصاق می کند همراه با مقدار yes یاno

فایل PHP

صفحه ی مورد نظر از اپلیکیشن تحت وب و مستقر در سرویس دهنده را کد JavaScript بالا فراخوانی می کند. این صفحه در واقع یک فایل PHP به نام "poll_vote.php" با محتوای ذیل می باشد:

```
<!--?php
$vote = $_REQUEST['vote'];
//get content of textfile
$filename = "poll result.txt";
$content = file($filename);
//put content in array
$array = explode("||", $content[0]);
$yes = $array[0];
$no = $array[1];
if ($vote == 0) {
$yes = $yes + 1;
}
if ($vote == 1) {
$no = $no + 1;
}
//insert votes to txt file
$insertvote = $yes."||".$no;
$fp = fopen($filename,"w");
fputs($fp,$insertvote);
fclose($fp);
?-->
<h2>Result:</h2>
```

```
<table>
<tbody><tr>
<td>Yes:</td>
<td>
<img src="poll.gif" width="<?php echo(100*round($yes/($no+$yes),2)); ?>"
height="20">
<!--?php echo(100*round($yes/($no+$yes),2)); ?-->%
</td>
</tr>
<tr>
<td>No:</td>
<td>
<img src="poll.gif" width="<?php echo(100*round($no/($no+$yes),2)); ?>"
height="20">
<!--?php echo(100*round($no/($no+$yes),2)); ?-->%
</td>
</tr>
</tbody></table>
```

مقدار ورودی کاربر توسط کد JavaScript به سرویس دهنده ارسال می شود و به تبع آن اتفاقات زیر به ترتیب رخ می دهد:

- محتوای فایل متنی به نام "poll_result.txt" بازیابی می شود.
- محتوای فایل مزبور داخل دو متغیر قرار داده شده و مقدار 1 به متغیر انتخابی اضافه می شود .
- نتیجه یا خروجی را داخل فایل "poll_result.txt" درج (write) می شود .
- نمود گرافیکی از نتیجه ی نظرسنجی را در خروجی (سمت کلاینت و داخل نمایشگر) قرار می دهد.

فایل متنی

اطلاعات واکشی شده از سرویس دهنده و درواقع نتیجه ی نظرسنجی داخل فایل متنی (poll_result.txt) ذخیره می شود.

این داده ها با فرمت و شکل زیر ذخیره می شود:

0||0

عدد اول از سمت چپ، مقادیر "Yes" را تشکیل داده و عدد دوم نشانگر مقادیر "No" در نظر سنجی می باشد .

نکته :

لازم است مجوز ویرایش فایل متنی را به سرویس دهنده ی وب (web server PHP) اعطا نمایید .