



آموزش PHP

تألیف: مهندس افشین رفوآ
www.tahlildadeh.com



فهرست

10	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؟
11	PHP چه کارهایی می تواند انجام دهد؟
11	چرا از PHP استفاده کنیم؟
11	آموزش نصب زبان PHP
12	نصب PHP Parser
12	پیکربندی پا Apache config
12	پیکربندی فایل PHP.INI
12	پیکربندی IIS WINDOWS
12	آموزش ساختار کدنویسی PHP
12	آموزش ساختار دستوری یا Syntax زبان PHP
13	آموزش درجه توضیحات Comments در زبان PHP
14	اهمیت بزرگ یا کوچک بودن حروف در PHP
14	آموزش تعریف و استفاده از متغیرها در زبان PHP
15	قوانین نام گذاری متغیرها در PHP
15	نمایش مقدار متغیرها در خروجی PHP
16	یک زبان برنامه نویسی بدون نیاز به تعریف نوع داده ای است PHP
16	میدان دید (Scope) در متغیر های PHP
16	متغیر های محلی Local و عمومی Global
17	آموزش کاربرد واژه کلیدی Global در PHP
18	آموزش استفاده از واژه کلیدی static در PHP
19	آموزش چاپ خروجی در زبان PHP
19	آموزش چاپ خروجی با دستورات echo و print در PHP
19	آموزش کار با دستور echo در PHP
20	آموزش کار با دستور Print در PHP
20	آموزش انواع داده ای Data Types در PHP
21	آشنایی با متغیر متنی String در PHP
21	آشنایی با متغیر عددی Integer در PHP

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

22	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
23	آشنایی با متغیر تهی یا Null در: PHP
24	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 یا String
24	شمارش تعداد کلمات موجود در یک رشته متن یا String
24	بر عکس کردن یک رشته متی یا String در: PHP
25	آموزش جستجو برای یک کلمه خاص در یک رشته متی String در PHP
25	آموزش جایگزین کردن یک کلمه درون متغیر متی string در: PHP
25	مرجع کامل کار با متغیر متی String در PHP
25	آموزش تعریف و کار با مقادیر ثابت یا Constant در PHP
26	آموزش تعریف و کار با مقادیر ثابت یا Constant در PHP
27	آموزش کار با Constant ها در PHP عمومی یا Global هستند:
27	آموزش کار با عملگرها یا operators در زبان: PHP
27	آموزش کار با ساختارهای شرطی if else در PHP
29	آموزش ساختار شرطی if ... else statement
29	آموزش ساختار شرطی: if ... else if ... else statement
30	آموزش کار با ساختار شرطی switch در PHP
30	آموزش کار با ساختار دستوری switch در: PHP
32	آموزش کار با حلقه های loop در PHP
32	آموزش کار با حلقه while loop در PHP
33	آموزش کار با حلقه do...while در PHP
34	آموزش کار با تابع Function در PHP
34	آموزش تعریف تابع های دلخواه کاربر در PHP
35	آموزش ارسال پارامتر به تابع در زبان PHP
36	مقدار پیش فرض پارامترها در تابع PHP
37	آموزش مقدار برگشتی Return Value در تابع PHP
37	آموزش کار با آرایه یا Array در زبان: PHP
37	یک آرایه چیست؟...
38	آموزش تعریف آرایه Array در PHP

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

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

78	آموزش خواندن یک کاراکتر در فایل PHP با() fgetc
78	آموزش ایجاد (create) و نوشتن (write) در فایل های PHP
79	آموزش ایجاد (create) فایل ها در PHP -تابع() fopen
79	آموزش اجازه دسترسی یا Permission در فایل های PHP
79	آموزش نوشتن (write) در یک فایل PHP با تابع(): fwrite
80	عملیات overwriting در PHP
80	آموزش ارسال فایل Upload در زبان PHP
80	مرحله اول تنظیم فایل "php.ini"
81	مرحله دوم-ایجاد فرم Html لازم جهت آپلود فایل
81	مرحله سوم- تعیین اسکریپت لازم جهت آپلود فایل
82	آموزش چک کردن این که فایل از قبل وجود داشته یا نه:
83	آموزش تعیین حجم مجاز فایل جهت آپلود:
83	آموزش تعیین نوع داده ای فایل جهت آپلود
83	کد کامل اسکریپت لازم جهت آپلود فایل
84	مرجع کامل کار با FileSystem در PHP
85	آموزش کار با کوکی Cookies در زبان PHP
85	آموزش ایجاد کوکی Cookie در زبان PHP
86	آموزش تغییر مقدار یک کوکی در زبان PHP
87	آموزش حذف یک کوکی در PHP
87	آموزش چک کردن فعال بودن کوکی در مرورگر کاربر
88	مرجع کامل HTTP در PHP
88	آموزش کار با Session در زبان PHP
88	مفهوم Session در PHP چیست؟
89	آموزش نحوه کار شروع یک Session در PHP
89	آموزش خواندن مقادیر متغیر Session در PHP
90	تغییر یک متغیر PHP Session Variable
91	آموزش از بین بردن یک Session در زبان PHP
91	آموزش کار با Filters در زبان PHP
92	آموزش کار با تابع () filter_var در PHP
92	آموزش نحوه پاکسازی یک String در PHP
93	آموزش چک کردن یک متغیر integer در PHP
93	آموزش چک کردن درستی یک IP با تابع() filter_var
94	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 در PHP
94	آموزش پیشرفته Filter کردن اطلاعات در زبان PHP
95	آموزش چک کردن مقدار یک عدد Integer در محدوده مورد نظر
95	آموزش اعتبارسنجی یک IPv6 با تابع filter_var()
95	آموزش چک کردن Query String در URL
95	آموزش حذف کاراکترهای کد ASCII از یک String
96	مرجع کامل کار با Filter ها در زبان PHP
96	آموزش مدیریت خطای PHP (Error Handling)
97	ایجاد یک مدیریت کننده خطای خواه (Custom Error Handling)
98	جدول درجه های خطای PHP
99	آموزش تنظیم تابع مدیریت خطای PHP
100	فعال کردن یک خطای error trigger
101	آموزش ثبت خطای error logging
101	اموزش نحوه ارسال یک پیام خطای Error Message با ایمیل
102	آموزش کار با استثناء ها (Exception) در زبان PHP
102	یک استثناء یا Exception چیست؟
103	آموزش نحوه ساده استفاده از Exception
104	آموزش کار با ساختارهای دستوری Try ، throw و Catch
108	آموزش ارسال مجدد خطای exception
110	آموزش تنظیم یک مدیریت کننده Exception سطح بالا
110	قوانين مربوط به exception در زبان PHP
111	بخش چهارم : آموزش پایگاه داده MySQL در PHP
111	آموزش کار با پایگاه داده MySQL در PHP
111	پایگاه داده MySQL چیست؟
112	سیستم کارکرد پایگاه داده MySQL با PHP
112	آموزش جستجو در پایگاه داده MySQL با Database Query
113	واقعیت هایی درباره پایگاه داده MySQL
113	آموزش اتصال Connect به پایگاه داده MySQL
113	کی و کجا از MySQL یا PDO استفاده کنیم؟
114	ارائه مثال های عملی با ساختار دستوری PDO و MySQLi
114	آموزش نصب MySQLi
114	آموزش نصب PDO

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

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



آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

بخش اول : آموزش مقدماتی PHP

معرفی زبان برنامه نویسی PHP

PHP چیست؟

- PHP مخفف عبارت "Hypertext Preprocessor" است به معنای زبان پیش پردازندۀ 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 :

- زبان طراحی صفحات وب
- CSS روش قالب دهنده 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 : یک زبان برنامه نویسی جذاب و پرطرفدار است. زبان PHP، به حدی قدرتمند است که می تواند هسته اصلی برنامه نویسی بزرگترین تیم وبلاگ دهنی 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.

چه کارهایی می تواند انجام دهد؟ PHP

- 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 رمزگاری کند.

با استفاده از زبان PHP، شما فقط محدود نیستید تا خروجی HTML تولید کنید. بلکه می توانید تصاویر، فایل های PDF و حتی فایل های تصویری فلاش 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 زبان PHP

به منظور راه اندازی صفحات وب PHP سه جزو اساسی باید بر روی سیستم نصب شده باشد .

: Microsoft's Internet Server (IIS) سرور وب _ سرور PHP تقریبا با تمام برنامه های وب سرور کار می کند از جمله Apache Server 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 چگونه PHP پارسر را روی سیستم نصب کنید.

نصب PHP 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 طراحی PHP بدانید.

آدرس زیر را در آدرس باکس مرورگر خود وارد نمایید.

<http://127.0.0.1/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 سرور apache هستید، این بخش چگونگی ویرایش پیکربندی فایل های apache را تشریح می کند.

پیکربندی فایل PHP.INI

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

IIS Windows پیکربندی

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 کدنویسی PHP

آموزش ساختار دستوری یا Syntax زبان: PHP

همانطور که قبلا اشاره کردیم، کدهای PHP بر روی سرور اشاره شده و سپس تایج 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 های PHP ، بایستی با عبارت پایان یابند. همانند کد مثال زیر:

```
<?php
<! --?php
// کدهای پی اج پی را در اینجا می نویسید
?-->
?>
```

پسوند پیش فرض فایل های PHP ، 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 "Hello world" را در خروجی چاپ کرده و نمایش دهد:

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

</h1><h1>ولین صفحه پی اج پی من</h1>

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

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

نکته مهم:

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

آموزش درجه توضیحات Comments در زبان: PHP

یک توضیح یا Comment در کدهای PHP ، 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 مواقع ممکن است پس از یک مدت طولانی به کدهای نوشته شده خود بازگرددید، در اینجاست که بایستی بفهمید هر بخش چه کاری انجام می دهد.

در زبان PHP ، 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
    جند
```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```

    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 زبان PHP ، تمامی کلیدواژه های اصلی مثل echo , whi le , i f ، el se Case Senet i ve نمی باشند. یا کوچک حساس نیستند. به عبارت دیگر در کد مثال زیر، هر دستور echo 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 vari abl e به حروف بزرگ و کوچک حساس هستند .

در کد مثال زیر، فقط دستور خط اول مقدار متغیر \$Col or \$COLORR و به عنوان متغیرهای دیگری محسوب خواهند شد.

آموزش تعریف و استفاده از متغیرها در زبان PHP

همانطور که می دانید، متغیرها، فضاهایی در حافظه برای نگهداری اطلاعات هستند. در زبان PHP ، 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 Hello world! را خواهد داشت، متغیر \$x مقدار 5 و متغیر \$y نیز شامل مقدار 10.5 می شود.

نکته کاربردی:

در هنگام تعریف متغیرهای متنی در زبان PHP ، 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.

قوانين نام گذاری متغیرها در: PHP

یک متغیر در PHP ، می تواند یکم نام بسیار کوتاه مثل x یا y و یا یک نام بلند و بامفهوم مثل username ، age و ... داشته باشد. در نام گذاری متغیرهای PHP ، 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 زبان PHP ، 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 Sensitive هستند. بنابراین متغیرهای \$AGE و \$Age یا \$age با هم متفاوت خواهند بود.

نمایش مقدار متغیرها در خروجی: PHP

در زبان PHP ، معمولا می توان با استفاده از دستور echo ، 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.

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 متغیر PHP ، مجبور نیستیم نوع داده ای آن را به برنامه اعلام کنیم.

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

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

میدان دید (Scope) در متغیرهای PHP:

در زبان PHP ، یک متغیر را می توانید در هر بخشی از اسکریپت که بخواهید، تعریف کنید. میدان دید یا Scope یک 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 زبان PHP ، سه نوع میدان دید یا Scope برای متغیرها به شرح زیر وجود دارد:

- local یا محلی
- global یا سراسری
- static یا 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 function 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 در خارج از تابع myTest ، 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>";
}

myTest();
```

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```
<?php
<!--?php
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 خطای 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 خطای 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، **تمای متغیرهای سراسری** یا `$_GLOBAL` در یک آرایه به نام `[index]` برنامه را در نگهداری می‌کند. در این آرایه `[index]` در واقع نام متغیرها می‌باشد. از این آرایه می‌توان برای دسترسی به متغیرهای سراسری برنامه، خواندن و یا تغییر مقدار آن‌ها درون توابع استفاده کرد. مثال زیر نحوه استفاده از این آرایه را در عمل نشان می‌دهد:

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

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

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

آموزش استفاده از واژه 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 می‌شوند. اما گاهی اوقات ممکن است بخواهیم یک متغیر محلی `local` پس از اتمام تابع آن، پاک نشود. برای این منظور بایستی از واژه **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 می‌دهیم:

```
<?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 `local` برای تابع فوق محسوب شده و در سایر تابع‌ها قابل استفاده نیست.

آموزش چاپ خروجی در زبان PHP

آموزش چاپ خروجی با دستورات echo و print در:

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 Print و echo برای چاپ خروجی بر روی صفحه استفاده خواهیم کرد. تابع های echo و print تقریباً عملکرد یکسانی داشته و از هر دو برای چاپ و نمایش خروجی بر روی صفحه شود.

می

استفاده

تفاوت اندکی بین توابع Print و echo به صورت زیر وجود دارد:
تابع echo دارای مقدار برگشتی یا return value نیست، در حالی که تابع print عدد ۱ 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:

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 يا Text به وسیله دستور echo 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 : Variabiles در مثال عملی زیر، نحوه نمایش متن (Text) به همراه متغیر e variale 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. مثال نمایش متن : کد زیر نحوه نمایش متن (Text) در خروجی را به وسیله دستور print 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 Types در: PHP

متغیرها در زبان php ، 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 یا string
- متغیر عددی معمولی یا integer
- متغیر عددی اعشاری یا float که به آن double هم می گویند
- متغیر درست یا غلط یا Boolean
- متغیر آرایه ای یا Array

- متغیر شی گونه یا `obj ect`
- متغیر تهی یا `Nul l`
- متغیر ارجاع به آدرس یا `Resource`

آشنایی با متغیر متنی String در PHP

یک متغیر متنی یا `String`، 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 متغیر عددی Integer در PHP

یک متغیر عددی ساده یا `integer`، عددی غیر اعشاری در محدوده ۰-۲۰۱۴۷.۶۴۸ تا ۰.۱۴۷.۶۴۸ 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_dump()`، 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 یا **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 زبان PHP ، جهت استفاده از متغیر شی یا object بایستی آن را کامل و صریح تعریف نمود. برای این منظور ما ابتدا بایستی یک کلاس class برای شی تعریف کنیم. یک کلاس با واژه کلیدی class تعیین شده و ساختاری است که می‌تواند خواص (properties) و متدهای (methods) مربوط به متغیر object را در خود نگهداری کند.

مثال عملی زیر، نحوه تعریف کلاس Car و سپس ایجاد یک متغیر یا object از روی آن کلاس به نام \$herbie را نشان می‌دهد. پس از تعریف متغیر شی، می‌توان 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 داده‌ای null یک متغیر خاصی است که فقط می‌تواند یک مقدار داشته باشد، خالی، تهی یا. یک متغیر از نوع داده‌ای null ، متغیری است که مقداری به آن نسبت داده نشده است.

راهنمایی:

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

همچنین با نسبت دادن مقدار null به یک متغیر، می‌توان 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 ویژه Resource ، 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 ای Resource ، فرآخوانی پایگاه داده یا database یا cal dat می باشد. ما در این درس، از آنجا که متغیر Resource یک مبحث کاملاً تخصصی است، به شرح توضیحات بیشتر پرداخته و آن را به بخش های بعدی موكول می کنیم.

آموزش کار با متن یا String در زبان: PHP

یک String ، 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()` در PHP ، 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 ۱۲ است را بر می گرداند:

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

شمارش تعداد کلمات موجود در یک رشته متن یا String

به وسیله تابع `str_word_count()` در زبان PHP ، 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 زبان PHP ، 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
```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

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

خروجی کد مثال فوق مقدار 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 می‌پردازد. اگر مورد یکسانی با گزینه در حال جستجو، پیدا شود، بُرناهه شماره اولین کاراکتر کلمه را بر می‌گرداند. اگر هم کلمه مورد نظر در رشته‌ی متنی نباشد، مقدار FALSE بازگشت داده می‌شود. به وسیله کد مثال زیر، به دنبال کلمه "World" در متغیر متنی "Hello World!" 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_replace()` در PHP، می‌توانید یک سری از کاراکترها را با کاراکترهای جدیدی در یک رشته متنی String، جایگزین کنید. در کد مثال عملی زیر، کلمه "world" را با کلمه "Dolly" عوض کرده ایم:

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

مرجع کامل کار با متغیر متنی String در PHP

برای دریافت اطلاعات راجع به لیست کامل توابع کار با متن‌ها یا String در PHP، 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 variables هستند با این تفاوت که پس از تعریف و مقداردهی، نمی‌توان 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 Constant با بایستی 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 Global هستند.

آموزش تعریف و کار با مقادیر ثابت یا Constant در PHP

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)
?>
```

• نام مورد نظر جهت Constant را تعیین می کند.

• مقدار مورد نظر برای Constant را مشخص می کند.

• تعیین می کند که آیا نام Constant به حروف بزرگ و کوچک حساس باشد یا خیر. به صورت Case sensitive 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;
?-->
?>
```

آموزش کار با Global ها در PHP عمومی یا Constant

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

- Arithmetic Operators
- Assignment Operators
- Comparison Operators
- Increment/Decrement Operators
- String 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 5 بود، یک مقدار در خروجی چاپ شده و در صورت کوچکتر بودن آن از 5، مقدار دیگری نمایش داده شود.

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

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

مقدار یا عدم اجرای آن ها، استفاده کنید.
در زبان PHP دارای ساختارهای شرطی مختلفی هستیم که در لیست زیر معرفی شده اند:

- ساختار شرطی : if statement : این ساختار شرطی در صورتی که شرط دستور if 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 if یک سری دستورات و در صورت درست نبودن آن، دستورات بخش else 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 شرطی switch ، می توان بر حسب مقادیر مختلف یک متغیر، دستورات خاصی را برای اجرا، تعیین کرد.

آموزش کار با ساختار شرطی: if

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

) f { أشرط }

کدهایی که در صورت درست بودن شرط اجرا می شوند

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 می‌کند. شکل کلی نوشتگان یک ساختار شرطی `if ... else` به صورت زیر است:

```
<?php
    شرط ( )
        کد هایی که در صورت درست بودن شرط اجرا می شوند
    }
    else
    {
        کد هایی 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 night" را در خروجی نشان می‌دهد:

```
<?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 شرطی `if ... else if ... else`، کدهای مختلف برای اجرا در بیش از ۲ حالت مختلف تعیین می‌شود. برنامه از دستور `if` اول شروع به چک کردن شرط‌ها می‌کند، شرط هر `if` یا `else if` درست باشد، دستور آن بخش اجرا شده و مابقی بدون اجرا رها می‌شود. اما اگر شرط هیچ کدام از بخش‌های `if` یا `else` تعیین شده، درست نبود، در نهایت دستور `else` بخش اجرا می‌شود.

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

```
<?php
    شرط ( 1 )
        کد هایی که در صورت درست بودن شرط اجرا می شوند
    }
    شرط ( 2 )
        کد هایی که در صورت درست نبودن شرط 1 و درست بودن شرط 2 اجرا می شوند
    }
```

```

else
{
    کدهایی که در صورت درست هیچ یک از شرط های ۱ و ۲ اجرا 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 night" و در غیر این صورت عبارت "good day" 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 دستوری switch ، 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 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;
}

```

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

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 معمولاً یک متغیر بوده و در ابتدای اجرای ساختار یک بار سنجیده شده و مقدار آن در حافظه قرار می‌گیرد. سپس مقدار عبارت با مقدار هر یک از Case های ساختار مقایسه می‌شود. اگر مقدار عبارت با هر یک از مقادیر Case ها یکسان بود، دستورات آن Case اجرا می‌شود. از دستور `break` در پایان هر Case، برای جلوگیری از ادامه اجرای سایر Case های ساختار استفاده نکنیم، دستورات Case های بعد از Case ای که مقدار آن با متغیر ساختار برابر بوده است، نیز اجرا خواهد شد که معمولاً نتیجه مورد نظر ما نیست.

در صورتی که مقدار عبارت `n` با مقدار هیچ یک از Case های ساختار `switch` برابر نباشد، دستورات بخش `default` اجرا 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 می‌شود. 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 در PHP

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

صورت پشت سر هم تکرار می می 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 زبان PHP ، ۴ نوع حلقه تکرار داریم که عبارتند از:

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

- 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 ، مجموعه دستورات بدن خود را به تعداد تعیین شده، تکرار می کند.

- حلقه : foreach ، 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 نوشتمن 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:

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```
<?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 نوشتمن 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 می 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 داشته باشید که شرط حلقه do...while در انتهای حلقه و پس از اجرای کامل دستورات آن چک می شود. بنابراین حتی اگر شرط حلقه do...while از ابتدا هم نادرست باشد، دستورات آن حداقل یک بار اجرا خواهند شد.

مثال

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

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

خودداری می‌شود:

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

حلقه‌های 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 تابع‌ها یا function می‌باشد. زبان PHP دارای بیش از ۱۰۰۰ تابع پیش ساخته برای انجام امور مختلف برنامه نویسی است.

در کنار تابع‌های از پیش تعریف شده زبان PHP، کاربر می‌تواند تابع‌های دلخواه خود را نیز تعریف کند. یک تابع، مجموعه یا بلوکی از دستورات است که می‌توان 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 می‌کنید.

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 تابع‌های دلخواه کاربر در PHP

با استفاده از واژه کلیدی function و با ساختار زیر می‌توانید 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 بعداً بتوانید آن را راحت تر در کد تشخیص دهید.

نکته:

نام تابع ها در زبان PHP ، حساس به حروف بزرگ یا کوچک (Case-sensitive) 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() در خروجی چاپ کرده و برای اجرای آن کافی است تا با نوشتن نام عبارت "Hello world" 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 آرگومان را بایستی در پرانتز جلوی نام تابع تعریف کنید. هر تعداد آرگومان یا پارامتر که نیاز داشته باشد، 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 \$name می باشد. هنگامی که تابع familyName() فراخوانی یا Call می شود، به وسیله آرگومان یا پارامتر \$name 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 \$fname و \$year تعریف شده است. تابع 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 setHeight() را بدون آرگومان فراخوانی کنید (مثل فراخوانی دوم تابع در مثال)، پارامتر مقدار پیش فرض

بای defaul 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 در PHP ، می تواند پس از اجرای کامل کدهای خود مقداری را به عنوان مقدار بازگشتی یا **Return Value** به برنامه بازگرداند.

برای این منظور بایستی از دستور `return` در انتهای کد تابع استفاده کرد. همانند کد مثال عملی زیر که تابع

`sum()` در انتهای 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 یا `Array` ، نوع خاصی از متغیرهاست که می تواند در یک زمان واحد 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 ماشین‌ها حرکت کرده و نام یک خودرو خاص را پیدا کنید، بدتر این که به جای ۳ ماشین، ۳۰۵ ماشین در لیست خود داشته باشید؟!

راه حل، ایجاد یک آرایه یا **Array** است. یک آرایه می‌تواند تعداد زیادی مقدار یا **value** را با یک نام مشابه در خود ذخیره کند. سپس می‌توانید 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 می‌شود:

```
array();
```

در زبان PHP، ۳ نوع آرایه **Array** داریم:

- آرایه اندیس دار : **Indexed Arrays** آرایه‌ای که در آن هر عضو یا یک اندیس **i** مشخص می‌شود.
- آرایه رابطه‌ای یا : **Associative Arrays** در این نوع آرایه هر کلید دارای یک نام است.
- آرایه‌های چند بعدی : **Multidimensional Arrays**) در این نوع آرایه دو یا چند آرایه به صورت تو در تو و چند بعدی تعریف شده است.

آموزش کار با آرایه‌های اندیس وار : **Indexed Arrays** دو راه برای تعریف آرایه‌های اندیس وار یا **Indexed Arrays** وجود دارد:

اندیس یا **i** را می‌توان به صورت خودکار به هر عضو آرایه نسبت داد. در این حالت اندیس آرایه از صفر شروع می‌شود، همانند کد زیر:

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

یا می‌توانید اندیس یا **i** 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:

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

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:

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

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

با

```
<?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

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

آموزش مرتب سازی آرایه ها (Sorting Array) در PHP

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 `value` آن ها مرتب می کند.
- تابع `ksort()` ، آرایه های رابطه ای یا associative arrays را در جهت صعودی بر حسب مقدار کلید `key` یا 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 `key` یا 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 **descending** 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` می مرتب 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 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

تابع `assort()`، 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 آن 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 variables` دسترسی 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 Superglobal variable است که برای دسترسی و مدیریت متغیرهای سراسری global variable در هر کجای کد برنامه PHP حتی درون تابع ها یا متدها استفاده می شود. زبان PHP ، کلیه متغیرهای سراسری global variable 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 \$GLOBALS را در کد superglobal variable \$GLOBALS نشان 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 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 المنشتهای 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 المتّهای متغیر \$SERVER را که می‌توانید از آن‌ها استفاده کنید، به همراه توضیح نشان داده است:

- \$ SERVER [PHP-SELF] این المتّ نام فایلی که هم اکنون اسکریپت آن در حال اجراست را بر می‌گرداند.
- \$_SERVER['GATEWAY_INTERFACE'] این المتّ ورژن CGI که هم اکنون سرور در حال استفاده از آن است را برمی‌گرداند.
- \$_SERVER['SERVER_ADDR'] این المتّ آدرس هاست سرور جاری را بر می‌گرداند.
- \$_SERVER['SERVER_NAME'] این المتّ نام هاست سرور جاری برای مثال (www.tahlildadeh.com) را بر می‌گرداند.
- \$_SERVER['SERVER_SOFTWARE'] این المتّ مقدار متنی کد شناسایی سرور یا server را مثل Apache/2.2.24 identification string بر می‌گرداند.
- \$_SERVER['SERVER_PROTOCOL'] این المتّ ورژن و نام پروتکل اطلاعات سرور یا HTTP/1.1 information protocol را بر می‌گرداند، مثل.
- \$_SERVER['REQUEST_METHOD'] این المتّ روش درخواست صفحه جاری یا Request را بر می‌گرداند مثل mode Post.
- \$_SERVER['REQUEST_TIME'] این المتّ شروع درخواست اجرای صفحه را بر می‌گرداند، برای مثال 1377687496.

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

- \$_SERVER['PATH_TRANSLATED'] این المنش مقدار آدرس مبتنی بر فایل سیستم file مربوط به اسکریپت جاری را برمی‌گرداند.
- \$_SERVER['SCRIPT_NAME'] این المنش آدرس یا path اسکریپت جاری را برمی‌گرداند.
- \$_SERVER['SCRIPT_URI'] این المنش مقدار URI مربوط به صفحه جاری را برمی‌گرداند.
- SERVER ADM N directive در فایل تنظیمات برنامه یا Configuration file اگر هاست شما به صورت مجازی Virtual host اجرا می‌شود، 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 در PHP ، برای جمع آوری اطلاعات پس از Submit یک فرم HTML استفاده می‌شود.

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 می‌کند، این data به آدرس فایلی که در خاصیت action تگ تعیین شده است، ارسال می‌شود. در این مثال، ما اطلاعات فرم را برای پردازش به خود صفحه جاری ارسال کرده ایم. اگر شما می‌خواهید این اطلاعات به صفحه یا آدرس دیگری برود، بایستی آدرس مورد نظر خود را به جای فایل جاری در خاصیت action بنویسید.

سپس می‌توانیم از متغیر سراسری ویژه \$_REQUEST برای جمع آوری اطلاعات یا value کادر متنی input استفاده 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 آن به روش "POST" استفاده می شود. همچنین از متغیر \$_POST می توان برای ارسال متغیرها یا vari ables نیز استفاده کرد.

مثال:

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

آدرس آموزشگاه: تهران - خیابان شریعتی - بالاتر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

هنگامی که کاربر بر روی لینک "Test \$GET" کلیک می‌کند، پارامترهای "subject" و "web" به صفحه "test_get.php" ارسال می‌شوند. سپس در صفحه "test_get.php" می‌توانید به این مقادیر یا values از طریق متغیر \$_GET دسترسی داشته باشید.

کد مثال زیر، اسکریپت مربوط به صفحه "test_get.php" را نشان می‌دهد:

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

درباره متغیرهای \$_POST و \$_GET در بخش آموزش فرم‌های PHP بیشتر صحبت خواهیم کرد.

بخش دوم: آموزش مدیریت فرم (Form) در PHP

آموزش مدیریت فرم‌ها در PHP

متغیرهای سراسری ویژه \$_GET و \$_POST برای جمع آوری اطلاعات فرم‌ها در زبان PHP کار به می‌روند.

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

```
<?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 می‌کند با اطلاعات فرم به فایل "welcome.php" که در خاصیت action فرم تگ Form Top of Form

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

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

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

الته کد فایل "welcome.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 validation 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 validation 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 اصول امنیتی را در هنگام پردازش فرم های PHP آموخته خواهیم داد. اعتبارسنجی یا validation صحیح اطلاعات برای محافظت سایت شما از رخنه و نفوذ هکرهای اسپرها 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

`)key=>val ue, key2=>val ue2, key3=>val ue3` (array) ایجاد می کنند. این آرایه جفت های مقدار/نام یا key/value را نگهداری می کند که در هر دو، کلیدها یا keys کنام کنترل های موجود در فرم و مقادیر یا values 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 superglobal variables 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 متodi بدون نیاز به انجام کار خاصی و بدون توجه به scope یا میدان دید در کل کد صفحه دسترسی داشته باشید.

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

چه زمانی از روش GET استفاده کنیم؟

اطلاعاتی که به وسیله متدهای GET ارسال می شوند، برای همگان قابل رویت هستند) زیرا نام متغیرها و مقدار آنها یا `val` 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 متدهای ۲۰۰۰ کاراکتر است. روش GET در برخی مواقع کاربرد مناسب دارد، مثل زمانی که بخواهید کاربر به راحتی بتواند صفحه را `Bookmark` کند، زیرا می توانید مقادیر مورد نظر خود را همراه با URL ارسال کنید. همچنین این روش برای ارسال اطلاعات غیرحساس 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 جفت‌های نام/مقدار به بدن 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 متدهای POST بر خلاف متدهای GET وجود ندارد. از طرف دیگر، متدهای POST از قابلیت‌های پیشرفت‌های مثل امکان ارسال چند‌تیکه اطلاعات یا `multipart` در هنگام آپلود فایل‌ها به سرور پشتیبانی می کند. ولی به دلیل این که مقدار متغیرهای اطلاعات در آدرس یا URL صفحه قرار نمی‌گیرند، روش POST برای صفحات مناسب نیست.

راهنمایی:

اکثر برنامه‌نویسان از روش POST برای ارسال اطلاعات فرم‌ها استفاده می‌کنند. در درس بعدی، به آموزش روش اعتبارسنجی و ارسال امن اطلاعات فرم‌های وب در PHP خواهیم پرداخت.

آموزش اعتبارسنجی فرم‌ها در زبان PHP

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

راهنمایی:

در هنگام پردازش اطلاعات فرم‌های وب به بحث امنیت اطلاعات یا security بسیار جدی فکر کنید. در این درس با تفکر کنترل و اعتبارسنجی اطلاعات ورودی توسط کاربر در فرم‌های وب کار خواهیم کرد. اعتبارسنجی درست اطلاعات وارد شده در فرم‌ها، بسیار مهم بوده و شما را از حمله هکرهای اسپم‌ها در امان نگه می‌دارد.

فرم HTML ای که در این درس‌ها از آن استفاده خواهیم کرد، به صورت زیر بوده و شامل چندین فیلد ورود اطلاعات مختلف می‌باشد. 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 Required 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.
- **فیلد E-mail :** این کادر اجباری بوده و بایستی شامل یک آدرس معتبر ایمیل با فرمت صحیح (با کاراکترهای @ و .) باشد.
- **فیلد website :** این کادر از نوع اختیاری (optional) 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 (Multi-line input) یا text area می باشد.
- **فیلد Gender :** این فیلد از نوع دکمه های رادیویی یا Radio Button بوده و کاربر بایستی حتما یکی از دو مورد را انتخاب کند، به عبارت دیگر اجباری است.

ابتدا بباید نگاهی به سورس کد خام HTML فرم بیندازیم.

بررسی کادرهای متن فرم یا Text Fields

کادر متن های name، website و email از نوع فیلد متن معمولی یا text input بوده و کادر متن به صورت چند خطی یا textarea از HTML 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">
```

آدرس آموزشگاه: تهران - خیابان شریعتی - بالاتر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```
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 المنش 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 (به سرور ارسال می شوند.)

سوال:

متغیر سراسری ویژه `$_SERVER["PHP_SELF"]` یک متغیر `variable superglobal` است که نام فایل گرداند. اجراین، متغیر `$_SERVER["PHP_SELF"]` اطلاعات فرم را به خود صفحه جاری ارسال می کند تا این که به صفحه دیگری پرسش 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 `function htmlspecialchars()` کارکترهای ویژه را به موجودیت های HTML تبدیل می کند. این کار بدین معناست که کاراکترهایی مثل < و > به صورت `$h;` 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 `?????` خوبی 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 اسکریپت های کلاینت 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_form.php" 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 شد:

```
<?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:

```
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 های PHP

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 خالی کند. به این نوع کادرهای متن در اصطلاح RequiredFields گفته و در این درس با نحوه کار آن ها آشنا خواهید شد.

مثال عملی کادر متن اجباری در PHP

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

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

در درس اول کلیه کادرهای متن فرم اختیاری یا بودند optional

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 آبا 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 و اگر هم خالی نباشد اطلاعات فرم را با کمک تابع `test_input` به مقصد

ارسال می‌کند:

```
<!--?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 می‌کنیم 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:

```
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 اعتبارسنجی 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 با خبر.

یا در مثالی دیگر آیا کاربر یک آدرس ایمیل با فرمت صحیح در کادر متن "Email" وارد کرده یا خیر) آدرسی که دارای کاراکترهای a و o 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

در این درس قصد داریم تا نحوه اعتبارسنجی مقادیر وارد شده توسط کاربر در کنترل های متن Email ، Name و آدرس 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";
}

```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

}

?>

نکته:

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

آموزش اعتبارسنجی مقدار Email در فرم PHP

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

در کد مثال زیر، اگر فرمت ایمیل وارد شده توسط کاربر درست نباشد، یک پیام خطای متغیری خطای مربوط به 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) الگوی Regulare expression به کار رفته 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 خطای متغیری خطای مربوطه ذخیره خواهد شد:

```
<?php
$website = test_input($_POST["website"]);
if (!preg_match("/\b(?:https?|ftp):\/\/|www\.)[-a-z0-9+\&\#46\%?=~_|!:,.]*[-a-z0-9+\&\#46\%?=~_|]/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?:\/\/|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 خواهیم پرداخت.

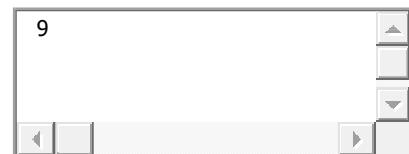
آموزش کامل مثال عملی کار با فرم ها در PHP

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 `value` کادرهای متن `name` و `website` اضافه می کنیم.

در فیلد توضیح یا `Comment` که به صورت کادر متن چندخطی `text area` 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 `$comment`, `$email`, `$name` و `$website` را نمایش خواهند داد.

همچنین برای نمایش این کدام گزینه از دکمه های رادیویی `Radio Button` مربوط به جنسیت کاربر یا انتخاب شده است، بایستی با خاصیت `checked` این کنترل کار کنیم دقیق کنید که در این نوع کنترل مقدار 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 *

بخش سوم : آموزش دستورات پیشرفته PHP

آموزش کار با تاریخ (Date) و ساعت (Time) در PHP

از تابع date() در زبان PHP برای قالب بندی و کار با تاریخ (Date) و یا ساعت (Time) استفاده می شود.

درواقع تابع date() در PHP ، 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)
?>

```

- پارامتر `format` قالب بندی کلی برچسب زمانی یا `timestamp` را تعیین کرده و اجباری (Required) است.

- این 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:

- کاراکتر `d`: این کاراکتر بیانگر تاریخ روز مورد نظر در ماه جاری (عددی بین ۱ تا ۳۱ است).
- کاراکتر `m`: این 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 چهار رقمی 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 یک علامت کپی رایت و عدد سال جاری نمایش در کنار برنده 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 می‌توانید استفاده کنید

پرداخته ایم:

- کاراکتر h از این کاراکتر برای نمایش عدد ساعت جاری (عددی بین ۱۰ تا ۱۲) استفاده می‌شود.
- کاراکتر m از این کاراکتر برای نمایش عدد دقیقه یا minute ساعت جاری (بین ۰۰ تا ۵۹) استفاده می‌شود.
- کاراکتر s از این کاراکتر برای نمایش عدد ثانیه یا seconds ساعت جاری (بین ۰۰ تا ۵۹) استفاده می‌شود.
- کاراکتر a این کاراکتر قبل از ظهر بودن ساعت (am) و بعد از ظهر بودن آن (pm) را در فرمت ۱۲ ساعتی تعیین می‌کند.

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:s");
?-->
?>
```

نکته مهم:

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

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

اگر مقدار ساعت یا تاریخی که از سرور دریافت می‌کنید 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 می‌خواهید زمان (time) صحیح را بر حسب محل خودتان به کاربر نمایش دهید بایستی از تابع date() در timezone استفاده کنید.

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

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

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

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

تعیین یک تاریخ با استفاده از تابع mkttime در PHP

همانطور که در بخش ابتدای این درس بیان کردیم پارامتر دلخواه `stamp time` در تابع `date()` یک برچسب زمانی خاص را تعیین می‌کند. اگر مقدار خاصی برای پارامتر `timestamp` تعیین نشود سیستم تاریخ و زمان جاری سیستم را مورد استفاده قرار می‌دهد (همانطور که در کد مثال عملی زیر نشان داده شده است) تابع `mkttime` برچسب زمانی `Unix` را برای تاریخ بر می‌گرداند. برچسب زمانی `Unix` تعداد ثانیه‌های سپری شده از دوره زمانی `Unix` یا تاریخ (`January/1970 00:00:00 GMT`) و تاریخ تعیین شده را بر می‌گرداند. همانند کد مثال زیر:

ساختار کلی تابع `mkttime` به صورت زیر است:

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

مثال عملی، کد مثال عملی زیر، یک زمان و تاریخ را بر حسب پارامترهای تعیین شده برای تابع `mkttime` ایجاد می‌کند:

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

آموزش ایجاد یک تاریخ (Date) بر حسب یک string با استفاده از تابع PHP strtotime()

از تابع `strtotime` در PHP برای تبدیل یک رشته متنی قابل خواندن برای انسان یا `string` یک زمان Unix استفاده می‌شود.

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:s", $d);
```

?-->
?>

زبان PHP در تبدیل رشته یا string به زمان بسیار باهوش است بنابراین می‌توانید 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:s", $d) . "<br-->";

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

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

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

مثال های عملی بیشتر کار با تابع date() در PHP

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

```
<?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 می‌دهد:

```
<?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 در PHP

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) در PHP

از تابع `date()` در زبان PHP برای قالب بندی و کار با تاریخ (Date) و یا ساعت (Time) استفاده می شود. در واقع تابع `date()` در PHP، 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)  
?>
```

- پارامتر `format` قالب بندی کلی برچسب زمانی یا `timestamp` را تعیین کرده و اجباری است (Required).
- این 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 کarakترهای 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 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:

- کاراکتر `d` این کاراکتر بیانگر تاریخ روز مورد نظر در ماه جاری (عددی بین ۰۱ تا ۳۱ است).
- کاراکتر `m` این 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 چهار رقمی 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:

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک ۶۵۱ طبقه دوم - واحد ۷

```
<!--?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 معمولاً 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 ۱۰ تا ۱۲) 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 می‌دهد نه کامپیوتر کلاینت کاربر.

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

اگر مقدار ساعت یا تاریخی که از سرور دریافت می‌کنید 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 (مثلاً سرور در آلمان یا آمریکاست)

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

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

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

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

تعیین یک تاریخ با استفاده از تابع `mktime` در PHP

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

ساختار کلی تابع `mktime` به صورت زیر است:

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

مثال عملی، کد مثل عملی زیر، یک زمان و تاریخ را برحسب پارامترهای تعیین شده برای تابع `mktme` ایجاد می‌کند:

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

آموزش ایجاد یک تاریخ (Date) برحسب یک string با استفاده از تابع PHP در `strtotime()`

از تابع `strtotime` در PHP برای تبدیل یک رشته متنی قابل خواندن برای انسان یا `string` ه یک زمان Unix استفاده می‌شود.

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 برحسب `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:s", $d);
?-->
?>
```

زبان PHP در تبدیل رشته یا `string` به زمان بسیار باهوش است بنابراین می‌توانید 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:s", $d) . "<br-->";

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

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

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

مثال های عملی بیشتر کار با تابع date() در PHP

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);
}
?->
```

مثال ۲: 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 Date در PHP

برای دریافت اطلاعات کامل برای کار با تاریخ Date در PHP به مرجع آموزش زمان در بخش آموزش 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 require و include در PHP

دستور require و include 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 require و include برای تکرار یک دستور یا محتوی تکراری در درون چند صفحه استفاده می شود و عملکردی مانند قابلیت متذیل در Asp.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 سایت یا تکرار کنید. این امکان وجود دارد که با استفاده از دستورات require یا include 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 `require` و `include` کاملاً شبیه هم هستند و تنها تفاوت آن‌ها واکنش در هنگام بروز

خطا است که به صورت زیر تقسیم می‌شوند:

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

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

بنابراین اگر می‌خواهید 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 `require` 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 نمی‌تواند باحذف عمدى یا اتفاقی یک فایل کلیدی یا بخشی از کد، صفحات را اجرا کند. برای درک بهتر یک مثال می‌زنیم. فرض کنید بخش چک کردن هویت کاربر از طریق یک دستور `require` به تمامی صفحات کپی می‌شود. اگر از دستور `include` استفاده شود کاربر می‌تواند 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 پردازه اجرا نخواهد شد و امنیت آن به خطر نمی‌افتد.

استفاده از دستورات `require` و `include` حجم زیادی از کار را کاهش می‌دهد به این معنا که می‌توانید 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 `require` یا `include` به صورت زیر است:

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

مثال های عملی کار با دستور include در PHP

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 باقیستی از این منو استفاده کنند. در کد مثال زیر نحوه قرار دادن این منو در تمامی صفحات فایل با استفاده از تگ را نشان داده ایم. بعده می توانید با CSS استایل و ظاهر منو را نیز تغییر 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" را در یک فایل دیگر include کنیم می‌توان 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

همانطور که در ابتدای درس نیز اشاره کردیم، از دستور require نیز می‌توان برای قرار دادن کدهای یک فایل در دیگر فایل 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 تواند 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 require از دستور include 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 می‌شود. 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 صفحات PHP

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 مانند ایجاد، 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 خطاهای 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 در PHP

تابع readfile 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 در PHP

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()` در PHP

راه بهتر برای باز کردن فایل ها در PHP استفاده از تابع `fopen()` به جای تابع `read file()` است زیرا تابع `fopen()` امکانات 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` است. تابع `fopen()` اگر تواند 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:
- ۲دراین 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 محتويات آن پاک شده و محتويات جدید در آن نوشته می شود.اگر هم فایل وجود نداشته باشد یک فایل جدید ايجاد می شود.در اين متدهم اشاره گر از ابتدای فایل شروع می كند.
 - aدر اين حالت نيز فایل برای نوشتن (`write`) باز شده با اين تفاوت که محتويات قبلی فایل حفظ شده و محتويات جدید به انتهای آن اضافه می شود.در الواقع اشاره گر از انتهای فایل شروع کرده و در صورت موجودنبوذن فایل ، یک فایل جدید اضافه خواهد شد.
 - Xاین حالت یک فایل جدید را جهت نوشتن (`write`) باز می کند.در صورتی که فایل مورد نظر از قبل وجود داشته `error` داده و مقدار `FALSE` را بر می گرداند.
 - +در اين حالت برنامه فایل را جهت خواندن و نوشتن (`read/write`) باز می کند. اشاره گر در ابتدای فایل قرار خواهد گرفت.
 - W+در اين حالت برنامه فایل را برای خواندن و نوشتن (`read/wife`) باز کرده و در صورتی که فایل موجود باشد محتويات آن را پاک نموده و در صورت عدم وجود فایل ، یک نمونه جدید ايجاد می کند. در اين حالت نيز اشاره گر از ابتدای فایل شروع می کند.

a : در این حالت نيز فایل برای خواندن و نوشتن (`read/write`) باز شده و محتويات قبلی آن حفظ شده و محتويات 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 گردداند.

آموزش کار با فایل `fread()` در PHP

تابع `fread()` محتويات یک فایل باز را می خواند.

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

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

```
<?php
```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

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

آموزش کار با تابع fclose() در PHP

تابع () 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 () 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 EOF یا end-of-life رسیده یا خیر. تابع `feof()` معمولا برای جستجو در دیتایی که طول آن را نمی مناسب دانیم در کد مثال عملی زیر، محتويات 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 با() باfgetc()

از تابع `fgetc()` در PHP برای خواندن یک کاراکتر تنها در فایل استفاده می شود. در کد مثال عملی زیر، محتويات فایل " web dictionary.text" را به صورت 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 (create) و نوشتن (write) در فایل های PHP

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 در PHP - تابع fopen()

از تابع `fopen()` در زبان PHP برای برای ایجاد یک فایل جدید در سرور استفاده می شود. ممکن است کمی کمی 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، PHP به صورت خودکار آن فایل را ایجاد کرده و آن را برای عمل نوشت (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 Permission در فایل های PHP

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

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

از تابع `fwrite()` در PHP برای نوشتند (write) در یک فایل استفاده می شود. تابع `fwrite()` دارای دو پارامتر بوده که پارامتر اول نام فایل مورد نظر جهت نوشتند و پارامتر دوم مقدار متن (string) را تعیین می کند.

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

```
<!--?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" نوشتندیم. در هر بار نوشتند درون این فایل، متغیر متنی `$txt` را به تابع ارسال کردیم که در مرحله اول شامل نام "John Doe" و در مرحله دوم شامل نام "Jane Doe" می باشد. پس از اتمام عملیات نوشتند در فایل ، با استفاده از تابع `fclose()` آن

را بسته ایم.

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

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

عملیات PHP overwriting

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 اجرا کد فوق، فایل "newfile.txt" را باز کرده، خواهید دید که محتويات 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 یا حجیم بسیار دقت کنید.

"php.ini" مرحله اول تنظیم فایل

در مرحله اول بایستی مطمئن باشید که سرور PHP شما جهت امكان 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 آپلود فایل

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:

- حتماً 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 محتويات 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 به آمد و یک دکمه Browse جهت پیدا کردن فایل مورد نظر را در کنار کنترل input نشان دهد.

فرم Html کد مثل 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 انتهای توضیح نکات مهم آن پرداخته ایم:

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```

<!--?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 `$uploadOK = 1;` در حال حاضر استفاده نمی شود و در مراحل بعدی به تشریح آن خواهیم پرداخت.
- خاصیت `$imageFileType` تعیین 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" قرار دارد ایجاد کنید. فایل های آپلود 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 `HTML` فوق به نام "file To Upload" می باشد. در بخش دوم بایستی حجم سایز جهت آپلود را چک کنیم. اگر حجم فایل بیشتر از ۵۰۰ کیلوبایت 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 پسوندهای `JPG`, `JPEG`, `PNG` و `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=0` عملیات 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"]);
```

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```

$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 != "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 زبان PHP

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

آموزش ایجاد کوکی Cookie در زبان PHP

کوکی در زبان PHP به وسیله تابع setcookie() ایجاد می‌شود. 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 می‌شود. مثال عملی : کد زیر یک کوکی به نام "user" با مقدار "John Doe" را ایجاد می‌کند. این کوکی بعد از 30 روز (30*86400) منقضی می‌شود. به کار بردن علامت (/) به این معناست که این کوکی در کل سطح برنامه یا سایت قابل دسترس خواهد بود (یا می‌توانید یک پوشه یا مسیر دیگری را تعیین کرده که کوکی فقط در آن بخش قابل دسترس باشد).

دسترس

قابل

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>
?>

```

نکته: 1

تابع () را بایستی قبل از تگ تعریف کنید.

نکته: 2

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 رخدادن 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 حذف یک کوکی، بایستی با استفاده از تابع `setcookie()` مجدداً آن را تنظیم کرده و لی 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 سایت‌های تحت وب این قابلیت را دارند که خواندن یا ایجاد کوکی‌ها را در آن‌ها غیر فعال کرد. بنابراین در برنامه‌هایی 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 می‌کند که آیا کوکی‌ها 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` می‌فهمیم 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 PHP در HTTP ، به بخش مرجع آموزش PHP HTTP در سایت تحلیل داده بروید.

آموزش کار با Session در زبان PHP

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

برخلاف کوکی یا Cookies ، 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 کند.

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 دائمی 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 می‌شود. متغیرهای `$_SESSION` درون متغیر سراسری `$_SESSION` نگهداری می‌شوند. 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 می‌کنیم، 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 نمی‌شوند. بلکه آن‌ها به یک باره توسط تابع `session_start()` ای که در آغاز هر صفحه قرار داده ایم، خوانده می‌شوند. همچنین توجه داشته باشید که تمامی متغیرهای Session درون متغیر سراسری `$_SESSION` نگهداری می‌شوند.

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

```
<?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:

Session چطوری کار می کند و از کجا می داند من کی هستم؟

بسیاری از Session 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-key اسکن 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

برای تغییر مقدار متغیر یک Session ، 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 session_destroy() و session_unset() 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 یا 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>';
```

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```

    }
?>
</table>

```

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

- اطلاعات ورودی توسط کاربر در فرم ها.
- Cookies کوکی ها یا.
- 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 ساختفاده از PHP Filter ها می توانید مطمین شوید که نرم افزار تحت وب شما همواره اطلاعات درست دریافت 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
```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```

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

آموزش چک کردن یک متغیر integer در PHP

در کد مثال عملی زیر، تابع `filter_var()` چک می کند که آیا متغیر `$int` یک متغیر عددی یا خیر. اگر متغیر `$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 آیا متغیر `$int` از نوع `Integer` هست یا خیر.

سپس بررسی کردیم آیا این عدد بین ۱ تا ۱۰۰ 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 دارد

اطلاعات اضافی ارسالی پس از آدرس صفحه هست یا خیر: `QueryString`

```
<?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 ۱۲۷ از متغیر `str` حذف می کند:

```
<?php
$str = "<h1>Hello WorldÃ¢Â!</h1>";
```

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک ۶۵۱ طبقه دوم - واحد ۷

```
$newstr = filter_var($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 زبان PHP ، به بخش آموزش کامل PHP Filter در سایت تحلیل داده بروید.

مرجع کامل آموزش PHP 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 خطا 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 تواند error های احتمالی را چک کند، برنامه تان بسیار غیر حرفلای به نظر رسیده و می تواند شما را با چالش های امنیتی رو به رو کند.

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

- دستورهای ساده با تابع "die()".
- error های خاص و بررسی تحریک کننده خطا (error triggers).
- آموزش نحوه گزارش کردن خطاهای خطا (error reporting).
- آموزش روش پایه مدیریت خطا-PHP Tابع 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، احتمالاً با خطاهایی هماند متن زیر مواجه خواهد 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 خطای 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:

1

File not

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 زبان PHP، کار نسبتاً 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 (error) و پیام خطا (error message) را مدیریت کند و تا پارامتر اختیاری را دریافت کند (پارامترهای اختیاری level) مثل نام فایل، شماره خط و محتويات خط.

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_context	error_line	error_file	error_message	error_level
پارامتر اختیاری. این پارامتر یک آرایه است که نام و مقدار کلیه متغیرهای برنامه را در هنگام بروز خطا ثبت و نگهداری می کند.	پارامتر اختیاری. این پارامتر شماره و آدرس خط	پارامتر نام تابعی که خطا رخ داده است را داده را تعیین می کند.	کدی که خطا در آن رخ داده شده است. این نوع اخطارها مثلاً هستند که توسعه کد کاربر رخ می دهد. این نوع اخطارها مثل یک مشکل توسعه کاربر و با استفاده از تابع trigger_error شده اند.	پارامتر اجباری بوده و درجه گزارش خطا را تعیین می کند. پارامتر پیام خطا مربوط به error رخ داده را مشخص می کند. تعیین 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_ALL	E_RECOVERABLE_ERROR	E_USER_NOTICE	E_USER_WARNING	E_USER_ERROR	E_NOTICE	E_WARNING	
این درجه خطا شامل خطاها می خواهد. این قابل ردیابی است. این نوع خطا شامل	این درجه شامل اعلان های غیر مخفوب می شود که همانند یک E_ERROR است	این درجه شامل اخطارهای غیر مخفوب می شود که همانند یک (Set_error_handler) در سطح برنامه می شود.	این درجه شامل اخطارهای غیر مخفوب می شود که همانند یک E_WARNING است	این درجه شامل اخطارهای غیر مخفوب می شود که همانند یک E_ERROR است	این درجه شامل اخطارهای غیر مخفوب می شود که همانند یک E_WARNING است	این درجه شامل اخطارهای غیر مخفوب می شود که همانند یک E_ERROR است	درجه خطاها غیر مخفوب در زمان اجرا است که باعث توقف اجرای اسکریپت نمی شوند.

اکنون باید یک تابعی ایجاد کنیم تا خطاها 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()` اجام 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:

1
2

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

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_NOTICE	E_USER_WARNING	E_USER_ERROR
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.	این نوع خطا شامل اخطارهایی می شود که توسط کد کاربر تولید شده اند. در این نوع خطاهای دیگر نمی توان برنامه را از نقصه بروز خطا باز گرداند و برداش اسکریپت متوقف خواهد شد.	این دسته شامل خطاهای مخربی می شوند که توسط کد کاربر تولید شده اند. در این نوع خطاهای دیگر نمی توان برنامه را از نقصه بروز خطا باز گرداند و برداش اسکریپت متوقف خواهد شد.

در این حالت اسکریپت ممکن است به موردي برخورد کرده باشد که از نظر آن خطا بوده، ولی امكان 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:

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

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

آموزش ثبت خطای (error logging) در PHP

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

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

آموزش نحوه ارسال یک پیام خطای (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
```

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```

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



2

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

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

1



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 زبان PHP

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

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 جدید PHP5 ، یک راه جدید شی گرا (object oriented) برای مدیریت خطاهای errors شد.

برنامه معرفی

برنامه

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

در لیست زیر، مراحل عادی که در هنگام فعال شدن یک Exception رخدانی 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 یا **handler** ، 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 باستثنی 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 **try...catch** دستوری خواهد بود. اگر **Exception** رخ داده در برنامه را تتوان ردگیری و دریافت کرد، یک خطای آسیب زننده (**fatal error**) به همراه یک پیام با مضمون "خطای خواهد نشده" یافت.

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 Try ، throw و Catch

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 خطای 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 exception فعال شود، در اصطلاح می گوییم که آن Exception صادر یا "thrown" شده است.
- بخش : Throw بخش مشخص کننده نحوه فعال سازی یک exception است. هر "throw" بخشی حداقل دارای یک بخش "catch" نیز باشد .
- بخش : Catch بلاک "Catch" استثناء یا exception رخداده را دریافت نموده و یکی شی (object) حاوی اطلاعات مربوط به exception را ایجاد می کند .

حال بباییید exception مثال قبل با یک ساختار کامل و درست مجدداً 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 exception را فعال کرده و آن را دریافت می کند. مراحل زیر در طی انجام کار رخ می دهند :

- تابع checkNum() 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، خطای exception روی خواهد داد.
- تابع CheckNum() درون ساختار "try" فراخوانی می شود.
- کد به همراه تابع CheckNum() صادر می شود.
- پیام خطای حاصل از فعال شدن exception ، به وسیله فراخوانی کد getMessage() از شی exception مربوط به ، 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 :

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

دلخواهی که ایجاد کرده اید، خواص (properties) خود را از کلاس اصلی Exception class

به 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 از کلاس exception class قدیمی است که تابع errorMessage() از کلاس اصلی و قدیمی exception class کپی شده است. از آنجایی که کلاس جدید یک کپی از کلاس اصلی و قدیمی دارد، خواص و متدهای آن را به ارث برده است، می توانید در کلاس جدید از متدهای کلاس قدیمی مثل تابع های errorMessage(), getMessage(), getFile() و getLine() استفاده کنید.

توضیح مثال : کد مثال عملی این بخش، یک exception را فعال کرده و سپس با استفاده از یک کلاس exception آن 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، exception تعیین شده روی می دهد.

5. بلوک دستوری "catch" ، خطای exception رخ داده را دریافت کرده و پیام خطای نمایش می دهد.

آموزش کار با exception چندگانه :

این امکان وجود دارد که در یک اسکریپت، از چندین exception که به آن exception می گوید، استفاده کرد که مختلف شرایط برای چک کردن می کنید.

همچنین می توانید از چندین دستور exception های چندگانه استفاده کنید. این خطای exception class می تواند از کلاس مختلف 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 آن‌ها درست نباشند، exception تعیین شده صادر می‌شود.

۱. کلاس () customException به عنوان یک نسخه جدید از کلاس قدیمی exception ایجاد می‌شود.

در این حالت، کلاس جدید تمامی خواص و متدهای کلاس قدیمی را به ارث می‌برد.

۲. تابع () errorMessage ایجاد شده و این تابع در صورتی که ایمیل وارد شده نادرست باشد، یک پیام خط را بر می‌گرداند.

۳. متغیر \$email با یک مقدار متنی (string) که حاوی یک ایمیل درست می‌باشد، پر شده ولی شامل کلمه "example" خواهد بود.

۴. بلوک دستوری "try" اجرا شده و به دلیل درست بودن شرط اول کد، خط را exception روی نمی‌دهد.

۵. شرط دوم کد به دلیل وجود کلمه "example" در متغیر ایمیل (\$email) یک خط را exception صادر می‌کند.

۶. بلوک دستوری "Catch" ، خط را exception صادر شده را دریافت کرده و یک پیام هشدار صادر می‌کند.

آموزش ارسال مجدد خطای exception

گاهی اوقات ممکن است در هنگام رخدادن یک exception ، آن را به شیوه‌ای متفاوت از حالت اسنادار مدیریت کنید. این امکان وجود دارد که یک exception خاص را به وسیله یک بلوک کد "Catch" مجدداً فعال نمایید.

ارسال

یا

یک اسکریپت بایستی خطاها را از کاربر مخفی نگه دارد. خطاها 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 exception را با ارسال یک پیام مناسب و کاربر پسند، مجددا نمایش دهید.

```
<!--?php
class customException extends Exception {
    public function errorMessage() {
        //error message
    }
}

$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 :

کد مثال عملی فوق، چک می کند آیا کلمه "example" در آدرس ایمیل وجود دارد یا خیر. اگر وجود داشته باشد، خطای 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. خطای exception به دلیل وجود کلمه "example" در ایمیل، فعال خواهد شد.

5. بلوک دستوری "customException" خطای exception رخ داده را دریافت کرده و یک "catch" دریافت شده و یک پیام هشدار صادر می کند.

اگر خطای exception در بلوک "try" جاری آن گرفته نشود، برنامه به دنبال دریافت آن در بلوک "catch" و "catch" 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 با exception های 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 خطای exception کننده سطح بالا یا top level exception handler فعال خواهد شد. این تابع کلیه خطاها را در برنامه را دریافت و مدیریت می کند.

قوایین مربوط به exception در زبان PHP

معمولاً کدها را در ساختارهای "try" قرار می دهیم تا امکان ردگیری بالقوه exception را افزایش دهیم.

هر بلوک دستوری "throw" یا "block" باقیستی حداقل یک بلوک کد "catch" متناظر داشته باشد.

از exception های چندگانه می‌توان برای گرفتن کلاس‌های مختلف exception های برنامه استفاده نمود. هر Exception را می‌توان درون یک ساختار try-Catch متعلق به یک بلوک try ، ارسال یا ارسال مجدد-Thrown (rethrown) کرد.

توجه :

اگر یک خطا یا exception را ارسال می‌کنید (throw) حتماً باستی آن را دریافت (catch) کنید .

بخش چهارم : آموزش پایگاه داده MySQL در PHP

آموزش کار با پایگاه داده MySQL در PHP

با استفاده از زبان PHP ، می‌توانید به یک پایگاه داده متصل شده و اطلاعات را ذخیره و ویرایش کنید. پایگاه داده MySQL محبوب ترین پایگاه داده‌ای است که در سیستم برنامه نویسی PHP استفاده می‌شود .

پایگاه داده MySQL چیست؟

پایگاه داده MySQL یک سیستم پایگاه داده است که بر روی وب استفاده می‌شود.

MySQL یک سیستم پایگاه داده می‌باشد که بر روی سرور اجرا می‌شود.

MySQL یک database ایده‌آل برای پروژه‌های 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 می‌کند.

پایگاه داده SQL را می‌توان بر روی طیف وسیعی از پلتفرم‌ها اجرا نمود.

پایگاه داده MySQL یک database است که این سورس بوده و دانلود و استفاده از آن رایگان می‌باشد.

پایگاه داده MySQL توسط شرکت اورالک طراحی، توزیع و پشتیبانی می‌شود.

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. معمولاً 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-platform است، 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 MySQL را 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 تایخ را دریافت رکورد چند یا یک در نمایید.

به زیر که با زبان استاندارد SQL نوشته شده است، دقیق کنید :

```
SELECT LastName FROM Employees
```

فوق کلیه اطلاعات درون ستون "Employees" پایگاه داده "LastName" را انتخاب می کند. برای یادگیری زبان SQL ، 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 داده MySQL ، database بالغه و در حال استفاده برای سیستم های عظیم با حجم اطلاعات گسترده و تعداد کاربران زیاد مثل سایت های [Facebook](#)، [Twitter](#) و [Wikipedia](#) 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 به پایگاه داده MySQL Connect

نسخه های PHP5 و بالاتر برای اتصال به یک پایگاه داده MySQL از موارد زیر استفاده می کنند:

- MySQLi extension • حرف A مخفف عبارت پیشرفته تر یا improved است
- PDO (PHP Data Objects). • اشیای داده ای PHP با.

نسخه های قدیمی تر PHP از MySQL Database برای اتصال به استفاده می کردند. اما این افزونه از سال 2012 به بعد، منسوخ شده است.

کی و کجا از PDO یا MySQL استفاده کنیم؟

اگر بخواهیم خلائق بگوییم کی و از کجا از PDO یا MySQLi استفاده کنیم، جواب اینه "هر جا و هرچور که دوست داشتید!".!

اما هر دو PDO و MySQLi 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 MySQLi و PDO 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 Statement پشتیبانی می کنند. دستورات امن SQL سیستم 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 MySQLi و PDO

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 و Linux ، هنگامی که پکیج PHP5 MySQL را نصب می کنند، معمولاً افزونه MySQL خودکار نصب می شود. برای دریافت اطلاعات بیشتر برویید .
php.net/manual/en/mysqli.installation.php

آموزش نصب PDO

برای دریافت آموزش PDO برویید .
php.net/manual/en/pdo.installation.php

آموزش باز کردن یا اتصال یا MySQL Connection

قبل از این که بتوانیم به اطلاعات درون پایگاه داده MySQL دسترسی داشته باشیم، بایستی یک اتصال یا Connection را به سرور برقرار کنیم. کد زیر نحوه اتصال به MySQL را به صورت عملی نشان داده است

:

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

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);
```

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```

    // 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();
}
?>

```

PDO روش نکته

در کد مثال روش PDO مثال فوق، ما یک پایگاه داده به نام myDB را نیز تعیین کرده ایم. به این دلیل که برای کارکرد صحیح، نیاز به یک پایگاه داده معتبر دارد. اگر پایگاه داده ای برای آن تعیین نشود، یک خطا رخ 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 پرس می کند.

آموزش بستن یا Close در MySQL Connection

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 MySQL

```
$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 در MySQL ، شما بایستی مجوز لازم (مثل کاربر ادمین بودن) را داشته باشید .

آموزش ایجاد یک پایگاه داده MySQL جدید در روش MySQLi و PDO

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 (پارامترهای Username ، Servername و Password)

: نکته 2

اگر مجبور هستید از یک Port خاص در هنگام تعریف پایگاه داده استفاده کنید، یک رشته متغیر خالی empty 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 روش PDO ، یک پایگاه داده جدید به نام "myDBPDO" ایجاد می کند.

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

توجه :

مزیت مهم استفاده از روش PDO در ایجاد یک پایگاه داده جدید MySQL این است که روش PDO دارای یک کلاس database query exception class خاص جهت مدیریت خطاهای احتمالی رخ داده در دستورات 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 پرس می کند.

برای مثال، در کد فوق و در بخش SQL دستور Catch{} به همراه پیام خطا تولید شده را در خروجی نمایش داده ایم.

آموزش ایجاد یک جدول (Table) جدید در MySQL

یک جدول پایگاه داده یا database table دارای یک نام منحصر به فرد بوده و اطلاعات خود را در قالب چندین ستون (row) می نگهداری (column) 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

تعیین 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 PHP بروید.

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					
PRIMARY KEY	AUTO INCREMENT	UNSIGNED	DEFAULT value	NOT NULL	
این خامیت 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 معقولاً 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 این حالت MySQL مقدار ستون مورد نظر را به ازای وارد کردن یک آیتم جدید، به صورت اتوماتیک، یک واحد افزایش می دهد.	این خامیت برای انواع عددی به کار رفته و کاربر را فقط مجاز به استفاده از اعداد مثبت و صفر می کند.	این خامیت 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.	با تعیین این خامیت، ستون مورد نظر حتماً بایستی دارای مقدار بوده و مقدار تهی یا NULL پذیرفته نمی شود.	

در کدهای مثال عملی زیر، نحوه تعریف یک جدول (table) جدید را در MySQL نشان می دهد:

کد مثال MySQL 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();
?>
<!--?php روش ای MySQLi
$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 در MySQL Data Insert

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 (SQL query) را با استفاده از دستورات SQL query باستی داخل دو " " قرار دهید .
- مقادیر متغیر یا String در دستورات SQL query باستی داخل دو " " قرار بگیرند.
- مقادیر عددی را با استفاده از دو " " 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 دستورات SQL ، 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 برجسب زمانی یا TIMESTAMP مثل ستون ("reg_data") باشد، نیازی نبست 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 INTO نشان داده ایم

کد مثال MySQL شی گر

```
<!--?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);
?>
```

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 (INSERT) یا ویرایش اطلاعات (UPDATE) را اجرا کنیم، می‌توان 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 MySQL 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);
}
?>

```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```

        }
        $sql = "INSERT INTO MyGuests (firstname, lastname, email)
                VALUES ('John', 'Doe', 'john@example.com')";
        if ($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 رویه‌ای:

```

<!--?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);
} 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();
} 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()` 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) {
```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```

    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 MySQL در SELECT به صورت زیر است :

```
SELECT column_name(s) FROM table_name
```

اگر از کاراوتر * 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 های id ، firstname و lastname 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 ابتدای یک SQL query را با استفاده از دستور `select` ایجاد کرده و اطلاعات فیلدهای `id`، `firstname` و `lastname` را از جدول `MyGuests` خوانده 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، حلقه `while()` به جستجو درون آرایه پرداخته و اطلاعات خروجی شامل فیلدهای `id`،

`firstname` و `lastname` را نشان می دهد.

کد مثال زیر، دستورات مثال قبل را با استفاده از روش عیناً انجام داده و MySQLi رویه ای همان خروجی را

بر می گرداند :

کد مثال MySQLi شی گرای:

```
<!--?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 SQL آماده

در کد مثال عملی زیر از روش MySQLi PDO و دستورات آماده (Prepared Statements) که در درس های بعدی به آن خواهیم پرداخت، استفاده کرده ایم. کد این مثال اطلاعات فیلدهای id، lastname و firstname را از جدول MyGuests خوانده و در یک جدول HTML نشان 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>";
    }
}
```

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```

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 **DELETE** به صورت زیر است :

```
DELETE FROM table_name
WHERE some_column = some_value
```

نکته مهم :

به کاربرد عبارت **WHERE** در دستور فوق دقت کنید !

عبارت **WHERE** در دستور فوق تعیین می کند چه رکورد یا رکوردهایی بایستی حذف شوند. اگر در دستور **WHERE** استفاده نکنید، کلیه اطلاعات جدول مورد نظر حذف خواهد شد.

برای دریافت اطلاعات کامل تر درباره 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
۱۰:۲۴:۳۰ 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 می‌کند:

عنوان جدول				
reg_date	email	lastname	firstname	id
۱۴:۲۶:۱۵ ۲۰۱۴-۱۰-۲۲	john@example.com	Doe	John	۱
۱۰:۳۲:۳۰ 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 می‌شود. ساختار کلی استفاده از دستور UPDATE در 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

عبارت **LIMIT** را برای محدود کردن تعداد رکوردهای خوانده شده و یا تعیین شمار رکوردهای مورد نظر دارد.

عبارت **LIMIT**، 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 SQL query زیر.

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 (OFFSET 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 می‌شوند.

آموزش استفاده از دستورات آماده (Prepared) در SQL

دستورات آماده در SQL یا Prepared 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 MySQL Prepared statement خواهیم پرداخت.

معرفی دستورات آماده (Bound Parameters) و پارامترهای متصل : Prepared statements

یک دستور آماده SQL یا Prepared statement قابلیتی است که به وسیله آن می‌توان یک دستور یکسان SQL یا دستوری مشابه آن را به صورت مکرر و با تاثیرگذاری بالا اجرا کرد.

دستورات آماده SQL به صورت کلی با روش زیر عمل می‌کنند:

- ۱- آماده سازی اولیه (Prapare) دستور : یک الگو برای دستور SQL مورد نظر ایجاد شده و به پایگاه داده ارسال می‌شود. مقادیر مشخصی به صورت تعیین نشده در دستور باقی می‌مانند که به آن‌ها پارامتر گفته و با نماد ? در کد SQL جایگزین می‌شوند. برای مثال، الگویی همانند کد زیر برای یک دستور درج اطلاعات ایجاد می‌شود :

```
INSERT INTO MyGuests VALUES (?, ?, ?)
```

- ۲- پایگاه داده الگوی دستوری SQL را خوانده و اجرا می‌کند. همچنین با استفاده از قابلیت بهینه سازی query optimization سریع ترین راه را برای اجرای آن مشخص می‌کند. در انتها نتایج حاصل از query را بدون اجرای نهایی در حافظه نگهداری می‌کند.

- ۳- مرحله سوم اجرا : (Execute) در مرحله آخر، برنامه مقادیر (values) را به پارامترها ارسال کرده و پایگاه داده دستور SQL را به صورت کامل اجرا می‌کند. برنامه می‌تواند یک دستور واحد SQL را با مقادیر مختلف، هر چند بار که نیاز داشته باشد اجرا کرده و خروجی‌های مختلف تولید کند. در مقایسه با اجرای مستقیم دستورات SQL، 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 می‌دهند (حتی با وجود این که دستورات مهم است چندین بار اجرا شوند.)

استفاده از پارامترها، حجم اطلاعات ارسالی به سرور را بسیار کاهش می‌دهند، زیرا در هر بار اجرای query فقط کافی است مقادیر جدید پارامترها را ارسال کنید نه کل دستور query را.

دستورات آماده 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 می‌کنند.

آموزش کار با دستورات آماده SQL در MySQLi

کد مثال عملی زیر از دستورات آماده و پارامترهای متصل در MySQLi استفاده می‌کند :

```
<?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 می‌پردازیم:

```
"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 مند. کاراکتر "s" در پارامتر فوق به mysql 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 یا integer

d : اعشاری یا double

S : رشته ای یا string

b : عددی بزرگ یا Binary 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 MySQLi PDO در روش Prepared SQL Statement استفاده می کند

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

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```

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 در PHP

آموزش کار با زبان XML در PHP چیست؟

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.

ساختاری شبیه HTML دارد، با این تفاوت که در آن می توانید تگ های مورد نظر خود را ایجاد

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

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 پروردید.

یک مفسر زبان XML یا XML Parser چیست؟

برای خواندن، ایجاد، به روزرسانی و تغییر یک سند XML، 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 دارخته XML با Tree-Based Parsers

- مفسس متن، بروگداد XML پارسرز با بندهاد

مفسر درختی XML پا Tree-Based Parsers

مفسر درختی XML یا Tree-Based Parsers کل سند XML را خوانده و در حافظه قرار می‌دهد. سپس ساختار سند XML را به صورت درختی ترسیم کرده و کل اطلاعات آن را آنالیز می‌کند. این ؟؟؟؟؟؟؟؟ امکان دسترسی به اعضای هر درخت را مبتنی بر ؟؟؟؟؟؟؟ فراهم می‌کند. مفسر درختی XML برای مدیریت اسناد ؟؟؟؟؟؟؟ کوچک مناسب بوده و در فایل‌های بسیار بزرگ XML کارآئی نداشته و می‌تواند عملکرد سیستم را مختل کند. از مدل‌های مفسر درختی XML می‌تواند به نمونه‌های زیر اشاره کرد:

- ## SimpleXML ل ساده XML

- DOM. مدل

Event-Based Parsers میتنی یز رویداد XML پا

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 پرای کار یا فایل های XML بسیار بزرگ مناسب بوده و اسناد XML را سریع تر لود می کند.

همجین حجم بسیار کمتری حافظه سیستم را اشغال خواهد کرد.

از نمونه های مفسر متغیر روئیداد XML می توان به موارد زیر اشاره کرد :

XML Reader •

XML Expat Parsers. •

آموزش کار با مفسر 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 SimpleXML یک مفسر درختی XML یا tree-based Parser است.

مفسر زبان XML نوع SimpleXML، یک راه ساده برای دریافت مقدار نام (name)، خواص (attribute) و محتوى متني هر عنصر (element) در یک سند XML را به شرط داشتن 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 Except Parser، مفسر SimpleXML برای خواندن یک عنصر در 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_string() برای خواندن اطلاعات XML از یک متغیر متني String استفاده می شود.

فرض کنید که یک متغیر متني 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>";
```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

در کد مثال عملی زیر، نحوه خواندن اطلاعات XML از یک متغیر متنی String با استفاده از تابع

`simplexml_load_string()` 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 Handling) 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 رخداده 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:

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```
<?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 () در PHP simplexml_load_file() 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 () را آموزش داده ایم:

```
$xml=simplexml_load_file("note.xml") or die ("Error: Cannot create object");
print_r($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 SimpleXML خواهیم پرداخت.

مثال هایی بیشتر درباره مفسر SimpleXML

برای دریافت اطلاعات بیشتر درباره مفسر SimpleXML به بخش مرجع آموزش SimpleXML در سایت تحلیل داده بروید.

آموزش خواندن مقادیر گره ها با استفاده از SimpleXML

همانطور که در درس قبلی اشاره کردیم، یک افزونه PHP است که امکان ویرایش، خواندن و نوشتן اطلاعات فایل های XML را در زبان PHP، 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 ابزار SimpleXML آموخته دارید.

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 المتن 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 `<book>` را در المتن `<title>` اول و دوم فایل

`books.xml` را 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 المتن های `<book>` در فایل

"حرکت کرده و مقدار گره های المتن 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 با SimpleXML (Attribute)

در کد مثال عملی زیر، مقدار خاصیت "Category" مربوط به عنصر اول `<book>` و مقدار خاصیت "lang"

مریبوط به عنصر `<title>` در دومین المتن `<book>` را با استفاده از SimpleXML خوانده ایم :

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```
<?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 المنشی 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 ساخته XML Expat Parser ، امکان خواندن و پردازش اطلاعات فایل های XML را در زبان PHP فراهم می کند.

همانطور که قبلا اشاره کردیم، XML Expat یک مفسر مبتنی بر رویداد یا event-based Parser است.

به قطعه کد XML زیر دقت کنید :

```
<from>Jani</from>
```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

یک مفسر مبتنی بر رویداد، قطعه کد XML فوق را به صورت یک مجموعه سه تایی از رویدادهای متوالی، گزارش می‌کند. به صورت زیر:

المنت آغازین یا Start element که "form" است.

شروع کننده بخش CDATA که دارای مقدار "Jani" می‌باشد.

المنت پایان دهنده Close element که باز هم "form" است.

مفسر زبان XML نوع Expat یکی از توابع درون ساخته زبان PHP بوده و برای استفاده از آن نیاز به نصب افزونه با 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 در XML Expat Parser

در کد زیر، ما مفسر XML Expat Parser را مقداردهی اولیه کرده و تعدادی مدیریت کننده رویداد تعیین کرده ایم. در نهایت هم فایل XML را خوانده و پردازش کرده (handler) را برای چندین رویداد

: ایم

```
<?php
// Initialize the XML parser
$paser=xml_parser_create();
// Function to use at the start of an element
function start($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 رخدادن خطاهای XML را اضافه کرده ایم تا خطای XML را در خطا (errors) داده را به خطای 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 Expat Parser

برای دریافت اطلاعات بیشتر درباره تابع های مفسر Expat Parser به بخش مرجع آموزش مفسر Parser در سایت تحلیل داده بروید.

آموزش کار با مفسر XML DOM در PHP

مفسر درون ساخته XML ، امکان پردازش و خواندن فایل های XML را در PHP فراهم می کند. مفسر XML DOM یک مفسر درختی یا tree-based parser است.

به قطعه کد XML زیر دقت کنید :

```
<?xml version="1.0" encoding="UTF-8" ?>
<from>Jani</from>
```

مفسر XML ، 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 یا <form> 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 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>
```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```
<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 (Object) 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 متغیر string قرار 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 عنصراها وجود دارد. هنگامی که فایل XML تولید می شود، برخی موقع فواصل سفید خالی بین گره ها به وجود می آید. مفسر XML با این عناصر همانند عناصرهای معمولی برخورد کرده و اگر از وجود آن ها اطلاع نداشته باشد، ممکن است برخی موقع برنامه را دچار مشکل کنند.

بخش ششم : آموزش کاربرد Ajax در PHP چیست؟

روشی 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.

در خلاصه به معنای استفاده غیر همزمان از جاوا اسکریپت و XML است. یک تکنیک جدید برای ایجاد و ساخت صفحات وب دینامیک می باشد. به 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 مجدد 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 مثل google map ، yahoo gmail

چگونه کار می کند؟ Ajax

دیاگرام زیر نشان می دهد که Ajax چگونه کار می کند؟

Ajax با استانداردهای موجود ایجاد شده است:

Ajax بر 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 کند.

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

تکنولوژی Ajax ، 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 وسیله Ajax ، بدون نیاز به لود شدن کامل صفحه، آموزش می دهیم. کد مورد نظر برای انجام کار را نیز به زبان PHP خواهیم نوش特.

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 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 مثال فوق، هنگامی که کاربر در کادر متن نوشته ای را وارد می کند، یک تابع به نام "showHint()" اجرا می شود. این تابع با رویداد keyup 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، برتابمه 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 `XMLHttpRequest` 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.

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 صفحه‌ی وب (از اپلیکیشن تحت وب) می‌تواند با فراخوانی توابع AJAX، داده‌ای از دیتابیس واکشی کند:

در این مثال کاربر اسم شخص مورد نظر را از لیست کشویی انتخاب کرده و اطلاعات مربوطه‌ی آن را بدون اینکه کل صفحه بروز رسانی شود، از دیتابیس می‌خواند و در زیر نمایش می‌دهد:

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.

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک ۶۵۱ طبقه دوم - واحد ۷

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 :

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

- یک آبجکت XMLHttpRequest ایجاد می شود.
- یک تابع تعریف می شود که به هنگام آماده شدن پاسخ سرویس دهنده(server response) را به فایل مستقر بر روی سرویس دهنده هدایت می شود.
- درخواست (request) را به فایل مستقر بر روی سرویس دهنده فراخوانی شده و اجرا 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

برمی 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,$sql);
echo "<table><tr><th>Firstname</th><th>Lastname</th></tr>";
while($row=mysqli_fetch_array($result)) {
    echo "<tr><td>" . $row["firstname"] . "</td><td>" . $row["lastname"] . "</td></tr>";
}
echo "</table>";
```

```

        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. یک PHP connection به سرور MySQL ایجاد می کند.
2. شخص مورد نظر و اطلاعات مربوطه‌ی آن یافت می شود.
3. یک جدول HTML ایجاد شده با داده های مورد نیاز پر می شود و متعاقباً به پارامتر مکان نگهدار (placeholder) "txtHint" ارسال می گردد.

آموزش XML و / XML – PHP – AJAX و واکنش اطلاعات از فایل

مثال XML و AJAX

مثال زیر نمایش می دهد یک صفحه‌ی وب چگونه اطلاعات لازم را از فایل XML با توابع AJAX ، بدون نیاز به بارگذاری مجدد کل صفحه، واکشی کرده و نشان می دهد:

Example

TITLE: Empire Burlesque
ARTIST: Bob Dylan
COUNTRY: USA
COMPANY: Columbia
PRICE: 10.90
YEAR: 1985

Example

TITLE: One night only
ARTIST: Bee Gees
COUNTRY: UK
COMPANY: Polydor
PRICE: 10.90
YEAR: 1998

Example

TITLE: The very best of
ARTIST: Cat Stevens
COUNTRY: UK
COMPANY: Island
PRICE: 8.90
YEAR: 1990

شرح مثال – صفحه‌ی HTML

زمانی که کاربر یک CD را از لیست کشویی حاضر انتخاب می‌کند، تابعی به نام "showCD()" فراخوانی و اجرا می‌شود.

تابع مورد نظر به دنبال رخداد "onchange" صدا زده می‌شود:

```
<?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();
    }
}

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 فایل XML ، "cd_catalog.xml" ، را بارگذاری کرده، یک کوئری بر روی فایل XML اجرا می کند و در پایان نتیجه ای 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 صفحه‌ی PHP ارسال می‌شود، اتفاقات زیر به ترتیب رخ می‌دهد:

- یک آبجکت XML DOM PHP ایجاد می‌کند.
- تمامی المان‌های که با اسم ارسالی از جاوا اسکریپت منطبق می‌باشد را پیدا می‌کند.
- اطلاعات آلبوم را به پارامتر جایگزین و مکان نگهدار "txtHint" ارسال می‌کند.

آموزش – PHP پیاده سازی قابلیت جستجوی تعاملی و زنده با توابع AJAX (AJAX Live search)

پیاده سازی قابلیت جستجوی تعاملی با توابع AJAX

مثال زیر یک کادر با قابلیت جستجوی تعاملی و زنده را پیاده سازی می‌کند که در آن به هنگام درج واژگان مورد نیاز در کادر و بدون بارگذاری مجدد کل صفحه، کاربر بلافصله نتایج دلخواه را مشاهده می‌کند.

قابلیت جستجوی تعاملی و زنده (Live search) در مقایسه با جستجو به روش قدیمی، از مزایای زیر برخوردار می‌باشد:

- همین که کاربر در حال تایپ واژگان جستجو در کادر می‌باشد، نتایج در لحظه برای وی نمایش داده می‌شوند.

- همین که کاربر تایپ را ادامه می‌دهد، نتایج جستجو محدود و محدودتر می‌شود.

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

- زمانی که نتایج بیش از حد محدود می‌شود، کاربر می‌تواند با حذف تعدادی از کاراکترها در کادر، موارد بیشتری را مشاهده کند.

نمونه‌ی زیر یک کادر با قابلیت جستجوی تعاملی و نمایش نتایج در لحظه را نمایش می‌دهد:



نتایجی که در مثال بالا مشاهده می‌کنید، داخل یک فایل XML قرار دارد که محتوای آن را در زیر مشاهده می‌کنید:

```

<?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 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

صفحه‌ی 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 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 مربوطه‌ی JavaScript ((`strlen($q) > 0`))، متنی را به سرویس دهنده ارسال می‌کند، اتفاقات زیر به ترتیب رخ می‌دهند:

- محتوای یک فایل XML داخل آبجکت XML DOM در حافظه بارگذاری می‌شود.
- داخل تمامی المان‌های title حلقه زده (می‌چرخد) و تا (زمانی که) مورد منطبق با متن ارسال شده از کد JavaScript را پیدا کند.
- url صحیح و title مناسب را در متغیر "\$response" ذخیره می‌کند. زمانی که مورد منطبق بیش از یک آیتم باشد، در آن صورت تمامی موارد منطبق و یافت شده داخل متغیر مذبور قرار داده می‌شوند.
- اگر هیچ مورد منطبقی یافت و استخراج نشد، مقدار متغیر \$response برابر رشته‌ی "no suggestion" می‌شود.

آموزش - PHP آموزش پیاده سازی خبر خوان RSS READER با PHP در AJAX

ایAJAX RSS Reader – RSS Reader
کل صفحه، در اپلیکیشن تحت وب بارگذاری می کند

مثال زیر یک RSS Reader را نمایش می دهد که در آن خبرهای RSS با فرآخوانی توابع ajax و بدون لود مجدد کل صفحه در اپلیکیشن بارگذاری می شود:



شرح مثال – فایل HTML

زمانی که کاربر یک آیتم RSS-feed را از لیست کشویی انتخاب می کند، تابعی به نام "showRSS()" فرآخوانی 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 می‌دهد:

- بررسی می‌کند آیا یک آیتم RSS-feed انتخاب شده است یا خیر.
- یک آبجکت XMLHttpRequest ایجاد می‌کند.
- یک تابع تعریف می‌کند و این تابع زمانی اجرا می‌شود که سرویس دهنده آماده ی پاسخ به درخواست ارسال شده به سرویس دهنده را به فایل مربوطه هدایت می‌کند.
- درخواست ارسال شده به سرویس دهنده از کد جاوا اسکریپت می‌باشد.
- یک پارامتر (`q`) به URL با محتوای لیست کشویی) اضافه می‌کند.

فایل PHP

صفحه‌ای که بر روی سرویس دهنده قرار داشته و کد JavaScript فوق آن را فراخوانی می‌کند، در واقع یک فایل PHP به نام "getrss.php" می‌باشد. محتوای فایل مذبور را در زیر مشاهده می‌کنید:

```

<!--?php
//get the q parameter from URL
$q=$_GET["q"];

```

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```

        //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 از المان channel استخراج کرده و به عنوان خروجی پاس می دهد.
- المان ها از المان item استخراج کرده و به عنوان خروجی بازمی گرداند.

AJAX (AJAX Poll) پیاده سازی قابلیت نظرسنجی با

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، برنامه‌ی زیر تیجه را از سرویس دهنده خوانده و بلافاصله بدون بروز رسانی کل صفحه، در اپلیکیشن تحت وب حاضر به نمایش می گذارد.

Do you like PHP and AJAX so far?

Yes:

No:

Result:

Yes: 82%

No: 18%

شرح مثال – فایل HTML

زمانی که کاربر یکی از دو گزینه‌ی 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 آجکت 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 یا no

فایل PHP

صفحه‌ی 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>

```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

```

<table>
<tbody><tr>
<td>Yes:</td>
<td>
<img src="poll.gif" width="php echo (100*round($yes/($no+$yes),2)); ?&gt;" height="20"&gt;
&lt;!--?php echo(100*round($yes/($no+$yes),2)); ?--&gt;%
&lt;/td&gt;
&lt;/tr&gt;
&lt;tr&gt;
&lt;td&gt;No:&lt;/td&gt;
&lt;td&gt;
&lt;img src="poll.gif" width="<?php echo (100*round($no/($no+$yes),2)); ?&gt;" height="20"&gt;
&lt;!--?php echo(100*round($no/($no+$yes),2)); ?--&gt;%
&lt;/td&gt;
&lt;/tr&gt;
&lt;/tbody&gt;&lt;/table&gt;
</pre

```

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) اعطای نمایید.