



وب پیشرفته با (PHP)

رشته: کامپیوتر ساینس - دیپارتمنت: سیستم معلوماتی
صنف سوم - سمستر پنجم

Ketabton.com

سال: 1404 هجری شمسی

فهرست

صفحه

عنوان

ط.....	مقدمه.....	
۱.....	فصل اول: اصول طراحی صفحات وب.....	
۲.....	مرحله اول.....	۱.۱
۲.....	طرح اولیه (Plan).....	۱.۲
۳.....	اصول طراحی اولیه وبسایت (Plan).....	۱.۲.۱
۳.....	در نظر گرفتن دیدگاه مخاطبان.....	۱.۲.۲
۴.....	مفاهیم اولیه صفحات وب.....	۱.۲.۳
۴.....	Web server.....	۱.۳
۶.....	URL.....	۱.۳.۱
۶.....	مراحل دریافت اطلاعات در صفحات وب.....	۱.۳.۲
۷.....	اجازه فضا و Donai n.....	۱.۴
۱۰.....	فصل دوم: معرفی PHP.....	
۱۱.....	صفحات وب static.....	۲.۱
۱۱.....	صفحات وب Dynamic.....	۲.۲
۱۱.....	توانایی های PHP.....	۲.۳
۱۲.....	تاریخچه PHP.....	۲.۴
۱۳.....	PHP به حیث یک لسان Open Source.....	۲.۴.۱
۱۴.....	سرعت PHP.....	۲.۵
۱۴.....	ساختار مناسب و امکانات بالا در PHP.....	۲.۶
۱۴.....	ضروریات برای استفاده از PHP.....	۲.۷
۱۵.....	نصب و فعال سازی apache server.....	۲.۸
۱۶.....	XAMPP.....	۲.۸.۱
۲۱.....	نوشتن کدهای PHP.....	۲.۹
۲۴.....	مسیر عملیاتی کدهای PHP.....	۲.۱۰
۲۵.....	توضیحات (Comments).....	۲.۱۱
۲۹.....	فصل سوم: نوعیت دیتا (Data type).....	
۳۰.....	متحول ها (Variabl es).....	۳.۱
۳۰.....	شرایط تعریف متحول ها در PHP.....	۳.۱.۱

۳۳.....	نوعیت دیتا (Data type)	۳.۲
۳۴.....	دیتاتایپ String	۳.۲.۱
۳۴.....	دیتاتایپ integer	۳.۲.۲
۳۶.....	دیتاتایپ float	۳.۲.۳
۳۶.....	دیتاتایپ Boolean	۳.۲.۴
۳۷.....	دیتاتایپ array	۳.۲.۵
۳۷.....	دیتاتایپ Object	۳.۲.۶
۳۸.....	دیتاتایپ Null	۳.۲.۷
۳۹.....	عملگرها در PHP	۳.۳
۳۹.....	عملگرهای حسابی (arithmetic operators)	۳.۳.۱
۴۰.....	عملگرهای جایگزینی (assignment operator)	۳.۳.۲
۴۲.....	عملگرهای مقایسوی (Comparison operators)	۳.۳.۳
۴۵.....	عملگرهای افزایشی یا کاهششی (Increment/Decrement)	۳.۳.۴
۴۶.....	عملگرهای منطقی (Logical operators)	۳.۳.۵
۴۷.....	عملگر منطقی and	۳.۳.۶
۴۷.....	عملگر منطقی or	۳.۳.۷
۴۸.....	عملگر XOR	۳.۳.۸
۴۹.....	عملگر NOT	۳.۳.۹
۵۰.....	بیانیه‌های شرطی (Condition statements)	۳.۴
۵۰.....	دستور IF	۳.۴.۱
۵۱.....	دستور if...else	۳.۴.۲
۵۱.....	دستور if...elseif	۳.۴.۳
۵۲.....	دستور switch	۳.۴.۴
۵۴.....	بیانیه‌های تکرار (Looping)	۳.۵
۵۴.....	For loop	۳.۵.۱
۵۵.....	While loop	۳.۵.۲
۵۶.....	Do while loop	۳.۵.۳
۵۷.....	بیانیه تکراری For each loop	۳.۵.۴
۵۸.....	صف (array)	۳.۶
۵۸.....	طریقه ایجاد array	۳.۶.۱
۵۸.....	اقسام array:	۳.۶.۲
۵۸.....	Index array:	۳.۶.۳

۵۹.....	Associ at i ve Arrays	۳.۶.۴
۶۱.....	متحول های جهانی PHP Gl obal Vari abl es – Supergl obal s	۳.۷
۶۴.....	فصل چهارم: PHP Forms	
۶۵.....	فرم در PHP	۴.۱
۶۶.....	\$_POST	۴.۱.۱
۶۷.....	\$_GET	۴.۱.۲
۶۸.....	میتودهای Get and Post	۴.۲
۶۸.....	میتود (Get)	۴.۲.۱
۶۸.....	میتود (Post)	۴.۲.۲
۶۸.....	Formval i dat i on	۴.۳
۷۱.....	PHP - Requi red Fi el ds	۴.۴
۷۳.....	Di spl ay The Error Messages	۴.۴.۱
۷۴.....	PHP - Val i dat e E-mai l	۴.۵
۷۸.....	فصل پنجم: PHP پیشرفته	
۷۹.....	PHP i ncl ude	۵.۱
۸۱.....	I ncl ude vs. requi red	۵.۲
۸۳.....	PHP fi l e handl i ng	۵.۳
۸۳.....	PHP mani pul at i ng fi l es	۵.۳.۱
۸۳.....	خواندن فایل (readfi l e)	۵.۳.۲
۸۶.....	Fcl ose() بسته نمودن فایل	۵.۴
۸۶.....	PHP fi l e upl oad	۵.۵
۹۳.....	PHP cooki es	۵.۶
۹۳.....	ایجاد کوکی توسط PHP	۵.۶.۱
۹۵.....	حذف کوکی	۵.۶.۲
۹۶.....	بررسی کردن کوکی	۵.۶.۳
۹۷.....	PHP sessi on	۵.۸
۹۸.....	آغاز در PHP sessi on	۵.۸.۱
۹۹.....	به دست آوردن مقادیر متحول های sessi on	۵.۸.۲
۱۰۰.....	نابود کردن PHP sessi on	۵.۸.۳
۱۰۱.....	PHP Error Handl i ng	۵.۹
۱۰۲.....	Set Error Handl er	۵.۹.۱

۱۰۴.....	اقسام اشتباهات.....	۵.۹.۲
۱۰۵.....	PHP Except i on Handl i ng	۵.۱۰
۱۰۵.....	:Use of Except i on	۵.۱۰.۱
۱۰۶.....	Try, throw and cat ch	۵.۱۰.۲
۱۱۰.....	فصل ششم: PHP & MySQL	
۱۱۱.....	PHP and MySQL	۶.۱
۱۱۱.....	MySQL	۶.۱.۱
۱۱۲.....	دیتابیس کیوری (dat abase query)	۶.۲
۱۱۲.....	Connect to MySQL	۶.۳
۱۱۳.....	Create a MySQL Dat abase	۶.۳.۱
۱۱۵.....	Create MySQL Tabl es	۶.۳.۲
۱۱۷.....	PHP I nsert Dat a i nt o MySQL	۶.۵
۱۱۸.....	:PHP Get I D of Last I nsert ed Record	۶.۶
۱۱۹.....	:PHP I nsert Mul t i pl e Records i nt o MySQL	۶.۷
۱۲۱.....	PHP Sel ect Dat a f rom MySQL	۶.۸
۱۲۱.....	تابع fetch_ assoc()/mysql i _f et ch_ assoc ()	۶.۸.۱
۱۲۲.....	PHP Del et e Dat a f rom MySQL	۶.۹
۱۲۴.....	PHP Updat e Dat a i n MySQL	۶.۱۰
۱۲۵.....	PHP Li mi t Dat a Sel ect i ons f rom MySQL	۶.۱۱
۱۲۷.....	پروژه (Proj ect)	۶.۱۲
۱۴۴.....	منابع و مأخذ (References)	

مقدمه

PHP یک زبان اسکریپت‌نویسی Server side است که برای ایجاد صفحات Dynamic website از آن استفاده می‌شود و بیشترین کاربرد را در سیستم‌های ویندوز و لینوکس دارد. PHP از جمله زبان‌های برنامه‌نویسی محبوب در بین استفاده‌کنندگان می‌باشد. علت این محبوبیت عواملی چون سرعت، انعطاف‌پذیری، قابلیت انتقال و رایگان بودن آن است.

امروز وبسایت به قسم گسترده‌یی رشد کرده، در محیط‌های علمی، اقتصادی، فرهنگی و اجتماعی کاربرد زیاد دارد و سهولت‌های زیادی را به‌میان آورده است. فعلاً، در جهان چندین میلیارد وبسایت در بستر اینترنت وظیفه نشر انبوه اطلاعات را بر عهده دارد. هر روز به تعداد و اهمیت وبسایت‌ها افزوده می‌شود، چون نیاز مندی‌های بشر رو به افزایش است.

پس برای آنانی که با مسلک طراحی وبسایت‌آشنایی دارند ضرورت است که دانش خویش را در این عرصه بیشتر رشد دهند، تا پاسخگوی نیازمندی‌های روز باشند. طراحی وبسایت بیشتر نیازمند اشخاص مبتکر و طراح می‌باشد و در کنار ابتکار به دانش فنی نیز ضرورت است. پس توصیه‌ما برای محصلان گرامی این است که بیشتر مطالعه کرده، عملاً روی سیستم‌های مورد نظر کار نمایند، چون طراحی وبسایت مهارتی است که به صورت عملی باید آن را آموخت.



هدف کلی کتاب

آشنایی محصلان با طراحی صفحات دینامیکی و قابلیت‌های زبان PHP.

فصل اول

اصول طراحی صفحات وب



هدف کلی: محصلان با نیازمندی‌های طراحی وبسایت آشنا می‌شوند.

اهداف آموزشی: در پایان این فصل محصلان قادر خواهند شد تا:

۱. اصول طراحی وبسایت را تعریف کنند.
۲. وب سرور را توضیح دهند.
۳. مراحل دریافت اطلاعات در صفحات را توضیح دهند.
۴. اجازه فضا یا domain را شرح دهند.

در این فصل محصلان در مورد اصول طراحی صفحات وب معلومات بدست خواهند آورد، هم‌چنان می‌توانند وظیفهٔ وب سرور را بدانند؛ چگونگی دریافت اطلاعات در صفحات وب را توضیح داده و اجازهٔ فضا یا Domain را می‌توانند تشریح کنند.

برای ایجاد یک وب‌سایت، مانند هر Software دیگر باید مراحل را طی نمود تا ضمن تعیین اهداف مورد نظر از ایجاد صفحات وب، مسیر رسیدن به اهداف نیز تا حدی روشن شود. در طراحی صفحات وب، توجه به موارد زیر ضروری است:

۱.۱ مرحلهٔ اول

در این مرحله باید مشخص شود که هدف از ایجاد وب‌سایت چیست؟ برای رسیدن به این پاسخ ضرورت است تا با سفارش‌دهنده‌ای وب‌سایت مصاحبه صورت گیرد تا پاسخ‌های روشن برای پرسش‌های زیر به دست آید.

- موارد استفاده وب‌سایت:
- وب‌سایت تعلیمی؛
- وب‌سایت تجارتي؛
- وب‌سایت خبری؛
- وب‌سایت نشراتی؛
- وب‌سایت معلوماتی؛
- مخاطبان وب‌سایت چه افراد و اشخاص هستند:
- استادان پوهنتون؛
- محصلان و یا دانش‌آموزان؛
- کودکان؛
- افرادی که شغل آزاد دارند.

۱.۲ طرح اولیه (Plan)

پس از شناخت کافی از اهداف ایجاد وب‌سایت، باید طرحی را آماده نمایید که موارد زیر را شامل شود:

بخش‌های اصلی و فرعی وب‌سایت

- بخش اصلی یک وب‌سایت همان اهداف وب‌سایت را شامل می‌شود.
- بخش فرعی به‌منظور جلب توجه مخاطبان و جذابیت صفحات وب می‌باشد.

آماده ساختن این طرح، یک طراح وب‌سایت را قادر می‌سازد تا یک وب‌سایت درست را طراحی نماید.

۱.۲.۱ اصول طراحی اولیه وبسایت (Plan)

طراحی صفحات وب ارتباط زیاد به خلاقیت، ابتکار و مهارت‌های حرفوی فرد طراح دارد، پس در این بخش کوشش شود تا طراح یک فرد مبتکر باشد. زیاد از وبسایت‌های طراحی شده پیروی نشود و کوشش شود تا طراح، طرح‌های جدید را ارائه نماید تا وبسایت بیشتر جلب توجه نماید.

- وبسایت، طوری دیزاین شود که در مرورگرهای وب‌های معروف دنیا به صورت درست و کامل نمایش داده شود، یعنی در جریان دیزاین با مرورگرهای مختلف Test شود.
- اگر می‌خواهید یک وبسایت را به زبان دری دیزاین نمایید، توجه کنید که این دیزاین در همه سیستم عامل‌ها قابلیت اجرایی (خواندن) را داشته باشد.
- صفحه اصلی یا Home page وبسایت را جذاب و مقبول دیزاین نمایید تا توجه افرادی را که به صورت تصادفی از آن بازدید می‌کنند، به خود جلب کند.
- برای ازدیاد جذابیت وبسایت از تصاویر، صداها و انیمیشن‌های گیرا استفاده شود.
- رنگ نوشته‌ها، رنگ چوکات صفحه، تصاویر و سایر اجزای صفحه باید با یک دیگر هماهنگی داشته باشند.

۱.۲.۲ در نظر گرفتن دیدگاه مخاطبان

- زمینه پیشنهاد و انتقاد برای مخاطبان وبسایت باز گذاشته شود.
- مطالب وبسایت متناسب با فعالیت‌های استفاده‌کنندگان وبسایت باشد.
- از تبلیغات تصویری که حجم زیادی را اشغال می‌کند تا حد امکان جلوگیری شود، چون در Load شدن صفحات وب نقش زیاد دارد، زیرا در صورت حجیم بودن، سرعت Load کم می‌شود.

۱.۲.۳ مفاهیم اولیه صفحات وب

یک وبسایت، عبارت از مجموعه صفحاتی است که در آن‌ها اطلاعات درج می‌شود. این صفحات به دو قسم است. صفحات اصلی (Home Page) و صفحات فرعی (Web Page).

صفحات وب معمولاً با این پسوندها (extinctions) ذخیره می‌شود: HTML, HTM, PHP, ASP, ASPX, JSP. این پسوندها وابسته به پروگرام‌هایی هستند که توسط آن، صفحات دیزاین می‌شود.

- به صفحه اول یک وبسایت Home page یا صفحه اصلی گفته می‌شود. این صفحه معمولاً به نام Index, Default است.

صفحات وب از دو بخش تشکیل شده است:

- صفحات ثابت (Static)
- صفحات متحرک (Dynamic)

۱.۳ Web server

عبارت از Server است که دیتا را با Frame‌های آن از دیتابیس‌های مختلف دریافت نموده، آن‌ها را پروسس کرده و به کدهای HTML, XML تبدیل می‌کند و سپس کدهای تولید شده را به سمت مرورگر وب ارسال می‌کند. استفاده‌کننده‌گان می‌توانند این اطلاعات را به کمک مرورگرهای وب از وبسایت دریافت کنند.



شکل (۱-۱): معرفی یک شبکه

محتویات Directory Root شامل این بخش‌ها می‌باشد:

- Index.aspx یا صفحه اصلی یک وبسایت.
- Image folder که حاوی تصاویر استفاده شده در وبسایت می‌باشد.
- Pages Folder که حاوی صفحات فرعی یا Web page می‌باشد.
- Database folder که حاوی دیتابیس است که محتویات وبسایت در آن ذخیره می‌باشد.

جهت تمرینات وبسایت‌تان از این Host به شکل رایگان استفاده کنید:

WWW.aspspider.net

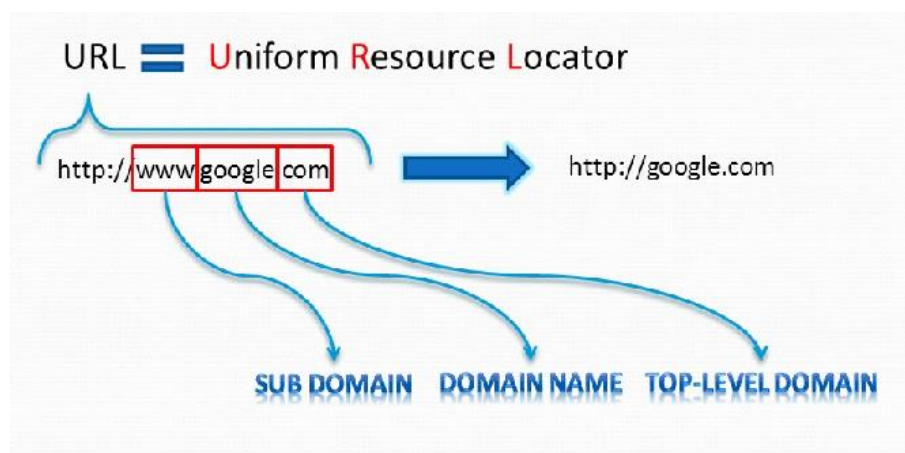


شکل (۱-۲): معرفی اتصال کامپیوتر به اینترنت

۱.۳.۱ URL

عبارت از یک آدرس است که در آدرس بار مرورگر وب نوشته شده و نظر به آن آدرس، وبسایت مورد علاقه استفاده کننده جستجو شده، در اختیارش قرار می گیرد.

URL دارای بخش های مختلف بوده که در شکل زیر به طور مکمل نشان داده شده است.



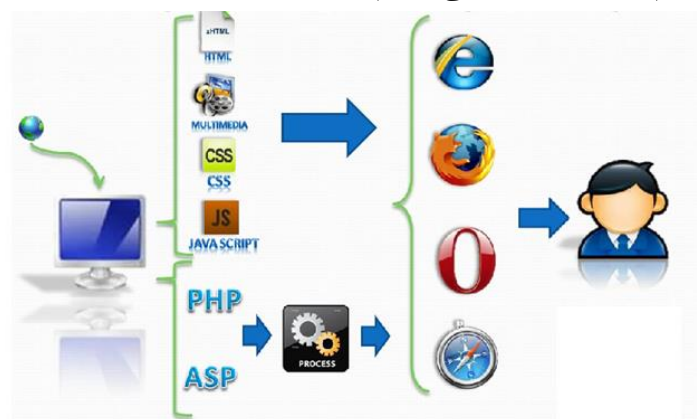
شکل (۱-۳): معرفی URL

۱.۳.۲ مراحل دریافت اطلاعات در صفحات وب

هر استفاده کننده یی که از اینترنت استفاده کند، محتویات وبسایت مراحل زیر را سپری نموده، به دسترس آن ها قرار می گیرد.

دیتا از داخل وب سرور به اینترنت قرار گرفته، سپس در هر نقطه از جهان قابل دسترس می باشد. برای دسترسی به اطلاعات ضرورت است تا کمپیوتر، موبایل هوشمند، تبلت یا آی پد داشته باشیم.

در کنار داشتن یکی از این وسایل، ضرور است تا نرم افزاری به نام web browser یا مرورگر وب نیز در آن نصب گردد. بعد از نصب نرم افزار مورد نظر، می توانیم دیتای مورد ضرورت خویش را از اینترنت به دست آوریم.



شکل (۱-۴): مراحل دریافت اطلاعات از اینترنت

فعال ساختن یا راه اندازی یک وبسایت به این مراحل تقسیم می شود:

- دیزاین یک وبسایت با استفاده از تکنالوژی های: PHP, HTML, CSS؛
 - اجازه دامنه (Domain)؛
 - اجازه فضای Host برای Upload کردن وبسایت؛
 - معرفی به Searching Engineها جهت دسترسی به اطلاعات داخل وبسایت.
- پس از تکمیل نمودن طراحی یک وبسایت ضرور است تا آن را فعال یا راه اندازی کنید که به دسترس تمام استفاده کنندگان در سطح جهان قرار گیرد. برای راه اندازی وبسایت به مراحل ذیل ضرورت است:

۱.۴ اجازه فضا و Domain

برای به دست آوردن فضا و Domain ضرور است تا به یک شرکت ارائه دهنده خدمات یا Web Hosting مراجعه نموده، از آن ها اجازه فضا و Domain را در یک وبسرور بگیرید. هدف از Domain همان نام وبسایت است که استفاده کنندگان با وارد کردن آن در مرورگر وب می توانند به صفحه اصلی وبسایت دیزاین شده شما دسترسی پیدا کنند. به آدرس ذیل توجه نمایید:

<http://www> نام وبسایت، پسوند وبسایت

این آدرس می تواند، نام سایت حاوی حروف انگلیسی، عدد، علامت باشد اما کاربرد Underscore در آن مجاز نیست. پسوند وبسایت به حروف اختصاری نوشته می شود که نشان دهنده نوعیت یک وبسایت می باشد. این ها از جمله پسوندهایی اند که در وبسایت ها استفاده می شود:

- Com (Commercial)
- Af (Afghanistan)
- Gov (government)
- Edu (Education)
- Org (Organization)

پس از تعیین نام وبسایت یا Domain نوبت به اجازه Host می رسد. در وبسایت، شرکت هایی که به فضای مورد نیاز برای راه اندازی وبسایت اجازه می دهند، بستگی به حجم فضا و امکانات Host دارد. برای وبسایت هایی که تنها حاوی صفحات HTML هستند و قرار نیست که فایل دارای حجم زیاد بر روی Server ذخیره گردد، حجم 10 تا 20 میگابایت کافی است.

خلاصه‌ی فصل اول



این فصل، برای معرفی بعضی از مفاهیم ابتدایی صفحات وبسایت است، در این فصل بیشتر از مفاهیمی بحث شده است که برای آشنایی شما، قبل از شروع با PHP ضرور است.

اصول طراحی صفحات وب یک بحث اساسی برای برنامه‌نویسان وب می‌باشد تا بدانند که یک وبسایت را چگونه طراحی کنند؛ هم‌چنان ضروریات برای طراحی صفحات نیز از جمله مفاهیم اساسی است که باید قبل از شروع به طراحی وب، آن را شناسایی نماییم. وب‌سرور و وب‌هاستنگ نیز بخش‌های مهمی‌اند که در جریان طراحی و توسعه‌دادن وب به آن‌ها نیاز داریم.

سوالات فصل اول



۱. برای طراحی وبسایت، کدام مراحل ضروری است؟
۲. فرق دینامیک وبسایتها و استاتیک وبسایتها در چیست؟
۳. ضروریات اولیه طراحی وبسایت کدامها اند؟
۴. چگونه می توانیم اطلاعات را از اینترنت به دست آوریم؟
۵. فضا یا Domain چه مفهوم دارد؟
۶. آیا فضا یا Domain رایگان است یا خیر؟
۷. چگونه وبسایت خود را فعال بسازیم تا در هر جا قابل دسترس باشد؟
۸. آیا در وبسایتها از URLهای یکسان استفاده می شود یا از URLهای متفاوت؟
۹. دومین Domain چه مفهوم دارد؟
۱۰. چرا در وبسایتها از extensionها استفاده می شود؟
۱۱. چند تا از extensionهای مشهور را نام بگیرید.

فصل دوم

معرفی PHP



هدف کلی: محصلان با قابلیت‌های PHP و Apache server آشنا می‌شوند.

اهداف آموزشی: در پایان این فصل محصلان قادر خواهند شد تا:

۱. PHP و قابلیت‌های آن را تعریف کنند.
۲. Apache server را نصب و configuration کنند.
۳. ساختار پروگرام PHP را توضیح دهند.

در این فصل از قابلیت‌های PHP بحث می‌شود، هم‌چنان **web server** اپاچی نصب و فعال‌سازی می‌شود، در مورد ساختار پروگرام‌های زبان برنامه‌نویسی PHP نیز بحث صورت گرفته و در بارهٔ اقسام کمنت‌ها در [PHP] نیز وضاحت داده شده است.

به‌صورت عموم صفحات وب به دو نوع است:

- صفحات **Static** یا ثابت.
- صفحات **Dynamic** یا متحرک.

۲.۱ صفحات وب **static**:

تنها با زبان HTML ایجاد می‌شوند. محتوا و ساختار کلی این صفحات ثابت است و در آن تغییر نمی‌آید. استفاده‌کننده‌ها با هر بار مراجعه به این صفحات همان صفحات قبلی را می‌بینند.

۲.۲ صفحات وب **Dynamic**:

عبارت از آن صفحات وب است که محتویات و ساختار صفحات در آن قابل تغییر است. این صفحات به کمک تکنالوژی‌های متعددی مانند: ASP.net, ColdFusion, PHP به وجود می‌آید.

PHP مخفف Hypertext preprocessor است، یک لسان Script نویسی **Server side** یا سمت سرویس‌دهنده می‌باشد. عبارت "Preprocessor" به این مفهوم است که اطلاعات قبل از تبدیل شدن به زبان HTML توسط PHP پروسس می‌شود. بیشتر Syntaxها و دستورات PHP از زبان‌های C, JAVA و Perl گرفته شده است. توسط PHP صفحات وب متحرک یا **Dynamic Website** طراحی می‌شود. شیوه کار آن مانند ASP.net می‌باشد.

۲.۳ توانایی‌های PHP

PHP قادر است تمام کارهایی را که سایر برنامه‌های CGI (Common Gateway Interface) اجرا می‌کند، اجرا کند. یکی از قوی‌ترین توانمندی‌های PHP پشتیبانی از Databaseها می‌باشد. ایجاد یک صفحه ارتباطی یا **Link** با دیتابیس در PHP بسیار کار آسان است. یکی از عوامل شهرت زبان PHP همین پشتیبانی از دیتابیس است که باعث ایجاد وب‌سایت متحرک توسط PHP گردیده است.

دیتابیس‌هایی که توسط PHP پشتیبانی می‌شود:

جدول (۲-۱): نام دیتابیس‌های مشهور

Adabas D	Ingres	Oracle
D Base	Interbase	Ovrimos
Empress	Frontbase	Postgre SQL
Filepro	M SQL	Solid
Hyperwave	DirectMS-SQL	Sybase
IBM DB2	My SQL	Velocis
Informix	ODBC	Unix Dbm

PHP برای ارتباط با دیگر سرویس‌ها از Protocol های ذیل پشتیبانی می‌کند:

IMAP: (Internet Message Access Protocol)

SNMP: (Simple Network Management Protocol)

(News Transfer Protocol [NNTP: Network](#)

POP3: (Post Office Protocol 3)

HTTP: (Hyper Text Transfer Protocol)

توسط PHP می‌توانیم:

- صفحات دینامیک ایجاد نماییم.
- فایل‌ها را به طرف سرور ایجاد نموده، ویرایش و یا حذف نماییم.
- اطلاعات را در دیتابیس درج نموده، تغییر و یا حذف نماییم.
- معلومات درج شده در فورم را جمع‌آوری نماییم.
- حق دسترسی به صفحات وب را محدود سازیم.

۲.۴ تاریخچه PHP

در سال 1995 برای اولین بار شخصی بنام راسموس [لردورف] "Rasmus Lerdorf" از لسان PHP در نوشتن صفحه شخصی خودش در وب استفاده کرد که آن را به نام Personal Home page (Tools) یاد کرد. در

سال 1997 تعداد استفاده‌کنندگان از لسان PHP برای دیزاین وبسایت به 50 هزار رسید. در این زمان PHP از حالت یک پروژه شخصی بیرون شده، به یک پروژه عمومی مبدل گشت و توسط تیمی توسعه‌کودهای آن بازنویسی شد. امروز برآورد می‌شود که در دیزاین بیشتر از شش میلیون وبسایت از لسان PHP استفاده شده است.

دلایل استفاده از PHP:

- PHP سرعت بالا دارد.
- Open source است و طبق ضرورت، هرکس می‌تواند به آن تغییرات بیاورد و از آن استفاده نماید و به قسم رایگان بدسترس هرکس قرار می‌گیرد.
- Syntax آن مشابه به لسان پروگرام‌نویسی C می‌باشد.
- نوعیت دیتا و ساختار کدهای PHP به آسانی آموخته می‌شود.
- نیازی به دانستن دستور خاصی برای Compile برنامه ندارد.
- PHP [سرویسی] از مجموعه فایل‌های کتابخانه‌یی زبان برنامه C را ارائه می‌دهد.
- توانمندی بالا دارد و بر روی سیستم‌های عامل مختلف، مانند: ویندوز، لینوکس، مکینتاش و یونکس قابل استفاده است که این خاصیت را به نام Multi plat form یا plat form independent یاد می‌کنند.
- سطح امنیت آن بالا است.

آنچه را که با زبان‌های دیگری برنامه‌نویسی انجام داده می‌توانیم، PHP هم قادر به اجرای آن می‌باشد؛ اما کار با PHP نظر به دیگر لسان‌ها، مانند Perl نسبتاً آسان است. این لسان برای طراحی صفحات وب Dynamic استفاده می‌شود.

PHP لسانی است که با تمام سیستم‌عامل‌ها سازگاری دارد. این یکی از برجستگی‌های PHP در بین دیگر لسان‌های پروگرام‌نویسی می‌باشد. برخی از دیگر لسان‌ها و تکنولوژی‌ها، مانند ASP.net محدود به سیستم عامل Windows است و پشتیبانی از آن در دیگر سیستم‌عامل‌ها بسیار پر هزینه و محدود می‌باشد. این انحصار را شرکت مایکروسافت برای محصولش به‌وجود آورده است.

۲.۴.۱ PHP به حیث یک لسان Open Source

دسترسی به PHP برای همه سیستم‌عامل‌ها رایگان است. علاقمندان می‌توانند آخرین نسخه مفسر این لسان را از سایت رسمی PHP که در ذیل نقل شده است، Download نمایند:

<http://www.php.net>

بسیاری از Editorهای این لسان نیز رایگان هستند و یا با هزینه ناچیز می‌توان آن را تهیه کرد. در حالی که تهیه Platformهای دیگر لسان‌های پروگرام‌نویسی به هزینه هنگفت ضرورت دارد، مانند لسان پروگرام‌نویسی JAVA و غیره.

۲.۵ سرعت PHP

PHP یکی از سریع‌ترین زبان‌ها در نوع خود است. تفسیر و اجرای یک Script در PHP به‌طور متوسط چهار برابر یک Script در ASP.net می‌باشد.

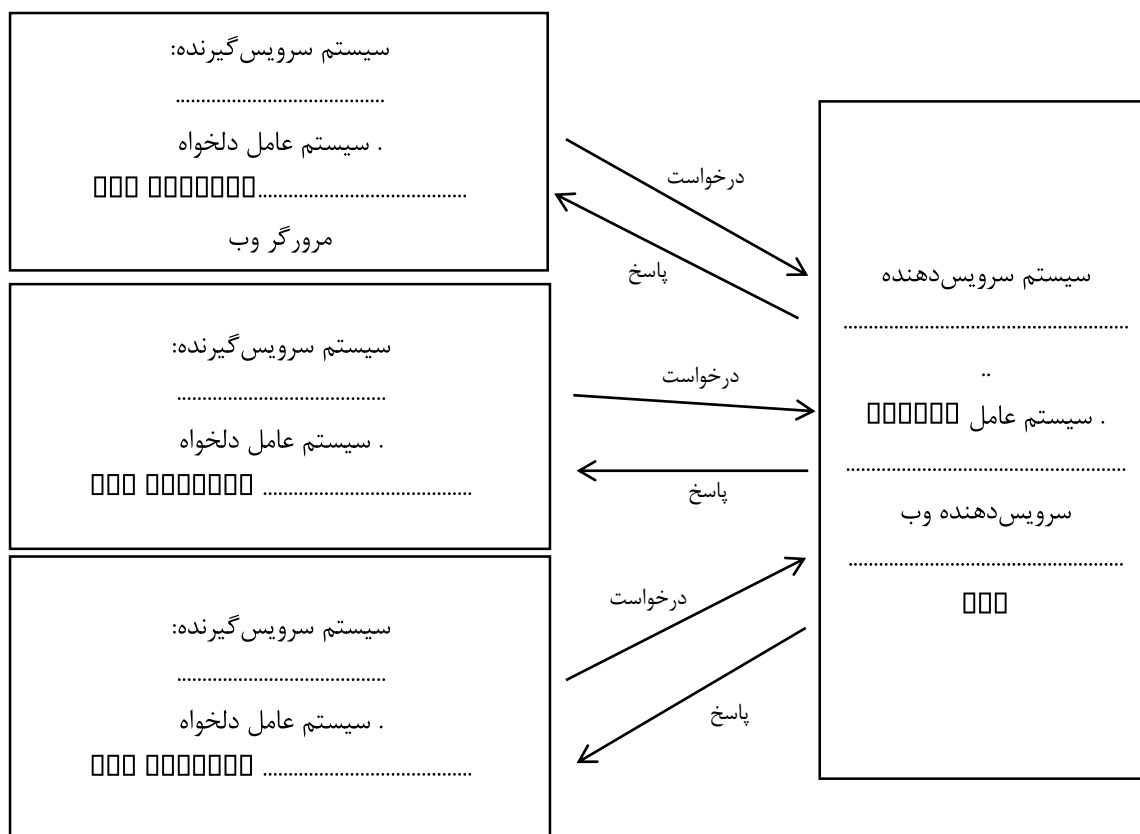
۲.۶ ساختار مناسب و امکانات بالا در PHP

در برنامه‌نویسی تحت وب، کمتر نیامندی‌هایی است که در PHP مرفوع نگردد، PHP شامل کتابخانه غنی از توابعی است که امکان پروسس اطلاعات، فورم‌ها، کار با دیتابیس، فایل‌های متنی و باینری، فایل‌های گرافیکی، PDF, ZIP و Protocolهای SMTP, FTP, DNS, TCP و غیره را برای برنامه‌نویس فراهم می‌کند.

۲.۷ ضروریات برای استفاده از PHP

- یک ویرایش‌گر متن "Text editor" مانند: Dream weaver, Notepad, Notepad++ Nusphere
- PHP Engine site, ضرورت است تا کدهای PHP را در آن بنویسید.
- یک پروگرام یا محیط که کدهای PHP بتوانند در آن اجرا شوند، چون PHP یک لسان پروگرام‌نویسی Server side است، ضرورت است که روی یک Server اجرا گردد. این کار را می‌توان از طریق نرم‌افزارهای Easy PHP, WAMPP, XAMPP اجرا نمود. این نرم‌افزارها به عنوان Server مجازی ایفای وظیفه می‌کنند. قبل از شروع کار با PHP یکی از این نرم‌افزارها را باید در سیستم نصب نمود.
- چندین مرورگر وب یا Web browser را در سیستم نصب نمایید تا در جریان کار از همه آنها جهت درست بودن صفحه وب خود استفاده کنید. مرورگرهای معروف و رایج مانند: Opera, Safari, Internet explorer, Mozilla Firefox, Net scape می‌باشند.
- قابل ذکر است که PHP نیاز به فضای زیاد حافظه اصلی RAM و HDD ندارد.

نوت: در صورتی که PHP را در یک شبکه کار می‌کنید، سیستم عامل Server و نرم‌افزار PHP را در یک سیستم نصب نموده، از این سیستم به عنوان Server سرویس‌دهنده استفاده کنید؛ اما مرورگرها را به سیستم‌های Client یا سرویس‌گیرنده نصب نموده از طریق آنها نتایج را مشاهده کنید.



دیاگرام (۱-۲): سروس های کامپیوتر سرور

توجه: این دیاگرام نحوه قرارگیری سیستم Server یا سرویس دهنده و سیستم های Client سرویس گیرنده را نشان می دهد:

نکته: بیشتر سایت های تجاری برای این که به درخواست های متعدد استفاده کنندگان پاسخگو باشند، به جای یک سیستم Server یا سرویس دهنده، مجموعه ای از سرویس دهنده ها را به کار می برند. این تکنیک Clustering نام دارد و در آن مجموعه ای از سیستم ها به عنوان یک سرویس دهنده واحد عمل کرده، درخواست های استفاده کنندگان را پروسس می کند.

آیا PHP به یک محیط خاص جهت کدنویسی ضرورت دارد یا خیر؟

نخیر، به محیط خاصی ضرورت ندارد، می توان کد آن را در Notepad یا Notepad++ نوشت اما محیط اجرای کد آن باید روی یک Server باشد. در غیر آن قابل اجرا نمی باشد. با نرم افزارهای WAMPP and XAMPP می توان یک Server را در سیستم شبیه سازی کرد.

۲.۸ نصب و فعال سازی apache server

برای نصب و فعال سازی apache server ضرورت است تا XAMPP را در کامپیوتر نصب نماییم.

XAMPP ۲.۸.۱

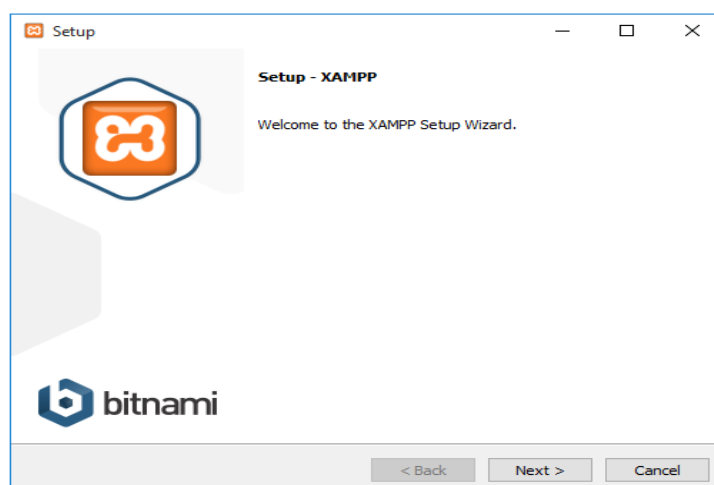
نرم افزار XAMPP یک سرور مجازی برای ساختن Local host بر روی کامپیوتر است. این نرم افزار برای سیستم عامل های ویندوز، لینوکس و مکینتاش نسخه های مختلفی دارد که می توانید از وبسایت apache آن را رایگان دانلود نمایید. این نرم افزار بستر یک Hosting را در کامپیوتر شما شبیه سازی خواهد کرد و می توانید کارهایتان را پیش از آنلاین کردن، به صورت آفلاین به پیش ببرید.

برای دانلود کردن XAMPP به این وبسایت مراجعه کنید:

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

بعد از دانلود شدن XAMPP روی آیکن آن کلیک کنید، تا پروسه نصب آغاز شود.

مرحله اول

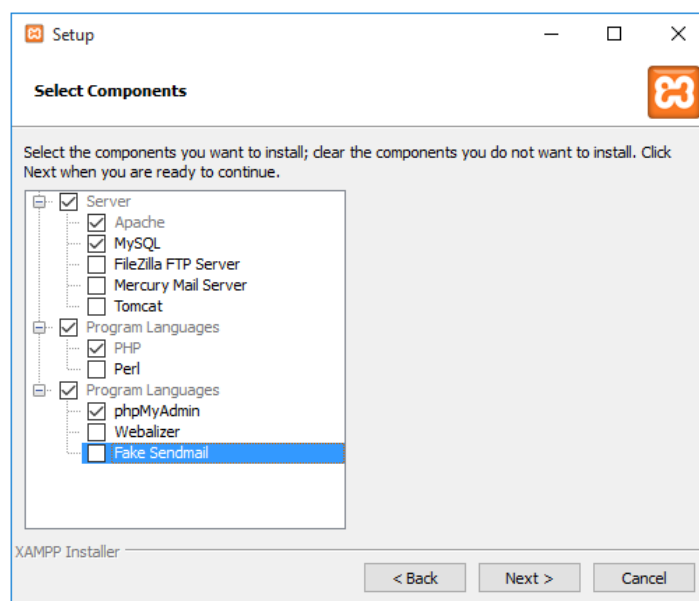


شکل (۱-۲)

مرحله دوم

در این مرحله از شما سؤال می شود که کدام قابلیت های نرم افزار را می خواهید نصب کنید. بعضی گزینه ها، مانند Apache و PHP برای اجرایی شدن نرم افزار ضروری هستند و به صورت پیش فرض نصب می شوند. بنابر این، آن ها به رنگ خاکستری هستند و نمی توانید آن ها را غیرفعال سازید.

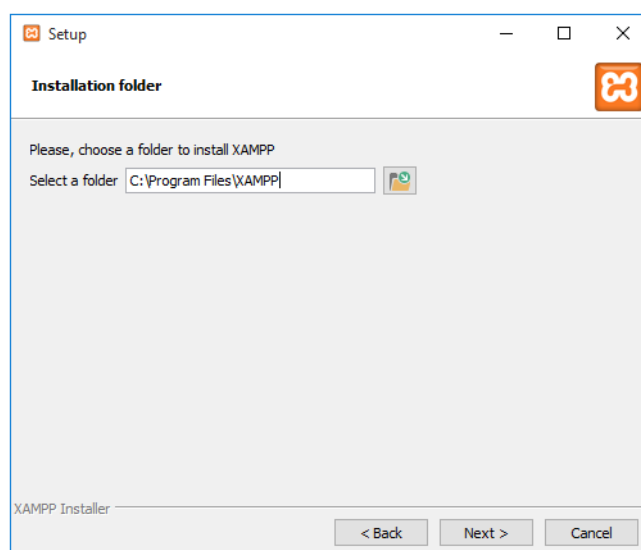
بخش بعدی به شما مربوط می شود، هر کدام از گزینه ها را که نیاز دارید انتخاب کنید. در صورتی که هدف شما از نصب XAMPP اجرای CMS هایی نظیر WordPress یا Joomla می باشد، تیک گزینه های MySQL و phpMyAdmin را فعال کنید و تیک مارک گزینه ها را بردارید.



شکل (۲-۲)

مرحله سوم

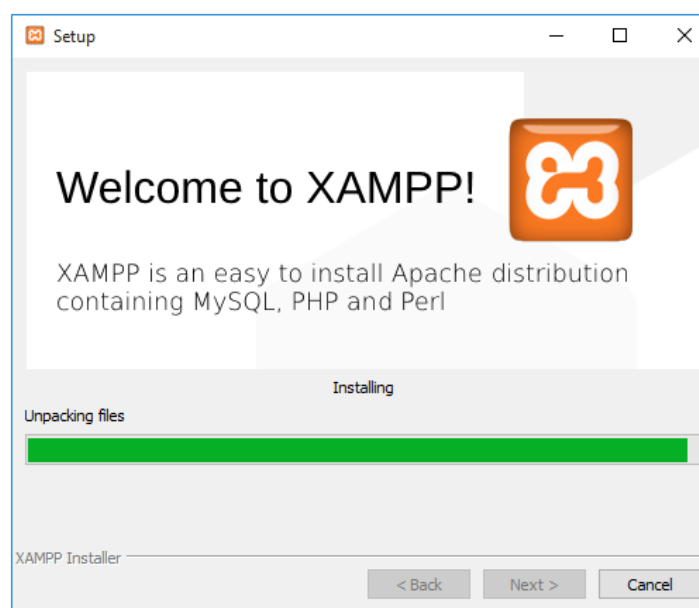
در این مرحله فولدر که می‌خواهید XAMPP در آن نصب شود را انتخاب کنید. من حالا آدرس **C:\Program Files\XAMPP** را انتخاب کردم اما بهترین آدرس همان **C:\XAMPP** می‌باشد چرا که در صورت فعال بودن UAC ویندوز، باز هم اشتباه دریافت می‌کنید. افزون بر این، در هنگام کار بر روی پروژه‌ها، به دلایل امنیتی، در صورتی که در درایو C و زیر پوشه‌هایی نظیر Program Files یا Windows یا حتی Users را برای نصب انتخاب کنید، به مشکلاتی بر خواهید خورد و این مشکلات می‌تواند باعث شود که حتی نتوانید، تغییرات را در فایل‌های مورد نظرتان ایجاد کنید.



شکل (۳-۲)

مرحله چهارم

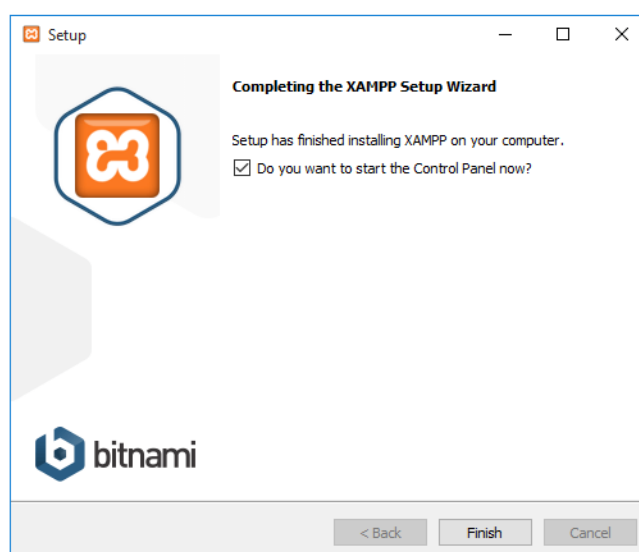
در این مرحله XAMPP آماده به نصب است، با کلیک کردن روی دکمه next پروسه نصب آغاز می شود.



شکل (۲-۴)

مرحله آخر

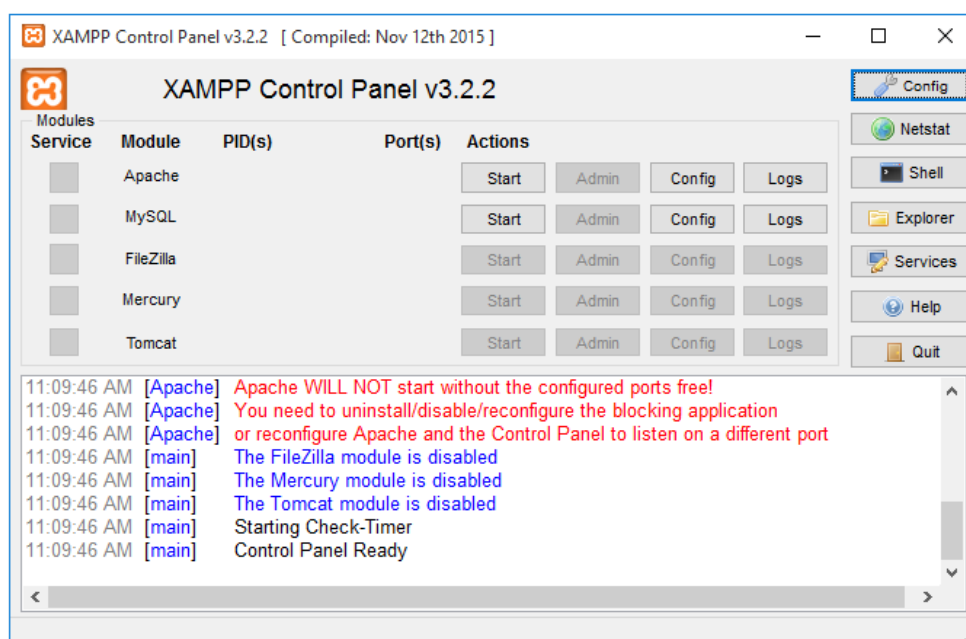
پروسه نصب تکمیل و در ویندوز باز شده، روی دکمه Finish کلیک کنید تا بسته شود.



شکل (۲-۵)

بعد از ختم پروسه نصب کنترل پنل XAMPP به شکل اتوماتیک باز می شود.

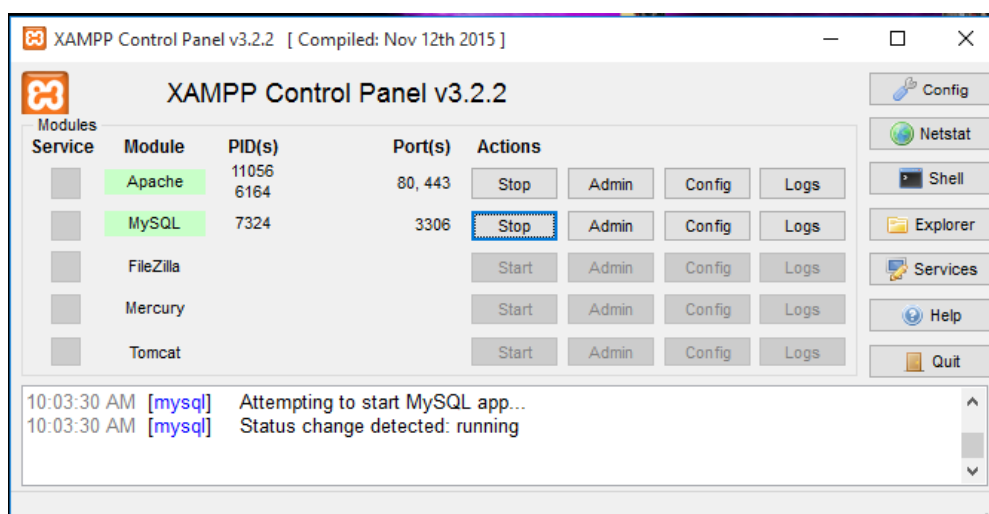
در صورتی که باز نشود به فولدر XAMPP رفته روی XAMPP Control Panel کلیک کنید تا باز شود. اگر نصب با موفقیت و کامل انجام شده باشد، کنترل پنل با متن‌های آبی و مشکی که در حال به‌روز شدن (Update) است، باز می‌شود و اگر مشکل داشته باشد پیام خطا ظاهر می‌گردد.



شکل (۶-۲)

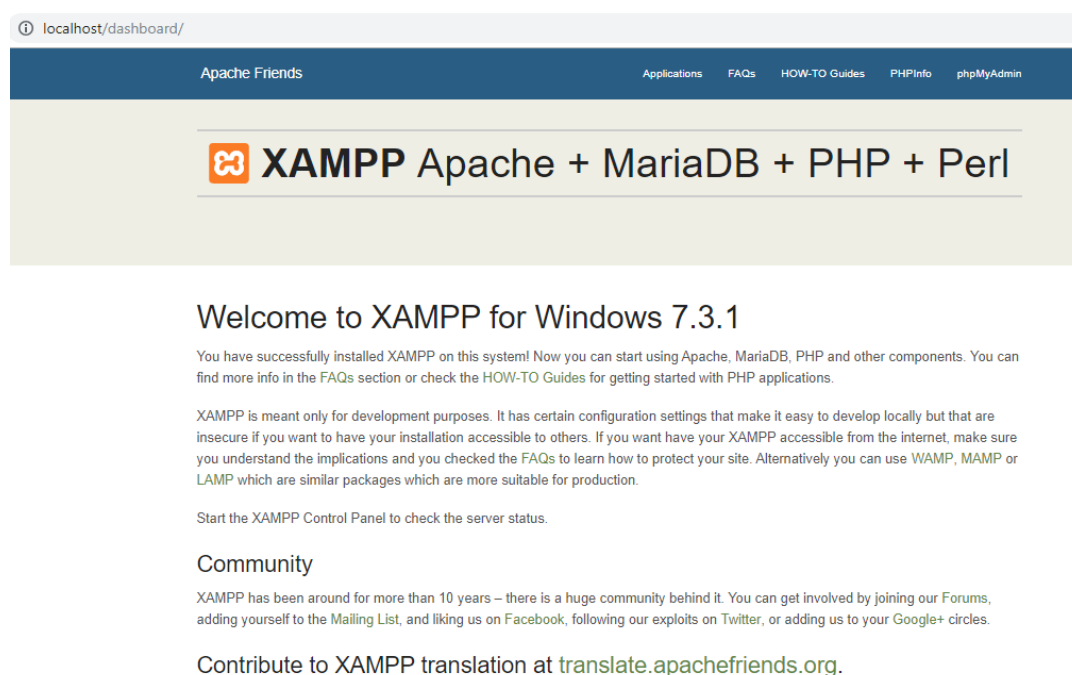
اجراء کردن XAMPP بر روی ویندوز

بعد از باز شدن پنجره کنترل پنل XAMPP روی گزینه‌های Apache و MySQL کلیک کنید تا فعال شود.



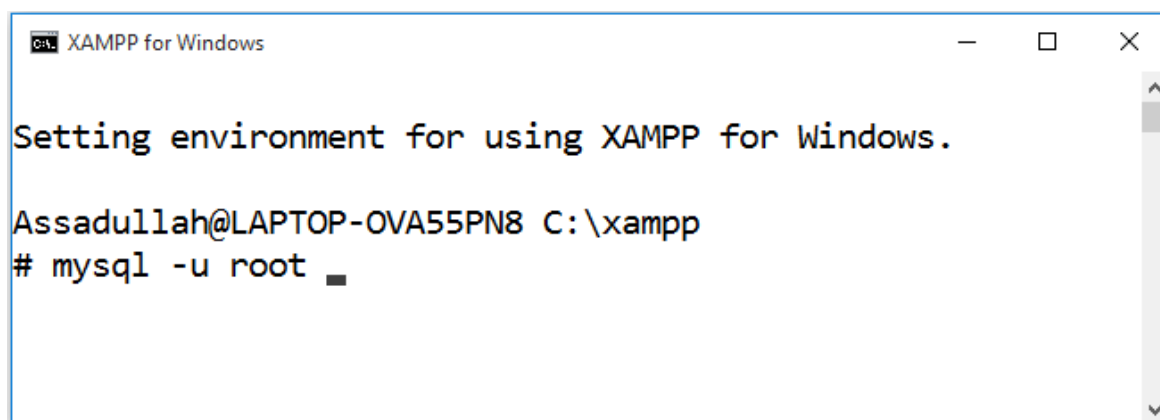
شکل (۷-۲)

برای تست کردن در browser این آدرس را بنویسید: `http://localhost` که در صورتی که Apache server فعال باشد، ویندوی زیر باز می‌شود:



شکل (۸-۲)

نوت: سمت راست کنترل پنل xampp کلیدی به نام shell است، وقتی بالای آن کلیک کنید وارد دیتابیس MySQL می‌شوید.



شکل (۹-۲)

با نوشتن `mysql -u root` و کلیک کردن `enter` وارد دیتابیس MySQL می‌شوید.
 -u: عبارت از user است. Root صورت حساب پیش فرض دیتابیس MySQL می‌باشد.

```

mysql -u root

Setting environment for using XAMPP for Windows.

Assadullah@LAPTOP-OVA55PN8 C:\xampp
# mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.

Your MySQL connection id is 12
Server version: 5.1.41 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

شکل (۲-۱۰)

بعد از نصب و فعال سازی apache server حالا برای آشنایی بیشتر، به نوشتن بعضی اسکریپت های کوچک PHP می پردازیم.

۲.۹ نوشتن کدهای PHP

در Script های PHP کدهایی را که می نویسم، باید در داخل Tag ها قرار دهیم. زمانی که PHP اسکریپت را اجرا می کند، به دنبال Tag های مشخص کننده کدهای PHP می شود و به کمک این Tag ها کدهای PHP را از سایر کدها تشخیص می دهد. به این ترتیب امکان استفاده از کدهای PHP در اسناد مختلف فراهم می شود. کدهای PHP را به سه روش در بین Tag های PHP می توان نوشت:

روش اول:

```
<? PHP code ?>
```

روش دوم:

```
<? PHP PHP code ?>
```

روش سوم:

```
<Script Language="PHP"> PHP code </script>
```

به مثال‌های ذیل توجه کنید:



برای نشان‌دادن نتایج کدهای PHP از این دو تابع استفاده می‌شود:

- .Echo
- .Print

هر دو تابع برای نشان‌دادن نتیجه، استفاده می‌شود؛ اما تابع Echo بیشتر، چون سرعت اجرای آن نسبت به تابع Print زیادتر می‌باشد.

```
<?php
echo ("welcome the first PHP program!!")
?>
```

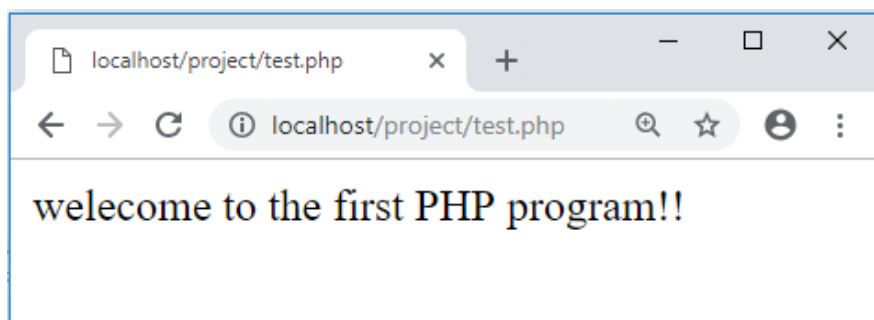
این کد را در داخل یک ادیتور نوشته، سپس آن را در درایف C داخل فولدر xampp/htdocs به نام text.php ذخیره کنید.

بعد یک Web browser را باز نموده، این آدرس را در Address bar وارد نموده، کلید Enter را بفشارید:

<http://localhost/test.php>

و یا این آدرس را وارد کنید: <http://127.0.0.1/test.php>:

بعد از فشردن کلید Enter، در صفحه مرورگر، نتیجه را به این شکل مشاهده خواهید کرد:



شکل (۲-۱۱)

کودهای PHP را می‌توان با کدهای HTML به قسم ترکیبی نیز استفاده کرد:

```

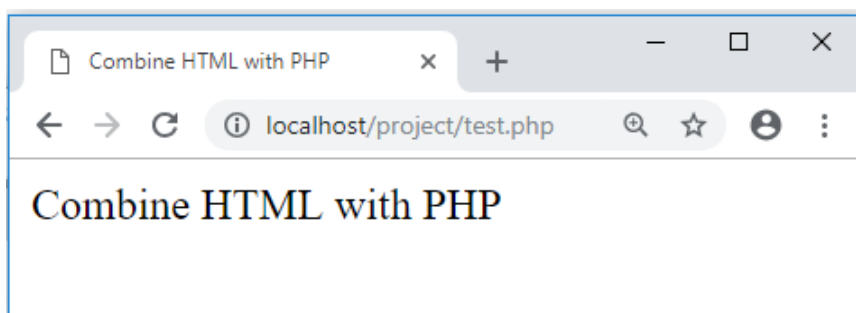
<!DOCTYPE html>
<html>
<body>
<?php

echo ("Combine HTML with PHP")

?>
</body>
</html>

```

آدرس را در Address bar نوشته، بعد نتیجه را به این شکل مشاهده خواهید کرد.



شکل (۲-۱۲)

اگر از فنکشن‌های Print و echo استفاده کنیم در خروجی یا نتیجه، کدام تغییر مشاهده نمی‌شود؛ فقط از لحاظ سرعت، این دو فنکشن باهم متفاوت است.

```

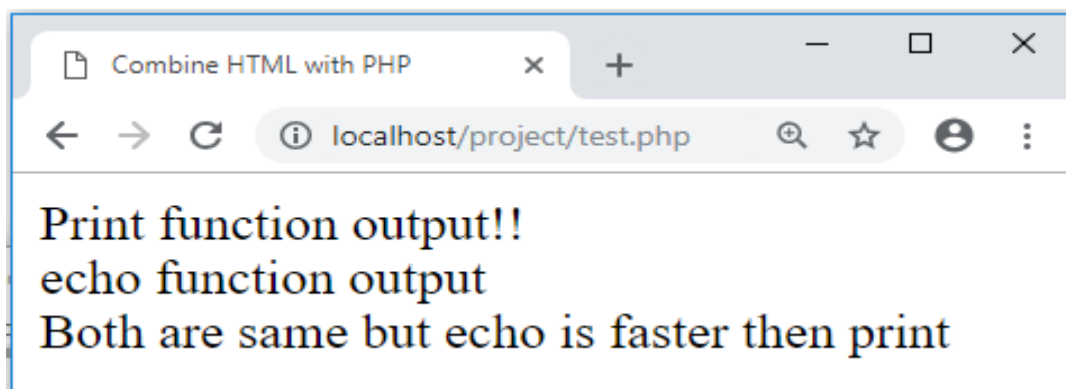
<!DOCTYPE html>
<html>
<body>

<?php print("print function output!!") ?> </br>
<?php echo ("echo function output") ?> </br>
<?php print("Both are same but echo is faster then print") ?>

</body>
</html>

```

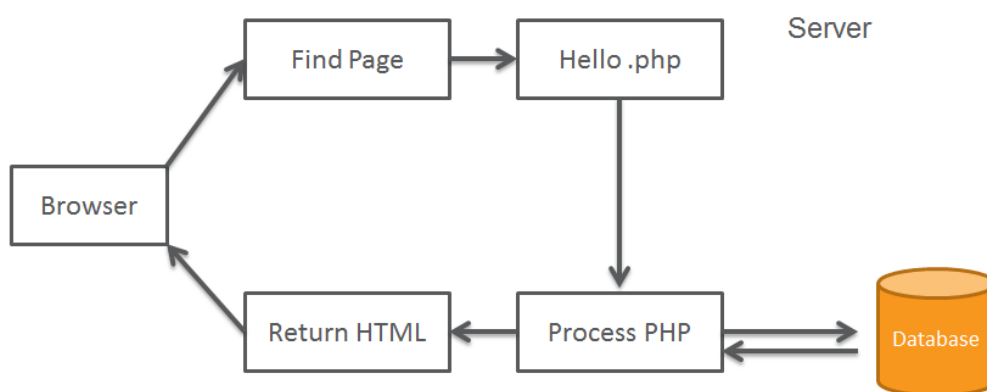
خروجی:



شکل (۲-۱۳)

۲.۱۰ مسیر عملیاتی کدهای PHP

مسیر عملیاتی و اجرای یک فایل PHP را در دیاگرام ذیل مشاهده کنید. این دیاگرام را همیشه به یاد داشته باشید تا تصور مسیر عملیاتی فایل های PHP برای تان آسان باشد.



شکل (۲-۱۴)

۲.۱۱ توضیحات (Comments)

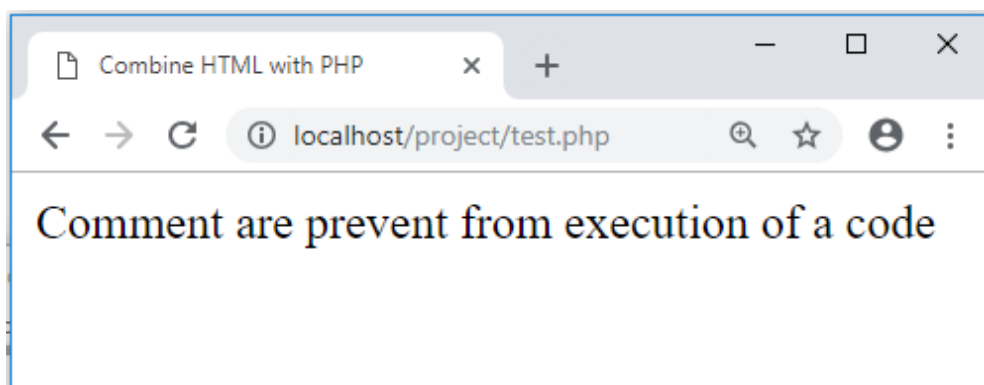
توضیحات یا Comment در تمام لسان‌های پروگرام‌نویسی کاربرد زیاد دارد. از Comments وقتی استفاده می‌شود که از اجرای یک قسمت پروگرام «پروگرامر» صرف نظر نماید و یا بخواهد یک یادداشت یا نوت اضافه نماید. وقتی که یک کود در صفحه Code editor به Comment تبدیل شود، دیگر قابل اجرا نیست و مفسر لسان پروگرام‌نویسی آن را ترجمه نمی‌کند. با این سمبول‌ها می‌توان یک کود را به Comment تبدیل نمود:

```
//.....Double forward slashes
#.....Shorp
/*.....Block comment. */
```

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

<?php // print("// comment will not execute") ?> </br>
<?php # echo ("# comment symbole") ?> </br>
<?php print("Comment are prevent from execution of a code") ?>
</body>
</html>
```

خروجی:



شکل (۲-۱۵)

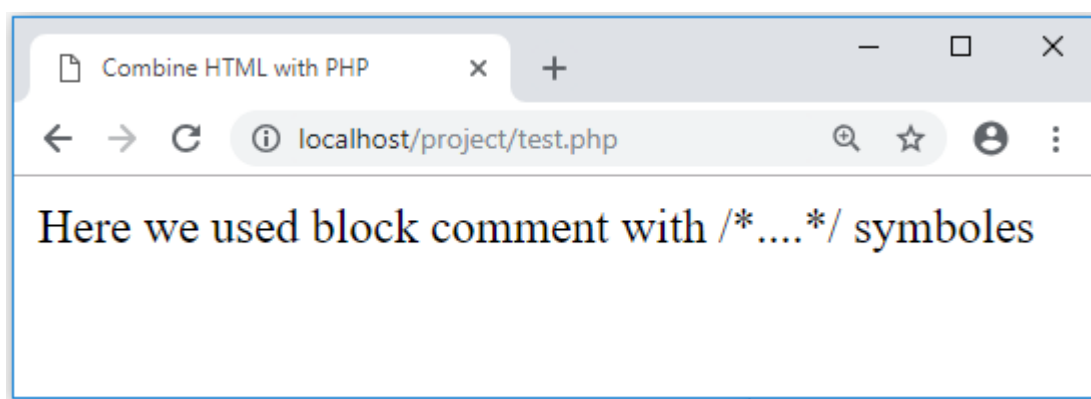
بین سمبول‌های "///" و "# " تفاوت وجود ندارد؛ هر دو وظیفه مشابه را انجام می‌دهند؛ اما معمولاً از سمبول "///" برای Comments ساختن بیشتر استفاده می‌شود. از سمبول /*.....*/ در حالتی استفاده می‌شود که بیشتر از یک سطر کود را به قسم یک ساحه به Comment تبدیل کنیم؛ از همین جهت این نوع Comment را به نام Block comment یاد می‌کند.

مثال بلاک کومنت:

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

<?php /* print("// comment will not execute") ?> </br>
<?php echo ("# comment symbole")  */?>    </br>
<?php print("Here we used block comment with /*...*/ symboles") ?>
</body>
</html>
```

خروجی در مرورگر وب:



شکل (۱۶-۲)

خلاصه‌ی فصل دوم



در این فصل با نصب XAMPP توانستیم که Apache server را در کامپیوتر خود شبیه‌سازی کرده، فعال‌سازیم. هم‌چنان با xampp دیتابیس MySQL نیز در کامپیوتر نصب شده است و در وقت ضرورت می‌توانیم از آن استفاده کنیم.

بعد از نصب و فعال‌سازی، بعضی از کدهای PHP نوشته و اجرا شده است. در مورد اقسام نظرات که یک بخش مهم در تمام زبان‌های برنامه‌نویسی است، بحث صورت گرفته و هر کدام آن‌ها با مثال توضیح داده شده است.



سوالات و فعالیت های فصل دوم

۱. ساختار پروگرام PHP را توضیح دهید.
۲. تگ های PHP کدام ها اند، شرح دهید.
۳. PHP مخفف چیست؟
۴. PHP در کدام سال و توسط کدام شخص معرفی شد؟
۵. ساختار پروگرام PHP به کدام زبان برنامه نویسی شباهت دارد.
۶. Parser چیست؟
۷. تفاوت بین زبان های برنامه نویسی وب و زبان های برنامه نویسی را شرح دهید.

فعالیت ها

- نرم افزار xampp را در کمپیوترهای تان نصب کنید.
- apache server را فعال کنید.
- MS SQL دیتابیس را فعال کرده، سپس آن را باز کنید.
- HTDOCS فولدر را توضیح دهید.
- پروگرام PHP را بنویسید که توسط آن Hello world در مرورگر وب نشان داده شود.
- تفاوت فنکشن های print, echo را در PHP شرح دهید.
- اقسام Comment ها را تشریح کرده، پروگرام هایی را بنویسید که در آن ها از Comment ها استفاده شود.

فعالیت گروهی

- مسیر عملیاتی کودهای PHP را در یک دیاگرام رسم کرده، باهم مباحثه کنید.
- در باره مفهوم client side and server side باهم بحث کنید.
- Web server های معروف جهان را نام گرفته، در باره تفاوت های آن ها، باهم گفتگو کنید.
- چرا کودهای PHP را در فولدر htdocs ذخیره می کنیم.

فصل سوم

نوعیت دیتا (Data type)



هدف کلی: محصلان در این فصل با متحول‌ها، عملگرها، شرط‌ها و array آشنا خواهند شد.

اهداف آموزشی: در پایان این فصل محصلان قادر خواهند شد تا:

۱. PHP variable را توضیح دهند.
۲. PHP Data type را تشریح کنند.
۳. PHP Operators را شرح دهند.
۴. PHP condition را تشریح کنند.
۵. PHP Switch را توضیح دهند.
۶. PHP Loop را تشریح کنند.
۷. PHP Array را شرح دهند.

در این فصل در مورد متحول‌ها، دیتاتایپ، عملگرهای کاربردی در PHP، شرط‌ها، حلقه‌ها و صف، بحث می‌شود. این فصل حاوی مطالب ابتدایی زبان برنامه‌نویسی PHP است. به محصلانی که می‌خواهند PHP را به صورت درست یاد بگیرند، توصیه می‌شود تا این فصل را به دقت خوانده، پروگرام‌های آن را تمرین کنند.

۳.۱ متحول‌ها (Variables)

برای تبدیل کردن صفحات وب ثابت (Static) به صفحات وب متحرک (Dynamic) و سایت‌های جذاب و زیبا، در ابتدا ضرورت است تا اطلاعات را در اختیار خود قرار دهید. متحول‌ها، عبارت از ابزاری است که شما با استفاده از آن‌ها می‌توانید، اطلاعات را در اختیار بگیرید. متحول‌ها یکی از مهمترین ابزارها و مفاهیم در هر زبان برنامه‌نویسی به شمار می‌رود. متحول‌ها به شما اجازه می‌دهد که اطلاعات را به طور موقت در حافظه اصلی سیستم ذخیره کنید و یا آن‌ها را تغییر دهید. از مسایلی که مورد توجه متحول‌ها قرار می‌گیرد، می‌توان به درک مفهوم متحول‌ها، انواع مختلف آن که توسط لسان‌های پروگرام‌نویسی پشتیبانی می‌شود، رسید و هم‌چنان به چگونگی استفاده آن پی برد. در لسان‌های مختلف پروگرام‌نویسی متحول‌ها به اشکال مختلف تعریف می‌شود.

۳.۱.۱ شرایط تعریف متحول‌ها در PHP

- متحول در PHP با سمبول "\$" شروع می‌شود.
- بعد از سمبول "\$" نام متحول نوشته می‌شود.
- نام متحول باید با یک حرف الفبا و یا اندرסקور شروع شود.
- نام متحول با اعداد آغاز نمی‌شود.
- نام متحول می‌تواند شامل حرف‌های الفبا، اعداد و اندرסקور باشد.
- نام متحول در PHP case sensitive می‌باشد و در برابر حروف بزرگ و کوچک حساس است.

متحول‌ها در تمام لسان‌های پروگرام‌نویسی مورد استفاده وسیع دارد. دانستن آن‌ها به شکل درست برای یک پروگرام‌نویس ضروری می‌باشد، با تعریف کردن یک متحول در تمام لسان‌های پروگرام‌نویسی یک قسمت از حافظه موقتی سیستم را ریزرف می‌کنیم تا آن پروگرام در آن موقعیت به صورت موقت ذخیره گردد. در بسیاری از لسان‌های پروگرام‌نویسی Data type در متحول‌ها نیز ضروری است؛ اما در PHP مشخص کردن نوعیت دیتا در متحول‌ها ضروری نیست. نوع اولین مقداری که وارد می‌شود به عنوان نوعیت دیتای متحول شناخته می‌شود. کاربرد متحول‌ها در لسان پروگرام‌نویسی PHP خیلی وسیع بوده، در یادگیری آن دقت بیشتر نیاز است. وقتی که از PHP در ساختن Interfaceها استفاده می‌شود، نقش متحول خیلی برجسته است. دانستن متحول‌ها به شکل درست آن کمک می‌نماید تا از ارتباط PHP با دیگر DBMSها استفاده شود. متحول‌ها در حقیقت نمایندگان مقادیری است که هر متحول به خود یک مقدار را اختیار می‌کنند.

مثال:

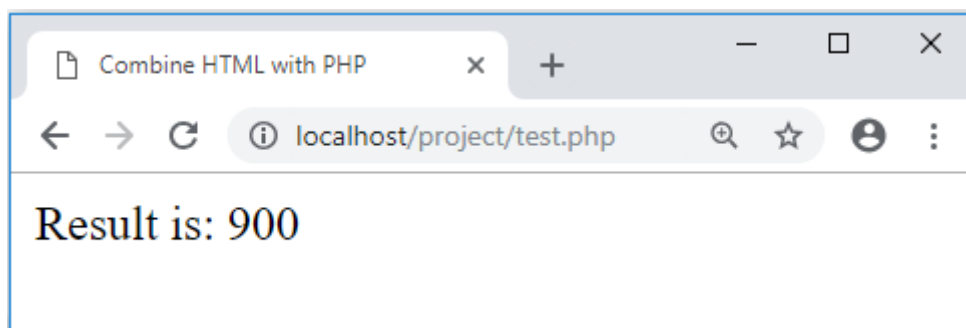
پروگرامی را بنویسید که در آن متحول‌ها تعریف شود و در داخل آن‌ها مقادیر ذخیره شوند.

```
<?php
$firstNo=100;
$secondNo=800;
$result=$firstNo + $secondNo;

echo "Result is: $result";

?>
```

بعد با کمک مرورگر وب، برنامه اجرا شده، نتیجه‌ای به شکل ذیل نشان داده می‌شود:



شکل (۱-۳)

تشریح پروگرام

\$firstNo: عبارت از متحولی است که با سمبول "\$" شروع می‌شود و نام متحول firstNo می‌باشد. متحول‌ها قادر به ذخیره‌سازی اطلاعات به صورت موقت در حافظه اصلی کامپیوتر است. این متحول عدد 100 را به صورت موقت در حافظه کامپیوتر ذخیره می‌کند.

\$secondNo: عبارت از متحول دومی است که با سمبول "\$" شروع شده و secondNo نام متحول است. در این متحول عدد 800 به طور موقت در حافظه کامپیوتر ذخیره شده است.

\$result: عبارت از متحول سومی است که با سمبول "\$" شروع شده و result نام متحول می‌باشد. این متحول نتیجه حاصل جمع عدد 100 را با عدد 800 به طور موقت در داخل حافظه اصلی کامپیوتر ذخیره می‌کند.

برای این که مقادیر را از داخل حافظه اصلی کامپیوتر در صفحه کامپیوتر مشاهده کنیم، از فنکشن های PHP که برای خروجی مقادیر از داخل حافظه کامپیوتر استفاده می شود، کار می گیریم.

Echo: عبارت از فنکشنی است که مقادیر را از داخل حافظه اصلی کامپیوتر به صفحه کامپیوتر نشان می دهد.

در نتیجه می بینیم که عدد 100 در متحول اول و عدد 800 در متحول دوم ذخیره است. برای این که حاصل جمع آن ها را در صفحه مرورگر ببینیم، توسط فنکشن echo متحول \$result را چاپ می کنیم. پس از اجرای پروگرام در مرورگر وب عدد 900 را که از حاصل جمع 100 و 800 می باشد مشاهده می کنیم.

در صورتی که بخواهید عملیات حسابی دیگر (تفریق، تقسیم، ضرب) را نیز انجام دهید، جای علامت ها را تبدیل کنید، عملیات اجرا می شود.

مثال هایی از شرایط نام گذاری متحول ها:

\$variable.....wright

\$Variable.....wrong

\$MyVariable.....wrong

\$this_variable.....wright

\$this-variable.....wrong

\$variable34.....wright

\$_book.....wright

\$__book.....wrong

\$4book.....wrong

\$variable: این طریقه، برای نام گذاری متحول درست است. می توان متحول ها را به این شکل نام گذاری کرد.

\$Variable: این هم یک متحول است؛ اما در برخی اوقات نام گذاری با حروف بزرگ به مشکل مواجه می شود؛ یعنی نام گذاری با حروف بزرگ و کوچک در PHP متفاوت است این خاصیت را به نام Case Sensitive یاد می کنند.

\$MyVariable: در صورتی که متحول متشکل از دو کلمه باشد، به این طریق نام گذاری درست نیست.

\$this_variable: در صورتی که نام متحول دو کلمه باشد، این طریقه نام گذاری درست است.

\$this-variable: در صورتی که نام متحول بیشتر از یک کلمه باشد، نام گذاری به این طریق درست نیست.

\$variable34: در صورتی که نام متحول متشکل از حروف و اعداد باشد، به این شکل نام گذاری درست است که اول کلمات و سپس اعداد نوشته می شود.

34variable: در صورتی که نام متحول متشکل از اعداد و حروف باشد، نوشتن اعداد در اول نام متحول درست نیست.

\$_book: در صورتی که با نام متحول Underscore هم موجود باشد، نوشتن آن در اول نام درست نیست.

\$_Book: در صورتی که دو تا Underscore را پشت سرهم در اول نام قرار دهید، درست نیست.

هنگام نام گذاری متحول ها، این قوانین را مراعات کنید، زیرا در صورت عدم رعایت آن ها به مشکل مواجه خواهید شد.

۳.۲ نوعیت دیتا (Data type)

مقادیری که در داخل حافظه کمپیوتر به شکل موقت به کمک یک متحول ذخیره می شود، نوعیت آن در زبان های برنامه نویسی دارای اهمیت خاص می باشد. برنامه نویسان باید مشخص کنند که چه نوع دیتا در داخل متحول تعریف شود تا در داخل حافظه کمپیوتر ذخیره گردد.

در PHP دیتاتایپ های زیر قابل استفاده می باشد:

String؛

Integer؛

Float (floating point numbers - also called double)؛

Boolean؛

Array؛

Object؛

NULL.

۳.۲.۱ دیتاتایپ String

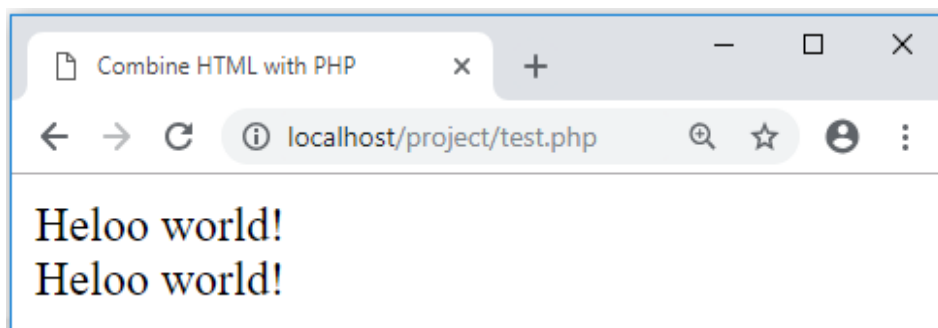
عبارت از دیتاتایپی است که گروهی از کرکترها را در داخل یک متحول ذخیره می‌کند. هر وقت که دیتاتایپ String تعریف شود، ضرورت است تا کرکترها را در داخل سینگل کوتیشن '' یا دبل کوتیشن "" نوشته شود.

مثال:

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

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

خروجی در مرورگر وب:



شکل (۲-۳)

تشریح پروگرام

:\$X عبارت از متحولی است که با سمبول \$ شروع می‌شود و نام متحول X می‌باشد. در داخل این متحول کرکترهای Hello word! به کمک دبل کوتیشن "" ذخیره می‌شود.

:\$Y عبارت از متحولی است که با سمبول \$ شروع می‌شود و نام متحول y می‌باشد. در داخل این متحول کرکترهای Hello word! به کمک سینگل کوتیشن '' ذخیره گردیده است.

برای مشاهده این کرکترها در داخل متحول از فنکشن echo استفاده می‌شود.

۳.۲.۲ دیتاتایپ integer

عبارت از دیتاتایپی است که توسط آن ارقام اعداد تام در داخل متحول ذخیره می‌شود.

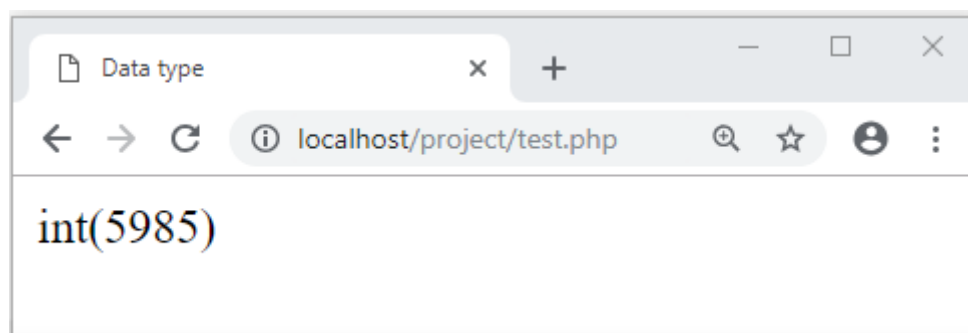
قوانین تعریف دیتاتایپ integer:

- integer باید حد اقل دارای یک رقم باشد.
- integer رقم اعشاری در آن استفاده نمی شود.
- Integer می تواند با ارقام مثبت و منفی تعریف شود.
- Integer می تواند به سه فارمت نوشته شود: دسیمیل به قاعده 10، هکذا دسیمیل به قاعده 16، اکتل به قاعده 8.

مثال:

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

خروجی توسط مرورگر وب:



شکل (۳-۳)

تشریح پروگرام

$\$X$: عبارت از متحولی است که با سمبول $\$$ شروع شده و نام متحول X می باشد. در این متحول عدد 5985 ذخیره شده است.

$\text{Var_dump}(\$x)$: عبارت از یک فنکشن است که محتویات متحول را خوانده، با دیتا تایپ در صفحه نشان می دهد.

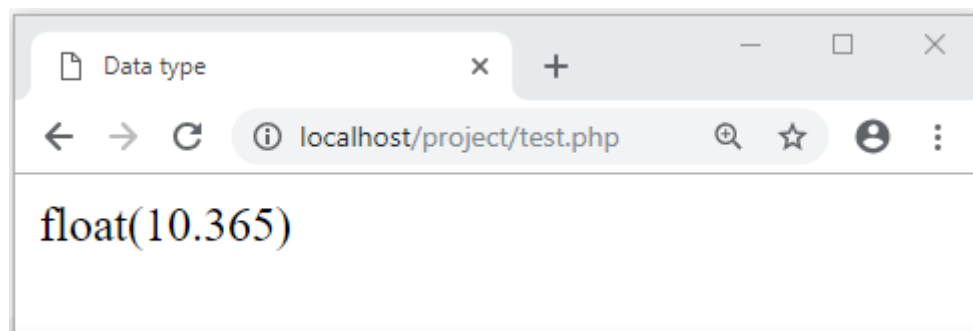
۳.۲.۳ دیتاتایپ float

عبارت از دیتاتایپی است که اعداد اعشاری را در داخل متحول ذخیره می کند.

مثال:

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

خروجی توسط مرورگر وب:



شکل (۳-۴)

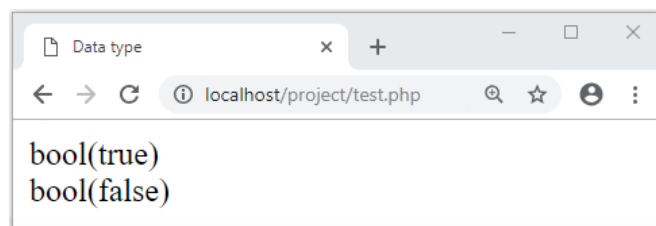
۳.۲.۴ دیتاتایپ Boolean

عبارت از دیتاتایپی است که دو حالت را نشان می دهد: True, False

از دیتاتایپ Boolean معمولاً برای بررسی شرایط استفاده می شود.

```
<?php
$x = true;
$y = false;
var_dump($x);
echo "<br>";
var_dump($y);
?>
```

خروجی توسط مرورگر وب:



شکل (۳-۵)

۳.۲.۵ دیتاتایپ array

عبارت از دیتاتایپی است که چندین مقدار را در یک متحول ذخیره می‌کند.

```
<?php
$city = array("Kabul", "Herat", "Kandahar", "Mazar");
var_dump($city);
?>
```

خروجی توسط مرورگر وب:



شکل (۳-۶)

۳.۲.۶ دیتاتایپ Object

عبارت از دیتاتایپی است که اطلاعات و معلوماتی را ذخیره می‌کند که چگونگی پروسس آن‌ها مورد بحث باشد.

در PHP یک اوجکت باید توسط برنامه‌نویس تعریف شود.

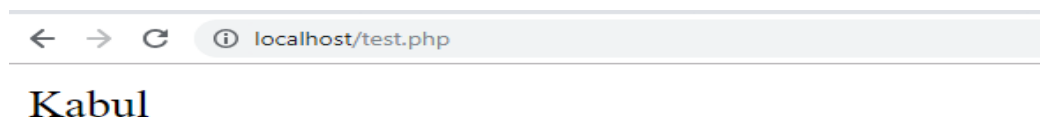
مثال:

```
<?php
class City {
    function City() {
        $this->city = "Kabul";
    }
}

// create an object
$herbie = new City();

// show object properties
echo $herbie->city;
?>
```

خروجی توسط مرورگر وب:



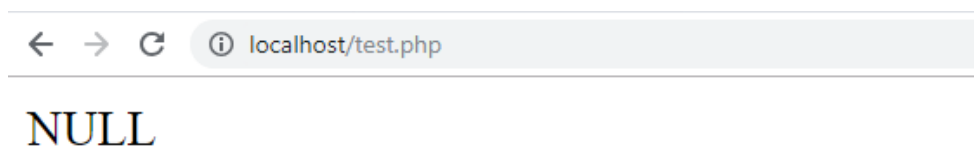
شکل (۳-۷)

۳.۲.۷ دیتاتایپ Null

عبارت از دیتاتایپی خاص است که فقط یک مقدار Null را می‌تواند در متحول ذخیره کند.

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

خروجی توسط مرورگر وب:



شکل (۸-۳)

۳.۳ عملگرها در PHP

عملگرها، برای اجرای عملیات بالای مقادیر که در متحولها ذخیره است استفاده می‌شود.

عملگرها در PHP از لحاظ اجرای عملیات به دسته‌های ذیل تقسیم می‌شود:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators
- Logical operators

۳.۳.۱ عملگرهای حسابی (arithmetic operators)

عبارت از عملگرها و یا سمبول‌هایی هستند که برای انجام عملیات بالای مقادیر عددی استفاده می‌شوند:

این عملگرها در جدول ذیل به صورت مکمل با وظایفشان توضیح داده شده‌اند.

جدول (۱-۳)

Operator	Name	Example	Result
+	Addition	$\$x + \y	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \y	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \y	Product of $\$x$ and $\$y$
/	Division	$\$x / \y	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \y	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \y	Result of raising $\$x$ to the $\$y$ 'th power (Introduced in PHP 5.6)

مثال:

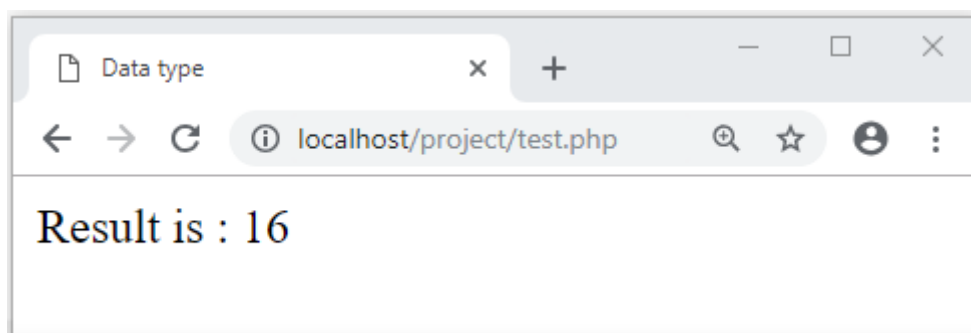
```
<!DOCTYPE html>
<html>
<body>

<?php
$x = 10;
$y = 6;

echo"Result is:", $x + $y;
?>

</body>
</html>
```

خروجی با استفاده از مرورگر وب:



شکل (۳-۹)

توجه: با تغییر علامات از + به -, *, /, %, ** عملیات دیگر نیز اجرا خواهند شد.

۳.۳.۲ عملگرهای جایگزینی (assignment operator)

عبارت از عملگرهایی هستند که برای نوشتن یک مقدار در داخل یک متحول استفاده می‌شوند. ساده‌ترین عملگر جایگزینی "=" می‌باشد. این بدان معنا است که یک مقدار از سمت راست به سمت چپ در داخل متحول کپی شده، جایگزین می‌شود.

عملگرهای جایگزینی در جدول ذیل تشریح شده است.

جدول (۲-۳)

Assignment	Same as...	Description
<code>x = y</code>	<code>x = y</code>	The left operand gets set to the value of the expression on the right
<code>x += y</code>	<code>x = x + y</code>	Addition
<code>x -= y</code>	<code>x = x - y</code>	Subtraction
<code>x *= y</code>	<code>x = x * y</code>	Multiplication
<code>x /= y</code>	<code>x = x / y</code>	Division
<code>x %= y</code>	<code>x = x % y</code>	Modulus

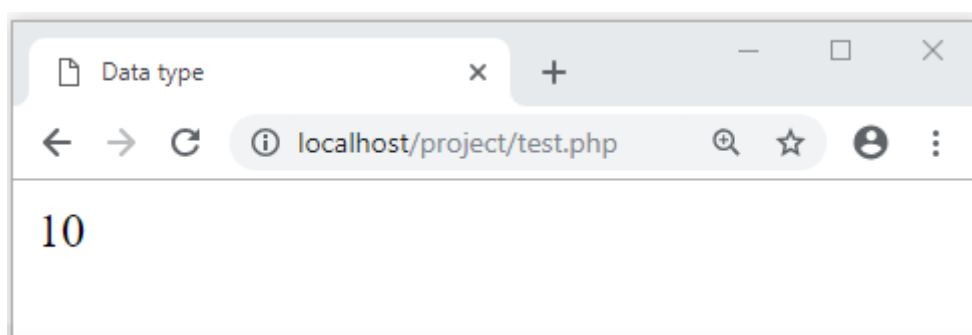
مثال:

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

<?php
$x = 10;
echo $x;
?>

</body>
</html>
```

خروجی در مرورگر وب:



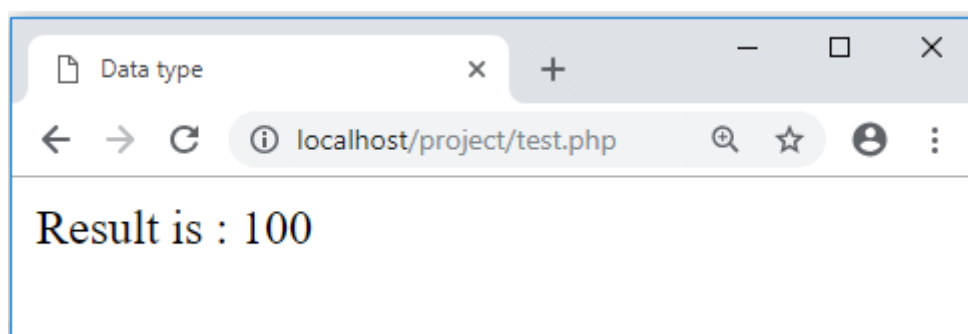
شکل (۱۰-۳)

مثال:

```
<?php
$x = 20;
$x += 100;

echo "Result is: ",$x;
?>
```

خروجی توسط مرورگر وب:



شکل (۳-۱۱)

توجه: در کنار علامت (=) علامت دیگر را تغییر دهید تا عملیات دیگر اجرا شود. مثلاً: (=, /=, *=, -=).

۳.۳.۳ عملگرهای مقایسوی (Comparison operators)

عبارت از عملگرهایی هستند که برای مقایسه دو مقدار عددی یا استرنگ استفاده می‌شود.

جدول ذیل عملگرهای مقایسوی را با مشخصات آن نشان می‌دهد.

جدول (۳-۳)

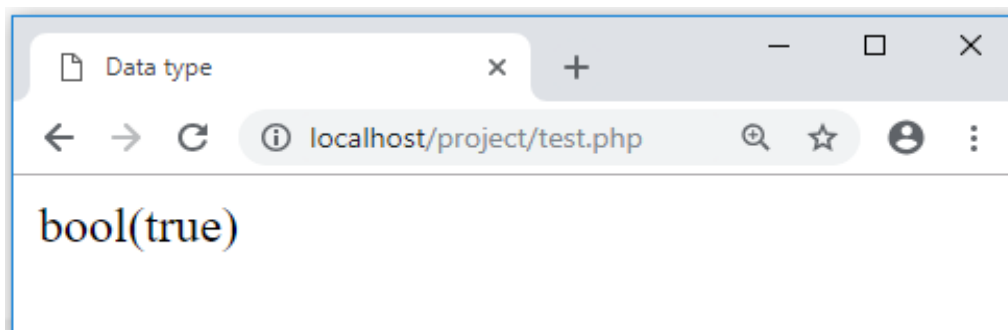
Operator	Name	Example	Result
==	Equal	<code>\$x == \$y</code>	Returns true if \$x is equal to \$y
===	Identical	<code>\$x === \$y</code>	Returns true if \$x is equal to \$y, and they are of the same type
!=	Not equal	<code>\$x != \$y</code>	Returns true if \$x is not equal to \$y
<>	Not equal	<code>\$x <> \$y</code>	Returns true if \$x is not equal to \$y
!==	Not identical	<code>\$x !== \$y</code>	Returns true if \$x is not equal to \$y, or they are not of the same type
>	Greater than	<code>\$x > \$y</code>	Returns true if \$x is greater than \$y
<	Less than	<code>\$x < \$y</code>	Returns true if \$x is less than \$y
>=	Greater than or equal to	<code>\$x >= \$y</code>	Returns true if \$x is greater than or equal to \$y
<=	Less than or equal to	<code>\$x <= \$y</code>	Returns true if \$x is less than or equal to \$y

مثال: عملگر ==

```
<?php
$x = 100;
$y = "100";

var_dump($x == $y); // returns true because values are equal
?>
```

خروجی توسط مرورگر وب:



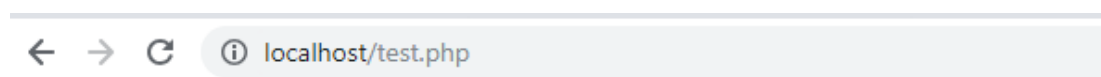
شکل (۳-۱۲)

مثال: عملگر ===

```
<?php
$x = 100;
$y = "100";

var_dump($x === $y); // returns false because types are not equal
?>
```

خروجی توسط مرورگر وب:



bool(false)

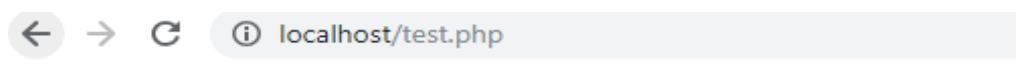
شکل (۳-۱۳)

مثال: عملگر !=

```
<?php
$x = 100;
$y = 600;

var_dump($x != $y); // returns true because values are not equal
?>
```

نتیجه در مرورگر وب:



bool(true)

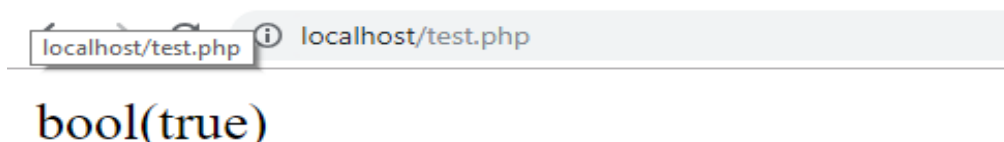
شکل (۳-۱۴)

مثال: عملگر <

```
<?php
$x = 100;
$y = 600;

var_dump($x < $y); // returns true because 100 is smaller than 600
?>
```

نتیجه در مرورگر وب:



شکل (۳-۱۵)

توجه: جای علامات را تغییر دهید، سپس نتایج را در مرورگر وب مشاهده خواهید کرد.

۳.۳.۴ عملگرهای افزایشی یا کاهش (Increment / Decrement)

عملگر افزایشی: عبارت از عملگری است که برای افزایش دادن مقادیر یک متحول استفاده می‌شود.

عملگر کاهش: عبارت از عملگری است که برای کاهش دادن مقادیر یک متحول استفاده می‌شود.

جدول زیر لست این عملگرها را با اجرای عملیات‌شان نشان می‌دهد:

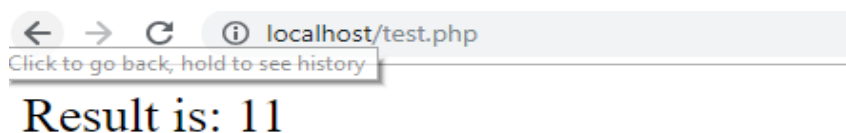
جدول (۳-۴)

Operator	Name	Description
++\$x	Pre-increment	Increments \$x by one, then returns \$x
\$x++	Post-increment	Returns \$x, then increments \$x by one
--\$x	Pre-decrement	Decrements \$x by one, then returns \$x
\$x--	Post-decrement	Returns \$x, then decrements \$x by one

مثال: عملگر افزایشی ++

```
<?php
$x = 10;
echo "Result is: ", ++$x;
?>
```

خروجی توسط مرورگر وب:



شکل (۱۶-۳)

مثال: عملگر کاهشی: --

```
<?php
$x = 10;
echo "Decrement result is: ",--$x;

?>
```

نتیجه در مرورگر وب:



شکل (۱۷-۳)

توجه: در صورتی که علامت افزایشی قبل از نام متحول در سمت چپ بیاید، آن را به نام pre increment یاد می‌کند.

در صورتی که علامت افزایشی بعد از نام متحول در سمت راست بیاید، آن را به نام post increment یاد می‌کند. نتایج در هر دو حالت یکسان می‌باشد.

۳.۳.۵ عملگرهای منطقی (Logical operators)

عبارت از عملگرهایی هستند که عملیات منطقی را بالای بیانیه‌های شرطی انجام می‌دهد.

جدول ذیل عملگرهای منطقی و مشخصات آن را نشان می‌دهد:

جدول (۳-۵)

Operator	Name	Example	Result
and	And	\$x and \$y	True if both \$x and \$y are true
or	Or	\$x or \$y	True if either \$x or \$y is true
xor	Xor	\$x xor \$y	True if either \$x or \$y is true, but not both
&&	And	\$x && \$y	True if both \$x and \$y are true
	Or	\$x \$y	True if either \$x or \$y is true
!	Not	!\$x	True if \$x is not true

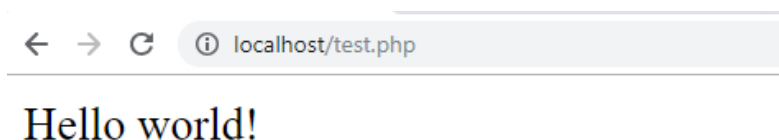
۳.۳.۶ عملگر منطقی and

عبارت از یک عملگر منطقی است که دو شرط را بررسی کرده، در صورتی که هر دو شرط درست باشد، نتیجه آن درست می‌باشد؛ در صورتی که یکی از شرطها اشتباه باشد، نتیجه آن اشتباه می‌باشد.

```
<?php
$x = 100;
$y = 50;

if ($x == 100 and $y == 50)
{
    echo "Hello world!";
}
?>
```

نتیجه در مرورگر وب:



شکل (۳-۱۸)

۳.۳.۷ عملگر منطقی or

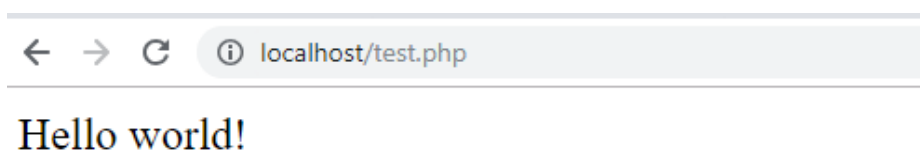
عبارت از عملگر منطقی است که دو شرط را بررسی می‌کند. در صورتی که هر دو شرط درست باشد و یا یکی از شرطها درست باشد، نتیجه را درست نشان می‌دهد؛ اما اگر هر دو شرط اشتباه باشد نتیجه آن اشتباه می‌باشد.

مثال:

```
<?php
$x = 100;
$y = 50;

if ($x == 100 or $y == 80) {
    echo "Hello world!";
}
?>
```

خروجی توسط مرورگر وب:



شکل (۳-۱۹)

در این مثال می‌بینیم که یکی از شرط‌ها درست، ولی شرط دوم اشتباه می‌باشد، پس نتیجه را درست نشان می‌دهد.

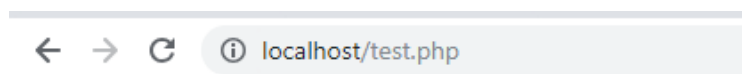
۳.۳.۸ عملگر XOR

عبارت از عملگری است که دو شرط را بررسی می‌کند و اگر یکی از شرط‌ها درست باشد، نتیجه را درست نشان می‌دهد. و اگر هر دو شرط درست باشد نتیجه را **False** نشان می‌دهد. از این عملگر زمانی استفاده می‌شود که نیاز به درست بودن یک شرط باشد.

```
<?php
$x = 100;
$y = 50;

if ($x == 100 xor $y == 80) {
    echo "This is XOR operator";
}
?>
```

نتیجه توسط مرورگر وب:



This is XOR operator

شکل (۳-۲۰)

عملگرهای &&، مانند & است، صرف تفاوتشان در این است که اگر && باشد، شرط اول را بررسی می‌کند و اگر درست باشد شرط دوم را نیز بررسی می‌کند؛ ولی اگر شرط اول اشتباه باشد، شرط دومی را بررسی نکرده، جواب را اشتباه نشان می‌دهد. استفاده && در زمان کمتر شرایط بیشتر را بررسی می‌کند.

عملگرهای || مانند | بوده صرف تفاوتشان در این است که اگر || باشد، شرط اول را بررسی می‌کند و اگر درست باشد، شرط دوم را بررسی نمی‌کند و نتیجه را درست نشان می‌دهد. این عملگر نیز در زمان کمتر شرایط بیشتر را بررسی می‌کند.

۳.۳.۹ عملگر NOT

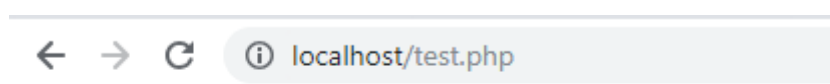
عبارت از عملگری است که نتیجه درست را اشتباه و نتیجه اشتباه را درست نشان می‌دهد، هدف استفاده از عملگر NOT برای برعکس کردن یک شرط می‌باشد.

مثال:

```
<?php
$x = 100;

if ($x !== 90) {
    echo "! not operator";
}
?>
```

خروجی توسط مرورگر وب:



! not operator

شکل (۳-۲۱)

۳.۴ بیانیه‌های شرطی (Condition statements)

زبان‌های برنامه‌نویسی ساختاریافته دارای دستورات شرطی می‌باشد. دستورات شرطی با برقراری شرط خاص، مجموعه‌ای از دستورات عمل‌ها را اجرا می‌کند. PHP دو نوع دستور شرطی را ارائه می‌کند. اولین دستور شرطی دستور If است که بر اساس شرط خاصی، مجموعه دستورات را اجرا می‌کند. در صورتی که بخواهید مقدار یک عبارت را با چند مقدار مقایسه کرده، بر اساس این مقایسه، مجموعه دستورات عمل‌هایی را اجرا کنید، باید از دومین دستور شرطی زبان PHP یعنی دستور Switch استفاده کنید.

۳.۴.۱ دستور IF

عبارت از دستوری است که در شرط‌های مشخص از آن استفاده می‌شود و یا در صورتی که در یک پروگرام یک شرط را تعریف کنیم از IF استفاده می‌شود.

Syntax

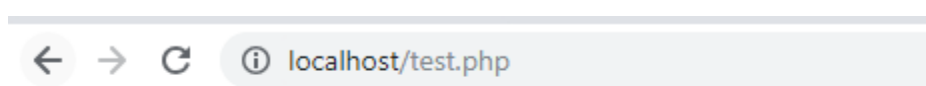
```
if (condition) {
    code to be executed if condition is true;
}
```

مثال:

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

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

نتیجه در مرورگر وب:



Have a good day!

شکل (۳-۲۲)

۳.۴.۲ دستور if..else

عبارت از دستوری است که دارای دو بلاک است. اگر شرط درست باشد، بلاک if اجرا می‌شود. در صورتی که شرط اشتباه باشد، بلاک else اجرا می‌شود.

Syntax:

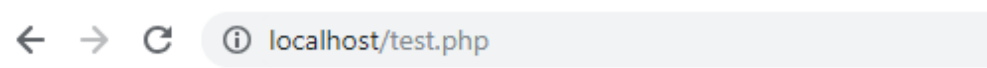
```
if (condition) {
    code to be executed if condition is true;
} else {
    code to be executed if condition is false;
}
```

مثال:

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

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

نتیجه در مرورگر وب:



Have a good night!

شکل (۳-۲۳)

۳.۴.۳ دستور if..elseif

عبارت از دستوری است که برای بررسی کردن چندین شرط از آن استفاده می‌شود. اگر دستور اول درست باشد، پروگرام اجرا می‌شود و نتیجه را نشان می‌دهد، در صورتی که شرط اول اشتباه باشد، شرط دوم بررسی می‌شود. به همین ترتیب تمام شرایطی که در پروگرام تعریف شده، بررسی می‌شود و در نهایت اگر تمام شرایط اشتباه باشد دستور else اجرا می‌شود.

Syntax

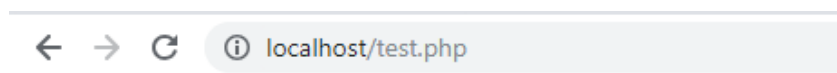
```
if (condition) {
    code to be executed if condition is true;
} elseif (condition) {
    code to be executed if condition is true;
} else {
    code to be executed if condition is false;
}
```

مثال:

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

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

نتیجه در مرورگر وب:



Have a good morning!

شکل (۳-۲۴)

۳.۴.۴ دستور switch

از دستور سویچ زمانی استفاده می‌شود که خواسته باشیم از چندین بلاک یا شرایط یک بلاک اجرا شود.

Syntax

```

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;
}

```

مثال:

```

<?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, or green!";
    }
?>

```

نتیجه در مرورگر وب:



Your favorite color is red!

شکل (۳-۲۵)

۳.۵ بیانیه‌های تکرار (Looping)

بیانیه‌های تکراری از عناصر اصلی زبان‌های برنامه‌نویسی ساختاریافته می‌باشد. توسط بیانیه‌های تکراری می‌توانیم مجموعه‌ای از دستورات را به‌دفعات مشخص یا تا برقراری شرط خاصی اجرا کنیم.

بیانیه‌های تکراری که در PHP استفاده می‌شود، عبارت‌اند از:

- For loop
- While loop
- Do while loop
- Foreach loop

For loop ۳.۵.۱

عبارت از بیانیه تکراری است که کدهای یک بلاک را به‌دفعات مشخص نظر به یک شرط تکرار می‌کند.

Syntax

```
for (init counter; test counter; increment counter) {
    code to be executed;
}
```

مثال:

```
<?php
for ($x = 0; $x <= 10; $x++) {
    echo "The number is: $x <br>";
}
?>
```

نتیجه توسط مرورگر وب:

← → ↻ ⓘ localhost/test.php

```
The number is: 0
The number is: 1
The number is: 2
The number is: 3
The number is: 4
The number is: 5
The number is: 6
The number is: 7
The number is: 8
The number is: 9
The number is: 10
```

شکل (۳-۲۶)

تشریح پروگرام

بیانیه تکراری for دارای سه آرگومان^۱ می باشد.

- Initialization: برای مشخص کردن اولین مقدار که شروع loop است، استفاده می شود.
- Condition: برای تعیین شرط است که آخرین رقم loop در آن مشخص می شود.
- Increment/decrement: برای افزایش یا کاهش تکرار مقادیر استفاده می شود. یک Loop می تواند افزایشی باشد و یا کاهشی.

Echo: عبارت از فنکشنی است که برای خروجی مقادیر تکراری استفاده می شود.

۳.۵.۲ While loop

عبارت از بیانیه تکراری است که برای تکرار دستورات نظر به یک شرط استفاده می شود.

^۱ Argument

Syntax

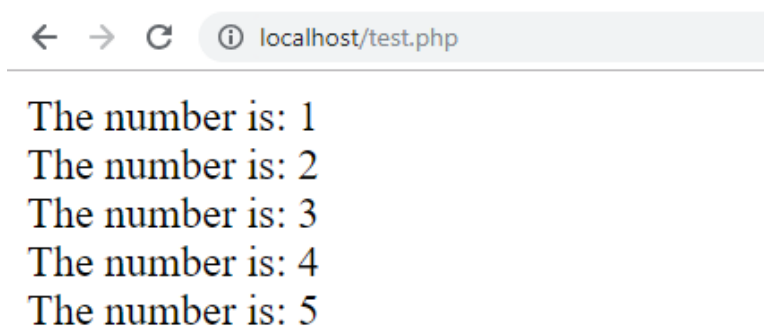
```
while (condition is true) {
    code to be executed;
}
```

مثال:

```
<?php
$x = 1;

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

نتیجه را در مرورگر وب ببینید:



```
← → ↻ ⓘ localhost/test.php

The number is: 1
The number is: 2
The number is: 3
The number is: 4
The number is: 5
```

شکل (۳-۲۷)

۳.۵.۳ Do while loop

عبارت از بیانیه تکراری است، که برای تکرار دستورات استفاده می‌شود. مقادیر تا زمانی تکرار می‌شوند که شرط درست باشد. اگر شرط درست نباشد صرف یک‌بار اجرا می‌شود.

Syntax

```
do {
    code to be executed;
} while (condition is true);
```

مثال:

```
<?php
$x = 1;

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

تشریح پروگرام

عدد 1 شروع loop را نشان می‌دهد. عدد 5 پایان loop است.

۳.۵.۴ بیانیه تکراری For each loop

عبارت از بیانیه تکراری است که در array استفاده می‌شود و هر مقدار کلیدی را تکرار می‌کند.

Syntax

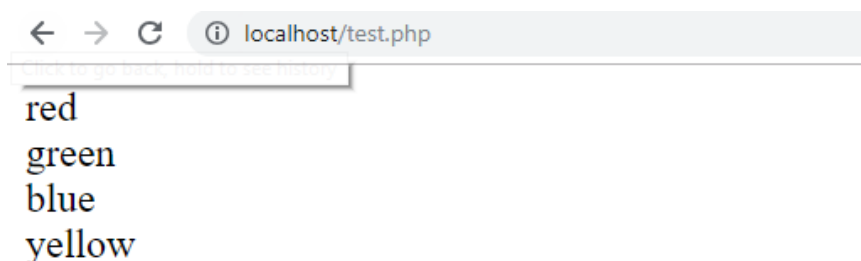
```
foreach ($array as $value) {
    code to be executed;
}
```

مثال:

```
<?php
$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {
    echo "$value <br>";
}
?>
```

نتیجه در مرورگر وب:



شکل (۳-۲۸)

۳.۶ صف (array)

عبارت از نوع متحول است که چندین مقدار را در یک زمان ذخیره می‌کند؛ اما تنها مشکل آن این است که صرف یک نوع دیتا را ذخیره می‌تواند.

۳.۶.۱ طریقهٔ ایجاد array

در PHP صف، صرف با فنکشن (array) ایجاد می‌شود.

```
Array();
```

۳.۶.۲ اقسام array:

در PHP سه نوع array وجود دارد:

1. Indexed arrays - Arrays with a numeric index
2. Associative arrays - Arrays with named keys
3. Multidimensional arrays - Arrays containing one or more arrays

۳.۶.۳ Index array

عبارت از نوع array است که با اندیکس^۲ یا یک فهرست ایجاد می‌شود. اندیکس‌ها از (0) آغاز می‌شوند.

```
$city = array("kabul", "Herat", "Mazar");
```

1. Index

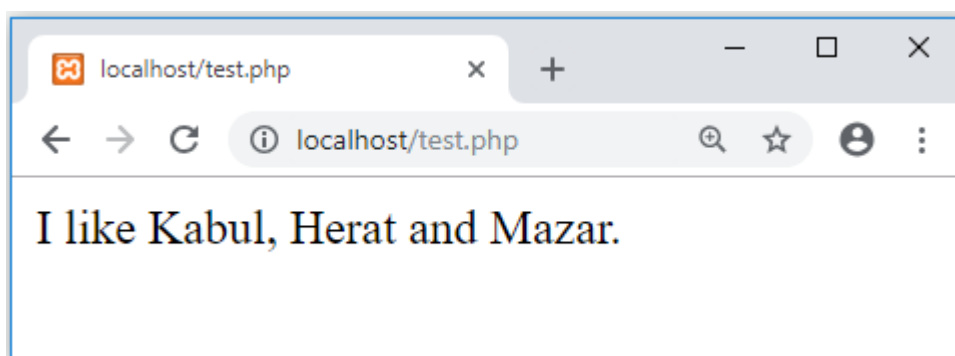
همچنان می‌توانیم اندیکس‌ها را خود ما بنویسیم:

```
$city[0] = "Kabul";
$city[1] = "Herat";
$city[2] = "Mazar";
```

مثال:

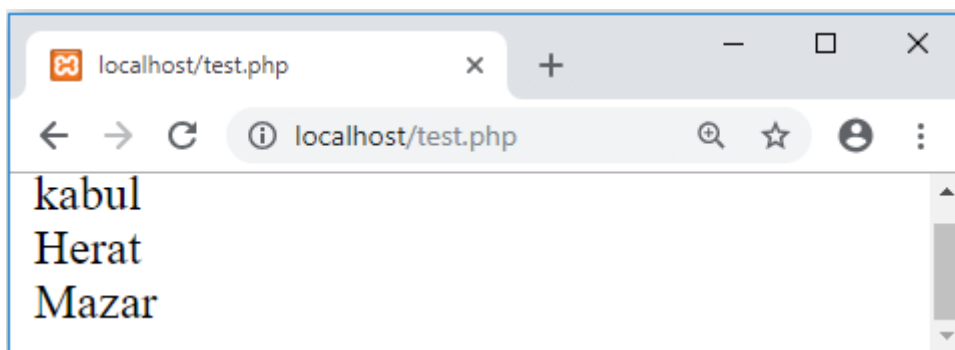
```
<?php
$cities = array("Kabul", "Herat", "Mazar");
echo "I like " . $cities[0] . ", " . $cities[1] . " and " . $cities[2] . ".";
?>
```

نتیجه را در مرورگر وب مشاهده کنید:



شکل (۳-۲۹)

اگر عناصر array را توسط loop نشان دهیم، از مثال ذیل استفاده می‌کنیم.



شکل (۳-۳۰)

۳.۶.۴ Associative Arrays

عبارت از نوع array است که بر اساس نام‌های تعریف‌شده، شناسایی می‌شود. اندکس‌های آن از نوع حروف بوده و توسط پروگرامر تعریف می‌گردد

Syntax

```
$age = array("Mateen"=>"35", "Sajeed"=>"37", "Jamshid"=>"43");
```

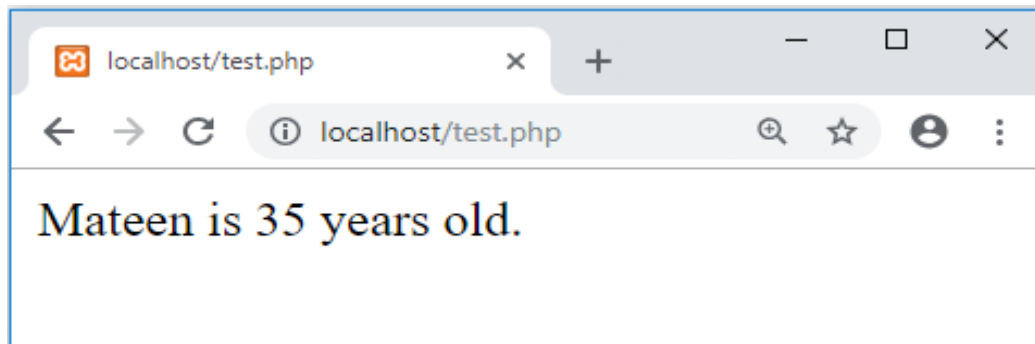
به طریقهٔ ذیل نیز می‌توان آن‌را تعریف کرد:

```
$age['Mateen'] = "35";
$age['Sajeed'] = "37";
$age['Jamshid'] = "43";
```

مثال:

```
<?php
$age = array("Mateen"=>"35", "Sajeed"=>"37", "Jamshid"=>"43");
echo "Mateen is " . $age['Mateen'] . " years old.";
?>
```

نتیجه را در مرورگر وب مشاهده کنید:



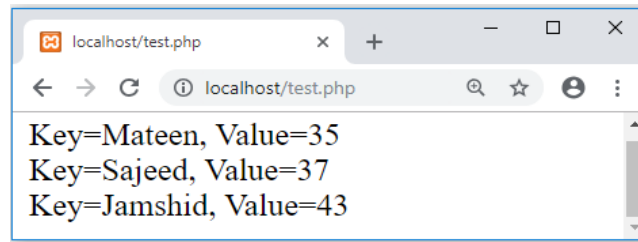
شکل (۳-۳۱)

Associative array را توسط loop foreach نیز می‌توان به شکل ذیل نشان داد؛

```
<?php
$age = array("Mateen"=>"35", "Sajeed"=>"37", "Jamshid"=>"43");

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

خروجی در مرورگر وب:



شکل (۳-۳۲)

۳.۷ Global Variables – Superglobal PHP متحول‌های جهانی

عبارت از متحول‌های از قبل تعریف شده‌ی زبان PHP بوده که همیشه و در هر پروگرام قابل دسترس می‌باشد و هر کدام آن برای مقاصد مختلف استفاده می‌شود. مهم‌ترین آن‌ها قرار ذیل است:

- \$_SERVER
- \$_REQUEST
- \$_POST
- \$_GET
- \$_FILES
- \$_COOKIE
- \$_SESSION

\$_SERVER: این متحول برای ذخیره نمودن اطلاعات سرور استفاده می‌شود.

مثال:

>?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'];
<?

```

از بقیه متحول‌ها در فصل‌های بعدی این کتاب نظر به نیاز استفاده گردیده است.

خلاصه‌ی فصل سوم



در این فصل در مورد مفاهیم اساسی زبان PHP بحث گردید. با فراگیری محتویات این فصل، محصلان قادر خواهند شد که در مورد مفاهیم اساسی و ابتدایی PHP دانش کافی را به دست آورند. هم‌چنان در این فصل در بارهٔ متحول‌ها، دیتاتایپ‌ها، عملگرهای کاربردی PHP، شرط‌ها، حلقه‌ها، صف یا Array بحث مفصل صورت گرفت.



سوالات و فعالیت های فصل سوم

۱. وظیفه متحول را تشریح کنید.
۲. دیتاتایپ‌ها در PHP چند نوع هستند؟
۳. از عملگرهای منطقی در کجا استفاده می‌شود؟
۴. حلقه‌ها در PHP چند نوع می‌باشند؟
۵. فرق بین حلقه `While` و `Do While` را شرح دهید.
۶. فرق بین `switch` و `else if` را شرح دهید.
۷. اقسام `array` را در PHP توضیح دهید.

فعالیت انفرادی

- متحول‌ها در PHP با کدام سمبول تعریف می‌شوند.
- پروگرامی را بنویسید که در آن اعداد با هم جمع شوند.
- پروگرامی را بنویسید که در آن مقادیر از هم تفریق شوند.
- پروگرامی را بنویسید که در آن مقادیر در داخل `array` ذخیره شوند.
- پروگرامی را بنویسید که یک متن ده بار تکرار چاپ شود.

فعالیت گروهی

- در مورد انواع دیتاتایپ‌های PHP باهم مباحثه کنید.
- اگر حلقه‌ها در PHP تعریف نشود، بجای آن از کدام بیانیه باید استفاده شود؟
- در مورد فواید `array` در PHP باهم گفتگو کنید.

فصل چهارم

PHP Forms



هدف کلی: محصلان در پایان این فصل با فورم و کاربرد آن آشنا خواهند شد.

اهداف آموزشی: در پایان این فصل محصلان قادر خواهند شد تا:

۱. Form را تعریف کنند.
۲. FormHandl i ng را شرح دهند.
۳. Formval i dat i on and f ormr equi red را شرح دهند.
۴. PHP For mURL/Enai l را تشریح کنند.

در این فصل در مورد فورم در PHP و موضوعاتی که متعلق به فورم است، بحث می‌شود. فورم یکی از مفاهیم کاربردی در PHP است، پس در آموزش این فصل دقت بیشتر داشته باشید.

۴.۱ فورم در PHP

فورم عبارت از عنصر کاربردی در PHP است که برای مرتب‌ساختن کنترل‌ها استفاده می‌شود. فورم، مانند یک فضایی است که در آن برنامه‌نویسان وب کنترل‌های مورد ضرورت را دیزاین می‌کنند. فورم به کمک تگ‌های HTML نوشته می‌شود. محتویاتی که در فورم درج می‌شود، سپس به کمک پروتوکول HTTP به سمت سرور به فایل‌های PHP ارسال می‌شود.

برای وضاحت بیشتر به مثال ذیل توجه کنید:

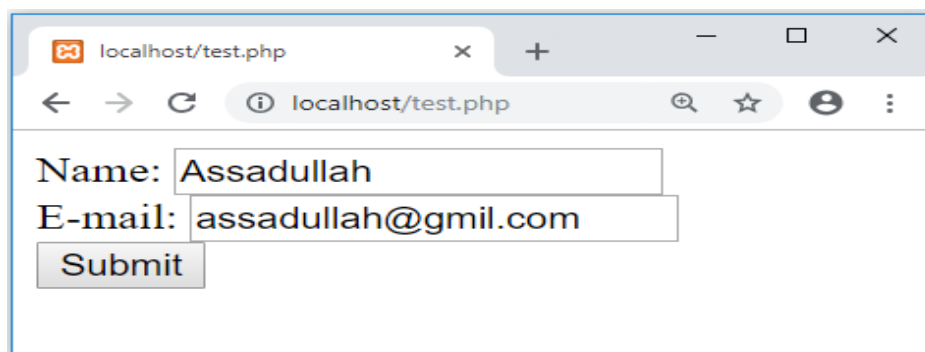
```
<html>
  <body>

  <form action="welcome.php" method="post">
  Name: <input type="text" name="name"><br>
  E-mail: <input type="text" name="email"><br>
  <input type="submit">
  </form>

  </body>
</html>
```

نتیجه را در مرورگر وب با دو تکست‌باکس^۳ و یک دکمه submit مشاهده می‌کنید که با کلیک کردن بالای دکمه submit محتویات تکست‌باکس‌ها به سمت سرور به فایل welcome.php ارسال می‌شود.

1. Text box



شکل (۱-۴)

بعد از کلیک کردن محتویات تکست باکس ها به کمک پروتوکول HTTP به سمت سرور ارسال شده، توسط فایل PHP پروسس می شود و بعد از پروسس در مرورگر وب، نتیجه نشان داده می شود.

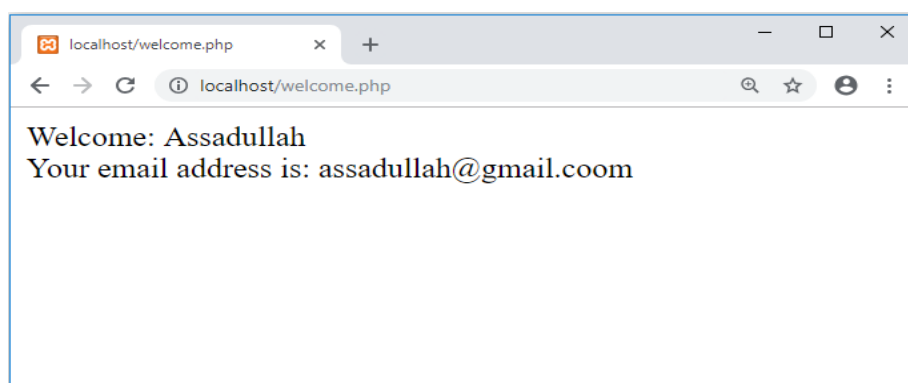
محتویات فایل welcome.php

```
<html>
<body>

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

</body>
</html>
```

خروجی در مرورگر وب:



شکل (۲-۴)

۴.۱.۱ \$_POST

از جمله متحول های Superglobal بوده که برای جمع آوری معلومات فرم استفاده می شود در صورت که میتود فرم get باشد.

```

<form method="post" action="">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>
?>
$name=                                     $_POST['fname'];
                                           echo          $name;
{
<?

```

\$_GET ۴.۱.۲

از جمله متحول‌های Superglobal بوده که برای جمع آوری معلومات فرم استفاده می‌شود در صورت که میتود فرم get باشد و در URL نیز نشان داده می‌شود.

```

<form method="get" action="">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>
?>
$name = $_GET['fname'];
  echo $name;
{
<?

```

۴.۲ میتودهای Get and Post

عبارت از میتودهایی است که برای ارسال اطلاعات از form به سمت سرور استفاده می شود.

۴.۲.۱ میتود (Get)

عبارت از میتودی است که اطلاعات را از form کمپیوتر کلاینت به سمت کمپیوتر سرور به کمک پروتوکول HTTP ارسال می کند.

مشکل این میتود، این است که تمام اطلاعات ارسال شده در URL قابل رویت می باشد. هم چنان ظرفیت این میتود در ارسال مقدار اطلاعات محدود بوده تا 2000 کرکت در یک وقت از طریق آن قابل ارسال می باشد.

از میتود Get وقتی استفاده می شود که اطلاعات حساس نباشد.

نوت: اطلاعاتی، مانند پسورد و یا دیگر اطلاعات حساس را هیچ وقت با میتود Get ارسال نکنید.

۴.۲.۲ میتود (Post)

عبارت از میتودی است که از آن برای ارسال اطلاعات از form به سمت سرور استفاده می شود. اطلاعاتی که با میتود (Post) ارسال گردد در URL قابل دید نمی باشد. هم چنان در زمینه مقدار اطلاعات محدودیت ندارد.

۴.۳ Form validation

عبارت از پروسه‌یی است که کنترل‌های مهم توسط آن نشانی می شود و برای وارد کردن اطلاعات ضروری از آن استفاده می شود.

فورم ذیل را مشاهده کنید که در آن، در برابر کنترل‌هایی که عکس ستاره به رنگ سرخ نشانی شده است، از جمله کنترل‌های مهم‌اند که باید دیتا در آن‌ها وارد شود.

PHP Form Validation Example

*** required field.**

Name: *

E-mail: *

Website:

Comment:

Gender: Female Male *

Your Input:

شکل (۳-۴)

قواعد فورم validation را در جدول ذیل مشاهده کنید:

جدول (۱-۴)

Field	Validation Rules
Name	Required. + Must only contain letters and whitespace
E-mail	Required. + Must contain a valid email address (with @ and .)
Website	Optional. If present, it must contain a valid URL
Comment	Optional. Multi-line input field (textarea)
Gender	Required. Must select one

تکست باکس نام: باید در آن حروف الفبا و کرکترها وارد شود.

تکست باکس ایمیل: باید ایمیل درست با علامه @ وارد شود.

تکست باکس وبسایت: باید یک URL درست در آن وارد شود.

تکست ایریاکومنت: در آن متنی وارد می شود که کدام محدودیت ندارد.

رادیو بتن جندر: باید یکی از بتن ها انتخاب شود.

Text Fields •

نام، ایمیل، وبسایت عناصر ورودی است، اما در فیلد کومنت تکست ایریا می باشد. کدهای HTML آن ها قرار ذیل اند:

```
<html>
<body>

Name: <input type="text" name="name">
E-mail: <input type="text" name="email">
Website: <input type="text" name="website">
Comment: <textarea name="comment" rows="5" cols="40"></textarea>

</body>
</html>
```

Radio Buttons •

جنر توسط رادیو بتن دیزاین می شود. کدهای HTML آن قرار ذیل می باشد.

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

Validate Form Data With PHP •

توسط قابلیت های PHP نیز می توانیم اطلاعاتی را که در فورم وارد می شود، validate کنیم.

مثال:

```
<?php
// define variables and set to empty values
$name = $email = $gender = $comment = $website = "";

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

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>
```

در این پروگرام می‌بینیم که فیلدهای نام، ایمیل، وبسایت، کامنت و جندر توسط فنکشن (test_input) بررسی می‌شود.

در صورتی که در اطلاعات وارد شده، اشتباهات وجود داشته باشد و یا مطابق شرط وارد نشود، در آن صورت پیام اشتباه ظاهر می‌شود.

۴.۴ PHP – Required Fields

قواعد فورم validation بالای جدول‌ها در بحث قبلی ما (name, E-mail and gender fields) ضرورت است. این فیلدها خالی بوده نمی‌تواند، باید حتمی خانه‌پری شود.

جدول (۲-۴)

Field	Validation Rules
Name	Required. + Must only contain letters and whitespace
E-mail	Required. + Must contain a valid email address (with @ and .)
Website	Optional. If present, it must contain a valid URL
Comment	Optional. Multi-line input field (textarea)
Gender	Required. Must select one

در پروگرام زیر بعضی متحول‌های جدید افزوده شده که عبارتند از:

\$nameErr, \$emailErr, \$genderErr and \$websiteErr. این متحول‌ها اشتباهات را از فیلدهای ضروری انتقال می‌دهند. در این پروگرام یک بلاک if else نیز افزوده‌ایم.

```

<?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"]);
    }
}

```

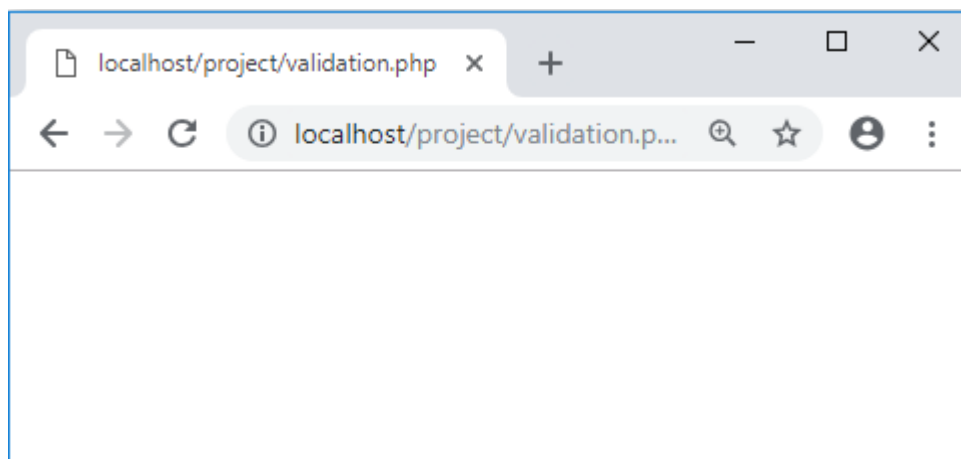
```

    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"]);
    }
}
?>

```

بعد از اجرای این پروگرام می بینیم که اشتباه وجود ندارد و پروگرام اجرا می شود.



شکل (۴-۴)

۴.۴.۱ Display The Error Messages

در HTML form بعد از هر فیلد یک اسکریپت نوشته شده که یک اشتباه را ایجاد کند.

```

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

  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>
  <label>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">

</form>

```

بعد از نوشتن این کد در مرورگر وب مشاهده کنید.

The screenshot shows a web browser window with the address bar displaying 'localhost/project/form.php'. The form contains the following elements:

- Name: *
- E-mail: *
- Website:
- Comment:
- Gender: Female Male *
- Submit button

شکل (۴-۵)

در صورتی که در وقت خانه‌پری فیلدها، یکی از آن‌ها خالی باشد، پیام اشتباه برای شما ظاهر می‌شود.

۴.۵ PHP – Val i dat e E-mai l

آسان‌ترین و مصون‌ترین راه برای بررسی یک ایمیل است که دارای فارمت درست است یا خیر. از فنکشن (filter_var) استفاده می‌کنیم.

این فنکشن برای Validate نمودن و Sanitize نمودن بوده و دارای دو پارامتر مهم می‌باشد:

- متحول که باید چک شود.
- و کدام نوع چک صورت بگیرد.

در پروگرام ذیل متحول \$email چک شود و تنها فارمت ایمیل اجازه داده شود.

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

در صورت که فارمت ایمیل درست نباشد، بعد از Submit پیام اشتباه را مشاهده کنید:

E-mail: * Invalid email format

شکل (۴-۶)

برای پاک سازی کرکترهای غیر مجاز از ایمیل (FILTER_SANITIZE_EMAIL) استفاده می شود.

```
<?php
```

```
$email = "hamida.ahmadi@gmail.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");
```

```
}
```

```
?>
```

خلاصه‌ی فصل چهارم



در این فصل در مورد فورم و قواعدی که برای فورم در زبان برنامه‌نویسی PHP استفاده می‌کنیم بحث گردید. یکی از بخش‌های کاربردی PHP در وقت توسعه‌دادن وبسایت‌ها، همین بخش فورم است. کنترل‌هایی که در فورم تعریف می‌شوند، روش خانه‌پری آن‌ها و هم‌چنان می‌تودهایی که توسط آن محتویات فورم به سمت کمپیوتر سرور ارسال شود، در این فصل مورد بحث قرار گرفت.



سوالات فعالیت های فصل چهارم

۱. چرا از فورم validation استفاده می کنیم؟
۲. در صورتی که فورم validation در کنترلر ها تعریف نشود، چه اتفاق می افتد؟
۳. اگر بخواهیم بعضی کنترلر ها را توسط فنکشن ها کنترل کنیم، کدام فنکشن ها را باید بنویسیم؟
۴. Validation email را توضیح دهید.
۵. Sanitize email را توضیح دهید.

فعالیت انفرادی

- پروگرامی را بنویسید که یک فورم با کنترلر های نام، آدرس، ایمیل آدرس و جنسیت باشد.
- پروگرامی را بنویسید که برای خانه پری کنترلر ها، یک تصدیق بخواهد.
- پروگرامی را بنویسید که validation را در کنترلر های email, gender نشان دهد.
- پروگرامی را بنویسید که فورم با validation های کنترلر ایجاد شود.

فعالیت گروهی

- در مورد PHP Filters جستجو نمایید.
- کاربرد \$_REQUEST را تشریح نمایید.

فصل پنجم

PHP پیشرفته



هدف کلی: محصلان در پایان این فصل با مدیریت فایل‌ها، session, cookie آشنا خواهند شد.

اهداف آموزشی: در پایان این فصل محصلان قادر خواهند شد تا:

۱. include را تعریف کنند.
۲. PHP file handling را شرح دهند.
۳. PHP-file upload را شرح دهند.
۴. PHP cookies را تشریح کنند.
۵. PHP session را توضیح دهند.
۶. PHP-error handling را تشریح کنند.
۷. PHP exception را شرح دهند.

در این فصل در مورد مفاهیم پیشرفته PHP بحث شده و مطالب کاربردی را یادآوری کرده‌ایم. با تکمیل این فصل با چگونگی آپلود کردن فایل‌ها در صفحات وب، با نرم افزارهای کوچک cookie و هم‌چنان session آشنا می‌شوید. نکات مهم PHP در این فصل بحث شده است، برای فراگیری آن دقت بیشتر داشته باشید.

۵.۱ PHP include

Include یا required تمام متن کود و فارمت را که در فایل مشخص موجود است، در فایلی که include استفاده می‌شود، کپی می‌کند.

Including خیلی کاربردی است، وقتی که بخواهیم بعضی فایل‌های PHP, HTML یا متن را به چندین صفحه وبسایت شامل بسازیم از include استفاده می‌کنیم.

PHP دارای قابلیت است که با استفاده از آن، می‌توانیم، محتویات یک فایل را به فایل دیگر اضافه کنیم؛ اما قبل از این که در سرور اجرا شود، این قابلیت توسط include یا required اجرا می‌شود.

Syntax

```
include 'filename';

or

require 'filename';
```

مثال:

فرض کنیم، یک فایل به نام footer.php داریم؛ اگر بخواهیم آن را در فایل دیگر include کنیم، پروسه ذیل را اجرا می‌کنیم.

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

حالا فایل footer.php را در فایل دیگر توسط include شامل می‌سازیم.

```

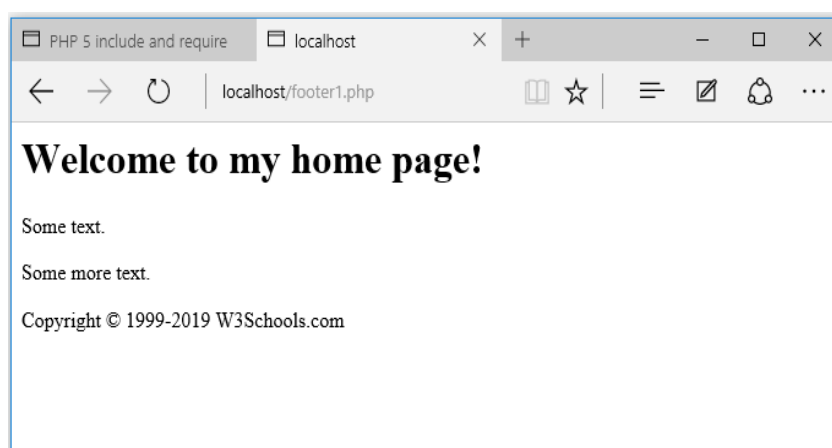
<html>
  <body>

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

  </body>
</html>

```

خروجی را در مرورگر وب مشاهده کنید.



شکل (۵-۱)

مثال:

فایلی که در آن مینوبار استاندارد تعریف شده، و به نام menu.php ذخیره گردیده است.

```

<?php
echo '<a href="/default.asp">Home</a> -
<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>';
?>

```

در صورتی که بخواهیم تمام صفحات وبسایت، از این مینوبار استاندارد استفاده شود، از کدهای ذیل استفاده می‌کنیم:

```

<html>
  <body>

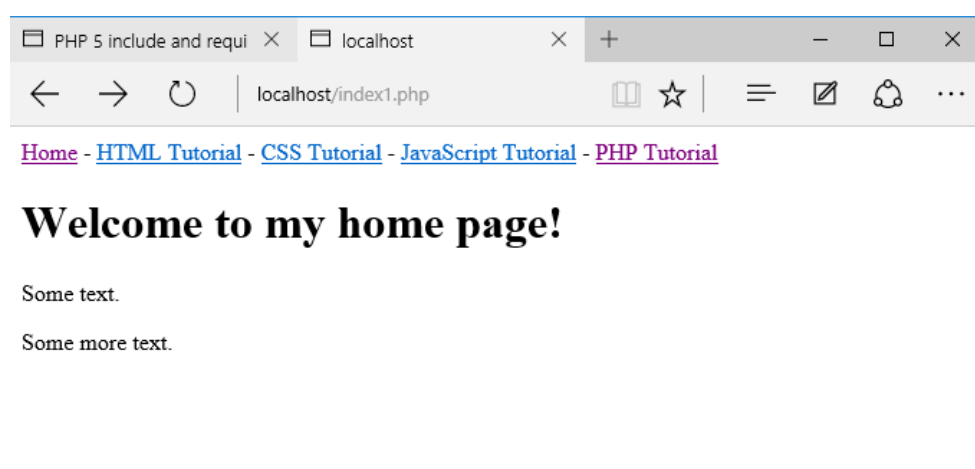
    <div class="menu">
      <?php include 'menu.php';?>
    </div>

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

  </body>
</html>

```

خروجی توسط مرورگر وب:



شکل (۵-۲)

۵.۲ Include vs. required

بیانیهٔ required نیز برای شامل سازی یک فایل به کدهای PHP استفاده می شود.

اما در این جا یک تفاوت بزرگ بین include و require وجود دارد.

وقتی که یک فایل توسط بیانیهٔ include شامل فایل دیگر می شود، php آن را پیدا نتوانسته اسکریپت به شکل دوامدار اجرا می شود.

مثال:

```
<html>
<body>

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

</body>
</html>
```

خروجی:



Welcome to my home page!

I have a .

اگر همین مثال را با بیانیه `require` اجرا کنیم، در آن صورت بیانیه `echo` اجرا نشده، در جریان اجرا اسکریپت `dies` می شود. در صورتی که فایل ضمیمه نگردد.

مثال:

```
<html>
<body>

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

</body>
</html>
```

خروجی:

```
Welcome to my home page!
```

PHP file handling ۵.۳

انتقال فایل‌ها در وب‌اپلیکیشن یک مفهوم کاربردی است که برای بازکردن و پروسس فایل‌ها به اهداف خاصی استفاده می‌شود.

PHP manipulating files ۵.۳.۱

PHP دارای فنکشن‌های زیادی است که برای ایجاد کردن، خواندن، آپلود کردن و ایدیت کردن فایل‌ها به طرف سرور استفاده می‌شود.

۵.۳.۲ خواندن فایل (readfile())

فنکشن (readfile) یک فایل را خوانده و محتویات آن را به buffer خروجی نشان می‌دهد.

مثلاً یک فایل متنی که به نام webdictionary.txt در سرور ذخیره می‌باشد.

```
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
```

محتویات این فایل توسط فنکشن (readfile) خوانده، در buffer خروجی قرار می‌دهد.

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

خروجی را در مرورگر وب مشاهده کنید:

```
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 Language236
```

باز نمودن فایل (fopen)

توسط فنکشن (fopen) می‌توانیم یک فایل را باز کنیم که به مودهای مختلف، برای ایجاد نمودن فایل استفاده می‌شود.

مثال:

فایلی را که حاوی متن است به نام webdictionary.txt در سرور ذخیره کنید.

```
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
```

فایل php که در آن فنکشن (fopen) تعریف شده است و توسط این فنکشن می‌توانیم فایل را باز نموده، متنی را که در «فایل متنی» ذخیره شده است، توسط تابع (fread) بخوانیم.

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

خروجی آن را در مرورگر وب مشاهده کنید:

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

تابع () Fopen دارای دو پارامتر مهم است که اول آن نام فایل و دوم آن تعیین مود می باشد که مود "r" به حالت تنها خواندنی (read only) و مود "w" به حالت تنها نوشتنی (Write only) استفاده می شود.

لست مودهایی که درین تابع استفاده شده می تواند قرار ذیل است:

جدول (۵-۱)

Modes	Description
r	Open a file for read only. File pointer starts at the beginning of the file
w	Open a file for write only. Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a	Open a file for write only. The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x	Creates a new file for write only. Returns FALSE and an error if file already exists
r+	Open a file for read/write. File pointer starts at the beginning of the file
w+	Open a file for read/write. Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a+	Open a file for read/write. The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x+	Creates a new file for read/write. Returns FALSE and an error if file already exists

۵.۴ fclose() بسته نمودن فایل

این تابع برای بسته نمودن فایل استفاده می‌شود.

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

نوشتن در فایل fwrite()

توسط تابع fwrite() اطلاعات در فایل نوشته می‌شود.

```
<?php
$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
$txt1 = "Mohammad\n";
fwrite($myfile, $txt1);
$txt2 = "Hamida\n";
fwrite($myfile, $txt2);
fclose($myfile);
?>
```

۵.۵ PHP file upload

توسط PHP به آسانی می‌توانیم، فایل را در server آپلود کنیم.

اما بعضی اوقات، با این آسانی خطرانی نیز فایل را تهدید می‌کند، وقتی که فایل را برای آپلود اجازه می‌دهید، دقت داشته باشید.

Configure The "php.ini" File

نخست، عیارسازی php را برای آپلود فایل آماده کنید.

مثال:

```
file_uploads = On
```

:Create the HTML Form

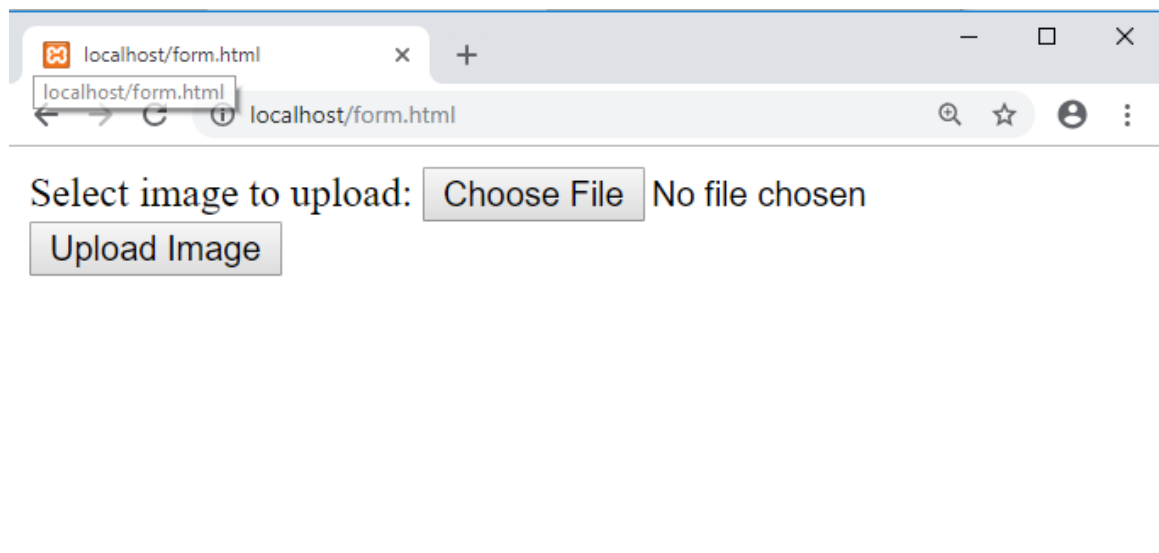
یک فرم HTML را ایجاد می‌کنیم تا استفاده‌کنندگان بتوانند، یک فایل تصویری را در آن آپلود کنند:

```
<!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>
```

خروجی در مرورگر وب:



شکل (۵-۳)

بعضی قوانین برای فرم HTML

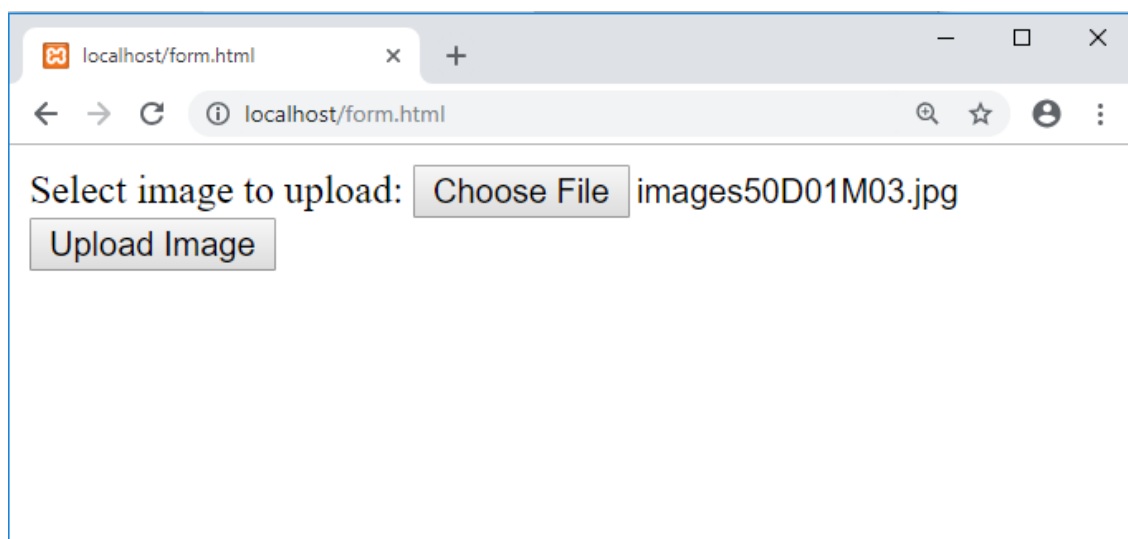
- برای فرم HTML از میتود post استفاده کنید.
 - فرم به بعضی مشخصات زیر نیاز دارد.
 - "Enctype="multipart/form-data"
- از این مشخصه برای ارسال محتویات یک فایل، استفاده می‌شود.

Upload File PHP Script •

Upload.php دارای کدهایی است که برای آپلود فایل استفاده می‌شود.

```
<?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;
    }
}
?>
```

بعد از اجرای کدها در مرورگر وب نتایج را مشاهده می‌کنیم:



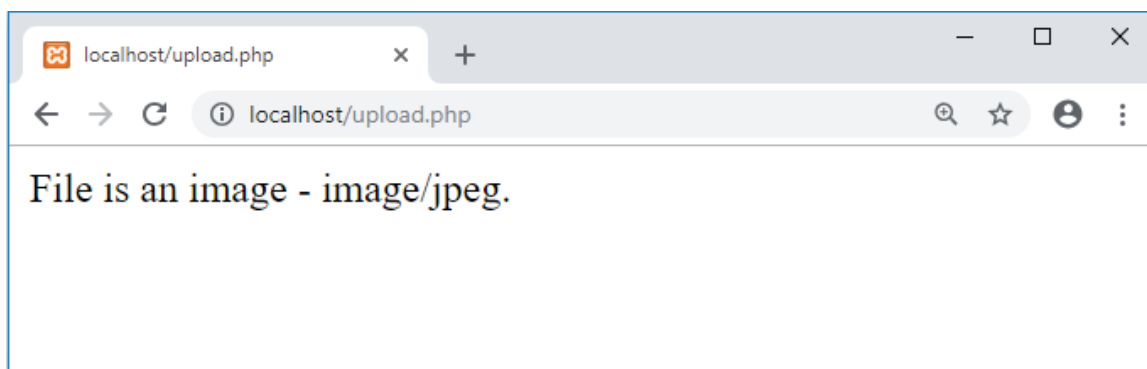
شکل (۴-۵)

در این پنجره دو دکمه موجود است.

Brows: برای افزودن فایل تصویری از کامپیوتر استفاده می‌شود. بعد از انتخاب فایل تصویری آدرس آن در باکس پهلوی دکمه brows نشان داده می‌شود.

Upload: از این دکمه برای ارسال فایل تصویری به سمت سرور استفاده می‌شود.

با کلیک کردن روی دکمهٔ upload فایل را به سمت سرور ارسال کرده، نوعیت فایل را نیز توسط یک پیام نشان می‌دهد.



شکل (۵-۵)

برای بررسی این که آیا فایلی از قبل موجود می‌باشد یا خیر؟ ضرورت به تعریف بعضی محدودیت‌ها می‌باشد. بعد از نوشتن این کدها در فایل PHP در صورتی که فایل از قبل موجود باشد، آن را دوباره نمی‌توانیم Upload کنیم.

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

محدودیت در سایز فایل

در صورتی که سایز فایل بیشتر از 500kb باشد و استفاده کنندگان نتوانند آن را آپلود کنند.

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

محدودیت در نوعیت فایل

اگر بخواهیم فایل را نظر به نوعیت آن آپلود کنیم، از کدهای ذیل استفاده می‌کنیم:

با نوشتن این کود صرف فایل‌هایی آپلود می‌شوند که دارای پسوند jpg,jpeg,png,gif باشند.

در صورتی که فایل با پسوند دیگر باشد، پیام اشتباه ظاهر می‌شود.

```
// 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;
}
```

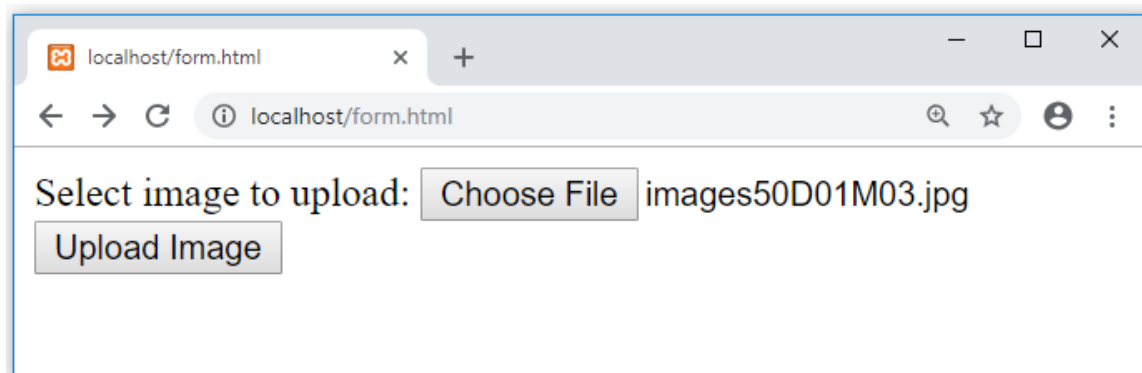
حالا کود مکمل PHP را برای آپلود کردن فایل، قرار ذیل می‌نویسیم:

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

```
// Check if file already exists
if (file_exists($target_file)) {
    echo "Sorry, file already exists.";
    $uploadOk = 0;
}
// Check file size
if ($_FILES["fileToUpload"]["size"] > 500000) {
    echo "Sorry, your file is too large.";
    $uploadOk = 0;
}
// Allow certain file formats
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
&& $imageFileType != "gif" ) {
    echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
    $uploadOk = 0;
}
// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0) {
    echo "Sorry, your file was not uploaded.";
```

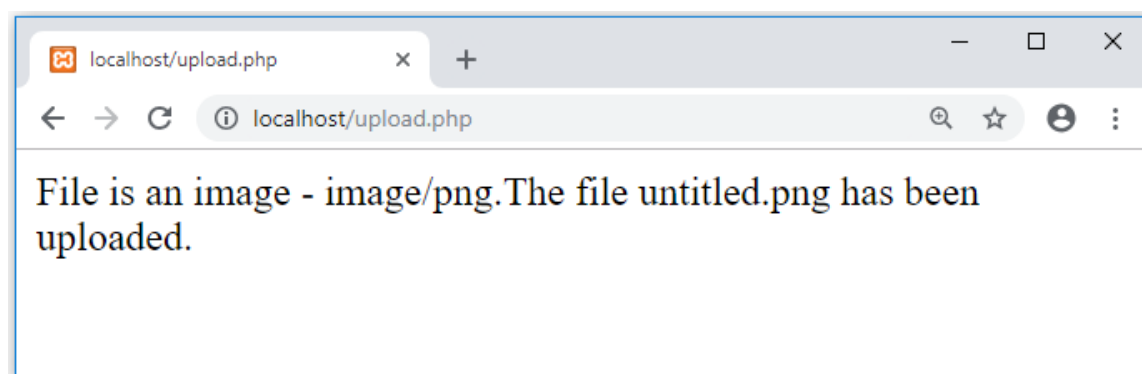
```
// if everything is ok, try to upload file
} else {
    if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)) {
        echo "The file ". basename( $_FILES["fileToUpload"]["name"]). " has been uploaded."
    } else {
        echo "Sorry, there was an error uploading your file.";
    }
}
}
?>
```

خروجی را در مرورگر مشاهده کنید:



شکل (۵-۶)

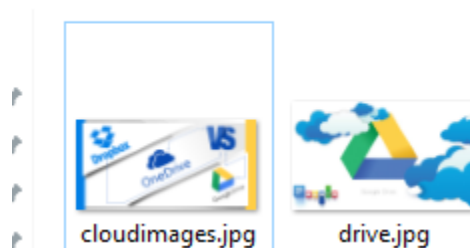
بعد از کلیک کردن بالای دکمه Upload تصویر انتخاب شده به سرور آپلود می شود و پیام در مرورگر به این شکل ظاهر می شود.



شکل (۵-۷)

بعد از اجرای این پروسه در داخل فولدر htdocs، فولدری که به نام uploads است، باز شده، تصاویری که در آنجا آپلود شده است، نشان داده می شود.

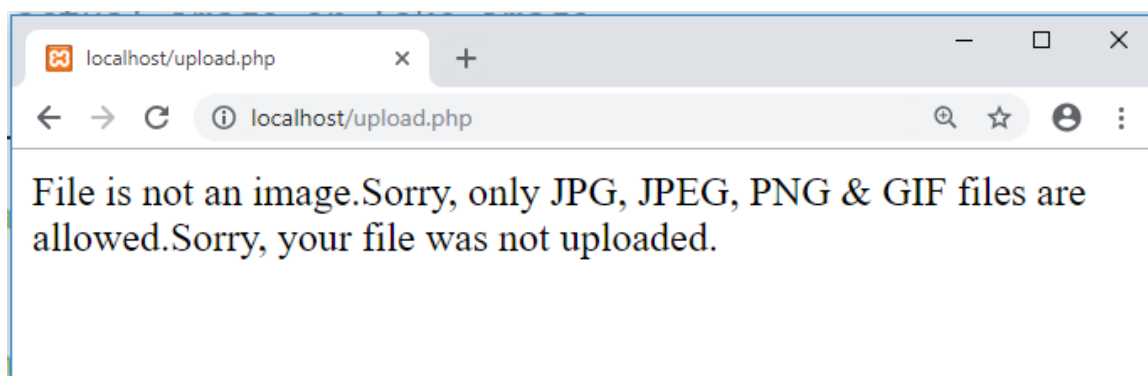
This PC > Windows (C:) > xampp > htdocs > uploads



به مراحل ذیل توجه کنید:

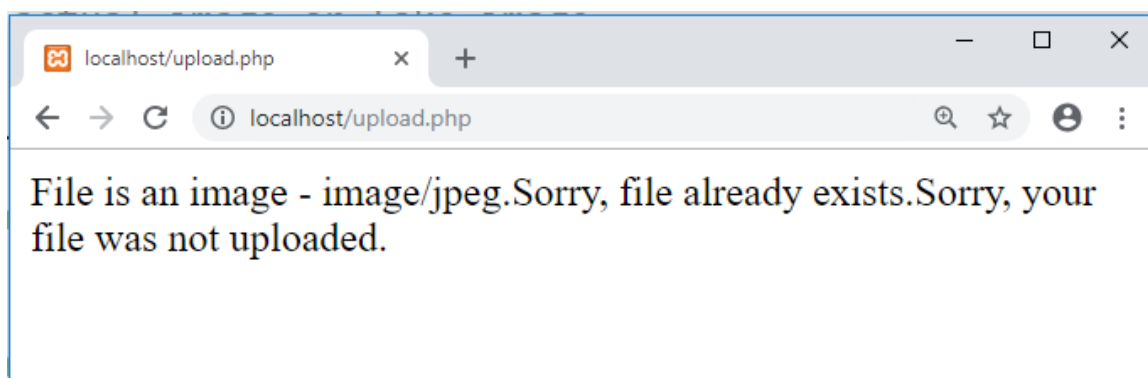
- نخست یک فایل به نام form.html ایجاد کرده، در داخل فولدر htdocs ذخیره کنید.
- یک فایل به نام upload.php ایجاد کرده، سپس کودهای آپلود را در آن بنویسید؛ بعد آن را در فولدر htdocs ذخیره کنید.
- یک فولدر به نام uploads در داخل فولدر htdocs ایجاد کنید تا تصاویر در داخل آن آپلود شود.
- مرورگر وب را باز کرده، آدرس localhost/form.html را در آن بنویسید، بعد اجرای فورم، نشان داده می‌شود.

در صورتی که فایل آپلود به فارمت تصاویر نباشد، پیام اشتباه را مشاهده خواهید کرد:



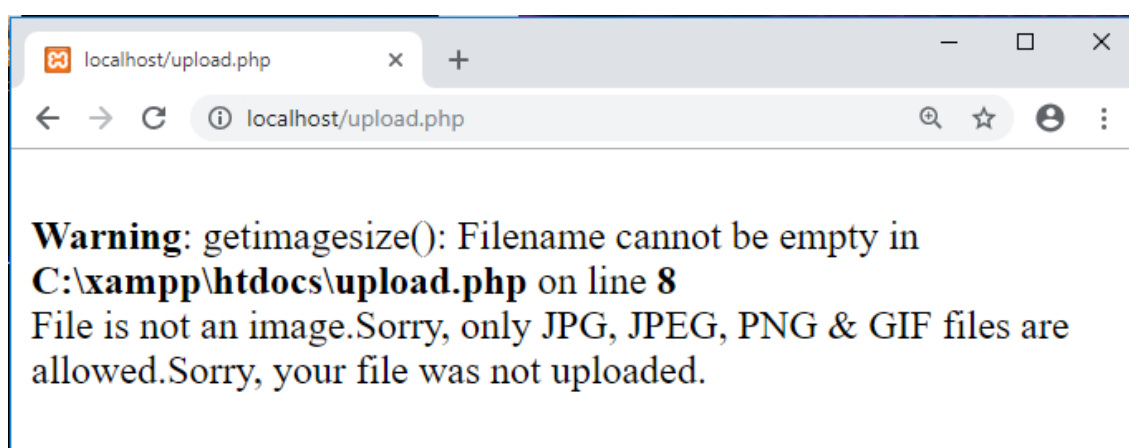
شکل (۵-۸)

در صورتی که فایل از قبل در داخل فولدر موجود باشد، پیام را به این شکل مشاهده خواهید کرد:



شکل (۹-۵)

در صورتی که سایز فایل از حد معین بزرگ باشد، پیام اشتباه را به این شکل مشاهده خواهید کرد:



شکل (۱۰-۵)

۵.۶ PHP cookies

کوکی‌ها معمولاً برای تشخیص هویت استفاده‌کننده، استفاده می‌شوند. کوکی‌ها، فایل‌های کوچکی‌اند که توسط سرور به کامپیوترهای استفاده‌کنندگان جا‌جا می‌شوند.

هر بار که کامپیوتر استفاده‌کننده، از کامپیوتر سرور درخواست می‌کند، سرور در کنار ارسال پاسخ، یک کوکی را نیز ضمیمه آن ارسال می‌کند.

توسط PHP می‌توانیم، یک کوکی را ایجاد و سپس مقادیر آن را دریافت کنیم.

۵.۶.۱ ایجاد کوکی توسط PHP

در PHP یک کوکی توسط فنکشن (setcookie) ایجاد می‌شود.

Syntax

```
setcookie(name, value, expire, path, domain, secure, httponly);
```

در این ساختار تنها پارامتر نام ضروری است، بقیه پارامترها دلخواه می‌باشد.

ایجاد و دریافت کوکی توسط PHP

مثال ذیل یک کوکی به نام user با «مقدار» ایجاد می‌کند. کوکی بعد از 30 روز انقضاء می‌شود، "/" معنای این سمبول این است که کوکی در تمام وب‌سایت موجود است.

بعد می‌توانیم، مقداری را که در کوکی موجود است، دریافت کنیم. با فنکشن (isset) چک نموده که آیا کوکی ست گردیده و مقدار کوکی را از متحول \$_COOKIE دریافت کنیم.

\$_COOKIE از جمله متحول‌های Super global است که برای ذخیره نمودن اطلاعات کوکی استفاده می‌شود.

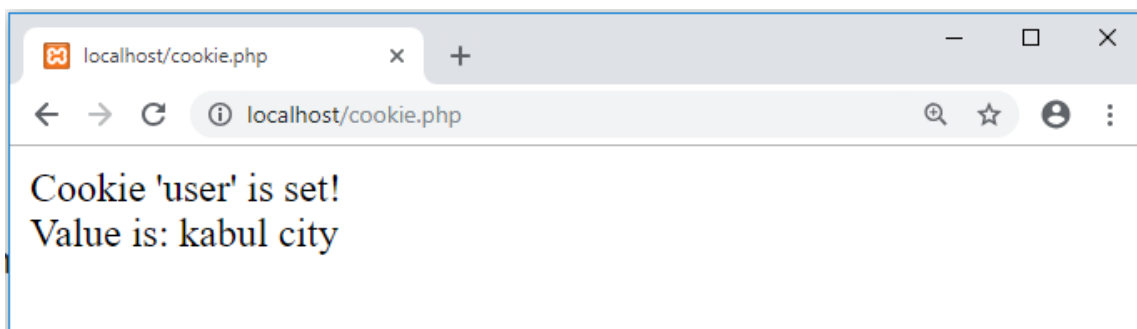
مثال:

```
<?php
$cookie_name = "user";
$cookie_value = "kabul city";
setcookie($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>
```

خروجی را در مرورگر وب مشاهده کنید:



شکل (۵-۱۱)

نوت: فنکشن (setcookie) را قبل از تگ‌های HTML بنویسید.

۵.۶.۲ حذف کوکی

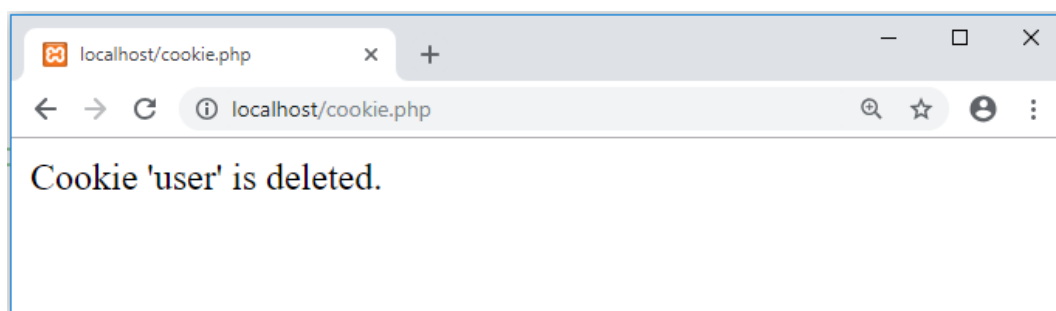
برای حذف کوکی از فنکشن (setcookie) استفاده کرده و تاریخ انقضاء آن را در گذشته بنویسید.

```
<?php
// set the expiration date to one hour ago
setcookie("user", "", time() - 3600);
?>
<html>
<body>

<?php
echo "Cookie 'user' is deleted.";
?>

</body>
</html>
```

خروجی را در مرورگر وب مشاهده کنید:



شکل (۵-۱۲)

۵.۶.۳ بررسی کردن کوکی

برای این که بدانیم که کوکی فعال است یا غیرفعال، فنکشن (setcookie) را نوشته، بعد تعداد صف‌های کوکی را محاسبه کنید.

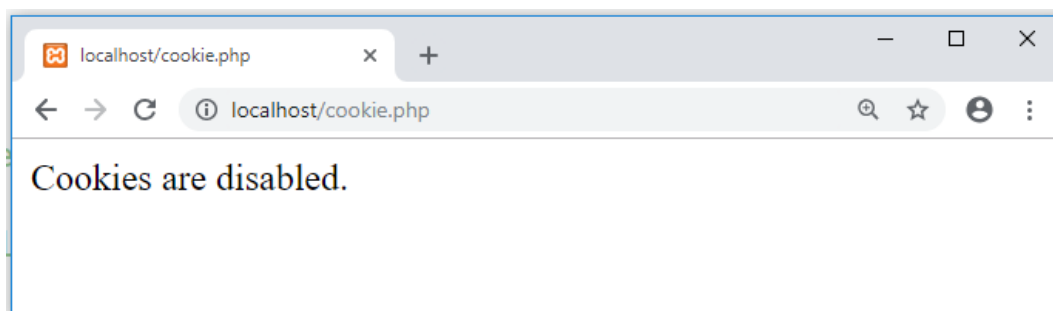
مثال:

```
<?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>
```

خروجی را در مرورگر وب مشاهده کنید:



شکل (۵-۱۳)

۵.۷ PHP session

Session عبارت از روشی است که برای ذخیره کردن «مقادیر» در متحول‌ها به کار می‌رود و در چندین صفحه استفاده می‌شود.

Session، مانند کوکی نیست و معلومات را در کامپیوتر استفاده کننده، ذخیره نمی‌کند.

PHP session

وقتی که ما با یک application کار می‌کنیم، آن را باز کرده، بعضی تغییرات را در آن وارد می‌کنیم و سپس می‌بندیم. این پروسه را به نام session گویند.

کامپیوتر، استفاده کننده را شناخته، هم‌چنان می‌داند که چه وقت کار با application شروع و چه وقت ختم شده است.

اما در اینترنت یک مشکل است؛ وب‌سرور نمی‌داند که شما چه کسی هستید، زیرا آدرس http حالت را نشان نمی‌دهد.

متحول session این مشکل را با ذخیره کردن معلومات استفاده کننده در چندین صفحه حل ساخته است،

متحول‌های session معلومات را به یک استفاده کننده انتقال داده، آن را در تمام application قابل دسترس می‌سازد.

۵.۷.۱ آغاز session در PHP

یک session با فنکشن (session_start) آغاز می‌شود.

\$_SESSION از جمله متحول‌های از قبل تعریف شده زبان PHP است که برای ذخیره نمودن اطلاعات Session استفاده می‌شود.

مثال:

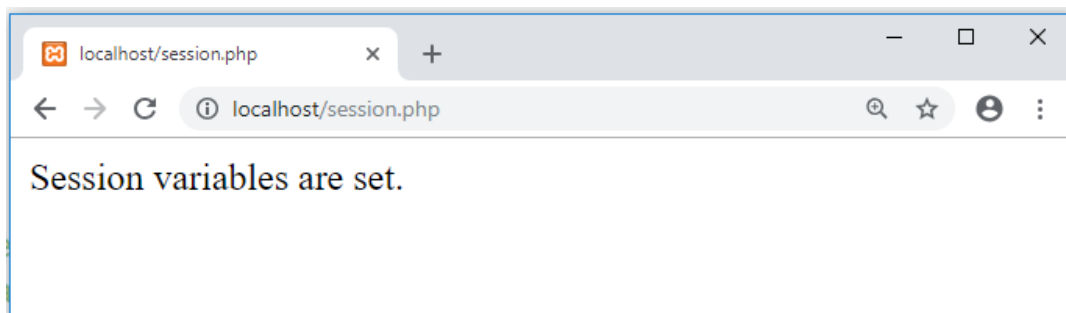
حالا یک صفحه جدید به نام session.php ایجاد کرده، در آن sessionها و متحول‌های session را تعریف می‌کنیم.

```
<?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>
```

خروجی را در مرورگر وب مشاهده کنید:



شکل (۵-۱۴)

نوت: فنکشن (session_start) در شروع فایل، قبل از تگ‌های HTML تعریف شود.

۵.۷.۲ به دست آوردن مقادیر متحول‌های session

یک صفحه جدید به نام demo_session.php ایجاد کرده، توسط آن می‌توانیم، معلوماتی را که در صفحه session.php ذخیره کرده‌ایم، به دست آوریم.

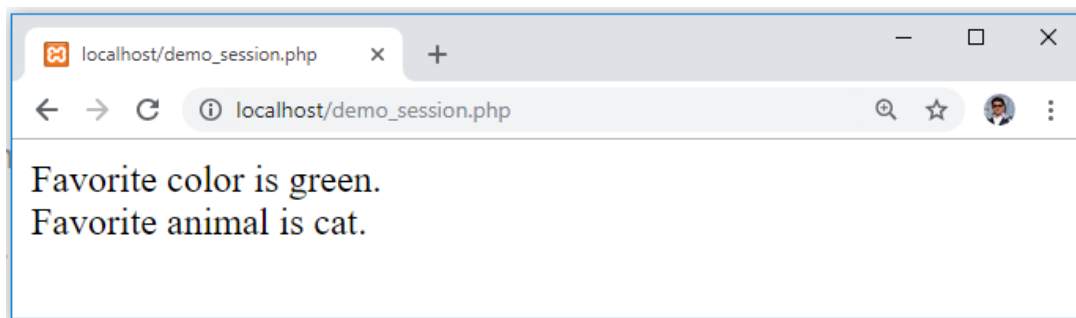
مثال:

```
<?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>
```

خروجی را در وب مشاهده کنید:



شکل (۵-۱۵)

رنگ green و هم‌چنان حیوان دلخواه (پیشک) در فایل دیگر که به نام session است وجود داشته، توسط فایل demo_session.php توانستیم، آن را به دست آوریم.

۵.۷.۳ نابود کردن session PHP

برای حذف متحول‌های session و نابود کردن آن‌ها، از فنکشن‌های (session_unset) و (session_destroy) استفاده می‌شود.

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

<?php
// remove all session variables
session_unset();

// destroy the session
session_destroy();
?>

</body>
</html>
```

خروجی را در مرورگر وب مشاهده کنید:

All session variables are now removed, and the session is destroyed.

۵.۸ PHP Error Handling

Error handling در PHP بسیار ساده است. یک پیام اشتباه با نام و شماره سطر که اشتباه در آن وجود دارد، با تشریح پیام در مرورگر وب نشان داده می‌شود.

وقتی که یک وب اپلیکیشن ایجاد می‌کنیم، Error handling یک بخش اساسی و مهم آن می‌باشد. اگر پروگرام بخش بررسی اشتباهات را نداشته باشد، پروگرام‌تان غیرمسلکی بوده و خطرات امنیتی متوجه آن خواهد شد.

در این جا میتودهای بررسی اشتباهات را نشان می‌دهیم.

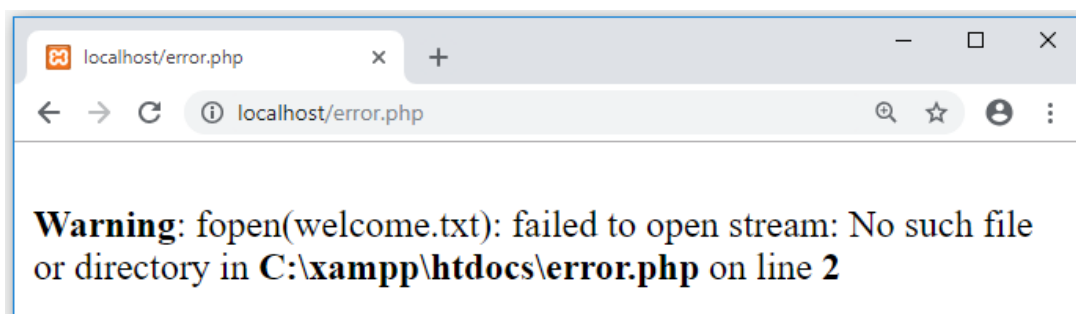
- Die() statement
- Custom errors and error triggers
- Error reporting

Error handling ابتدایی توسط فنکشن (die)

به مثال توجه کنید:

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

در صورتی که فایل موجود نباشد، در مرورگر وب پیام اشتباه به شکل ذیل نشان داده می‌شود:

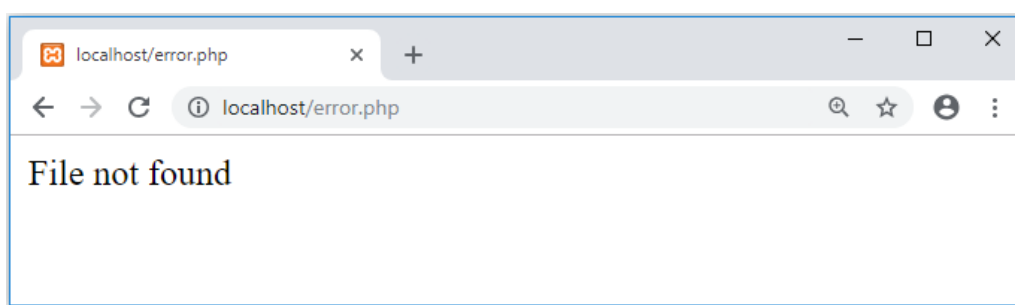


شکل (۵-۱۶)

برای جلوگیری از پیام اشتباه برای استفاده کننده، مانند پیام بالا، اگر فایل موجود باشد می توانیم به آن دسترسی داشته باشیم.

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

خروجی در مرورگر وب به شکل ذیل نشان داده می شود:



شکل (۵-۱۷)

پروگرام دومی نسبت به پروگرام اولی بیشتر کاربردی است، چون در این پروگرام یک error handling ساده و با ساختار منظم برای نشان دادن اشتباه استفاده شده است.

۵.۸.۱ Set Error Handler

Error handler پیش فرض در PHP عبارت از error handler از قبل تهیه شده؛ ما می خواهیم با تعریف یک فنکشن error handler جدید را معرفی کنیم.

اگر بخواهیم، error handler را برای بعضی اشتباهات تغییر دهیم، هم امکان پذیر می باشد. در آن صورت اسکریپت ها اشتباهات مختلف را از روش های مختلف دریافت می کنند.

مثال: برای تعریف error handler انتخابی برای تمام اشتباهات

Syntax

```
set_error_handler("customError");
```

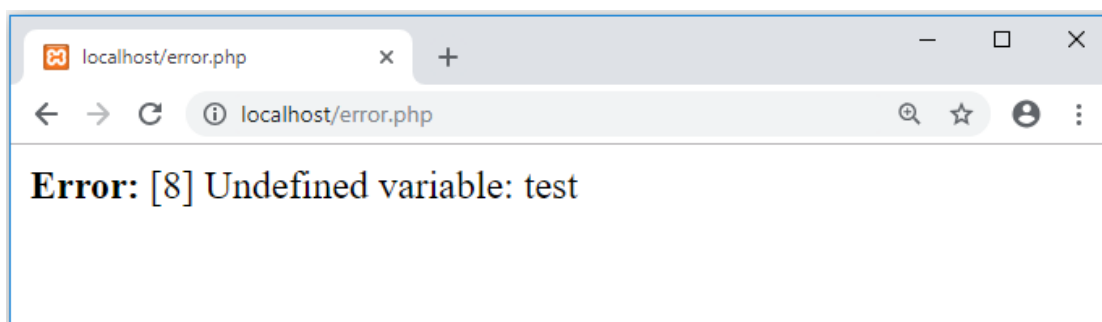
مثال:

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

//set error handler
set_error_handler("customError");

//trigger error
echo($test);
?>
```

خروجی را در مرورگر وب مشاهده کنید:



شکل (۵-۱۸)

Trigger an Error

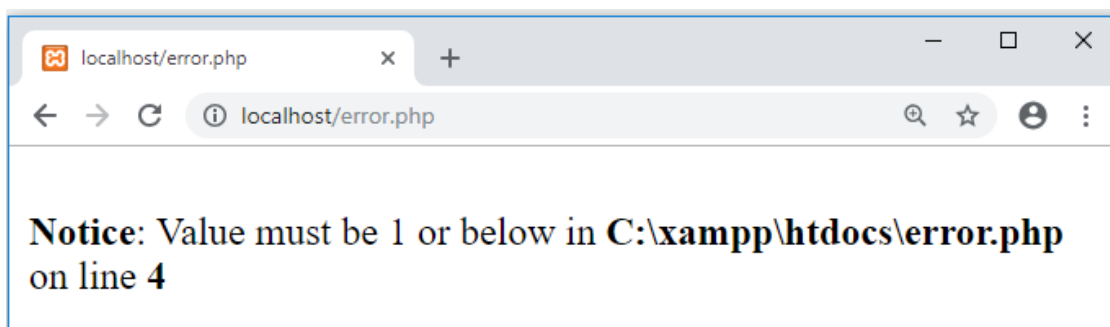
یک اسکریپت که استفاده کنندگان اطلاعات را به آن وارد نمایند، می‌توانیم، در آن trigger errors را تعریف کنیم.

در PHP توسط فنکشن (trigger_error) تعریف می‌شود.

مثال:

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

خروجی را در مرورگر وب مشاهده کنید:



شکل (۵-۱۹)

۵.۸.۲ اقسام اشتباهات

E_USER_ERROR: عبارت از fatal error است که در وقت اجرای پروگرام ایجاد می‌شود.

E_USER_WARNING: یک اشتباه fatal نبوده، در صورتی که exception درست نباشد، این نوع error به‌وقوع می‌پیوندد.

E_USER_NOTICE: عبارت از نوعی پیام اشتباه است و در صورتی رخ می‌دهد که بعضی کودها اشتباه نوشته شده باشند.

مثال:

```

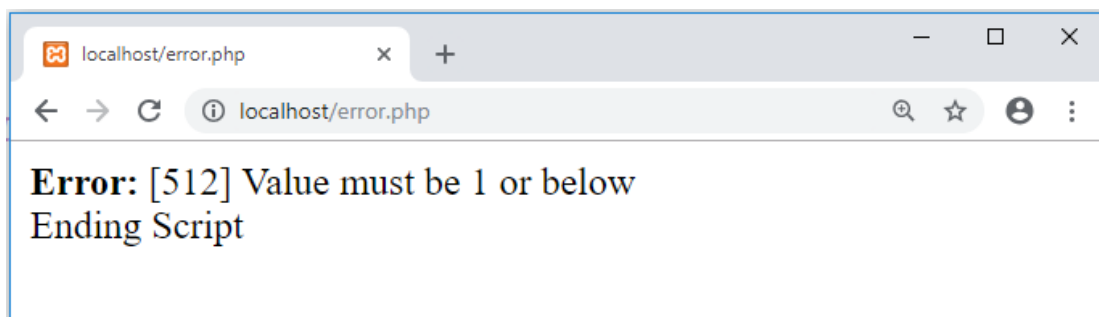
<?php
//error handler function
function customError($errno, $errstr) {
    echo "<b>Error:</b> [$errno] $errstr<br>";
    echo "Ending Script";
    die();
}

//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 را در مرورگر وب مشاهده کنید:



شکل (۵-۲۰)

۵.۹ PHP Exception Handling

عبارت از پروسه‌یی است، برای تبدیل کردن جریان کودها به شکل نورمال و در صورتی که اشتباه رخ دهد، استفاده می‌شود.

یک اشتباه در صورتی که با یک شرط بررسی گردد، آن را به نام exception گویند.

۵.۹.۱ Use of Exception

وقتی که یک اکسپشن thrown باشد، کودهای ذیل اجرا نمی‌شود، PHP تلاش می‌کند تا بلاکی را که با آن مطابقت می‌کند، پیدا کند.

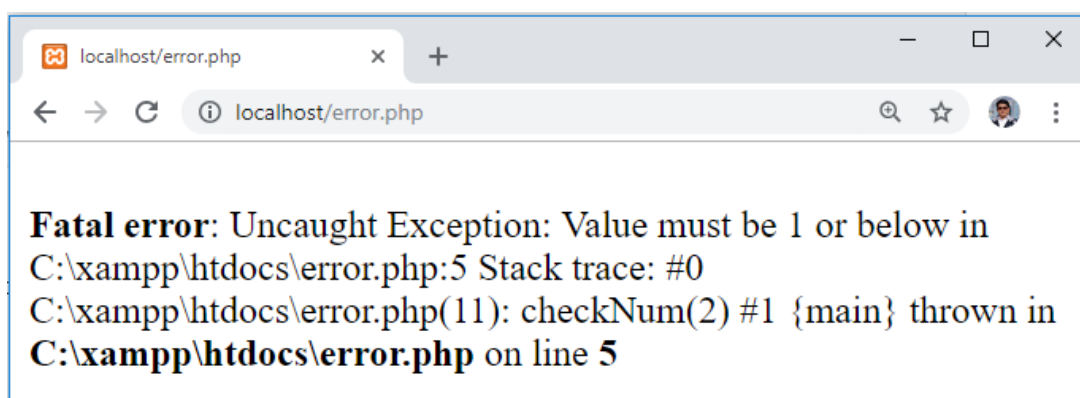
اگر یک اکسپشن گرفته نشود، یک fatal error با پیام uncaught exception رخ می‌دهد.

مثال:

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

خروجی را در مرورگر وب مشاهده کنید:



شکل (۵-۲۱)

۵.۹.۲ Try, throw and catch

برای جلوگیری از اشتباهاتی که در مثال بالا دیده می‌شود، ضرورت است، یک کود مناسب ایجاد گردد تا اشتباه را دریافت کند.

کود مناسب شامل بخش‌های ذیل می‌باشد:

- **Try:** یک فنکشن باید در بلاک try استفاده شود، اگر اکسپشن trigger نباشد، کود به شکل نورمال ادامه می‌یابد. اگر اکسپشن trigger باشد، در آن صورت اکسپشن thrown می‌باشد.
- **Throw:** در این جا نشان می‌دهد که یک اکسپشن trigger است. هر throw باید یک بلاک catch داشته باشد.

- **Catch**: یک بلاک catch یک اکسپشن را دریافت کرده، یک آبجکت ایجاد می‌کند که محتویات آن معلومات در مورد اکسپشن می‌باشد.

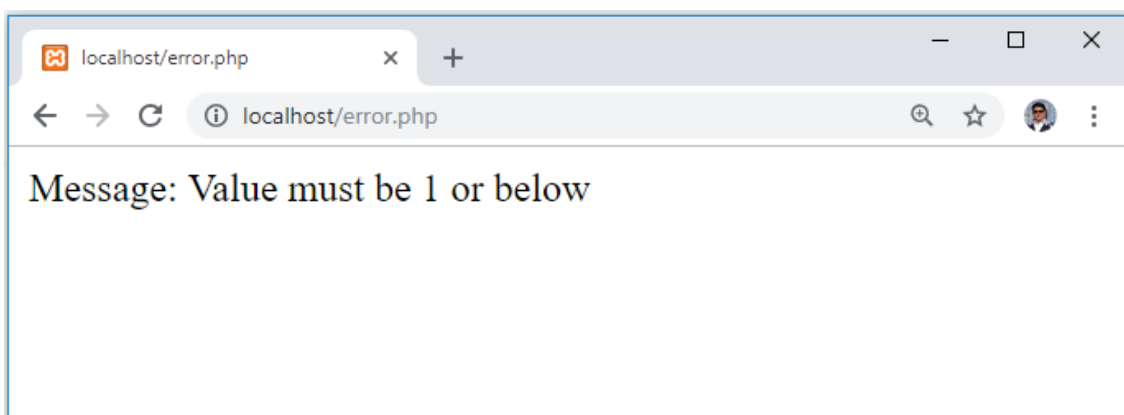
مثال:

```
<?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();
}
?>
```

خروجی را در مرورگر وب مشاهده کنید:



شکل (۵-۲۲)



خلاصه‌ی فصل پنجم

در این فصل با مفاهیم پیشرفته‌ی زبان برنامه‌نویسی PHP آشنایی حاصل کرده، مهم‌ترین مطالب آن را مورد بحث قرار دادیم. برای برنامه‌نویسان وب، دانستن مطالب PHP including یکی از بحث‌های کاربردی است و استفاده بسیار زیاد دارد. هم‌چنان بحث PHP file handling نیز کاربردی است و در دیزاین و توسعه صفحات وب بیشتر استفاده می‌شود. PHP file upload بحث مهم و کاربردی جهت آپلود فایل‌ها از کمپیوترهای کلاینت به کمپیوترهای سرور می‌باشد. PHP cookies and session نیز از بحث‌های مهم و کاربردی به شمار می‌رود و برنامه‌نویسان وب از آن‌ها در توسعه وبسایت‌ها استفاده زیاد می‌کنند. تمام مفاهیم مهم و کاربردی در این فصل به‌طور مفصل با ارائه مثال‌ها و کاربرد علمی، اجرا شدند.



سوالات فعالیت های فصل پنجم

۱. موارد استفاده PHP include را شرح دهید.
۲. موارد استفاده PHP file handling را توضیح دهید.
۳. برای آپلود کردن فایل ها، کدام شرایط مهم می باشد؟
۴. از cookie به چه هدف استفاده می شود؟
۵. از session در کجا استفاده می شود؟
۶. مفهوم exception چیست؟
۷. چرا از exception handling استفاده می شود؟

فعالیت انفرادی

- پروگرامی را بنویسید که توسط آن یک فایل به شکل include در فایل دیگر استفاده شود.
- پروگرامی را بنویسید که توسط آن File های تصویری با پسوندهای JPG, JPEG, PNG و غیره از کمپیوتر کلاینت به کمپیوتر سرور آپلود شود.
- پروگرامی را بنویسید که توسط آن یک cookie ایجاد شود.
- پروگرامی را بنویسید که توسط آن یک session ایجاد شود.
- پروگرامی را بنویسید که در آن از exception handling استفاده شود.

فعالیت گروهی

- در مورد کاربرد تابع های ذیل جستجو نمایید.
- fgets();
- fgetc();
- feof();
- rename();
- unlink();

فصل ششم

PHP & MYSQL



هدف کلی: در این فصل، محصلان به مفاهیم PHP و MySQL آشنا خواهند شد.

اهداف آموزشی: در پایان این فصل محصلان قادر خواهند شد تا:

۱. PHP – MySQL را تعریف بتوانند.
۲. MySQL Connect i on را شرح دهند.
۳. MySQL- DDL را بدانند.
۴. MySQL- DM را تشریح کنند.
۵. MySQL-DRL را شرح دهند.
۶. MySQL-Get l i s t I D را تشریح کنند.
۷. MySQL l i n i t d a را شرح دهند.

در این فصل بحث اساسی بالای اتصال صفحات انترفس با دیتابیس^۴ MySQL، ذخیره کردن ریکاردها^۵ در دیتابیس، حذف کردن ریکاردها در دیتابیس، آپدیت^۶ کردن ریکاردها در دیتابیس و انتخاب کردن ریکاردها از داخل دیتابیس صورت گرفته است. این فصل مهم ترین بخش برای توسعه دادن صفحات دینامیک در وبسایت ها می باشد، پس نظر به فصول دیگر، روی این فصل تمرکز بیشتر داشته باشید.

۶.۱ PHP and MySQL

یکی از قابلیت های PHP اتصال Application به دیتابیس است که توسط آن می توانیم در دیتابیس تغییرات به وجود بیاوریم.

MySQL، یکی از دیتابیس های معروف است که با PHP در وبسایت های دینامیک استفاده می شود.

۶.۱.۱ MySQL

- MySQL، عبارت از آن سیستم دیتابیس است که در وب استفاده می شود.
- MySQL، عبارت از آن سیستم دیتابیس است که در سرور اجرا می شود.
- MySQL، عبارت از سیستم دیتابیس مناسب برای اپلیکیشن های کوچک و بزرگ می باشد.
- MySQL، عبارت از سیستم دیتابیس سریع قابل اطمینان و آسان می باشد.
- MySQL، عبارت از سیستم دیتابیس رایگان است.
- MySQL، عبارت از آن سیستم دیتابیس است که توسط شرکت «اوراکل» مدیریت می شود.

اطلاعات در دیتابیس MySQL، در داخل جدول ها ذخیره می شود. جدول عبارت از مجموعه دیتای باهم مرتبط است که شامل سطر و ستون می باشد.

دیتابیس اطلاعات را به شکل دسته بندی در جدول ها ذخیره می کند.

مثلاً، دیتابیس یک شرکت می تواند دارای جدول های ذیل باشد:

- Employee؛
- Product؛
- Customers؛
- Orders.

⁴ Database

⁵ Record

⁶ Update

۶.۲ دیتابیس کیوری (database query)

کیوری عبارت از سؤال یا یک درخواست است.

به کیوری‌های ذیل ببیند که با استاندارد SQL نوشته شده‌اند:

```
MariaDB [(none)]> SELECT LastName FROM Employees;
```

این کیوری دیتا را از ستون LastName جدول Employee انتخاب می‌کند.

۶.۳ Connect to MySQL

بازکردن یک ارتباط با دیتابیس MySQL

قبل از این که به اطلاعات یک دیتابیس از طریق اپلیکیشن دسترسی داشته باشیم، باید نخست اپلیکیشن را به دیتابیس سرور وصل کنیم.

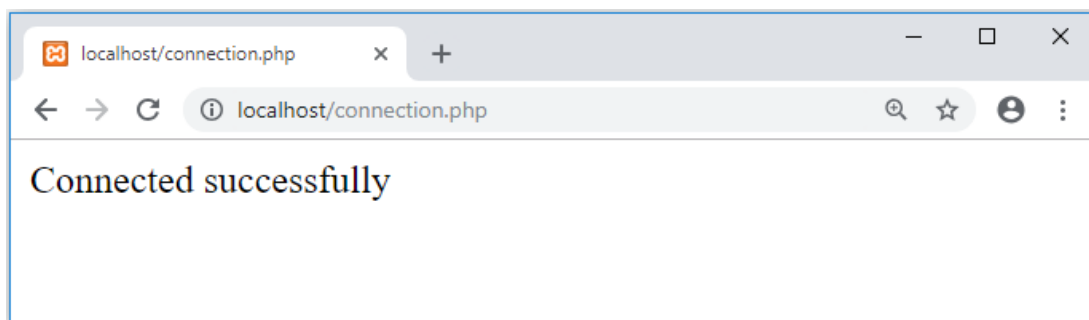
مثال:

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

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

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

خروجی را در مرورگر وب مشاهده کنید:



شکل (۱-۶)

برای اتصال به دیتابیس، سه پارامتر را در نظر بگیرید.

نام سرور: localhost

نام استفاده کننده: در این جا خالی است.

پسورد: در این جا خالی است.

بعد از بررسی پارامترها، به شرایط رفته، در صورتی که درست باشد، پیام connected successfully نشان داده می شود.

۶.۳.۱ Create a MySQL Database

برای ایجاد یک دیتابیس در MySQL توسط اسکریپت PHP

این مثال، یک دیتابیس را به نام myDB در MySQL ایجاد می کند.

```

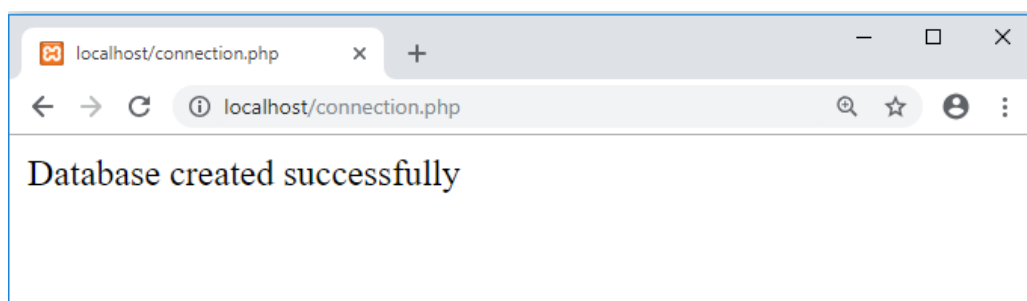
<?php
$servername = "localhost";
$username = "root";
$password = " ";

// Create connection
$conn = mysqli_connect($servername, $username);
// Check connection
if (!$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);
?>

```

خروجی را در مرورگر وب مشاهده کنید:



شکل (۶-۲)

در دیتابیس MySQL لیست دیتابیس‌ها را مشاهده کنید:

```

MariaDB [(none)]> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| mydb              |
| mysql             |
| performance_schema |
| phpmyadmin        |
| test              |
+-----+
6 rows in set (0.00 sec)

```

۶.۳.۲ Create MySQL Tables

جدول در دیتابیس برای ذخیره کردن اطلاعات استفاده می‌شود. در داخل یک دیتابیس می‌توانیم، چندین جدول داشته باشیم؛ اما نام‌هایشان باید متفاوت باشد.

یکی از خصوصیت‌های مهم جدول این است که کم‌از کم دارای یک سطر و ستون باشد.

مثال:

در این مثال، یک جدول به نام MyGuests با ستون‌های ID, FirstName, LastName, Reg_date ایجاد می‌شود.

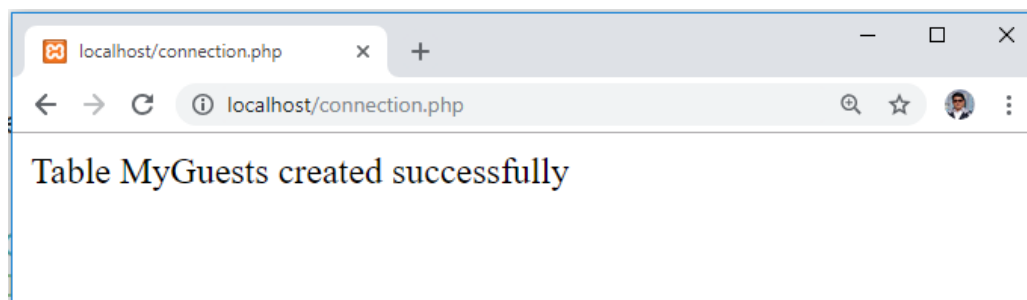
```

<?php
$servername = "localhost";
$username = "root";
$password = "123";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// 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();
?>

```

بعد از اجرای کدها نتیجه را در مرورگر وب مشاهده کنید:



شکل (۳-۶)

۶.۵ PHP Insert Data into MySQL

برای ذخیره کردن اطلاعات در داخل دیتابیس، از قابلیت‌های PHP استفاده می‌کنیم؛ البته بعد از ایجاد دیتابیس، سپس در داخل دیتابیس جدول ایجاد کرده، بعد داخل جدول می‌توانیم دیتا را ذخیره کنیم. برای ذخیره کردن اطلاعات در داخل جدول از insert استفاده می‌کنیم که ساختار آن در PHP قرار ذیل می‌باشد:

Syntax

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

مثال:

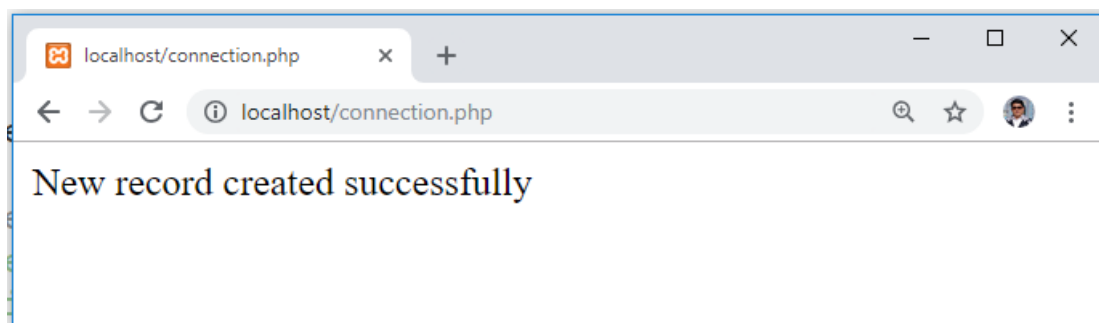
در این مثال یک ریکارد در داخل جدول MyGuest ذخیره می‌شود.

```
<?php
$servername = "localhost";
$username = "root";
$password = "123";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Insert record in table MyGuest
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Nargiss', 'Sahil', 'sahil@gmail.com)";

if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>
```

بعد از اجرای کود، نتیجه را در مرورگر وب مشاهده کنید:



شکل (۴-۶)

در داخل جدول myguests می‌توانیم ریکاردهای ذخیره‌شده را مشاهده کنیم:

```
MariaDB [MyDB]> select * from myguests;
```

id	firstname	lastname	email	reg_date
4	Mateen	Jalal	jalal@gmail.com	2019-02-06 20:46:45
5	Nargiss	Sahil	sahil@gmail.com	2019-02-06 20:47:38

```
2 rows in set (0.00 sec)
```

۶.۶ PHP Get ID of Last Inserted Record

اگر ما بیانیه‌های insert, Update را در یک جدول توسط auto_increment اجرا کنیم، برای دیدن آخرین ID که در جدول ثبت شده است، از این قابلیت PHP استفاده می‌کنیم.

مثال:

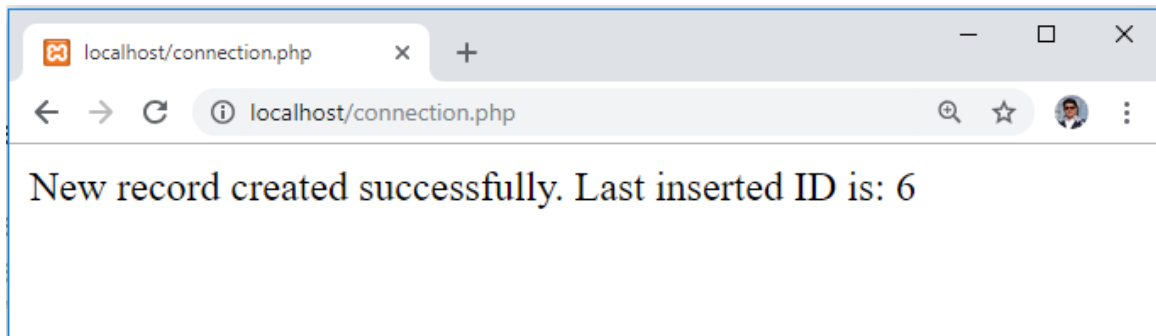
این مثال یک ریکارد ثبت‌شده و هم‌چنان ID نمبر را نشان می‌دهد.

```
<?php
$servername = "localhost";
$username = "root";
$password = "123";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Insert record in table MyGuest
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Nadim', 'Shoja', 'shoja@gmail.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();
?>
```

نتیجه را در مرورگر وب مشاهده کنید:



شکل (۵-۶)

همچنان در جدول داخل دیتابیس نیز تعداد ریکاردها را می بینیم:

```
MariaDB [MyDB]> select * from myguests;
```

id	firstname	lastname	email	reg_date
4	Mateen	Jalal	jalal@gmail.com	2019-02-06 20:46:45
5	Nargiss	Sahil	sahil@gmail.com	2019-02-06 20:47:38
6	Nadim	Shoja	shoja@gmail.com	2019-02-06 20:58:33

۶.۷ PHP Insert Multiple Records into MySQL

اگر بخواهیم چندین ریکارد به طور همزمان در جدول ثبت شود، توسط فنکشن (mysqli_multi_query) این عمل را انجام می دهیم.

مثال:

در این مثال، چندین ریکارد به طور همزمان در جدول myguests ثبت می شود:

```

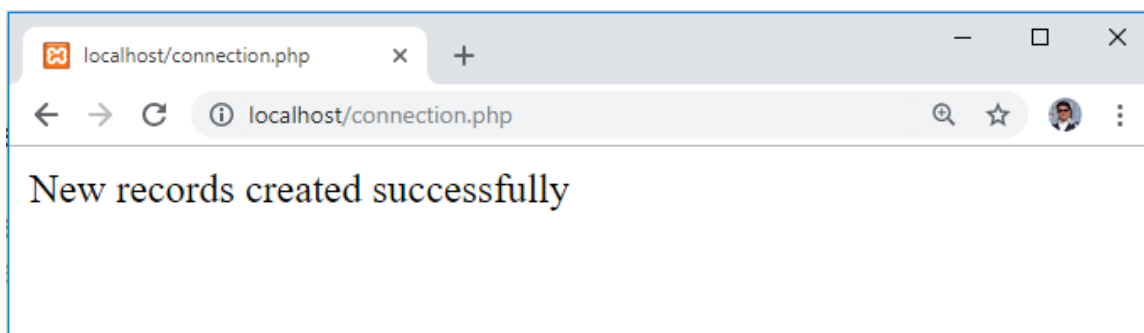
<?php
$servername = "localhost";
$username = "root";
$password = "123";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Insert record in table MyGuest
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Ali', 'Sadat', 'sadat@gmail.com')";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Mary', 'Mobeen', 'mobeen@gmail.com')";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Shahlla', 'Zolal', 'zolal@gmail.com')";

if ($conn->multi_query($sql) === TRUE) {
    echo "New records created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>

```

نتیجه را در مرورگر وب قرار ذیل مشاهده می‌کنیم:



شکل (۶-۶)

در جدول داخل دیتابیس نیز این تغییرات را مشاهده می‌کنیم:

```
MariaDB [MyDB]> select * from myguests;
```

id	firstname	lastname	email	reg_date
4	Mateen	Jalal	jalal@gmail.com	2019-02-06 20:46:45
5	Nargiss	Sahil	sahil@gmail.com	2019-02-06 20:47:38
6	Nadim	Shoja	shoja@gmail.com	2019-02-06 20:58:33
7	Ali	Sadat	sadat@gmail.com	2019-02-06 21:08:52
8	Mary	Mobeen	mobeen@gmail.com	2019-02-06 21:08:52
9	Shahlla	Zolal	zolal@gmail.com	2019-02-06 21:08:52

در جدول مشاهده می‌شود که سه ریکارد اضافه شده است.

۶.۸ PHP Select Data from MySQL

توسط این قابلیت PHP می‌توانیم، ریکاردهای داخل جدول را مشاهده کنیم:

Syntax

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

مثال:

توسط این مثال، می‌توانیم با انتخاب ID مشخصات آن را مشاهده کنیم.

۶.۸.۱ تابع fetch_assoc()/mysqli_fetch_assoc()

این تابع برای آوردن یک سطر جدول به شکل Associative Array استفاده می‌شود.

```

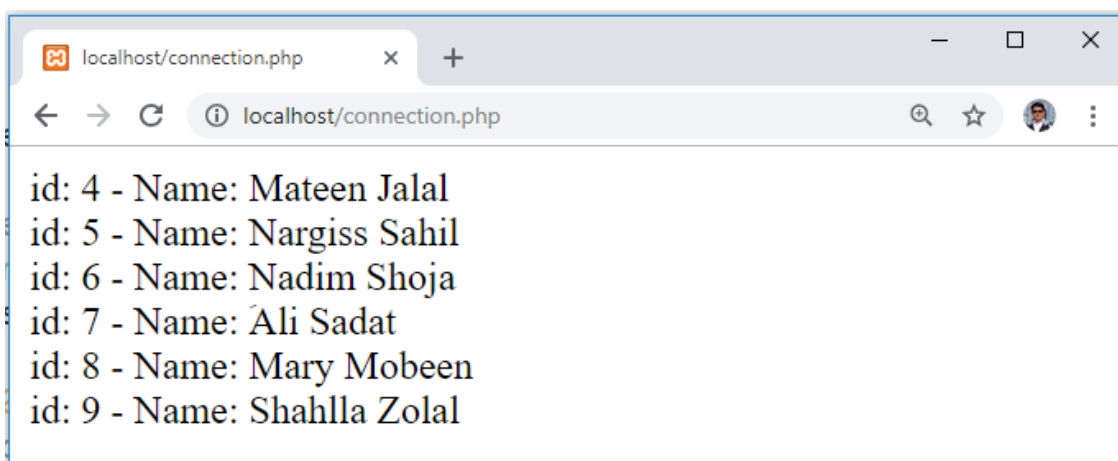
<?php
$servername = "localhost";
$username = "root";
$password = "123";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Insert record in table MyGuest
$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();
?>

```

نتیجه را در مرورگر وب مشاهده کنید.



شکل (۶-۷)

۶.۹ PHP Delete Data from MySQL

از بیانیه delete برای حذف کردن ریکاردها از جدول، استفاده می شود:

Syntax

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

مثال:

برای حذف کردن یک ریکارد مشخص از جدول، از این مثال استفاده کنید:

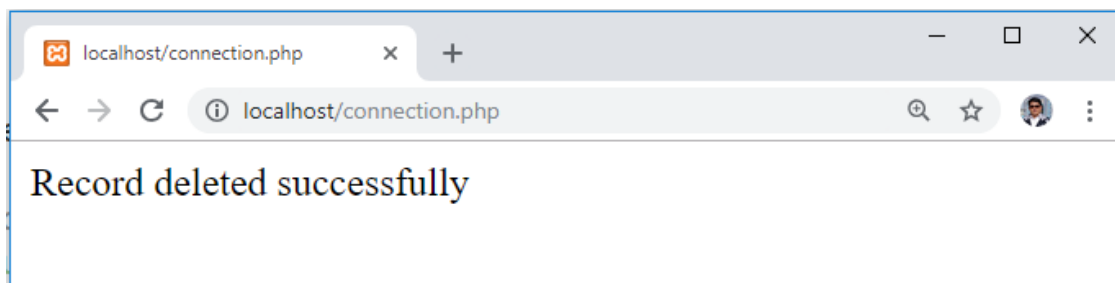
```
<?php
$servername = "localhost";
$username = "root";
$password = "123";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=6";

if ($conn->query($sql) === TRUE) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . $conn->error;
}

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

نتیجه را در مرورگر وب مشاهده کنید:



شکل (۸-۶)

می‌بینیم که یک ریکارد مطابق ID که در کد مشخص شده بود، حذف شد.

حالا جدول را در دیتابیس مشاهده کنید.

```
MariaDB [MyDB]> select * from myguests;
```

id	firstname	lastname	email	reg_date
4	Mateen	Jalal	jalal@gmail.com	2019-02-06 20:46:45
5	Nargiss	Sahil	sahil@gmail.com	2019-02-06 20:47:38
7	δ?Ali	Sadat	sadat@gmail.com	2019-02-06 21:08:52
8	Mary	Mobeen	mobeen@gmail.com	2019-02-06 21:08:52
9	Shahlla	Zolal	zolal@gmail.com	2019-02-06 21:08:52

می بینیم که ریکارد ۶ حذف شده است.

۶.۱۰ PHP Update Data in MySQL

این قابلیت PHP برای update کردن یک ریکارد در داخل دیتابیس استفاده می شود.

Syntax

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

مثال:

در این مثال، بعضی مشخصات یک ریکارد را Update می کنیم:

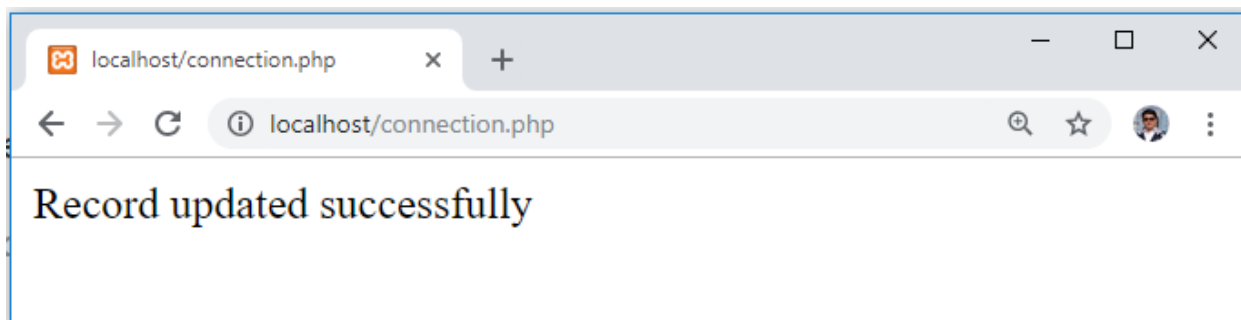
```
<?php
$servername = "localhost";
$username = "root";
$password = "123";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// sql to update a record
$sql = "UPDATE MyGuests SET lastname='Ahmadi' WHERE id=8";

if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . $conn->error;
}

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

نتیجه را در مرورگر وب مشاهده کنید:



شکل (۹-۶)

پروژه موفقانه اجرا شده و ریکارد هشتم تخلص شان از Mobeen به احمدی تغییر کرده است.

تغییرات را در جدول دیتابیس مشاهده کنید:

```
MariaDB [MyDB]> select * from myguests;
```

id	firstname	lastname	email	reg_date
4	Mateen	Jalal	jalal@gmail.com	2019-02-06 20:46:45
5	Nargiss	Sahil	sahil@gmail.com	2019-02-06 20:47:38
7	δ?Ali	Sadat	sadat@gmail.com	2019-02-06 21:08:52
8	Mary	Ahmadi	mobeen@gmail.com	2019-02-06 21:54:15
9	Shahlla	Zolal	zolal@gmail.com	2019-02-06 21:08:52

۶.۱۱ PHPLimit Data Selections from MySQL

در MySQL می‌توانیم انتخاب ریکاردها را توسط LIMIT clause محدود بسازیم. این clause ما را کمک می‌کند تا ریکارد مورد نیاز خویش را انتخاب کرده، از نمایش ریکاردهای غیر ضروری جلوگیری کنیم.

Syntax

```
$sql = "SELECT * FROM Orders LIMIT 30";
```

وقتی که این کود اجرا گردد، ریکارد 1 تا 30 نشان داده شده، بقیه ریکاردها را نشان نمی‌دهد.

مثال:

در این مثال تعداد ریکاردها را محدود می‌سازیم به 2 ریکارد، بعد از اجرای کود، از بین ریکاردها صرف دو ریکارد نشان داده می‌شود.

```

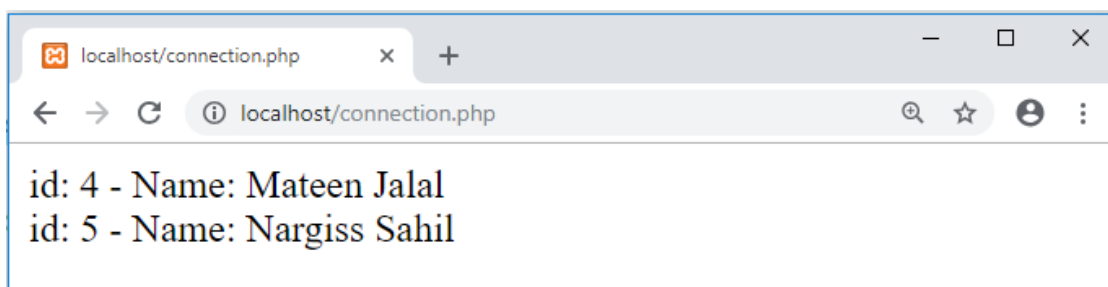
<?php
$servername = "localhost";
$username = "root";
$password = "123";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// sql to update a record
$sql = "SELECT id, firstname, lastname FROM MyGuests LIMIT 2";
$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();
?>

```

نتیجه را در مرورگر وب مشاهده کنید:



شکل (۶-۱۰)

۶.۱۲ پروژه (Project)

برای ذخیره کردن، حذف کردن، ویرایش کردن و نشان دادن ریکاردها ضرورت است که یک محیط گرافیکی ایجاد گردد تا استفاده کنندگانی که غیرمسلکی هستند، بتوانند، ریکاردها را به دیتابیس ذخیره، حذف، ویرایش و انتخاب کنند.

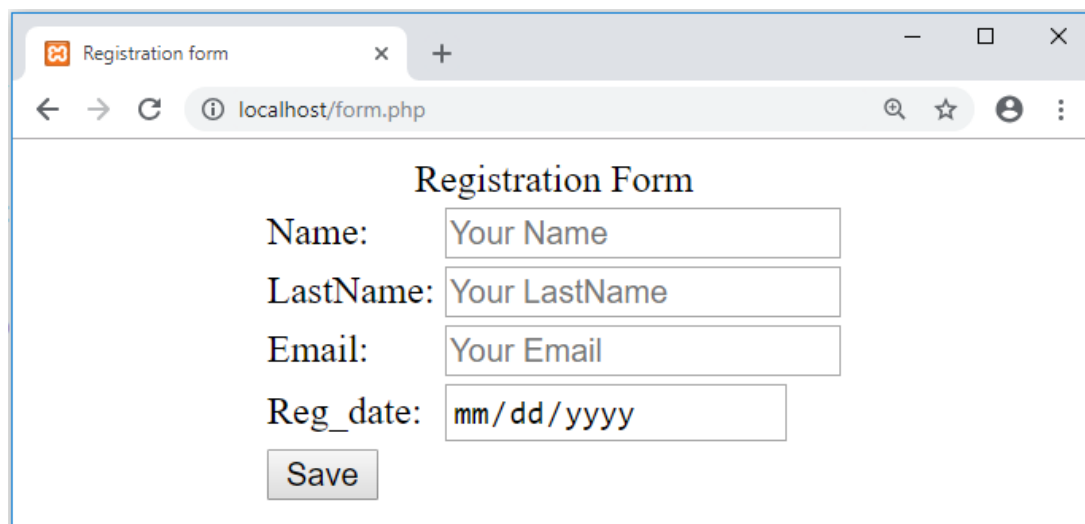
برای ایجاد یک محیط گرافیکی از تگ‌های HTML استفاده می‌شود. در صورتی که بخواهیم شکل ظاهری صفحات گرافیکی را «استایل» دهیم، از قابلیت‌های CSS استفاده می‌کنیم.

ایجاد فورم

توسط تگ‌های HTML یک فورم ایجاد می‌کنیم که در آن تکست باکس‌ها برای ورود دیتا و بتن، برای ارسال دیتا از کامپیوتر کلاینت به کامپیوتر سرور باشد.

```
<html>
<head>
<title>Registration form</title>
</head>
<body>
<table align="center">
<caption>Registration Form</caption>
<form>
<tr><td>
Name:</td><td><input type="text" name="name" placeholder="Your Name">
</td></tr>
<tr><td>
LastName:</td><td><input type="text" name="lname" placeholder="Your LastName">
</td></tr>
<tr><td>
Email:</td><td><input type="email" name="email" placeholder="Your Email">
</td></tr>
<tr><td>
Reg_date:</td><td><input type="date" name="rgdate" placeholder="Reg_date">
</td></tr>
<tr><td>
<input type="submit" value="Save">
</td></tr>
</form>
</table>
</body>
</html>
```

نتیجه را در مرورگر وب مشاهده کنید:



Registration Form

Name:

LastName:

Email:

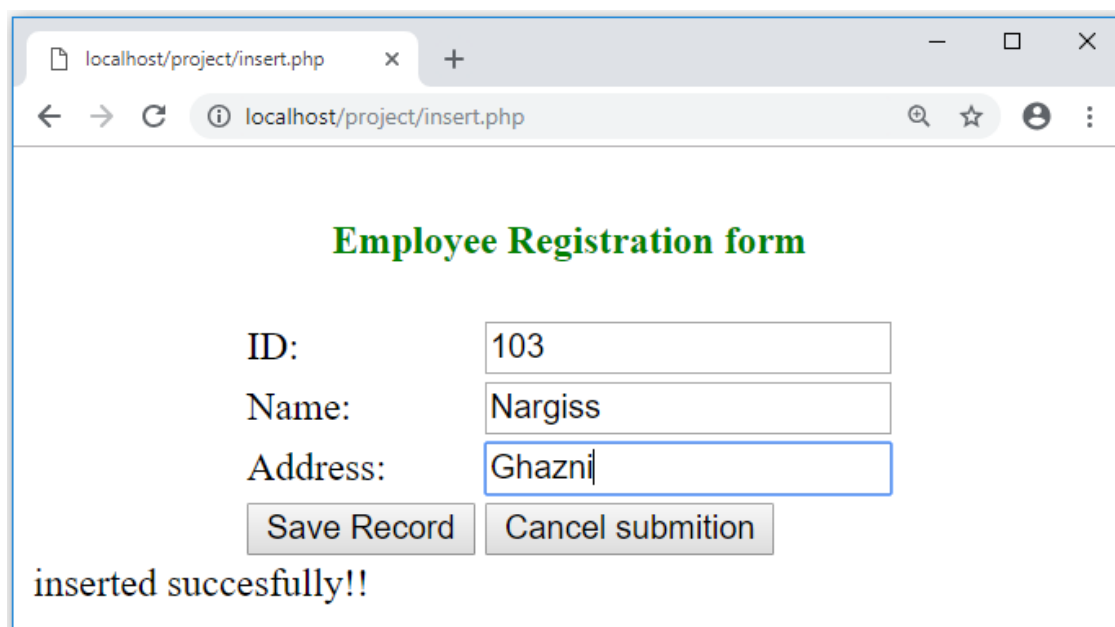
Reg_date:

شکل (۶-۱۱)

ذخیره کردن Insert

برای ذخیره کردن دیتا از محیط انترفیس به دیتابیس ضرورت است تا از قابلیت‌های PHP نیز استفاده کنیم.

پس صفحه «راجستریشن» فورم را به کمک PHP به دیتابیس MySQL وصل می‌کنیم.



Employee Registration form

ID:

Name:

Address:

inserted successfully!!

شکل (۶-۱۲)

بعد از فشردن دکمه save record اطلاعات داخل فورم، به کمک کدهای PHP در دیتابیس داخل جدول employee ذخیره می‌شود.

کدهای HTML که برای ایجاد فرم استفاده شده است:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
</head>
<body>
<table align="center">
<caption><h4 style="color:green;">Employee Registration form </h4></caption>
<form method="post" action="insert.php">
<tr><td>
ID:</td><td><input type="text" name="id">
</td></tr>
<tr><td>
Name:</td><td><input type="text" name="name">
</td></tr>
<tr><td>
Address:</td><td><input type="text" name="address">
</td></tr>
<tr><td>
<input type="submit" value="Save Record" > </td>
<td><input type="reset" value="Cancel submission">
</td> </tr>
</table>
</body>
</html>
```

کدهای زبان PHP برای ذخیره کردن اطلاعات در داخل دیتابیس استفاده می‌شود.

نوت: این هر دو کد در یک فایل به امتداد همدیگر نوشته می‌شوند. در این جا داخل فایل به نام insert.php نوشته شده و در داخل فولدر htdocs ذخیره شده است.

```
<?php
$conn=mysql_connect("localhost","root","");
mysql_selectdb("mydb");
//-----
$id=$_POST['id'];
$name=$_POST['name'];
$address=$_POST['address'];

$insert=mysql_query("insert into employee values('$id','$name','$address')");

if($insert)
{
echo"inserted succesfully!!" ;
}
else
{
echo "Not inserted";
}
?>
```

حالا جدول employee را که اطلاعات داخل آن ذخیره شده است، نشان می‌دهیم:

id	name	address
100	Mateen	Kabul
101	Najeeb	Herat
102	Omid	Balkh
103	Nargiss	Ghazni

انتخاب کردن ریکارد Select

بعد از ذخیره کردن ریکاردها نیاز است تا آن‌ها را دوباره در داخل جداول دیتابیس مشاهده کنیم. برای اجرای این عملیه، از select استفاده می‌کنیم.

کیوری select یکی از کاربردی‌ترین کیوری‌هایی است که در زبان SQL استفاده می‌شود. در این جا صرف از قابلیت انتخاب کردن ریکاردها توسط select استفاده می‌کنیم:

Syntax

```
select * from tablename;
```

مثال:

توسط اسکریپت‌های PHP ریکاردها را از داخل جدول‌های دیتابیس در مرورگر وب مشاهده کنید:

فورم را توسط تگ‌های HTML ایجاد می‌کنیم:

```
<html>
<head>
</head>
<body bgcolor="lightgreen">
<table border="2" align="center" cellpadding="10" cellspacing="10">
<caption>Employee Report</caption>
<tr>
<td>ID</td>
<td>Name</td>
<td>Address</td>
</tr>
```

بعد از ادامه این تگ‌ها، کدهای PHP را می‌نویسیم:

```

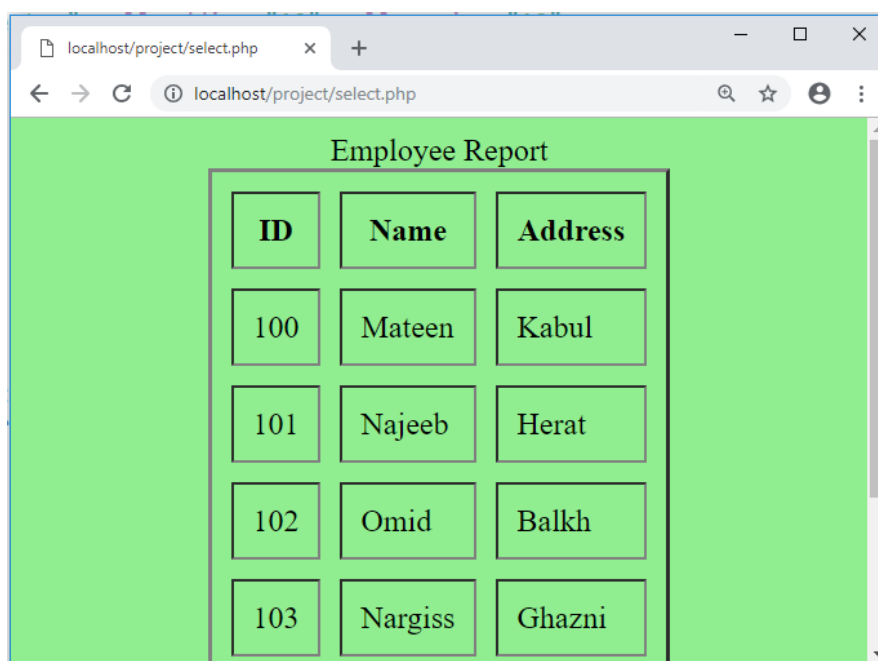
<?php
mysql_connect("127.0.0.1","root","")or die("Not connected");
mysql_selectdb("mydb");
$select=mysql_query("select * from employee");
while($rows=mysql_fetch_array($select)){
echo "<tr>";
echo "<td>".$rows['id']. "</td>" ;
echo "<td>".$rows['name']. "</td>" ;
echo "<td>".$rows['address']. "</td>" ;
}
?>
</table>

```

تابع `mysql_fetch_array()`:

این تابع برای آوردن یک سطر جدول به شکل Associative Array و یا Numeric Array استفاده می‌شود.

نتیجه را در مرورگر وب مشاهده کنید:



ID	Name	Address
100	Mateen	Kabul
101	Najeeb	Herat
102	Omid	Balkh
103	Nargiss	Ghazni

شکل (۶-۱۳)

حذف کردن ریکارد Delete

ریکاردهایی که استفاده نمی‌شوند، اگر بخواهیم آن‌ها را از دیتابیس حذف کنیم، توسط بیانیه Delete این کار را اجرا می‌کنیم.

اگر بخواهیم، تمام ریکاردهای یک جدول حذف شود، از این ساختار استفاده می‌شود:

```
delete from tablename;
```

اگر بخواهیم یک ریکارد مشخص حذف شود، از این ساختار استفاده می‌شود؛ البته در این ساختار از clause به نام where استفاده می‌شود که این clause ریکاردها را sort کرده، ریکارد مورد نظر را حذف می‌کند.

```
delete from tablename where id= idnumber;
```

برای حذف یک ریکارد از داخل جدول دیتابیس، مراحل ذیل را در نظر بگیرید:

مرحله اول

یک فایل را به نام Delete.php ایجاد کرده، سپس توسط کدهای PHP این فایل را به دیتابیس وصل می‌کنیم. بعد از اتصال جدولی را که بخواهیم ریکارد از آن حذف شود، انتخاب می‌کنیم.

```
<?php
//variable declaration
$servername="localhost";
$username="root";
$password="";
$dbname="mydb";
$tblname="employee";
//connect to server
mysql_connect("$servername","$username","$password")or die(mysql_errno());
mysql_selectdb("$dbname");
$selecttb=mysql_query("select * from $tblname");
?>
```

مرحله دوم

توسط تگ‌های HTML یک جدول ایجاد می‌کنیم تا ریکاردهایی، از داخل جدول دیتابیس به این جدول نشان داده شود. توجه داشته باشید که تعداد fieldهایی که در فورم ایجاد می‌شود، با تعداد فیلدهایی که در جدول دیتابیس است، یکسان باشد.

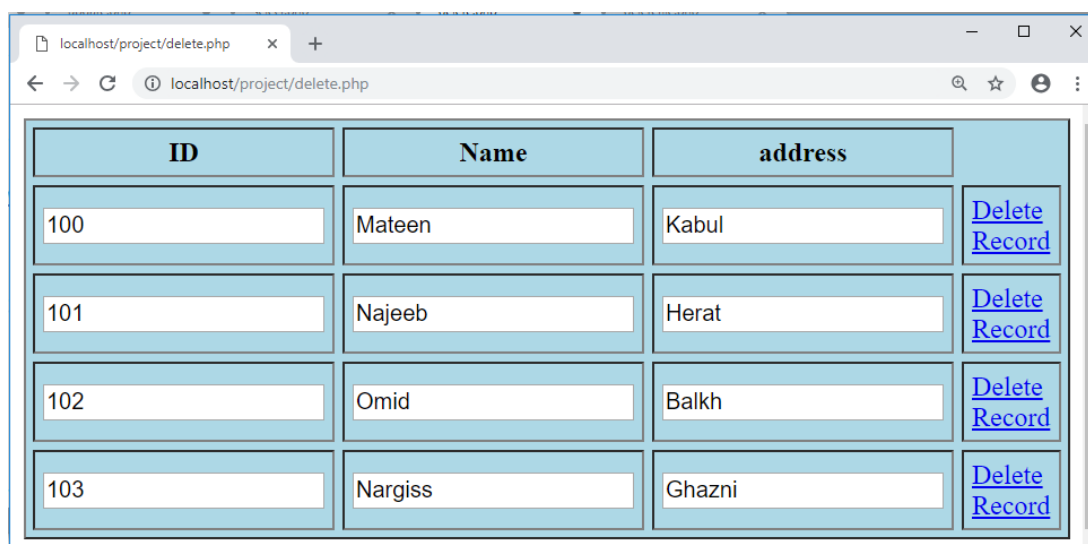
```
<table border="1" bgcolor="lightblue" align="center"
cellpadding="5" cellspacing="5">
<tr>
<th>ID</th><th>Name</th><th>address</th>
</tr>
```

مرحله سوم

اسکرپت‌های PHP را جهت انتخاب کردن ریکاردها از داخل جدول دیتابیس، می‌نویسیم:

```
<?php
mysql_connect("127.0.0.1","root","")or die("Not connected");
mysql_selectdb("mydb");
$select=mysql_query("select * from employee");
while($rows=mysql_fetch_array($select)){
extract($rows);
?>
<tr>
<td><input value="<?php echo$id?>"></td>
<td><input value="<?php echo$name?>"></td>
<td><input value="<?php echo$address?>"></td>
<td><a href=deletefile.php?id=<?php echo$id?>>Delete Record</a></td></tr>
<?php
}
?>
</table>
```

نتیجه را در مرورگر وب مشاهده کنید:



ID	Name	address	
<input type="text" value="100"/>	<input type="text" value="Mateen"/>	<input type="text" value="Kabul"/>	Delete Record
<input type="text" value="101"/>	<input type="text" value="Najeeb"/>	<input type="text" value="Herat"/>	Delete Record
<input type="text" value="102"/>	<input type="text" value="Omid"/>	<input type="text" value="Balkh"/>	Delete Record
<input type="text" value="103"/>	<input type="text" value="Nargiss"/>	<input type="text" value="Ghazni"/>	Delete Record

شکل (۶-۱۴)

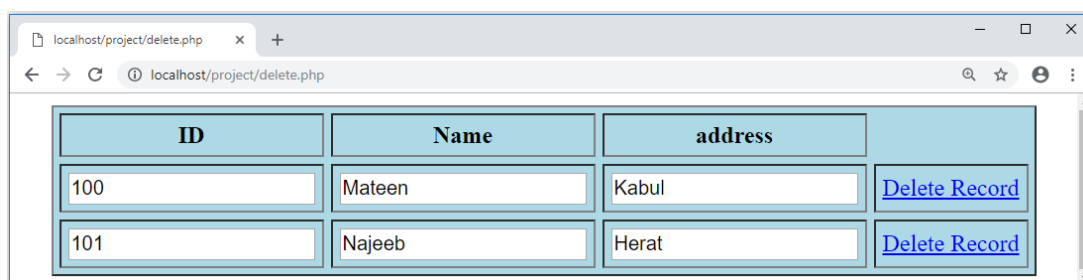
مرحله چهارم

کدهای PHP را در فایل دیگری به نام Deletefile.php نوشته، ذخیره کنید:

```
<?php
mysql_connect("127.0.0.1","root","")or die("Not connected");
mysql_selectdb("mydb");
$id=$_GET['id'];
$delete=mysql_query("delete from employee where id='$id'");
header("location:delete.php");
?>
```

بعد از کلیک کردن بالای کلید Delete Record، ریکارد مورد نظر از دیتابیس حذف می‌شود.

در اینجا می‌بینیم که ریکاردهای 102 و 103 از جدول employee حذف می‌شود:



ID	Name	address	
100	Mateen	Kabul	Delete Record
101	Najeeb	Herat	Delete Record

شکل (۶-۱۵)

وقتی که جدول employee را در دیتابیس مشاهده کنیم، می‌بینیم که دو ریکارد حذف شده است:

```
+-----+-----+-----+
| id   | name  | address |
+-----+-----+-----+
| 100  | Mateen | Kabul   |
| 101  | Najeeb | Herat   |
+-----+-----+-----+
```

ویرایش کردن ریکارد (Update)

عبارت از کیوری است که برای ویرایش یا تغییرآوردن در ریکاردها از آن استفاده می‌شود. برای اجرای این کیوری به مراحل ذیل توجه کنید:

مرحله اول

یک فایل به نام updatefile.php ایجاد کرده، سپس کدهای PHP را بنویسید تا فایل به دیتابیس MySQL وصل شود:

```
<?php
mysql_connect("127.0.0.1","root","")or die("Not connected");
mysql_selectdb("mydb");
$select=mysql_query("select * from employee");
while($rows=mysql_fetch_array($select)){
extract($rows);
?>
```

مرحله دوم

توسط تگ‌های THML جدولی را ایجاد کنید تا ریکاردها از داخل جدول دیتابیس در آن نشان داده شود:

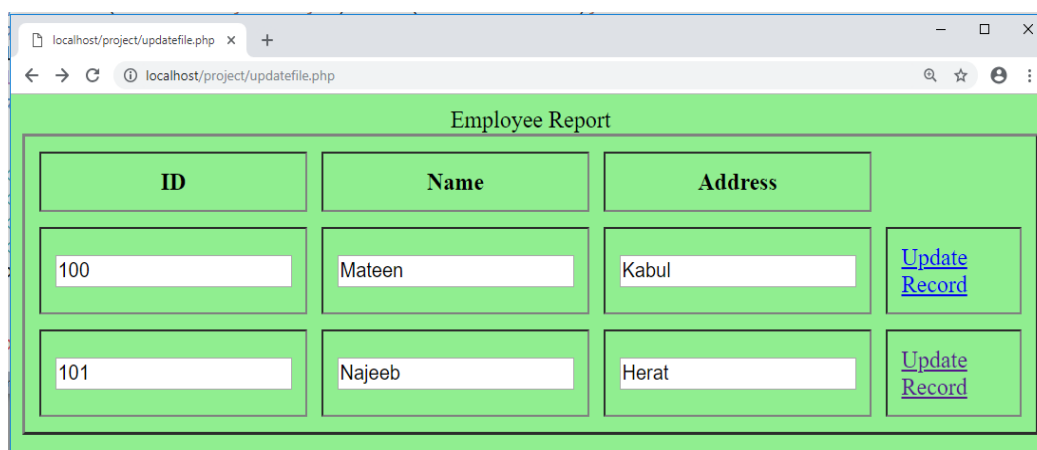
```
<html>
<head>
</head>
<body bgcolor="lightgreen">
<table border="2" align="center" cellpadding="10" cellspacing="10">
<caption>Employee Report</caption>
<tr>
<th>ID</th>
<th>Name</th>
<th>Address</th>
</tr>
```

مرحله سوم

برای نشان دادن ریکاردها از جدول دیتابیس، اسکریپت‌های PHP را می‌نویسیم:

```
<?php
mysql_connect("127.0.0.1","root","")or die("Not connected");
mysql_selectdb("mydb");
$select=mysql_query("select * from employee");
while($rows=mysql_fetch_array($select)){
extract($rows);
?>
<tr>
<td><input value="<?php echo$id?>"></td>
<td><input value="<?php echo$name?>"></td>
<td><input value="<?php echo$address?>"></td>
<td><a href=update.php?id=<?php echo$id?>>Update Record</a></td></tr>
<?php
}
?>
</table>
```

نتیجه را در مرورگر وب مشاهده کنید:



شکل (۶-۱۶)

مرحله چهارم

یک فایل به نام update.php ایجاد کرده، سپس کدهای PHP را بنویسید که فایل را به دیتابیس وصل کند، بعد کیوری Update را بنویسید تا ریکاردها را آپدیت کند.

نخست فایل را به دیتابیس وصل کنید:

```

<?php
$id=$_GET['id'];

$cn=mysql_connect("localhost","root","");
$selectdb=mysql_select_db("mydb");

$selectttb=mysql_query("select * from employee where id='$id' ");

while($rows=mysql_fetch_array($selectttb)) {
    extract($rows);
}
?>

```

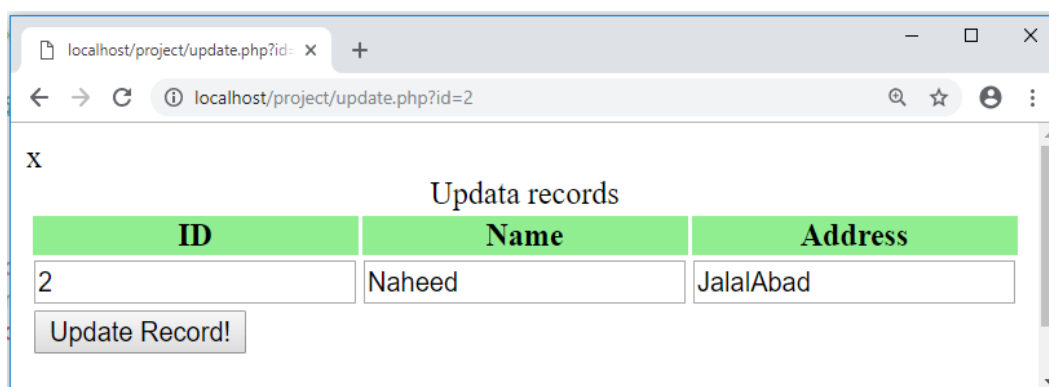
توسط تگ‌های HTML، جدولی را ایجاد کنید تا ریکاردهایی را که بخواهیم آپدیت شود، در آن نشان داده شود:

```

<form action="update.php" method="post" id="edit">
<table id="viewdat1" align="center">
<caption>Updata records</caption>
<tr>
<th width="20" bgcolor="lightgreen"> ID</th>
<th width="20" bgcolor="lightgreen">Name</th>
<th width="20" bgcolor="lightgreen">Address</th>
</tr>
<tr>
<td><input type="text" name="id" value=<?php echo $id ?></td>
<td><input type="text" name="name" value=<?php echo $name ?></td>
<td><input type="text" name="address" value=<?php echo $address ?></td>
</tr>
<tr>
<td><input type="submit" value="Update Record!"></td>
</tr>
</table>
</form>

```

بعد از نوشتن تگ‌های HTML نتیجه را در مرورگر وب مشاهده کنید:



شکل (۶-۱۷)

حالا اسکریپت PHP را بنویسید که توسط آن ریکارد قبلی Update شود:

```
<?php
$id1=$_POST['id'];
$name=$_POST['name'];
$lastname=$_POST['lastname'];
$address=$_POST['address'];

$update=mysql_query("update employee set id='$id1',
name='$name',address='$address' where id='$id'");
?>
<p><a href="updatefile.php">Back to main page</a></p>
```

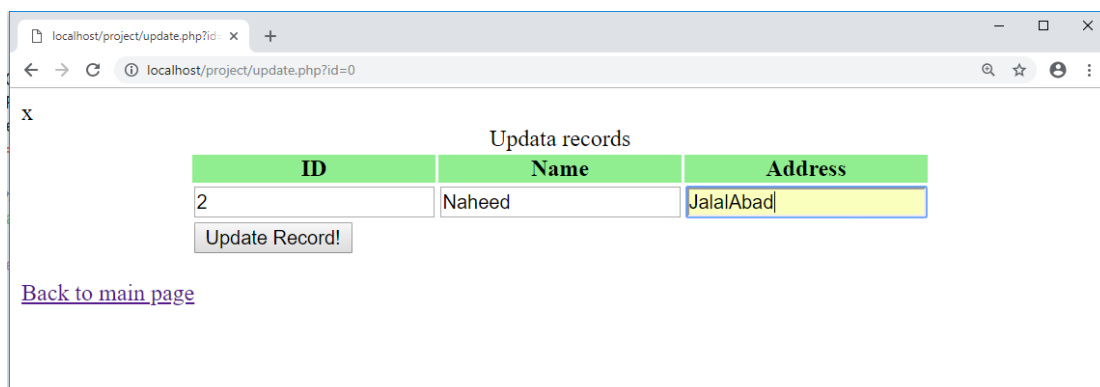
حالا نتیجه را در مرورگر وب مشاهده کنید:

ID	Name	Address	
<input type="text" value="1"/>	<input type="text" value="Assadullah"/>	<input type="text" value="Kabul"/>	Update Record
<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>	Update Record

شکل (۶-۱۸)

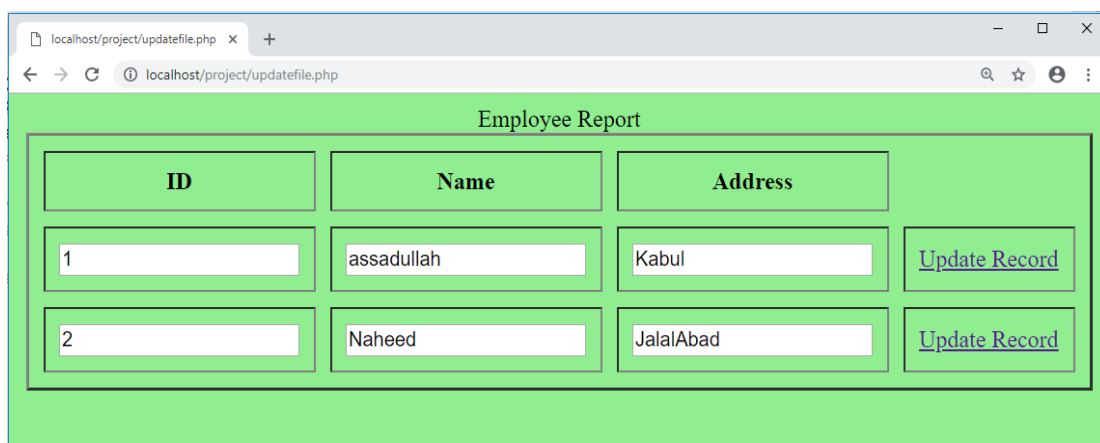
وقتی که بالای لینک Update Record کلیک کنیم، صفحه جدیدی باز می شود که در آن می توانیم ریکارد را آپدیت کنیم.

می خواهیم در ریکاردی که ID آن 0 است، یک ریکارد را ثبت کنیم:



شکل (۱۹-۶)

بعد از فشردن دکمه Update Record در صفحه اصلی رفته، می بینیم که ریکارد 0 به شماره 2 تبدیل شده و فیلدهای خالی، پر شده است:



شکل (۲۰-۶)

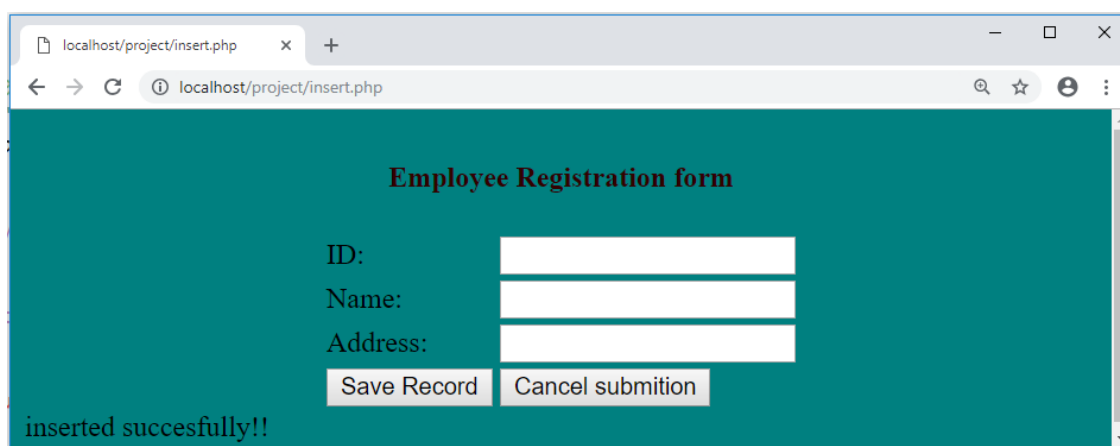
حالا یک پنل ایجاد می کنیم که به تمام گزینه های insert, select, delete and update دسترسی پیدا کنیم.



شکل (۶-۲۱)

در قسمت بالا می‌بینیم، برای هر کدام Insert, Delete, Update, select یک مینو ایجاد شده است، وقتی که بالای هر کدام از آن‌ها کلیک کنیم، صفحاتی که به آن‌ها لینک بوده‌اند، باز می‌شود.

وقتی که بالای مینوی insert کلیک کنیم، صفحه به این شکل باز می‌شود:



شکل (۶-۲۲)

وقتی که بالای مینوی delete کلیک کنیم، صفحه به این شکل باز می‌شود:

ID	Name	address	
1	assadullah	Kabul	Delete Record
2	Naheed	JalalAbad	Delete Record
0			Delete Record

شکل (۶-۲۳)

وقتی که بالای مینوی Update کلیک کنیم، دریچه به این شکل باز می شود:

ID	Name	Address	
1	assadullah	Kabul	Update Record
2	Naheed	JalalAbad	Update Record
0			Update Record

شکل (۶-۲۴)

وقتی که بالای مینوی selection کلیک کنیم، صفحه به این شکل باز می شود:

ID	Name	Address
1	assadullah	Kabul
2	Naheed	JalalAbad

شکل (۶-۲۵)



خلاصه‌ی فصل ششم

مفاهیمی که در این فصل از آن‌ها بحث شد، از جمله کاربردی‌ترین و مهم‌ترین مفاهیم زبان برنامه‌نویسی PHP اند، زیرا یکی از قابلیت‌های مهم و کاربردی زبان برنامه‌نویسی PHP توسعه‌دادن وب‌سایت‌های دینامیک و اتصال آن با Database می‌باشد. بعد از اتصال به دیتابیس، ما می‌توانیم محتویات صفحات وب را در داخل دیتابیس ذخیره کنیم. مطالبی که در این فصل بحث شد، اتصال انترفیس به دیتابیس و هم‌چنان اجرای کیوری‌های ذخیره‌سازی اطلاعات، حذف اطلاعات، انتخاب و مشاهده اطلاعات و تغییرآوردن در اطلاعات می‌باشد.

برای مطالعه این فصل، باید بیشتر وقت گذاشته، تمام مثال‌های آن را به شکل عملی کار کنید تا قادر شوید، صفحات استاتیک وب‌سایت را به صفحات دینامیک تبدیل کنید.



سوالات فعالیت های فصل ششم

۱. با کدام فنکشن می توانیم، صفحه انترفیس را به دیتابیس MySQL وصل کنیم؟
۲. برای ثبت ریکاردها، کدام کیوری ضرورت است، توضیح دهید.
۳. برای حذف ریکاردها، کدام کیوری ضرورت است، توضیح دهید.
۴. برای تغییرآوردن در ریکاردها، کدام کیوری ضرورت است، شرح دهید.
۵. هدف از Local host چیست؟
۶. چگونه می توانیم استاتیک وبسایتها را به دینامیک وبسایتها تبدیل کنیم؟

فعالیت انفرادی

- پروگرامی را بنویسید که انترفیس را به دیتابیس وصل سازد.
- پروگرامی را بنویسید که یک دیتابیس به نام Kabul Ci ty ایجاد کند.
- پروگرامی را بنویسید که جدولها را به نامهای empl oye e, st aff ایجاد کند.
- پروگرامی را بنویسید که ریکاردها را در داخل جدول empl oye e ذخیره کند.
- پروگرامی را بنویسید که یک ریکارد را از جدول empl oye e حذف کند.
- پروگرامی را بنویسید که ریکاردها را در جدول st aff تغییر دهد.
- پروگرامی را بنویسید که تمام ریکاردها را از جدول empl oye e نشان دهد.

منابع و مأخذ (References)

1. Lerdorf, (2009), Programming PHP, united states of Amrica: Addison Westley.
2. Matt Zandstr (2012), PHP in 24 hours, Associates inc.
3. Crstian Darie and Emilian Balanescu(2010),PHP amd MySQL,USA,Bjorklden.
4. www.prdev.com
5. www.majidonline.com
6. www.w3schol.com

Thank you for reading

Find more e-books and articles on Ketabton - your multilingual digital library.

www.ketabton.com

Ketabton - Pashto, Farsi, Arabic & English