

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

بسته‌های نرم افزارى (۳)

(جلد دوم)

رشته کامپیوتر

گروه تحصیلى کامپیوتر

زمینه خدمات

شاخه آموزش فنى و حرفه‌ای

سرشناسه: سلیمی زاده، زهرا، ۱۳۵۹-
عنوان و نام پدیدآور: بسته‌های نرم افزارى (۳) رشته کامپیوتر زمینه خدمات شاخه فنى و حرفه‌ای/مؤلف: زهرا سلیمی زاده؛ برنامه‌ریزی محتوا و نظارت بر تألیف دفتر تألیف کتاب‌های درسى فنى و حرفه‌ای و کار دانش. مشخصات نشر: تهران: شرکت چاپ و نشر کتاب‌های درسى ایران، ۱۳۹۵.
مشخصات ظاهرى: ج ۲: ۲۹×۲۲ س. م.
شابک: ۹۷۸-۹۶۴-۰۵-۲۲۴۷-۹
وضعیت فهرست‌نویسى: فیا
یادداشت: ج ۲، (چاپ اول: ۱۳۹۰) (فیا).
مندرجات: ج ۲، رشته کامپیوتر گروه تحصیلى کامپیوتر...،
موضوع: وب-سایت‌ها-طراحی.
شناسه افزوده: سازمان پژوهش و برنامه‌ریزی آموزشى. دفتر تألیف کتاب‌های درسى فنى و حرفه‌ای و کار دانش.
رده‌بندى کنگره: ۱۳۹۰ س ۸ب ۵۱۰۵/۸۸۸ TK
رده‌بندى دیوبى: ۰۰۴/۶۷۸
شماره کتاب شناسى ملی: ۲۳۶۱۳۲۰

همکاران محترم و دانش آموزان عزیز :

پیشنهادات و نظرات خود را درباره محتوای این کتاب به نشانی
تهران - صندوق پستی شماره ۴۸۷۴/۱۵ دفتر تألیف کتاب های درسی
فنی و حرفه ای و کاردانش، ارسال فرمایند.

info@tvoccd.sch.ir

پیام نگار (ایمیل)

www.tvoccd.sch.ir

وبگاه (وب سایت)

محتوای این کتاب در کمیسیون تخصصی رشته کامپیوتر دفتر تألیف کتاب های درسی
فنی و حرفه ای و کاردانش با عضویت: بتول عطاران، محمد رضا شکرریز، شهناز علیزاده، نیلوفر
بزرگ نیا، زهره سهرابی، مریم حسکوئیان، شرمین الوندی و زهرا محرابی تأیید شده است.

وزارت آموزش و پرورش

سازمان پژوهش و برنامه ریزی آموزشی

برنامه ریزی محتوا و نظارت بر تألیف: دفتر تألیف کتاب های درسی فنی و حرفه ای و کاردانش

نام کتاب: بسته های نرم افزاری (۳) (جلد دوم) - ۴۵۲/۲

مؤلف: زهرا سلیمی زاده

آماده سازی و نظارت بر چاپ و توزیع: اداره کل نظارت بر نشر و توزیع مواد آموزشی

تهران: خیابان ایرانشهر شمالی - ساختمان شماره ۴ آموزش و پرورش (شهید موسوی)

تلفن: ۹-۸۸۸۳۱۱۶۱، دورنگار: ۹۲۶۶-۸۸۳، کدپستی: ۱۵۸۴۷۴۷۳۵۹

وبسایت: www.chap.sch.ir

مدیر امور فنی و چاپ: لیدا تیکروش

طراح جلد: طاهره حسن زاده

صفحه آرا: سمیه قنبری

حروفچین: زهرا ایمانی نصر

مصحح: علی نجمی، رضا جعفری

امور آماده سازی خیر: فریبا سیر

امور فنی رایانه ای: ناهید خیام بانسی، سیده شیوا شیخ الاسلامی

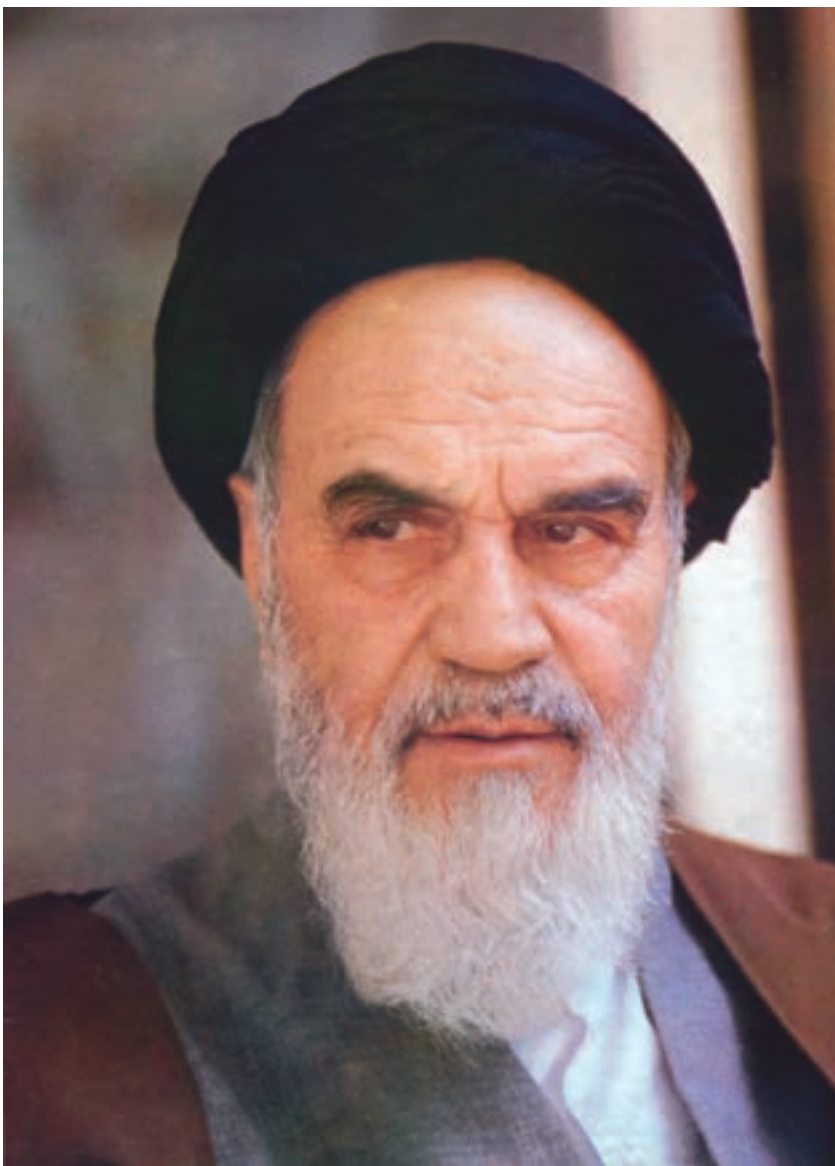
ناشر: شرکت چاپ و نشر کتاب های درسی ایران: تهران - کیلومتر ۱۷ جاده مخصوص کرج - خیابان ۶۱ (داروبخش)

تلفن: ۵-۴۴۹۸۵۱۶۱، دورنگار: ۴۴۹۸۵۱۶۰، صندوق پستی: ۱۳۹-۳۷۵۱۵

چاپخانه: شرکت چاپ و نشر کتاب های درسی ایران «سهامی خاص»

سال انتشار و نوبت چاپ: چاپ چهارم ۱۳۹۵

حق چاپ محفوظ است.



شما عزیزان کوشش کنید که از این وابستگی بیرون آید و احتیاجات کشور خودتان را برآورده سازید، از نیروی انسانی ایمانی خودتان غافل نباشید و از اتکای به اجانب بپرهیزید.

امام خمینی «قدس سرّه الشریف»

یکی از مهم‌ترین ویژگی‌های عصر حاضر پیشرفت چشم‌گیر علم و فناوری است در این راستا ایجاد ارتباط شبکه‌هایی مانند اینترنت یک رکن اساسی به شمار می‌رود و برای هنرجویان رشته کامپیوتر، آموزش طراحی صفحات وب به منزله فراهم کننده بسته اصلی این ارتباط یک ضرورت اجتناب ناپذیر است در کتابی که پیش رو دارید سعی شده ضمن معرفی مفاهیم معمول پایه، روند طراحی وب سایت‌های ایستا و پویا به طور مجزا آموزش داده شود.

لذا به هنرجویان عزیز توصیه می‌شود برای یادگیری مؤثر هر فصل تمرین‌ها و مثال‌های بیان شده را به طور کامل اجرا نمایند.

در خاتمه از تمامی کارشناسان، معلمان و هنرجویان عزیز خواهشمندم نظرات ارزشمند خود را به منظور رفع نارسایی احتمالی این کتاب از طریق پست الکترونیکی tvoccd@roshd.ir به دفتر تألیف کتاب‌های درسی فنی و حرفه‌ای و کاردانش ارسال فرمایند.

مؤلف

۳۸	۳-۱-۲ دستور elseif	مقدمه
۴۰	۳-۱-۳ عملگر شرطی؟	
۴۰	۳-۲ دستور switch	
۴۳	۳-۳ حلقه های تکرار	
۴۳	۳-۳-۱ حلقه for	
۴۷	۳-۳-۲ حلقه foreach	
۴۸	۳-۳-۳ حلقه while	
۴۹	۳-۳-۴ حلقه do...while	
۵۱	چکیده فصل	
۵۲	خودآزمایی	
فصل چهارم : انواع		
۵۴	۴-۱-۱ تابع رشته ای	
۵۴	۴-۱-۱-۱ تابع strlen	
۵۵	۴-۱-۲ تابع () strtoupper	
۵۵	۴-۱-۳ تابع () strtolower	
۵۵	۴-۲-۱ تابع تاریخ و زمان	
۵۵	۴-۲-۱-۱ تابع () date	
۵۷	۴-۲-۲ تابع () time	
۵۷	۴-۳-۱ تابع ریاضی	
۵۷	۴-۳-۱-۱ تابع () abs	
۵۸	۴-۳-۲ تابع () floor	
۵۸	۴-۳-۳ تابع () fmod	
۵۹	۴-۳-۴ تابع () max	
۵۹	۴-۳-۵ تابع () min	
۶۰	۴-۳-۶ تابع () rand	
۶۱	۴-۳-۷ تابع () round	
۶۱	۴-۴-۱ تعریف توابع جدید	
۶۶	چکیده فصل	
۶۷	خودآزمایی	
فصل پنجم : طراحی برنامه های تحت وب		
۶۸	کار با فرم ها	
۶۹	۵-۱-۱ \$_post	
۷۰	۵-۲-۱ \$_Get	
۷۱	۵-۳-۱ \$_REQUEST	
فصل اول : معرفی PHP و نصب و راه اندازی آن		
۲	۱-۱ زبان های برنامه نویسی وب	
۲	۱-۲ انواع سرویس دهنده های وب برای تولید صفحات وب پویا	
۴	۱-۳ پارامترهای مهم در انتخاب سرویس دهنده زبان برنامه نویسی وب	
۵	۱-۴ ویژگی های زبان برنامه نویسی PHP	
۷	۱-۵ نصب و راه اندازی برنامه های سرویس دهنده	
۷	۱-۶ نصب برنامه wamp server	
۸	چکیده فصل	
۱۲	خودآزمایی	
۱۳		
فصل دوم : ساختار PHP		
۱۴	۲-۱ شکل نوشتن دستورات PHP	
۱۵	۲-۲ نوشتن اولین برنامه به زبان PHP	
۱۶	۲-۲-۱ چاپ یک عبارت در خروجی	
۱۷	۲-۳ درج توضیحات در کدهای PHP	
۱۸	۲-۴ متغیرها در PHP	
۱۹	۲-۴-۱ قواعد نامگذاری متغیرها در PHP	
۱۹	۲-۵ انواع داده ها	
۲۰	۲-۵-۱ string	
۲۲	۲-۵-۲ array	
۲۴	۲-۶ دو تابع کاربردی در ارتباط با متغیرها	
۲۶	۲-۷ ثابت ها	
۲۷	۲-۸ انواع عملگرها	
۲۸	۲-۸-۱ عملگرهای ریاضی (محاسباتی)	
۲۸	۲-۸-۲ عملگرهای مقایسه ای	
۳۰	۲-۸-۳ عملگرهای منطقی	
۳۱	چکیده فصل	
۳۴	خودآزمایی	
۳۵		
فصل سوم : کنترل روند برنامه		
۳۶	۳-۱ دستور if	
۳۷	۳-۱-۱ دستور else	

فصل هشتم : ارتباط با پایگاه داده در

۱۲۰

Dreamweaver

- ۱۲۱ ۸-۱ تنظیمات اولیه برای ارتباط با پایگاه داده
- ۱۲۵ ۸-۱-۱ ایجاد Recordset
- ۱۲۶ ۸-۲ مشاهده اطلاعات پایگاه داده
- ۱۳۰ ۸-۳ درج اطلاعات در پایگاه داده
- ۱-۳-۸ درج اطلاعات در پایگاه داده با استفاده از
- ۱۳۱ Wizard
- ۱۳۲ ۸-۴ حذف رکورد از پایگاه داده
- ۱۳۶ ۸-۵ ویرایش اطلاعات پایگاه داده
- ۱۳۸ ۸-۶ فرم‌های Master و Detail
- ۱۴۰ چکیده فصل
- ۱۴۱ خودآزمایی

۱۴۲

فصل نهم : پروژه

- ۱۴۳ ۹-۱ نیازسنجی
- ۱۴۳ ۹-۲ معماری اطلاعات
- ۱۴۴ ۹-۳ طراحی
- ۱۴۵ ۹-۴ ثبت Domain و خرید فضا (Host)
- ۱۴۵ ۹-۵ طراحی گرافیکی و قالب وب سایت
- ۱-۵-۹ ایجاد صفحه وب توسط نرم افزار
- ۱۴۶ Photoshop
- ۱۴۹ ۹-۶ مدیریت کاربران
- ۱۴۹ ۹-۷ بانک اطلاعاتی
- ۱۵۲ ۹-۸ طراحی بخش مدیریت سیستم
- ۱۵۳ ۹-۸-۱ طراحی صفحه ورود مدیر فروشگاه
- ۱۵۵ ۹-۸-۲ طراحی صفحه اصلی مدیر فروشگاه
- ۱۵۶ ۹-۸-۳ طراحی صفحه درج کالا جدید
- ۱۵۷ ۹-۸-۴ طراحی صفحه ویرایش مشخصات کالا
- ۱۵۸ ۹-۸-۵ طراحی صفحه حذف کالا
- ۱۵۸ ۹-۸-۶ طراحی صفحه گزارشات فروشگاه
- ۱۵۹ ۹-۹ طراحی بخش مشتریان
- ۱۵۹ ۹-۹-۱ ورود کاربر
- ۱۶۱ ۹-۹-۲ عضویت کاربر
- ۱۶۱ ۹-۹-۳ مشاهده لیست کالاها
- ۱۶۱ ۹-۹-۴ سفارش خرید
- ۱۶۲ منابع

- ۷۱ ۵-۴ ایجاد فرم بارگذاری فایل
- ۷۵ ۵-۵ ذخیره فایل بارگذاری شده
- ۸۲ چکیده فصل
- ۸۳ خودآزمایی

فصل ششم : به کارگیری کوکی‌ها (Cookies)

۸۴

و جلسه‌ها (sessions)

- ۸۵ ۶-۱ Cookie
- ۸۶ ۶-۱-۱ ایجاد Cookie
- ۸۸ ۶-۱-۲ دسترسی به اطلاعات Cookie
- ۸۹ ۶-۱-۳ حذف یک Cookie
- ۸۹ ۶-۲ جلسه session
- ۹۰ ۶-۲-۱ ایجاد جلسه
- ۹۱ ۶-۲-۲ متغیرهای جلسه
- ۹۴ ۶-۲-۳ حذف جلسه و متغیرهای آن
- ۹۴ ۶-۲-۴ نگهداری تنظیمات شخصی کاربر
- ۹۶ چکیده فصل
- ۹۷ خودآزمایی

۹۸

فصل هفتم : MySQL

- ۱۰۰ ۷-۱ ایجاد پایگاه داده
- ۱۰۲ ۷-۲ ساخت جداول پایگاه داده
- ۱۰۳ ۷-۳ ایجاد ارتباط با سیستم مدیریت پایگاه داده MySQL
- ۱۰۵ ۷-۴ ایجاد یک پایگاه داده و اجزای آن
- ۱۰۵ ۷-۴-۱ ایجاد یک پایگاه داده
- ۱۰۶ ۷-۴-۲ ایجاد یک جدول در پایگاه داده
- ۱۰۸ ۷-۴-۳ فیلد کلیدی و خاصیت افزایش خودکار
- ۱۰۹ ۷-۴-۴ درج اطلاعات در جدول
- ۱۱۰ ۷-۴-۵ درج اطلاعات در جدول از طریق فرم
- ۱۱۲ ۷-۵ انتخاب و نمایش اطلاعات جدول
- ۱۱۳ ۷-۵-۱ نمایش اطلاعات پایگاه داده در جدول
- ۱۱۵ ۷-۶ به روز رسانی اطلاعات جدول
- ۱۱۶ ۷-۷ حذف رکورد
- ۱۱۸ چکیده فصل
- ۱۱۹ خودآزمایی

فصل ۱

معرفی PHP و نصب و راه‌اندازی آن

هدف‌های رفتاری:

- ۱- پس از آموزش این فصل هنرجو می‌تواند: مفهوم زبان‌های client side و server side را توضیح دهد.
- ۲- PHP و ویژگی‌های آن را بشناسد.
- ۳- برنامه WAMP را به منظور استفاده از امکانات زبان PHP، نصب و راه‌اندازی کند.
- ۴- برنامه PHP را تست و بررسی نماید.
- ۵- تفاوت صفحات وب ایستا و پویا را بشناسد.

طراح وب می‌تواند انواع مختلفی از صفحات وب را با زبان‌های متفاوتی طراحی کند. صفحات وبی که با طراحی آنها در جلد اول این کتاب آشنا شدید، صفحاتی بودند که محتوای آنها در هر بار مشاهده توسط کاربر یا کاربران مختلف، تغییر نمی‌کرد. به این نوع صفحات، صفحات وب ایستا^۱ گفته می‌شود. در این فصل با صفحات وبی آشنا می‌شویم که ضمن آن که درخواست کاربر (سرویس گیرنده) را دریافت می‌کنند، اطلاعات مورد درخواست او را در صفحه وب به نمایش می‌گذارند. به این ترتیب محتوای این صفحات در هر بار مشاهده توسط کاربر یا کاربران مختلف، با توجه به درخواست آنها، متغیر خواهد بود. به عبارت دیگر به چنین صفحاتی که با کاربر (سرویس گیرنده) تعامل دارند، صفحات وب پویا^۲ گفته می‌شود.

به طور مثال فرض کنید کارنامه هفتگی دانش‌آموزان در وب سایت یک مدرسه نشان داده شود. از آنجا که نمرات آنها به طور هفتگی اضافه می‌شود، باید دانش‌آموزان و اولیای آنها بتوانند انتخاب کنند که نمرات کدام هفته را می‌خواهند مشاهده کنند. برای ساخت چنین صفحات پویایی نیاز به نرم‌افزارهای بیشتری نسبت به آنچه در جلد اول خوانده‌ایم، احساس می‌شود. در این فصل با این نرم‌افزارها، محل نصب و چگونگی نصب آنها آشنا می‌شویم.

۱-۱- زبان‌های برنامه‌نویسی وب

از دیدگاه ماهیت تعامل با کاربر (سرویس گیرنده)، صفحات وب همانگونه که قبلاً اشاره شد به دو دسته زبان‌های ایستا و پویا تقسیم می‌شوند. اما از دیدگاهی دیگر می‌دانیم که مدل ارتباطی در اینترنت، مدل client/server (سرویس دهنده / سرویس گیرنده) است یعنی client (سرویس گیرنده) درخواست می‌دهد و سرویس دهنده پاسخ او را آماده کرده برای او می‌فرستد. درخواست سرویس گیرنده غالباً به صورت یک آدرس وب یا URL است و پاسخی که دریافت می‌کند به صورت محتویات درخواستی در آن صفحه است که توسط مرورگر سرویس گیرنده نمایش داده می‌شود. بر اساس این مدل زبان‌های برنامه‌نویسی وب به طور کلی به دو دسته تقسیم می‌شوند:

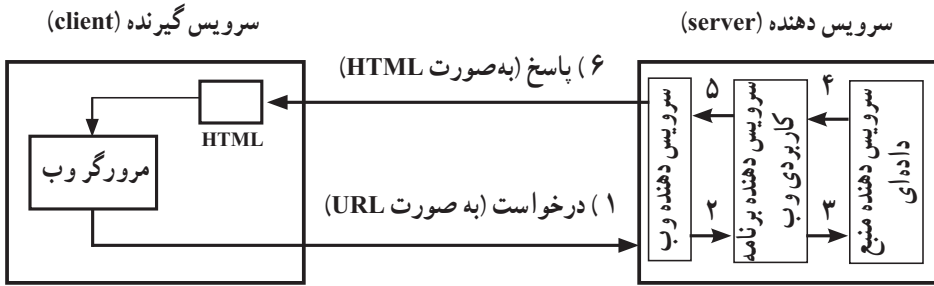
۱- Client side (سمت سرویس گیرنده): به زبان‌های برنامه‌نویسی تحت وب گفته می‌شوند که بر روی دستگاه سرویس گیرنده و به وسیله برنامه مرورگر وب اجرا می‌شوند. HTML و

۱_ Static

۲_ Dynamic

JavaScript جزء این دسته هستند.

۲- Server side (سمت سرویس دهنده): زبان‌های برنامه‌نویسی سمت سرویس دهنده، زبان‌هایی هستند که به‌طور کامل بر روی دستگاه سرویس دهنده اجرا می‌شوند و حاصل اجرای آنها در قالب کدهایی که برای سرویس گیرنده قابل اجرا است، (غالباً HTML) برای او ارسال می‌شوند. از انواع زبان‌های برنامه‌نویسی سمت سرور می‌توان PHP و ASP.Net را نام برد. با این توضیحات متوجه می‌شویم که برای ساخت صفحات وب پویا، نیاز به زبان‌های سمت سرور داریم تا درخواست کاربر را انجام و نتیجه را ارسال نمایند. برای درک بهتر این مفاهیم توجه شما را به شکل زیر جلب می‌کنیم.



شکل ۱-۱

همان‌طور که در شکل ۱-۱ ملاحظه می‌کنید در سمت سرویس گیرنده، تنها کدهای HTML را برای نمایش توسط مرورگر وب داریم. در حالی که در سمت سرویس دهنده به مجموعه‌ای از نرم افزارها برای دریافت درخواست از سرویس گیرنده، پردازش آن و آماده کردن پاسخ برای کاربر نیازمندیم. نرم افزارهای موجود در سمت سرویس دهنده به سه دسته کلی تقسیم می‌شوند:

۱- سرویس دهنده وب (Web Server):

در مرحله اول، این نرم افزار، درخواست کاربر را به‌صورت یک URL دریافت کرده و بررسی می‌کند و صفحه وب درخواست شده در URL را یافته، سپس آن را به سرویس دهنده برنامه کاربردی تحویل می‌دهد. (مرحله ۲ در شکل ۱-۱) بنابراین سرویس دهنده وب به‌صورت مستقیم درخواست کاربران را دریافت می‌کند و به آنان پاسخ می‌دهد.

۲- سرویس دهنده برنامه کاربردی وب (Web Application Programming Server)

یک نرم افزار است که به سرویس دهنده وب، کمک می‌کند که محتویات صفحه را به‌صورت پویا

تولید کند. به این صورت که سرویس دهنده برنامه کاربردی، باید صفحه وب پویای مورد درخواست کاربر (که به صورت کدهای زبان برنامه نویسی وب است) را پردازش کرده و سپس داده‌های مورد نیاز این صفحه را از سرویس دهنده منبع داده‌ای (Data base Server) درخواست نماید. دقت کنید که هنوز کار سرویس دهنده برنامه کاربردی تا اینجا تمام نشده است. (مرحله ۳ شکل ۱-۱)

نکته: به سرویس دهنده برنامه کاربردی وب، «سرویس دهنده زبان برنامه نویسی وب» هم می‌گوییم.

۳- سرویس دهنده منبع داده‌ای (Data Base Server)

این سرویس دهنده باید داده‌های درخواست شده را از درون بانک و جداول داده‌ای خود استخراج کرده و به سرویس دهنده برنامه کاربردی برگرداند. (مرحله ۴ شکل ۱-۱) سپس سرویس دهنده برنامه کاربردی، ترجمه نهایی کدهای زبان برنامه نویسی وب را به همراه اطلاعات استخراج شده از منبع داده‌ای، آماده می‌کند و در اختیار سرویس دهنده وب قرار می‌دهد (مرحله ۵ شکل ۱-۱) تا در قالب یک فایل HTML به‌عنوان پاسخ برای سرویس گیرنده ارسال شود (مرحله ۶ شکل ۱-۱).

نکته: به سرویس دهنده منبع داده‌ای، سرویس دهنده پایگاه داده هم گفته می‌شود.

۱-۲- انواع سرویس دهنده‌های وب برای تولید صفحات وب پویا

با توجه به این که در مراحل ساخت و طراحی صفحات وب، کدهای نوشته شده، باید تست و عیب یابی شوند، لازم است در یک سرویس دهنده بارگذاری شوند تا در هر مرحله بتوان نتیجه اجرا را مشاهده نمود^۱.

با توجه به محیط و زبان برنامه نویسی، سرویس دهنده‌های مختلفی قابل استفاده هستند. مهم‌ترین سرویس دهنده‌های وب، IIS, coldfusion (کلد فیوژن) و Apache (آپاچی) هستند.

این سرویس دهنده‌های وب تفاوت‌هایی از لحاظ ساختاری دارند. برای این که سرویس دهنده وب بتواند به درستی کار کند، باید فایل‌هایی که مربوط به صفحات وب سایت هستند، در یک مسیر مشخص که برای سرویس دهنده قابل شناسایی باشد، کپی شوند. چنانچه از سیستم عامل ویندوز استفاده شود، مسیر پیش فرض برای ذخیره داده‌های سایت در هر کدام از سرویس دهنده‌ها، مانند جدول ۱-۱ است.

۱- در واقع این سرویس دهنده که برای تست استفاده می‌شود، به‌صورت محلی، یعنی روی دستگاه client نصب می‌شود تا

برنامه‌نویس بتواند ارتباط بین سرویس گیرنده و سرویس دهنده را جهت تست به‌صورت محلی بر روی دستگاه انجام دهد.

جدول ۱-۱- مسیر پیش فرض در سرویس دهنده های مختلف

مسیر پیش فرض	سرویس دهنده وب
\Cold Fusion\wwwroot	Cold Fusion
\inetpub\wwwroot	IIS
\apache\htdocs	Apache

نکته: تمام مسیرها به طور پیش فرض در درایوی که ویندوز در آنجا نصب شده، ایجاد می شوند اما شما هنگام نصب می توانید آن را تغییر دهید.

برای تست و بررسی سایت و مشاهده صفحات وب در مرورگر از طریق هر یک از سرویس دهنده های وب، باید مسیرهای مشخصی را در نوار آدرس مرورگر درج کنید. این مسیر برای هر کدام از سرویس دهنده ها در جدول ۱-۲ درج شده است.

جدول ۱-۲- مسیر پیش فرض در سرویس دهنده های مختلف

آدرس مشاهده صفحات	سرویس دهنده وب
http://localhost:8500/	Cold Fusion
http://localhost/	IIS
http://localhost	Apache

این مسیرها برای مشاهده صفحه وبی که آن را به صورت پویا طراحی کرده اید به کار می رود. به این صورت که با توجه به نوع سرویس دهنده، باید مسیر محلی مناسب به همراه نام صفحه وب را در نوار آدرس مرورگر درج کرده، سپس آن را اجرا کنید.

۳-۱- پارامترهای مهم در انتخاب سرویس دهنده زبان برنامه نویسی وب

سرویس دهنده برنامه کاربردی، یک نرم افزار است که به سرویس دهنده وب کمک می کند تا محتویات صفحه را به صورت پویا تولید کند. طراحان وب چند نکته اساسی را برای انتخاب سرویس دهنده زبان برنامه نویسی تحت وب در نظر می گیرند:

۱- هزینه: هر سرویس دهنده زبان برنامه نویسی وب با توجه به نوع کارایی، هزینه های متفاوتی را برای شما ایجاد می کند، لذا باید قبل از هر چیز، میزان مجاز برای صرف هزینه را در نظر بگیرید.

۲- فن آوری سرویس دهنده وب : نرم افزار Dreamweaver امکان استفاده از فناوری های

سرویس دهنده وب مختلف وجود دارد که مهم ترین آنها عبارتند از :

● Cold Fusion

● IIS

● Apache

نوع فناوری سرویس دهنده وب که از آن بهره می گیرید مستقیماً در انتخاب سرویس دهنده زبان برنامه نویسی وب دخالت دارد و هر سرویس دهنده زبان برنامه نویسی وب نیز با زبان برنامه نویسی خاصی کار می کند.

هر سرویس دهنده وبی قادر نیست با تمام انواع سرویس دهنده های زبان برنامه نویسی وب به طور هماهنگ کار کند. لذا لازم است پس از انتخاب نوع سرویس دهنده وب، سرویس دهنده برنامه کاربردی را طوری انتخاب کند که بتواند به طور تعاملی با یکدیگر فعالیت کنند. به عنوان مثال Apache به خوبی می تواند با PHP server کار کند.

۳- نوع سرویس دهنده پایگاه داده (منبع داده ای) : از آنجا که سرویس دهنده زبان

برنامه نویسی وب داده های مورد درخواست کاربر را از آنجا تأمین می کند باید در انتخاب سرویس دهنده منبع داده ای دقت کنیم که با سرویس دهنده زبان برنامه نویسی وب مورد نظر سازگاری داشته باشد. از انواع سرویس دهنده های منبع داده ای می توان SQL Server , My SQL را نام برد.

۴- زبان برنامه نویسی وب : سرویس دهنده یک زبان برنامه نویسی، به طور کلی مترجم یا

مفسر آن زبان است. مثلاً PHP Server سرویس دهنده زبان PHP است.

جدول ۱-۳ نوع فناوری سرویس دهنده وب و سرویس دهنده برنامه کاربردی مورد استفاده برای

هر یک را نشان می دهد. این فناوری ها به عنوان نمونه ذکر شده اند و می توانند شامل طیف گسترده ای از انواع سرویس دهنده ها باشند.

جدول ۱-۳- نوع فناوری سرویس دهنده و سرویس دهنده برنامه کاربردی مورد استفاده آن

فناوری سرویس دهنده منبع داده ای	فناوری سرویس دهنده وب	سرویس دهنده زبان برنامه نویسی تحت وب	زبان برنامه نویسی
Mssql	Cold Fusion	Cold Fusion	CFML (Cold Fusion Markup Language)
MSSQL	IIS	ASP(Active Server Page)	ASP
MySQL	Apache	PHP server	PHP

۴-۱- ویژگی‌های زبان برنامه‌نویسی PHP

هر کدام از زبان‌های Server Side که برای طراحی صفحات وب به کار می‌روند، ویژگی‌هایی دارند که آنها را از سایر زبان‌ها متمایز می‌کند. برخی از این ویژگی‌ها عبارتند از امنیت، سرعت، سادگی اجرا، سادگی دستورات عمل‌های برنامه‌نویسی و قابلیت انعطاف.

زبان PHP یکی از ساده‌ترین و سریع‌ترین زبان‌های طراحی صفحات پویا است.

PHP مخفف عبارت Perl Hypertext Preprocessor است و برای نوشته شدن کدهای برنامه، نیاز به محیط خاصی ندارد و کدهای آن را به راحتی می‌توانید در محیط ساده Notepad بنویسید. این زبان یک رقیب قدرتمند برای زبان برنامه‌نویسی ASP از شرکت مایکروسافت به شمار می‌رود.

یکی دیگر از خصوصیات مهم PHP، *متن باز بودن* این زبان برنامه‌نویسی است و به طور کاملاً رایگان از طریق اینترنت در اختیار برنامه‌نویسان قرار دارد. ساختار و امکانات PHP به شکل مستقل از سیستم عامل طراحی شده است و این بدان معناست که روی هر سیستم عاملی اجرا می‌شود. به‌عنوان مثال برنامه‌نویس می‌تواند برنامه خود را در محیط سیستم عامل ویندوز بنویسد و سپس آن را بدون محدودیت به سیستم عامل لینوکس انتقال دهد. به این خاصیت اصطلاحاً Cross-platform گفته می‌شود.

۵-۱- نصب و راه‌اندازی برنامه‌های سرویس‌دهنده

همان‌طور که می‌دانید برای استفاده از امکانات یک زبان برنامه‌نویسی، باید قبل از هر کاری برنامه‌ها و نرم‌افزارهای لازم برای اجرای آن را روی سیستم خود، نصب کنید؛ زبان PHP نیز از این قاعده مستثنی نیست و برای شروع کار با آن و طراحی صفحات وب پویا باید نرم‌افزارها و سرویس‌های MySQL و Apache و PHP را به‌طور مجزا نصب و راه‌اندازی نمایید.

آماده کردن سیستم به منظور برنامه‌نویسی به زبان PHP و اجرای آن باید با دقت و حوصله فراوان انجام شود و حتی اگر یکی از مراحل کار به اشتباه انجام شود، ممکن است مجبور شوید تمام مراحل را از ابتدا انجام دهید.

پس از نصب هر کدام از برنامه‌های فوق، لازم است تنظیمات هر کدام، مطابق دستورات عمل مشخص، انجام شود.

برنامه‌های سرویس‌دهنده‌ای وجود دارند که مجموع امکانات و سرویس‌های لازم برای ساخت بدون دردسر صفحات وب پویا را به‌طور یکجا، در اختیار طراحان وب قرار می‌دهند.

در سیستم عامل ویندوز برنامه‌های متعددی وجود دارند که PHP را به همراه سایر نرم افزارهای مورد نیاز برای برنامه‌نویسان، ارائه می‌دهد. چند مورد از پر استفاده‌ترین این برنامه‌ها عبارتند از: WAMP Server, PHPtraid, EasyPHP, XAMP, و SQLWare. هر کدام از این برنامه‌ها را نصب کنید، برنامه‌های Apache و MySQL و PHP نیز نصب می‌شوند و نیاز به نصب به‌طور جداگانه ندارند. از بین این برنامه‌ها WAMP Server از محبوبیت بسیار زیادی بین برنامه‌نویسان PHP برخوردار است.

۶-۱- نصب برنامه WAMP Server

پس از تهیه دیسک فشرده مربوط به این برنامه و با دانلود آن از اینترنت، نرم افزار WAMP را روی سیستم خود نصب کنید.

نکته: در این کتاب از نسخه ۲.۱.۰ e WAMP Server به دلیل سادگی نصب و تنظیمات آن استفاده شده است. برای نصب کفایت پس از اجرای فایل نصب و در ویزارد مربوط به آن، مقادیر پیش فرض را با زدن دکمه Next قبول کرده و به این ترتیب نصب انجام می‌شود.

دقت کنید که مسیر ریشه برای قرار دادن فایل‌های مربوط به سایت، به‌طور پیش فرض در همان شاخه نصب WAMP، پوشه WWW انتخاب شده است. اگر مایل بودید می‌توانید این مسیر را عوض کرده و مسیر مورد نظر خودتان را برای ذخیره فایل‌ها و اطلاعات مربوط به سایت تعیین کنید. چگونگی این عمل در مطالعه آزاد در انتهای همین فصل توضیح داده شده است.

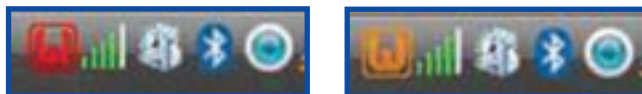
نکته: این مسیر را به خاطر بسپارید زیرا از این پس برای این که بتوانید برنامه‌های خود را اجرا کنید، حتماً باید آنها را در این پوشه ذخیره کنید.

پس از نصب WAMP، روی ناحیه تذکر از نوار وظیفه یک آیکن به شکل حرف W سبز رنگ ظاهر می‌شود. (شکل ۱-۲)



شکل ۱-۲

ظاهر شدن این آیکن با رنگ‌های قرمز یا نارنجی به این معناست که سرویس‌ها به‌طور کامل در حال اجرا نیستند و یا این که به درستی عمل نمی‌کنند. (شکل ۱-۳)



شکل ۱-۳

در این صورت برای اجرای مجدد سرویس‌ها، روی آیکن W کلیک کنید و مطابق شکل ۱-۴ پنجره برنامه باز می‌شود و سپس روی گزینه Restart All Services کلیک کنید.



شکل ۱-۴

نکته: قبل از نصب برنامه WAMP، از نصب نبودن سایر فناوری‌های سرویس‌دهنده وب مثل IIS اطمینان حاصل کنید. چرا که وجود IIS هم می‌تواند موجب شود که سرویس‌های WAMP کامل اجرا نشده و به رنگ سبز در نیاید. همچنین باید با کاربر Administrator به ویندوز وارد شده باشید.

نکته: پس از نصب برنامه WAMP دیگر نیازی به نصب Apache و MySQL و PHP به‌طور مجزا نیست می‌توانید بدون درگیر شدن مستقیم با تنظیمات این برنامه‌ها، کار با زبان PHP را شروع کنید. پس از نصب WAMP می‌توانید عملیات برنامه‌نویسی PHP را آغاز کنید و پس از ذخیره هر برنامه در مسیر ریشه سایت یعنی همان پوشه WWW، اجرای آن را توسط مرورگر مشاهده کنید. برای اجرای برنامه‌ای که نوشته‌اید باید روی آیکن W کلیک کرده و در بالاترین قسمت پنجره برنامه WAMP (شکل ۱-۴) گزینه Local Host را کلیک کرده سپس مرورگری برای شما باز می‌شود که در نوار آدرس آن Local Host را مشاهده می‌کنید، در مقابل Local Host نام صفحه وب (برنامه PHP) خود را نوشته و اجرا کنید. مثلاً اگر نام فایل خود را test.php گذاشته باشید در نوار آدرس باید بنویسید:

<http://localhost/test.php>

در صورت عدم استفاده از WAMP یا سرویس دهنده های مشابه، باید Apache, PHP و MySQL را یکی پس از دیگری نصب و پیکربندی نمایید. روش پیکربندی این برنامه ها به اختصار به صورت زیر است:

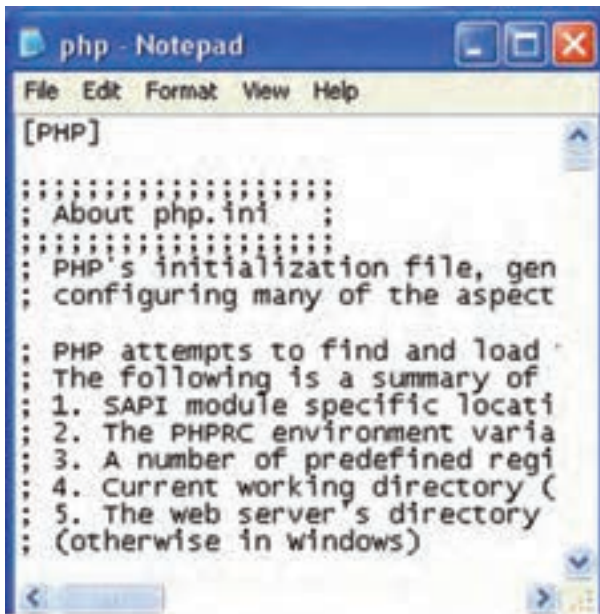
● پیکربندی PHP

در مسیر نصب PHP، فایل php.ini-dist را پیدا کنید.

این فایل را به php.ini تغییر نام داده، سپس آن را در مسیر نصب ویندوز و در پوشه system32 کپی کنید.

PHP برای تنظیم عملیات مختلفی که روی سرویس دهنده انجام می دهد، از دستورات عملیاتی که درون فایل php.ini قرار داده شده است، استفاده می کند و برنامه نویس می تواند بسته به نیاز خود این فایل را تغییر دهد.

دسترسی به این فایل از طریق WAMP نیز امکان پذیر است، برای باز کردن این فایل در شکل ۴-۱، در زیر منوی PHP، روی گزینه php.ini کلیک کنید.



شکل ۴-۱- فایل php.ini

● پیکربندی Apache

برای پیکربندی سرویس دهنده وب Apache، فایلی با نام `httpd.conf` در نظر گرفته شده است. این فایل دستورات مختلفی برای پیکربندی و مدیریت رفتارهای مختلف سرویس دهنده دارد. برای دسترسی به این فایل در شکل ۱-۴ در زیر منوی Apache، گزینه `httpd.conf` را انتخاب کنید.

در ابتدای هر سطر از فایل‌های مربوط به پیکربندی PHP یا Apache، یکی از علائم `#` یا `;` قرار گرفته است. این دو کاراکتر برای درج توضیحات یک خطی به کار برده می‌شوند، به این معنا که هنگام اجرا هر سطری که در ابتدای آن یکی از این دو علامت درج شده باشد، نادیده گرفته می‌شود و برای فعال کردن آن باید این کاراکترها از ابتدای سطر برداشته شوند. (شکل ۱-۵)

نکته: سعی کنید فقط در مواقع لزوم و مطابق دستورالعمل محتویات فایل‌های `httpd.conf` و `php.ini` را دست کاری کنید. در غیر این صورت حتی ممکن است عملکرد این برنامه‌ها به طور کلی مختل شود.

نکته: برای تغییر مسیر پیش فرض WWW می‌توانید در فایل `httpd.conf` در زیر قسمت Document Root، به جای مسیر WWW، مسیر مورد نظر خود را درج نمایید. البته این کار برای برنامه نویسان مبتدی توصیه نمی‌شود.

محتویات صفحات ایستا از طریق کدهایی تعیین می‌شود که از قبل نوشته شده‌اند. اما محتویات صفحات پویا از قبل به طور کامل و ثابت تعریف نمی‌شوند بلکه زمانی که یک داده یا گروهی از داده‌ها درخواست می‌شوند، درخواست مطرح شده به سرویس دهنده اعلام می‌شود. سرویس دهنده با استفاده از برنامه‌های مخصوص، جواب مناسب را یافته، آنها را به صورت کدهای HTML در می‌آورد و سپس آن را برای سرویس گیرنده ارسال می‌کند.

صفحات وب از نظر ماهیت تعامل با سرویس گیرنده به دو گروه تقسیم می‌شوند:

- صفحات وب پویا (Dynamic): دارای تعامل با کاربر هستند و محتویاتشان با درخواست کاربر تغییر می‌کند.

- صفحات وب ایستا (Static): دارای تعامل نیستند و محتویاتشان همواره ثابت است.

زبان‌های برنامه نویسی به طور کلی به دو دسته اصلی تقسیم می‌شوند:

- Client Side

- Server Side

مهم‌ترین سرویس دهنده‌های وب Coldfusion (کلد فیوژن)، IIS و Apache (آپاچی)

هستند.

سرویس دهنده برنامه کاربردی وب، یک نرم افزار است که به سرویس دهنده وب کمک می‌کند محتویات صفحه را به صورت پویا تولید کند.

نوع سرویس دهنده وبی که از آن بهره می‌گیرید مستقیماً در انتخاب سرویس دهنده برنامه کاربردی وب دخالت دارد و هر سرویس دهنده برنامه کاربردی نیز با زبان برنامه نویسی خاصی قابل دستیابی است.

PHP مخفف عبارت Perl Hypertext Preprocessor است و یکی از ساده‌ترین و

سریع‌ترین زبان‌های طراحی صفحات وب پویا است.

خودآزمایی

- ۱- مفهوم زبان‌های client side و server side را توضیح دهید. از هر کدام یک مثال بزنید.
 - ۲- انواع صفحات وب را از نظر ماهیت تعامل با کاربر بیان کنید.
 - ۳- چه عواملی در انتخاب سرویس دهنده برنامه کاربردی وب اهمیت دارد؟
 - ۴- ویژگی‌های زبان PHP را بیان کنید.
-

فصل ۲

ساختار PHP

هدف‌های رفتاری:

پس از آموزش این فصل هنرجو می‌تواند:

- ۱- برچسب‌های معین‌کننده زبان PHP را به شکل‌های مختلف بنویسد.
- ۲- برنامه‌های ساده PHP را در Notepad بنویسد.
- ۳- برنامه‌های PHP را در مرورگر اجرا کند.
- ۴- انواع داده‌ها و متغیرها را تعریف و آنها را در برنامه به کار گیرد.
- ۵- عملگرهای مختلف محاسباتی را شناخته و مورد استفاده قرار دهد.
- ۶- عملگرهای منطقی و مقایسه‌ای را بشناسد.
- ۷- عملگر رشته‌ای الحاق را بشناسد و مورد استفاده قرار دهد.

مقدمه

در جلد اول این کتاب با شکل نوشتن برجسبها در قالب یک فایل HTML آشنا شدید. همانطور که می‌دانید هر زبان برنامه‌نویسی از استانداردها و قواعد مشخصی پیروی می‌کند طراحان زبان PHP نیز مانند سایر زبان‌های اسکریپت‌نویسی و برنامه‌نویسی، ساختارهای مشخصی برای نوشتن دستورات، پیشنهاد داده‌اند که در ادامه با هر کدام از آنها آشنا خواهید شد.

تعریف: به زبانی که کدهای آن لابه‌لای کد HTML نوشته و گنجانده شوند ولی متمایز از HTML پردازش شوند، زبان اسکریپت‌نویسی گفته می‌شود.

۱-۲- شکل نوشتن دستورات PHP

برای نوشتن دستورات PHP چهار روش وجود دارد که در اینجا به معرفی دو روش مرسوم می‌پردازیم. شما به صورت دلخواه می‌توانید هر کدام از این ساختارها را در برنامه مورد استفاده قرار دهید.

روش اول:

در این روش که به روش کوتاه (short) معروف است، دستورات و کدهای PHP درون علائم `<?>` قرار می‌گیرند.

```
<?
```

```
دستورات و کدهای PHP
```

```
?>
```

روش دوم:

در این روش که به روش XML معروف است، دستورات و کدهای PHP درون علائم `<?php >` قرار می‌گیرند.

```
<?PHP
```

```
دستورات و کدهای PHP
```

```
?>
```

توصیه: به منظور تسهیل و حداکثر سازگاری کدها در مرورگرهای مختلف، توصیه می‌شود برای نوشتن کدهای PHP از روش دوم استفاده نمایید.

نکته: معمولاً کدهای PHP درون برچسب‌های HTML نوشته می‌شود و با این برچسب‌ها ترکیب می‌شود. اما پس از نوشتن کدها، باید فایل با پسوند .php ذخیره شود و در صورت ذخیره فایل با پسوند .html یا .htm. کدهای PHP اجرا نخواهند شد.

۲-۲- نوشتن اولین برنامه به زبان PHP

به منظور نوشتن اولین برنامه PHP و اجرای آن کدهای زیر را در برنامه Notepad بنویسید، سپس فایل را با پسوند . PHP در مسیری که پیش از این برای ریشه WAMP تعیین کرده بودیم (پیش فرض www)، ذخیره کنید :

```
<html>
<body>
<?php
Phpinfo();
?>
</body>
</html>
```

برای اجرای این فایل در مرورگر، با کلیک روی آیکن WAMP و انتخاب گزینه Localhost مسیر پوشه‌ها و فایل‌های موجود در مسیر سرویس‌دهنده نشان داده می‌شود، اگر روی عنوان فایل کلیک کنید حاصل اجرای برنامه در پنجره مرورگر، مشخصات کامل نرم افزار PHP نصب شده بر روی کامپیوتر شما خواهد بود. (شکل ۲-۱)



شکل ۲-۱

نکته: تابع (`phpinfo()`) مشخصات کاملی از سیستم شما به همراه اطلاعاتی در مورد نسخه PHP مورد استفاده را می‌دهد.

۱-۲-۲- چاپ یک عبارت در خروجی

در زبان PHP برای نمایش یک عبارت در خروجی می‌توان یکی از دستورات `echo` و `print` را به کار گرفت.

قطعه کد زیر عبارت `Hello World` را در خروجی نمایش می‌دهد:

```
<html>
<body>
<?php
echo "Hello World";
?>
</body>
</html>
```

نکته: در انتهای هر کدام از دستورات PHP باید علامت `"`; درج شود و در واقع علامت سمی کولن به عنوان جداکننده دو دستور مختلف از همدیگر به کار می‌رود. به مثال زیر توجه کنید:

```
<?PHP
echo "Hello World"; echo "welcome";
?>
```

خروجی Hello world welcome

در کد برنامه فوق، با استفاده از دستور `echo` عبارت `Hello World` و پس از آن `welcome` چاپ می‌شود، اما این دو دستور در یک سطر نوشته شده‌اند، یعنی درج علامت `"`; در انتهای هر دستور `echo` به معنای خاتمه دستور جاری و شروع دستور بعدی است و نیازی نیست برای درج دستور جدید، سطر دیگری ایجاد کنید. اما توصیه می‌شود برای بالا بردن خوانایی برنامه از درج دو دستور مختلف در یک سطر اجتناب کنید.

۲-۳- درج توضیحات در کدهای PHP

در زبان PHP برای درج توضیحات یک خطی از علامت // یا # و برای درج توضیحات چند خطی از علامت /* در شروع و از علامت */ برای خاتمه توضیحات استفاده می‌شود. توضیحات درون کدهای برنامه خوانایی آن را بالا می‌برد، از این رو سعی کنید هنگام نوشتن یک برنامه، توضیحات مناسبی را به همراه کدهای آن ارائه دهید. توضیحات در روند اجرای برنامه و خروجی آن تأثیری ندارد.

```
<html>
<body>
<?php
//this is a single line comment
/*
This is
a comment
Block
*/
#this is a single line comment
echo "Hello Word";
?>
</body>
</html>
```

نکته: کدهای php پس از اجرا به کدهای HTML تبدیل می‌شوند. به این معنا که اگر بر روی صفحه وب فوق که حاوی کد PHP است و در حال نمایش توسط مرورگر است، کلیک راست کنید و سپس از منوی باز شده، گزینه View Source را انتخاب نمایید، خواهید دید که تمام کدها به برچسب‌های HTML تبدیل شده‌اند و دستورات PHP در آن مشاهده نمی‌شود. به عبارت دیگر کدهای PHP فقط در سمت سرور دیده می‌شوند و ما در سمت سرور گیرنده هیچ کدی به غیر از HTML نخواهیم داشت.

۲-۴- متغیرها در PHP

یکی از ویژگی‌های اصلی زبان‌های برنامه‌نویسی، توانایی تعریف متغیرها است. متغیرها مکان‌هایی در حافظه هستند که برای ذخیره مقادیر انواع داده‌های مختلف به کار می‌روند. در برنامه به هر متغیر یک نام منحصر به فرد اختصاص داده می‌شود که شناسایی آن را از سایر متغیرها امکان پذیر می‌کند. پس از معرفی یک متغیر می‌توان بارها از آن در جاهای مختلف برنامه و بدون محدودیت استفاده کرد. برای تعریف متغیرها در زبان PHP، علامت \$ در ابتدای نام متغیرها درج می‌شود. در زبان PHP تعریف متغیرها همزمان با مقداردهی آنها صورت می‌گیرد. شکل صحیح و کلی تعریف متغیر به صورت زیر است:

مقدار متغیر = نام متغیر \$

مثال زیر یک متغیر از نوع رشته ای با مقدار hello world و یک متغیر از نوع عددی مقدار ۱۶ را تعریف می‌کند:

```
<?PHP
$txt = "Hello World";
$x=16;
?>
```

نکته: زبان PHP به نوع متغیرها وابسته نیست و لازم نیست هنگام تعریف متغیر، نوع آن نیز معرفی شود، بلکه پس از مقداردهی به طور خودکار نوع متغیر تشخیص داده می‌شود. این توانایی موجب انعطاف این زبان نسبت به بسیاری از زبان‌های برنامه‌نویسی دیگر شده است.

۲-۴-۱- قواعد نامگذاری متغیرها در PHP

برای نامگذاری متغیرهای PHP می‌توانید از حروف الفبای کوچک و بزرگ، اعداد و Underline (_) استفاده کنید.

نام هر متغیر باید با یکی از حروف الفبا یا "_" شروع شود و متغیرهایی که با عدد یا علامتی غیر از حروف الفبا یا "_" شروع می‌شوند، نامعتبر هستند.

لازم به ذکر است که از اعداد، فقط در ابتدای نام متغیرها نمی‌توان استفاده کرد و در هر جای دیگری از نام متغیرها، استفاده از اعداد بلامانع است.

نکته: PHP به حروف بزرگ و کوچک حساس^۱ است. یعنی متغیری که در ابتدا با ترکیب

^۱ Case sensitive

مشخصی از حروف کوچک و بزرگ معرفی می‌شود، در همه جای برنامه باید به همان صورت استفاده شود. در غیر این صورت به عنوان متغیر دیگری در نظر گرفته می‌شود. به طور مثال متغیرهای \$name و \$Name به عنوان دو متغیر متمایز از یکدیگر در نظر گرفته می‌شوند. به طور مثال متغیرهایی با نام‌های زیر معتبرند:

```
$_txt  
$a5Bc  
$A  
$a  
$str23
```

متغیرهایی با نام زیر نامعتبر هستند:

```
$2a  
$#txt  
$+abc  
$my var
```

برنامه زیر نحوه استفاده از متغیرها را نشان می‌دهد. در این برنامه با استفاده از متغیرها، محیط دایره‌ای به شعاع دلخواه r محاسبه شده است:

```
$PI= 3.14;  
$r= 5;  
$circumference = $PI * 2 * $r;  
echo $circumference;
```

خروجی: ۳۱/۴

۲-۵- انواع داده‌ها

در PHP هشت نوع داده مختلف قابل استفاده است. که ما در این کتاب با ۵ نوع آنها کار خواهیم داشت. ۳ نوع دیگر را در ادامه در مطالعه آزاد می‌توانید مطالعه کنید. در مباحث گذشته با روش مقداردهی متغیرها آشنا شدید. هر متغیر براساس مقداری که می‌گیرد، نوع داده‌ای متفاوتی را می‌تواند داشته باشد. شناخت انواع داده‌ها در به کارگیری متغیرهای استفاده شده در برنامه بسیار مؤثر و مفید است.

پنج نوع داده قابل پشتیبانی در PHP که در این کتاب مورد مطالعه قرار می‌گیرند، عبارتند از : Integer , Boolean , String , Float , Array که در ادامه در جدول ۲-۱ به معرفی ساختار هر یک خواهیم پرداخت.

جدول ۲-۱

نوع داده	مثال
Integer	240000 0xABCD 007 -100
Float	3.14 +0.9e-2 -170000.5 54.6E42
Boolean	True , false
String	در ادامه این نوع داده به طور دقیق بررسی شده است.
Array	در ادامه این نوع داده به طور دقیق بررسی شده است.

نکته: در جدول ۲-۱ در سطر integer مثال دوم 0x نشانه کد هگزادسیمال (مبنای ۱۶) می‌باشد.

علاوه بر انواع داده‌ای که مطرح شدند، سه نوع داده دیگر در PHP کاربرد دارند که عبارتند از:

● Object

این نوع داده در مباحث مربوط به شیء‌گرایی کاربرد دارد و همانند نوع داده آرایه، یک نوع ترکیبی به شمار می‌رود.

● Resource

این نوع داده اختصاصی بوده و در نسخه‌های جدید PHP تعریف شده است. متغیر Resource می‌تواند یک فایل باز شده، یک اتصال به پایگاه داده و یا هر نوع داده خارجی دیگری را شامل شود.

در برنامه برخلاف انواع داده دیگر، نمی‌توان این نوع داده را مستقیماً مقداردهی کرد و نسبت به سایر انواع، کاربرد محدودتری دارد.

● Null

اگر متغیری از این نوع داده تعریف شده باشد، شامل هیچ مقداری نیست، به این معنا که یا مقدار Null به آن اختصاص داده شده و یا اینکه مقدار آن توسط توابع ویژه‌ای در PHP حذف شده است.

1-5-2-string

به چند کاراکتر که در کنار یکدیگر قرار می‌گیرند، اصطلاحاً رشته (string) گفته می‌شود. برای معرفی و مقداردهی یک مجموعه کاراکتر به یک متغیر، از علائم " و یا ' استفاده می‌شود.

مثال:

```
<?php
$txt1="Hello World!";
echo $txt1;
?>
```

Hello World!

خروجی :

در یک دستور echo می‌توان دو یا چند عبارت را در خروجی در کنار هم نمایش داد، برای مشخص کردن چند متغیر رشته‌ای یا رشته در کنار هم، می‌توان از کاراکتر نقطه (.) بین هر دو مورد از آنها استفاده نمود.

کاراکتر نقطه، برای الحاق (چسباندن) دو رشته به یکدیگر به کار می‌رود و یکی از اصلی‌ترین عملگرهای مربوط به رشته‌ها به شمار می‌رود.

مثال زیر، روش به کارگیری این عملگر را به خوبی نشان می‌دهد :

```
<?php
$txt1="Hello World!";
$txt2="what a nice day!";
echo $txt1 ." ." $txt2;
?>
```

!Hello World! what a nice day

خروجی :

در این مثال بین دو متغیر رشته‌ای \$txt1 و \$txt2، یک کاراکتر فاصله (space) به منظور جدا کردن مقدار دو متغیر قرار داده شده و به متغیرهای \$txt1 و \$txt2 الحاق شده است.

تمرین : سه متغیر جداگانه تعریف کنید و به هر کدام از آنها به ترتیب روز، ماه و سال تولد خود را نسبت دهید، سپس با استفاده از یک دستور echo تاریخ تولد خود را به صورت روز/ماه/سال در خروجی نمایش دهید.

۲-۱-۵-۲- استفاده از متغیرها در دستور echo

بین علائم نقل قول جفتی (") و نقل قول تکی (') یک تفاوت مهم وجود دارد. اگر به هنگام انتساب یک مقدار به یک متغیر از " " استفاده کنید، در داخل نقل قول می‌توانید از متغیرهای مختلف استفاده نمایید، مادر دستور echo، در صورت استفاده از متغیرها در علائم ' ' نام متغیر به عنوان یک رشته ثابت در نظر گرفته شده، و بدون در نظر گرفتن محتوای آن چاپ می‌شود، با دقت در مثال زیر این موضوع به خوبی آشکار می‌شود :

```
<?php
$txt="Hello World!";
```

```

$x=16;
echo "string is: $txt". "<br/>";
echo ".....". "<br/>";
echo 'my variable is: $x';
?>

```

string is: Hello World! خروجی :

.....

my variable is: \$x

درج
 مقابل دستور echo یک سطر جدید در خروجی ایجاد می کند. نوشتن مثال فوق بدون
 خروجی را به صورت زیر نمایش خواهد داد :
خروجی :

string is: Hello World!..... my variable is: \$x

array-۲-۵-۲

در PHP هر آرایه مجموعه ای اندیس دار به همراه مقدار است و به عبارت دیگر هر آرایه گروهی از متغیرهای کنار هم است که ارجاع به آنها از طریق یک نام مشترک صورت می گیرد. آرایه ها روش بسیار مفیدی برای گروه بندی اطلاعات مرتبط به هم هستند. در PHP آرایه هایی از انواع مختلف را به صورت یک یا چند بعدی می توان ایجاد نمود.

اگر یک لیست از متغیرهای هم نوع را داشته باشید و بخواهید آنها را ذخیره کنید می توانید از ساختار آرایه ای استفاده نمایید.

آرایه ها قادرند تعداد زیادی متغیر را با یک نام یکسان ذخیره کنند و پس از آن برای شناسایی تک تک عناصر آرایه می توان از اندیس هر عنصر استفاده نمود.

به طور مثال برای ذخیره اسامی ۲۰ نفر از هم کلاسی های خود می توانید یک آرایه تعریف کنید که دارای ۲۰ عنصر از نوع رشته باشد :

رشته ۱	رشته ۲	رشته ۳	...	رشته ۲۰
--------	--------	--------	-----	---------

ساختار آرایه ای از نوع string با ۲۰ عنصر

شکل کلی تعریف یک آرایه به صورت زیر است :

```
$ (مقدار عنصر آخر... , مقدار عنصر دوم , مقدار عنصر اول) = نام آرایه
```

مثال :

در مثال زیر یک آرایه با نام cars تعریف شده است. این آرایه دارای چهار عنصر است. که مقاداردهی عناصر آن هنگام تعریف آرایه، صورت گرفته است :

```
<?php
$cars=array("Samand" , "Pars" , "L90" , "Pride");
?>
```

در PHP پس از ایجاد آرایه، برای هر کدام از عناصر آن به طور خودکار یک اندیس اختصاص داده می شود، اندیس های عناصر به طور خودکار از عدد صفر آغاز می شود.

با توضیحات فوق می توان گفت اندیس عناصر آرایه مثال فوق به صورت زیر در نظر گرفته می شود :

```
$cars[0]="Samand";
$cars[1]="Pars";
$cars[2]="L90";
$cars[3]="Pride";
```

مثال :

می خواهیم مورد اول از عناصر آرایه cars را در خروجی نمایش دهیم :

```
<?php
$cars=array("Samand" , "Pars" , "L90" , "Pride");
Echo $cars[0];
?>

Samand
```

خروجی :

تمرین : نام عناصر آرایه cars مربوط به مثال قبل را در خروجی نشان دهید.

دسترسی به کاراکترهای تشکیل دهنده یک رشته، مانند دسترسی به عناصر آرایه است. یک رشته همانند آرایه ای از کاراکترها ذخیره می شود و برای دسترسی به هر کدام از عناصر آن

می‌توان نام متغیر رشته‌ای به همراه اندیس کاراکتر را به کار برد، در این صورت اندیس کاراکتر را باید درون {} یا [] قرارداد.

مثال:

مثال زیر اولین و هفتمین کاراکتر یک رشته را در خروجی چاپ می‌کند.

```
<?php
$str= "Hello World";
echo "first character is: ".$str{0}." and ";
echo "seventh character is: $str[6]";
?>
```

خروجی: first character is: H and seventh character is: W

نکته: اندیس عناصر رشته مانند آرایه‌ها از صفر شروع می‌شود.

۶-۲- دو تابع کاربردی در ارتباط با متغیرها

• تابع isset()

یکی از توابع مهم و کاربردی در PHP تابع isset() است، توسط این تابع می‌توان از معرفی شدن متغیرها اطمینان حاصل کرد.

اگر متغیر مورد نظر قبلاً تعریف شده باشد، مقدار true و در غیر این صورت مقدار false را بر می‌گرداند.

فرض کنید در برنامه‌ای به مقدار یک متغیر نیاز داریم، اما نمی‌دانیم که این متغیر قبلاً معرفی و مقدار شده است یا خیر، در این صورت می‌توانیم از تابع isset() برای بررسی این مسأله استفاده می‌شود.

مثال:

```
if (isset($first_name))
{
    print "$first_name is set" ;
}
```


در PHP علاوه بر دستور echo می‌توان از دستور print برای نمایش در خروجی استفاده نمود.

• تابع () unset

این تابع برای حذف و از بین بردن متغیرها در حافظه به کار برده می‌شود.

مثال:

```
$a= si;  
unset($a);
```

پس از اجرای این کد \$a در برنامه شناخته شده نخواهد شد.

۲-۷- ثابت‌ها

یکی از امکانات اغلب زبان‌های برنامه‌نویسی قابلیت استفاده از ثابت‌ها است. ثابت‌ها مقادیری هستند که در طول اجرای برنامه بدون تغییر باقی می‌مانند. به منظور تعریف ثابت‌ها از کلمه کلیدی define استفاده می‌شود.

نکته: فقط داده‌هایی از نوع Integer , Boolean , String , Float قابل استفاده در

ثابت‌ها هستند.

شکل کلی تعریف ثابت‌ها به صورت زیر است:

```
define("نام ثابت", مقدار ثابت);
```

ثابت‌ها در همه جای برنامه قابل دسترسی بوده و اصطلاحاً به صورت سراسری (global) در نظر گرفته می‌شوند.

برای دسترسی به ثابت‌ها در طول برنامه، نیازی به علامت \$ نیست و دسترسی به آن توسط نام

ثابت صورت می‌گیرد.

در مثال زیر یک ثابت تعریف و توسط دستور echo در خروجی چاپ می‌شود:

```
<?php  
define("pi",3.14);  
echo "pi is: ".pi;  
?>  
pi is: 3.14
```

خروجی:

نکته: قوانین نامگذاری ثابت‌ها همانند متغیرهاست و باید با یکی از حروف الفبا و یا کاراکتر () شروع شود.

نکته: نام ثابت را در دستور echo نباید داخل "" گذاشت.

۲-۸- انواع عملگرها

عملگر نمادی است که عملیات خاصی را روی متغیرها انجام می‌دهد. زبان PHP بسته به نوع متغیرها، عملگرهای متعددی را پشتیبانی می‌کند. عملگرهای اصلی PHP عبارتند از:

- ۱- عملگرهای ریاضی (محاسباتی)
- ۲- عملگرهای مقایسه‌ای
- ۳- عملگرهای منطقی
- ۴- عملگرهای رشته‌ای

۲-۸-۱- عملگرهای ریاضی (محاسباتی)

در جدول زیر لیست انواع عملگرهای ریاضی که در PHP پشتیبانی می‌شوند ارائه شده است.

جدول ۲-۲

عملگرهای ریاضی			
عملگر	عملکرد	مثال به ازای $x=2$ و $y=5$	نتیجه
+	جمع	$x+2$	۴
-	تفریق	$5-x$	۳
*	ضرب	$x*5$	۱۰
/	تقسیم	$y/x=5/2$	۲/۵
%	باقیمانده تقسیم	$y\%x=5\%2$	۱
++	یک واحد افزایش و انتساب	$x++$	۳
--	یک واحد کاهش و انتساب	$x--$	۲

عملگرهای افزایش و کاهش به اندازه ۱ واحد به ترتیب به متغیر اضافه کرده یا از آن کم می‌کنند و مقدار جدید را داخل متغیر قرار می‌دهند.

مثال:

$a=5;$

$a++;$

$++a;$

در مثال بالا ابتدا متغیر a از نوع عددی با مقدار ۵ تعریف شده است، در خط دوم با استفاده از عملگر $++$ یک واحد به آن اضافه شده، یعنی مقدار a به ۶ تغییر پیدا می‌کند، در خط سوم نیز مجدداً یک واحد دیگر به a افزوده می‌شود که در نهایت مقدار a برابر با ۷ می‌شود. این دو دستور معادل هم هستند و فقط در یک مورد با هم تفاوت دارند، به مثال بعد توجه کنید:

مثال:

$a=5;$

$b=\$a ++ ;$

در اینجا پس از مقدار دادن به a ، به دلیل این که عملگر $++$ بعد از نام متغیر آمده، ابتدا مقدار فعلی a (یعنی ۵) در b ریخته می‌شود و بعد به متغیر a یک واحد افزوده می‌شود. یعنی در پایان برنامه متغیر a برابر با مقدار ۶ و متغیر b برابر با مقدار ۵ خواهد بود. به عبارت دیگر این یک دستور؛ $b=\$a++$ به ترتیب معادل دو دستور زیر می‌باشد.

$b=\$a;$

$a=\$a+1;$

مثال:

$a=5;$

$b=++\$a;$

به مجموعه دستورات فوق دقت کنید، این مثال به دلیل این که عملگر $++$ قبل از نام متغیر آمده، در ابتدا یک واحد به a اضافه می‌شود یعنی مقدار a به ۶ تغییر کرده و بعد مقدار a که در حال حاضر ۶ است، در b ریخته می‌شود و در پایان خط سوم متغیرهای a و b حاوی عدد ۶ هستند. به عبارت دیگر این دستور؛ $b=++\$a$ به ترتیب، معادل دو دستور زیر می‌باشد:

$a=\$a+1;$

$b=\$a;$

شیوه عملکرد -- نیز مشابه ++ است با این تفاوت که در -- عمل کاهش انجام می شود.

۲-۸-۲- عملگرهای مقایسه‌ای

این عملگرها برای مقایسه بین اعداد و متغیرها به شکل های مختلف صورت می گیرد و نتیجه این

مقایسه یک مقدار Boolean (درست یا غلط) است. (جدول ۲-۳)

جدول ۲-۳

مثال	عملگر
$8 == 5$ مقدار false برگردانده می شود. $2 == 2$ مقدار true برگردانده می شود.	==
$8 != 5$ مقدار true برمی گرداند.	!=
$8 < 5$ مقدار true برمی گرداند.	<
$8 < 5$ مقدار false برمی گرداند.	<
$8 > 5$ مقدار true برمی گرداند.	>
$8 >= 5$ مقدار true برمی گرداند.	>=
$8 <= 5$ مقدار false برمی گرداند.	<=
$5 === 5$ مقدار true برگردانده می شود. $2 === '2'$ مقدار false برگردانده می شود.	===
$5 !== 5$ مقدار false برگردانده می شود. $2 !== '2'$ مقدار true برگردانده می شود.	!==

رفتار عملگرهای $=!$ و $<>$ و مشابه است و با یکدیگر تفاوتی ندارند. به معنی «مخالف» یا «نامساوی» هستند.

عملگرهای $===$ و $==$ برای بررسی مساوی و یا نامساوی بودن پارامترهای مختلف به کار می‌روند، اما این دو تفاوت‌های ساختاری با یکدیگر دارند.

در عملگر $===$ مقایسه براساس نوع و مقدار انجام می‌شود، اما در عملگر $==$ مقایسه فقط براساس مقدار صورت می‌گیرد.

برای روشن شدن موضوع، به مثال‌های زیر دقت کنید.

مثال:

$a=3;$

$b="3";$

$a==b;$

در دستور سوم ارزش `true` برگردانده می‌شود زیرا ابتدا متغیر `b` نوع رشته‌ای به نوع عددی تبدیل می‌شود و سپس مقایسه صورت می‌گیرد.

همان طور که اشاره شد، اگر بخواهیم تبدیل نوع صورت نگیرد و مقایسه با انواع یکسان صورت گیرد، باید از عملگر $===$ استفاده کنیم:

مثال:

$a=7;$

$b="7";$

$a===b;$

دستور سوم مقدار `false` را بر می‌گرداند، زیرا `a` و `b` هم نوع نیستند.

۳-۸-۲- عملگرهای منطقی

نتیجه برگشت داده شده در این دسته از عملگرها مشابه عملگرهای مقایسه‌ای است.

عملگرهای منطقی نیز نتیجه `true` یا `false` را بر می‌گردانند.

جدول ۲-۴

عملگر	عملکرد	مثال به ازای $x=6$ و $y=3$
&&	and (معادل حرف «و») در جملات محاوره‌ای فارسی)	$x < 1 \text{ \&\& } y > 1$ مقدار true را برمی‌گرداند.
	or (معادل حرف «یا») در جملات محاوره‌ای فارسی)	$(x == 5 \text{ } y == 5)$ مقدار false را برمی‌گرداند.
!	not (نقیض - نفی کننده)	$(x != y)$ مقدار true را برمی‌گرداند.

زبان PHP در عملگر منطقی &&، دو عملوند && را بررسی می‌کند و اگر هر دو true بودند، نتیجه true را برمی‌گرداند. در غیر این صورت، مقدار برگشت داده شده در این عملگر، false خواهد بود. همچنین در دستور || دو عملوند || را بررسی می‌کند حتی اگر یکی از دو طرف true بودند، نتیجه true را برمی‌گرداند و تنها در صورتی نتیجه false می‌شود که هر دو متغیر یا عبارت منطقی، false باشند.

۲-۸-۴- عملگرهای رشته‌ای

عملگر الحاق یا چسباندن (Concatenation)

عملگر (.) برای چسباندن رشته‌ها به یکدیگر به کار می‌رود. با این عملگر، بیشتر در قسمت رشته‌ها آشنا شدید.

عملگرهای تجمعی

یکی از مسائلی که مورد توجه برنامه‌نویسان حرفه‌ای قرار می‌گیرد، سرعت نوشتن و سرعت اجرای دستورات توسط پردازنده است.

در اغلب زبان‌های برنامه‌نویسی که امروزه به کار می‌روند، امکان استفاده از عملگرها به شکل تجمعی است.

عملگرهای تجمعی با سرعت بالاتری توسط پردازنده اجرا می‌شوند و برنامه‌نویسان حرفه‌ای به جای شکل ساده عبارات، توسط عملگرهای محاسباتی، آنها را به صورت تجمعی به کار می‌برند.

جدول ۵-۲

عملگرهای تجمعی		
عملگر	مثال	صورت مشابه عملگر
=	$x=y$	$x=x=y$
+=	$x+=y$	$x=x+y$
-=	$x-=y$	$x=x-y$
=	$x=y$	$x=x*y$
/=	$x/=y$	$x=x/y$
.=	$x.=y$	$x=x.y$
%=	$x%=y$	$x=x.y$

در هر ردیف جدول فوق، یکی از عملگرهای تجمعی به همراه مثال بیان شده است. در ستون "صورت مشابه عملگر"، شکل ساده آن بیان شده است به طور مثال $x+=y$ مانند $x=x+y$ عمل می‌کند.

نوشتن عملگرها به صورت تجمعی، سرعت اجرای برنامه را بالا می‌برد.

پراستفاده‌ترین روش برای نوشتن کدهای PHP به شکل زیر است :

```
<?PHP  
دستورات و کدهای PHP  
?>
```

در زبان PHP برای نمایش یک عبارت در خروجی، یکی از دستورات `echo` و یا `print` استفاده می‌شود و در انتهای هر کدام از دستورات PHP، باید علامت `;` درج شود. در زبان PHP برای درج توضیحات یک خطی از علامت `//` یا `#` و برای درج توضیحات چند خطی از علامت `/*` در شروع و از علامت `*/` برای خاتمه توضیحات استفاده می‌شود. متغیرها مکان‌هایی در حافظه هستند که برای ذخیره مقادیر انواع داده‌های مختلف به کار می‌روند.

برای تعریف متغیرها در زبان PHP، علامت `$` در ابتدای نام متغیرها درج می‌شود. انواع داده قابل پشتیبانی در PHP عبارتند از: `Array`, `Float`, `String`, `Boolean`, `Integer`. در PHP هر آرایه، مجموعه‌ای اندیس دار به همراه مقدار است و آرایه‌ها قادرند تعداد زیادی متغیر را با نام یکسان ذخیره کنند و پس از آن برای شناسایی تک تک عناصر آرایه می‌توان از اندیس هر عنصر استفاده نمود.

به منظور تعریف ثابت‌ها، از کلمه کلیدی `define` استفاده می‌شود و فقط داده‌هایی از نوع `Float`, `String`, `Boolean`, `Integer` قابل استفاده در ثابت‌ها هستند.

عملگر، نمادی است که عملیات خاصی را روی متغیرها انجام می‌دهد و زبان PHP تمام عملگرهای اصلی محاسباتی و مقایسه‌ای و منطقی و رشته‌ای را پشتیبانی می‌کند.

- ۱- یک برنامه برای نمایش نام و نام خانوادگی خود بنویسید و در کنار هر سطر برنامه توضیحات مربوط به عملکرد آن را درج کنید.
 - ۲- با استفاده از متغیرها، مساحت دایره‌ای به شعاع 10 cm را محاسبه کنید.
 - ۳- نام ۴ شهر ایران را در یک آرایه تعریف کنید، سپس با استفاده از روش دسترسی به عناصر آرایه، عنوان شهر اول و سوم را با استفاده از یک دستور echo در خروجی نمایش دهید.
 - ۴- برنامه‌ای بنویسید که میانگین سه عدد مفروض را محاسبه و چاپ نماید.
-

فصل ۳

کنترل روند برنامه

هدف‌های رفتاری:

پس از آموزش این فصل هنرجو می‌تواند:

- ۱- ساختار دستور `if...else` را بشناسد و در مواقع لزوم آن را در برنامه به کار ببرد.
- ۲- با دستور `switch` کار کند.
- ۳- با انواع حلقه‌های تکرار برنامه نویسی کند.

معمولاً تمام زبان‌های برنامه‌نویسی، دستورات مختلفی برای کنترل روند برنامه دارند. زبان PHP تقریباً تمام دستوراتی که در زبان‌های دیگر، برای انجام عملیات کنترلی برنامه در نظر گرفته شده است را پشتیبانی می‌کند.

دستورات کنترلی در زبان‌های برنامه‌نویسی به دو گروه اصلی تقسیم می‌شوند :

– دستورات کنترلی شرطی

– حلقه‌های تکرار

دستورات شرطی، روی جریان اجرای بخشی از برنامه تأثیر می‌گذارند. اما حلقه‌های تکرار تکه‌ای از برنامه را به تعداد دفعات مشخص، یا رسیدن به شرط خاصی اجرا می‌کنند. انواع دستورات شرطی و انواع حلقه‌های تکرار، برای اجرا به متغیرها نیاز دارند و برای اجرای آنها باید متغیرها شرایط خاصی داشته باشند. لذا به علت وابستگی اجرای دستورات به مقدار متغیرها، لازم است در طول برنامه مقادیر متغیرها ارزیابی شوند.

۱-۳- دستور if

یکی از کاربردی‌ترین دستورات زبان‌های برنامه‌نویسی، دستور شرطی if است. این دستور مفهومی شبیه به همان کلمه "if" در زبان انگلیسی و کلمه "اگر" در زبان فارسی دارد. عملکردی کاملاً مشابه جملات شرطی محاوره‌ای دارد. ساختار کلی دستور if به صورت زیر است :

if (عبارت شرطی)

{

دستورات بدنه در صورت برقراری عبارت شرطی

}

else

{

دستورات بدنه در صورت عدم برقراری عبارت شرطی

}

کد زیر یک برنامه ساده است که با دستور شرطی if نوشته شده است :

```
if ( $a == 0 ) {
```

```
echo " found a 0 ! " ;  
}
```

شرط فوق مقدار a را با صفر مقایسه می‌کند، اگر a برابر با صفر بود پیغامی به صورت "found a 0" را در مرورگر نمایش می‌دهد.

نکته: اگر دستورات بدنه که باید در صورت برقراری شرط اجرا شوند، از یک خط بیشتر نباشد می‌توان از نوشتن { } صرف‌نظر کرد.

۱-۱-۳- دستور else

یکی از قابلیت‌های دستور شرطی if این است که برنامه‌نویس می‌تواند معین کند که اگر شرط برقرار بود، یک عمل خاص و در غیر این صورت عمل دیگری را انجام دهد.

else به معنای "در غیر این صورت" در زبان انگلیسی با همان معنا در برنامه‌نویسی نیز به کار برده می‌شود.

شرط if مثال اول را طوری دوباره‌نویسی می‌کنیم که اگر مقدار a برابر با صفر نبود پیغام دیگری را نمایش دهد:

```
if ( $a == 0 )  
{  
echo " found a = 0 ! " ;  
}  
else  
{  
echo " not found ! " ;  
}
```

۲-۱-۳- دستور elseif

اگر در دستور شرطی if بخواهید یک مجموعه از شرط‌ها را مدیریت کنید، باید از دستور elseif استفاده کنید.

این دستور، مانند چند دستور شرطی مستقل، عمل می‌کند و در صورت برقراری هر کدام از شرط‌ها، کدهای مشخصی را برای اجرا انتخاب می‌کند.

برنامه‌زیر برای عددی که در متغیر \$num قرار دارد ضمن انجام عمل مقایسه توسط دستور if، مثبت، منفی و یا مساوی صفر بودن را تعیین می‌کند:

```

$num=0;
if ($num < 0)
{
echo "$num is negative";
}
elseif ($num == 0)
{
echo "$num is zero";
}
elseif ($num > 0)
{
echo "$num is positive";
}

```

مطالعه آزاد

همان‌طور که می‌دانید کدهای PHP را می‌توان با HTML ترکیب کرد، اگر بخواهید کد مثال فوق را به صورت ترکیب کدهای PHP و HTML بازنویسی کنید، می‌توانید از دستور echo صرف نظر کرده، تنها از پرچسب‌های HTML برای نمایش متن استفاده نمایید. در این صورت می‌توان کد مثال قبل را به صورت زیر بازنویسی کرد :

```

<?php
$num=0;
if ($num < 0): ?>
<h1><?php echo "$num";?> is negative</h1>
<?php elseif ($num == 0): ?>
<h1><?php echo "$num";?> is zero</h1>
<?php elseif ($num > 0): ?>
<h1><?php echo "$num";?> is positive</h1>
<?php endif; ?>

```

در اینجا هنگام استفاده از این ساختار، به دلیل جدا شدن کدهای PHP از یکدیگر، می‌توان به جای { هنگام شروع بلوک کد، از علامت " : " استفاده نمود و در این صورت نیازی به تعیین مکان { برای خاتمه بلوک کد نیست و در انتها باید با استفاده از دستور endif تمام دستور شرطی معین شود.

۳-۱-۳ عملگر شرطی؟

یکی از قابلیت‌های جالب زبان PHP، وجود عملگر شرطی؛ است که از آن برای پیاده‌سازی دستورات شرطی می‌توان استفاده نمود. شکل کلی این دستور به صورت زیر است:

عملیات ۲: عملیات ۱؟ (عبارت شرطی)

عملیات ۲ در صورت غلط بودن عبارت شرطی و عملیات ۱، در صورت صحیح بودن عبارت شرطی انجام می‌شود.

مثال: در مثال زیر اگر مقدار نمره (\$num)، مساوی یا بیشتر از ۱۲ باشد، در خروجی رشته Pass (قبول) چاپ می‌شود و اگر کمتر از ۱۲ باشد، در خروجی رشته fail (به معنای مردود) چاپ می‌گردد.

```
<?php
$num =13;
$result=($num>=12)? "pass" : "fail";
print "$result";
?>
```

خروجی این مثال عبارت است از:

pass

۳-۲-۳ دستور switch

مثال زیر را با دستور if... else... در نظر بگیرید در این مثال به ازای مقادیر مختلف یک متغیر، دستورات متفاوتی اجرا می‌شود. یعنی در واقع چند شرط را برای یک متغیر بررسی می‌کنیم.

مثال:

```
$a=3;
if ( $a == 0 )
    echo " found a 0 ! " ;
elseif ( $a == 1 )
    echo " found a 1 ! " ;
elseif ( $a == 2 )
    echo " found a 2 ! " ;
elseif ( $a == 3 )
```

```
echo " found a 3 ! " ;
```

خروجی : found a 3!

اگر چه ساختار مثال فوق ساده است، اما در صورتی که تعداد شرط‌ها و تعداد دستوراتی که باید در صورت درستی هر شرط اجرا شوند زیاد باشد، ممکن است باعث پیچیدگی برنامه‌ها شود. برای چنین ساختارهایی در زبان‌های برنامه‌نویسی، از جمله PHP، دستور switch در نظر گرفته شده است. دستور if elseif مثال قبل با دستور switch به این صورت نوشته می‌شود.

مثال :

```
switch ( $a )
{
    case "0" :
echo " found a 0 ! " ;
    break;
case "1" :
    echo " found a 1 ! " ;
    break;
case "2" :
    echo " found a 2 ! " ;
    break;
case "3" :
    echo " found a 3 ! " ;
    break;
default:
    echo " not found ! " ;
}
```

اجزای دستور switch عبارتند از :

- **کلمه switch** : مقابل کلمه switch متغیر یا عبارتی که باید مقادیر مختلف آن بررسی شود قرار می‌گیرد.
- **کلمه case** : مقادیر مختلف و مرتبط با متغیر مقابل عبارت switch تعیین می‌شود.
- **دستور default** : برای تعیین دستورات پیش فرض ساختار switch استفاده می‌شود. در

واقع توسط این دستور می‌توان تعیین کرد که در صورت عدم برقراری تمام مقادیر موجود در case ها، چه عملی انجام شود.

نکته: در هر case از برنامه، برای جلوگیری از اجرا شدن case بعدی، باید از دستور break برای خروج از آن قسمت از برنامه استفاده کرد. اگر درون یک دستور case از break استفاده نشود، دستور مربوط به case بعدی اجرا می‌شود.

مثال: برنامه زیر محدوده نمره دانش‌آموز را بررسی می‌کند چنانچه بین صفر و کمتر از ۱۰ باشد رتبه F و اگر بین ۱۰ تا ۱۲ باشد رتبه E و اگر بیشتر از ۱۲ تا ۱۴ باشد رتبه D و ... را برای دانش‌آموز چاپ می‌کند.

```
$num = 15;
switch ($num)
{
case (($num<10)&& ($num>=0)):
    echo "your grade is F";
    break;
case (($num>=10)&& ($num<=12)):
    echo "your grade is E";
    break;
case (($num>12)&& ($num<=14)):
    echo "your grade is D";
    break;
case (($num>14)&& ($num<=16)):
    echo "your grade is C";
    break;
case (($num>16)&& ($num<=18)):
    echo "your grade is B";
    break;
case (($num>18)&& ($num<=20)):
    echo "your grade is A";
    break;
default:
    echo "It is a wrong number.";
}
```


خروجی :

your grade is C

۳-۳- حلقه‌های تکرار

فرض کنید می‌خواهیم یک عبارت را سه بار در خروجی نمایش دهیم، این برنامه را به صورت زیر خواهیم نوشت :

```
echo " hello " ;
```

```
echo " hello " ;
```

```
echo " hello " ;
```

حال فرض کنید همان عبارت را می‌خواهیم ۱۰۰ بار در خروجی نمایش دهیم، احتمالاً خواهید گفت صد بار یک خط از برنامه فوق را می‌نویسیم. اما در زبان‌های برنامه‌نویسی امکاناتی وجود دارد که به برنامه‌نویس کمک می‌کند بدون نیاز به کار تکراری، یک مجموعه عملیات را به تعداد دفعات مورد نیاز انجام دهد. به این ساختارها "حلقه" گفته می‌شود.

مفهوم حلقه : حلقه‌ها برای اجرای یک یا چند دستورالعمل به تعداد دفعات مشخص یا تا زمان رسیدن به شرط خاصی به کار برده می‌شوند. حلقه‌های زبان PHP دارای دو نوع کلی `for` و `while` هستند.

۱-۳-۳- حلقه `for`

این حلقه زمانی کاربرد دارد که می‌خواهیم مجموعه‌ای از عملیات، به تعداد دفعات معینی انجام شوند. ساختار حلقه `for` به صورت زیر است :

(میزان افزایش یا کاهش متغیر شمارنده حلقه؛ عبارت کنترلی؛ مقدار اولیه متغیر) `for`

```
{  
دستورات بدنه حلقه  
}
```

- **مقدار اولیه متغیر :** مقدار اولیه برای شروع اجرای حلقه را مشخص می‌کند.
- **عبارت کنترلی :** در این قسمت یک عبارت شرطی درج می‌شود که در هر بار اجرای حلقه مقدار متغیر شمارنده حلقه توسط این عبارت کنترلی که در واقع شرط انجام حلقه است، چک می‌شود، اگر جواب شرط `True` بود، اجرای حلقه ادامه می‌یابد و در غیر این صورت اجرای حلقه پایان می‌پذیرد.
- **میزان افزایش یا کاهش :** مقدار متغیر شمارنده حلقه را تعیین می‌کند که در هر بار اجرای دستورات مقدار این متغیر افزایش یا کاهش یابد.

نکته : از میان سه پارامتر اصلی ساختار حلقه `for`، نوشتن مقدار اولیه و میزان افزایش یا کاهش اختیاری است اما نوشتن عبارت کنترلی در ساختار `for` اجباری است.

در صورتی که میزان افزایش یا کاهش حلقه مقابل for نوشته نشود، می‌توان آن را در بدنه حلقه تعیین کرد و افزایش یا کاهش داد. مقدار اولیه را نیز می‌توان قبل از شروع حلقه for مقداردهی کرد.

```
<?php
for ( $n = 2 ; $n <= 5 ; $n++ )
{
    echo "number: " . $n . "<br/>";
}
?>
```

خروجی :

```
number: 2
number: 3
number : 4
number: 5
```

در مثال فوق مقدار اولیه متغیر \$n برابر با ۲ تعریف شده است و شرط خاتمه حلقه توسط عبارت کنترلی $n \leq 5$ تعیین شده است، مقدار \$n نیز در هر مرحله، ۱ واحد افزوده می‌شود.

مثال :

کد زیر اعداد ۱ تا ۳ را در خروجی چاپ می‌کند.

```
<?php
for ( $n = 1 ; $n <= 5 ; $n++ )
{
    echo("number: " . $n . "<br/>");
    if ( $n == 3 ) break ;
}
?>
```

خروجی :

```
number: 1
number: 2
```

number: 3

نکته: ممکن است گاهی اوقات هنگام اجرای حلقه، تحت شرط یا شرایط خاصی بخواهید از آن خارج شده و ادامه اجرا را متوقف نمایید. در این شرایط می‌توانید از دستور break استفاده کنید. مانند آنچه در مثال فوق انجام شد.

مثال:

کد زیر مجذور اعداد کوچک‌تر از ۱۰ را نمایش می‌دهد.

```
<?php
for ($i = 0; $i < 10; $i++) {
print "The square of $i is " . $i*$i . "<br/>";
}
?>
```

خروجی:

The square of 0 is 0

The square of 1 is 1

The square of 2 is 4

The square of 3 is 9

The square of 4 is 16

The square of 5 is 25

The square of 6 is 36

The square of 7 is 49

The square of 8 is 64

The square of 9 is 81

نکته: علاوه بر دستور break، دستور continue نیز در حلقه‌ها کاربرد دارد. دستور continue ادامه اجرای حلقه با مقدار شمارنده فعلی را متوقف می‌کند و اجرای حلقه را با مقدار شمارنده بعدی از سر می‌گیرد. برای فهم بهتر به مثال بعدی توجه کنید.

نکته: تأثیر break و continue بر دستورات بعد از خودشان است و دستورات قبل از آنها به طور کامل اجرا می‌شوند.

مثال:

```
<?php
```

```

for ( $n = 1 ; $n <= 5 ; $n++ )
{
    if ( $n == 2 ) continue ;
    echo("number: ".$n."<br />");
}
?>

```

خروجی :

```

number: 1
number: 3
number: 4
number: 5

```

نکته : برای چاپ چند عبارت یا متغیر توسط دستور echo باید عبارات یا متغیرهای مورد نظر توسط علامت . به یکدیگر چسبانده شوند . به عنوان مثال ; echo("number: ".\$n); عبارت number: و به دنبال آن مقدار متغیر \$n را چاپ می کند .

نکته : در دستور echo درج عبارت "
" باعث می شود که مکان نما پس از چاپ عبارت موجود، به ابتدای سطر بعد منتقل شود .

مثال :

چاپ اعداد ۵ تا ۲ با مقداردهی اولیه خارج از حلقه به صورت زیر است :

```

<?php
$n=5 ;
for ( ; $n >= 2 ; $n -- )
{
    echo("number: " . $n . "<br />");
}
?>

```

خروجی :

```

number: 5
number: 4
number: 3
number: 2

```

مثال: مثال فوق با انتقال عمل کاهش متغیر شمارنده به داخل بدنه حلقه به صورت زیر است.

```
<?php
$n=5 ;
for ( ; $n >= 2 ; )
{
    echo("number: " . $n . "<br/>");
    $n-- ;
}
?>
```

خروجی :

```
number: 5
number : 4
number: 3
number: 2
```

۲-۳-۳- حلقه foreach

این نوع از حلقه برای کار با عناصر آرایه در نظر گرفته شده است. ساختار کلی حلقه foreach به صورت زیر است :

```
foreach ($array as نام متغیر)
{
    بدنه حلقه و عباراتی که باید اجرا شوند
}
```

هنگام اجرای این حلقه، عناصر آرایه از ابتدا تا انتها یکی یکی در متغیری که به عنوان ورودی Foreach ذکر شده، قرار می گیرند در مثال زیر \$value است. نام متغیر \$value می تواند هر نام دلخواه برنامه نویس باشد. برای آشنایی با عملکرد این حلقه به مثال زیر توجه کنید :

مثال:

```
<?php
$x=array("one","two","three");
```

```
foreach ($x as $value)
```

```
{  
echo $value . "<br />";
```

```
    {  
?>
```

در اولین اجرای حلقه، اولین مقدار آرایه در متغیر \$value قرار می‌گیرد و به دلیل اینکه دستور بدنه foreach، چاپ این متغیر و سپس رفتن به ابتدای سطر بعد است، خروجی به صورت زیر خواهد بود:

```
one  
two  
three
```

۳-۳-۳- حلقه while

حلقه while نوعی حلقه تکرار است و زمانی از آن استفاده می‌شود که بخواهیم عملیات مشخصی، تا زمانی که شرط تعیین شده‌ای برقرار باشد، اجرا شود. ساختار حلقه while به این صورت است:

```
while (شرط حلقه)  
{  
    دستورات بدنه حلقه  
    به روز رسانی متغیر  
}
```

مثال:

برنامه زیر با استفاده از حلقه while اعداد ۲ تا ۵ را نمایش می‌دهد:

```
<?php  
$n=2 ;  
while ( $n <= 5 )  
{  
echo("number: " . $n. "<br/>");  
$n++;  
}  
?>
```

خروجی :

number: 2

number: 3

number: 4

number: 5

تمرین: با استفاده از حلقه تکرار while، اعداد زوج کوچک‌تر از ۱۰ را نمایش دهید.

۴-۳-۳- حلقه do ... while

ساختار حلقه do ... while مانند حلقه while است، تنها تفاوت این دو نوع حلقه در این است که حلقه do ... while شرط را در انتها کنترل می‌کند، یعنی، پس از هر بار اجرای دستورات حلقه شرط در انتها بررسی می‌شود و اگر شرط برقرار باشد دفعه بعد نیز اجرای حلقه از سر گرفته می‌شود، اما اگر شرط برقرار نباشد، بدون بازگشت به ابتدای حلقه، از حلقه while خارج می‌شوند.

نکته: در حلقه while در صورت عدم برقراری شرط، دستورات بدنه حلقه هرگز اجرا نمی‌شود اما در do ... while حتی در صورت برقرار نبودن شرط دستورات بدنه، حلقه حداقل یک بار اجرا می‌شود.

ساختار حلقه do ... while به صورت زیر است :

```
do
{
دستورات بدنه حلقه
}
while (شرط حلقه)
```

مثال:

برنامه زیر با استفاده از حلقه do اعداد ۲ تا ۵ را نمایش می‌دهد :

```
<?php
$n=2 ;
do
{
    echo("number: " . $n. "<br/>");
    $n++;
}
while( $n<=5)
?>
```

خروجی :

number: 2

number: 3

number: 4

number: 5

مثال: در کد زیر از یک حلقه do...while استفاده شده است که شرط خاتمه آن این است

که متغیر \$n کوچک تر یا مساوی ۳ باشد، اما همان طور که در کد ملاحظه می کنید، این متغیر قبل از شروع حلقه با عدد ۵ مقداردهی شده است. یعنی با وجود برقرار نبودن شرط اجرای حلقه، عملیات بدنه while یک بار اجرا می شود.

```
<?php
$n = 5 ;
do
{
    echo(" number : " . $n . "<br />");
    $n++;
}
while ( $n <= 3 )
?>
```

number : 5

خروجی :

تمرین: با استفاده از حلقه تکرار do ... while اعداد زوج کوچک تر از ۱۰۰ را نمایش دهید.

دستورات کنترلی در زبان‌های برنامه نویسی به دو گروه اصلی تقسیم می‌شوند:

– دستورات کنترلی شرطی

– حلقه‌های تکرار

دستورات شرطی روی جریان اجرای بخشی از برنامه تأثیر می‌گذارند اما حلقه‌های تکرار تکه‌ای

از برنامه را به تعداد دفعات مشخص یا تا زمان رسیدن به شرطی خاص، اجرا می‌کنند.

یکی از کاربردی‌ترین دستورات زبان‌های برنامه نویسی، دستور شرطی `if` است که عملکردی

کاملاً مشابه جملات شرطی محاوره‌ای دارند.

در دستور `switch` از `default` برای تعیین دستورات پیش‌فرض ساختار `switch` استفاده

می‌شود.

حلقه‌های زبان PHP دارای دو نوع کلی `for` و `while` هستند.

از حلقه `while` زمانی استفاده می‌شود که بخواهیم عملیات مشخصی، تا زمانی که شرط تعیین

شده‌ای برقرار باشد، اجرا شود.

حلقه `for` زمانی کاربرد دارد که می‌خواهیم مجموعه‌ای از عملیات به تعداد دفعات معینی

انجام شوند.

- ۱- برنامه‌ای بنویسید که اعداد کوچک تر از 100 و بزرگ‌تر از 20 را به همراه زوج یا فرد بودن آن با استفاده از دستورات `for` و `if`، چاپ کند.
 - ۲- زوج یا فرد بودن اعداد در سؤال اول را با دستور `switch` به جای `if` بررسی کنید.
 - ۳- برنامه سؤال اول را با استفاده از دستورات `do ... while` و `if` بازنویسی کنید.
 - ۴- یک آرایه 5 عنصری برای معرفی رنگ‌های مورد علاقه خود تعریف کنید، سپس با استفاده از دستور `foreach` آنها را در مرورگر نمایش دهید.
 - ۵- برنامه‌ای بنویسید که مجموع ارقام یک عدد مفروض را چاپ نماید.
 - ۶- برنامه‌ای بنویسید که براساس شماره‌های 1 تا 4 به ترتیب چهار عمل اصلی `+`، `-`، `÷` و `×` را روی دو عدد مفروض انجام دهد.
- یعنی اگر متغیر مورد نظر مقدار 1 داشت عمل جمع و اگر مقدارش 2 بود عمل تفریق و ... بر روی دو عدد دلخواه انجام گیرد.

فصل ۴

توابع

هدف‌های رفتاری:

پس از آموزش این فصل هنرجو می‌تواند:

۱- مهم‌ترین انواع توابع رشته‌ای مطرح شده را توضیح داده و هر

یک را به کار ببرد.

۲- در برنامه از توابع تاریخ و زمان استفاده کند.

۳- از توابع مهم ریاضی در برنامه PHP استفاده کند.

۴- با توجه به نیاز برنامه، توابع جدیدی را تعریف کند.

در PHP حدود ۷۰۰ تابع به صورت درون ساخته وجود دارد که برنامه نویسان به راحتی می توانند در برنامه های خود از آنها استفاده کنند. برخی از این توابع مربوط به متغیرهای رشته ای هستند و به منظور کار با رشته ها طراحی شده اند. در ادامه با مهمترین توابع کار با رشته ها و چند نمونه تابع پر کاربرد دیگر آشنا خواهید شد.

۴-۱- توابع رشته ای

۴-۱-۱- تابع strlen

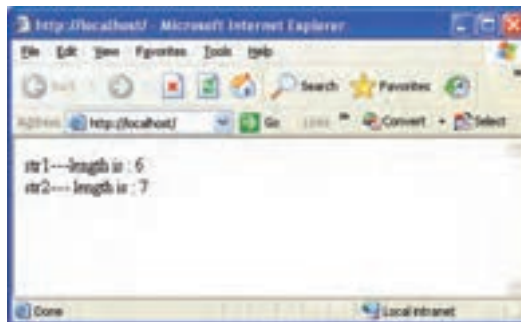
این تابع طول رشته ای را که به عنوان پارامتر ورودی به آن داده می شود برمی گرداند. در نتیجه مقدار برگشت داده شده آن یک عدد صحیح است. به عبارت دیگر کاراکترهای تشکیل دهنده یک رشته را شمرده و تعداد آنها را برمی گرداند. ساختار این تابع به صورت زیر است :

strlen ("پارامتر ورودی");

مثال :

```
<?php
$str1 = "abcdef";
echo "$str1" . "----length is : " . strlen($str1) . "<br/>";
$str2 = " ab cd ";
echo "$str2" . "----length is : " . strlen($str2);
?>
```

خروجی :



شکل ۴-۱- نتیجه اجرای تابع strlen

۲-۱-۴- تابع () strtoupper

این تابع، رشته‌ای را که به عنوان پارامتر ورودی می‌گیرد، به حروف الفبای بزرگ انگلیسی تبدیل می‌کند.

مثال:

```
<?php
$str1 = "aBc def";
echo "$str1". "----strtoupper is : ".strtoupper($str1). "<br/>";
?>
```

خروجی:

aBc def ---- strtoupper is: ABC DEF

۳-۱-۴- تابع () strtolower

این تابع عکس strtoupper عمل می‌کند و تمام حروف پارامترهای ورودی را به حروف کوچک الفبای انگلیسی تبدیل می‌کند.

مثال:

```
<?php
$str1 = "aBc def";
echo "$str1". "----strtolower is : ".strtolower($str1). "<br/>";
?>
```

خروجی:

aBc def ---- strtolower is: abc def

۲-۴- توابع تاریخ و زمان

۱-۲-۴- تابع () date

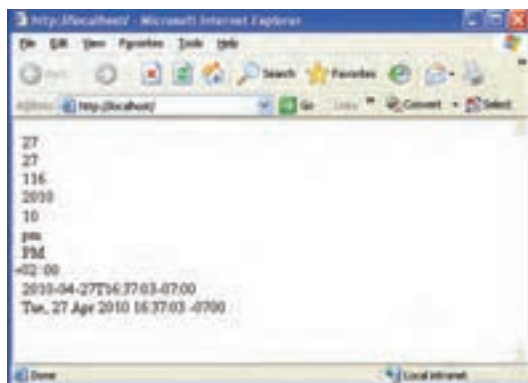
یک پارامتر را به عنوان ورودی دریافت می‌کند و سپس یک رشته را به عنوان خروجی برمی‌گرداند. پارامتر ورودی یک قالب درخواستی برای برگرداندن تاریخ است. هر کدام از قالب‌ها یک شناسه استاندارد دارند که در جدول ۴-۱ چند نمونه از این شناسه‌ها بیان شده‌اند.

جدول ۴-۱

شناسه	عملکرد	مثال (نمونه رشته خروجی)
d	شماره روز ماه به صورت عدد (در صورتی که شماره تک رقمی باشد به ابتدای آن صفر را اضافه می‌کند)	01 تا 31
j	شماره روز ماه به صورت عدد (بدون درج صفر)	1 تا 31
z	تعداد روزهای گذشته از سال	0 تا 365
Y	شماره سال را به صورت یک عدد ۴ رقمی برمی‌گرداند.	1999 یا 2003
y	شماره سال را به صورت یک عدد ۲ رقمی برمی‌گرداند.	99 یا 03
a	فقط تعیین می‌کند که قبل از ظهر است یا بعد از ظهر (با حروف الفبای کوچک انگلیسی)	am or pm
A	فقط تعیین می‌کند که قبل از ظهر است یا بعد از ظهر (با حروف الفبای بزرگ انگلیسی)	AM or PM
P	تفاوت زمانی منطقه جاری را با زمان به وقت گرینویچ نشان می‌دهد.	+02: 00
c	سال - روز - ماه - ساعت - دقیقه - ثانیه	2010_02_12T15: 19: 21+00: 00
r	سال - روز - ماه - ساعت - دقیقه - ثانیه	Thu, 21 Dec 2000 16: 01: 07 +0200

مثال:

```
<?php
echo date("d")."<br/>";
echo date("j")."<br/>";
echo date("z")."<br/>";
echo date("Y")."<br/>";
echo date("y")."<br/>";
echo date("a")."<br/>";
echo date("A")."<br/>";
echo date("P")."<br/>";
echo date("c")."<br/>";
echo date("r")."<br/>";
?>
```



شکل ۴-۲

۴-۲-۲- تابع time()

تابع time() در PHP تعداد ثانیه‌های گذشته از اولین روز سال ۱۹۷۰ میلادی تا لحظه جاری را نشان می‌دهد.

اگر کد زیر را اجرا کنید، شکل خروجی این تابع را مشاهده خواهید کرد.

مثال:

```
<?php
$Now = time();
echo $Now;
?>
```

خروجی:

۱۲۷۲۴۱۲۰۶۷

تابع time() معمولاً در محاسبه‌ی زمان انقضای یک عملیات خاص مثلاً در آزمون‌های online و یا همان‌طور که در فصل‌های آینده خواهید دید در بحث cookieها و جلسه‌ها کاربرد دارد.

۴-۳- توابع ریاضی

۴-۳-۱- تابع abs()

این تابع به منظور محاسبه قدر مطلق یک پارامتر به کار می‌رود.

مثال:

```
<?php
$num1 = -243;
```

```

echo (" $num1 " . "<br/>");
$num2 = abs($num1);
echo "$num2" . "<br/>";
?>

```

خروجی :

```

-۲۴۳
۲۴۳

```

۲-۳-۴- تابع floor ()

این تابع عددی را که به عنوان پارامتر ورودی دریافت می‌کند به عدد صحیح کوچک‌تر تبدیل می‌نماید و مقدار اعشار آن را از بین می‌برد.

مثال زیر با استفاده از پارامترهای مختلف، رفتار این تابع را به خوبی نشان می‌دهد.

مثال :

```

<?php
echo(floor(0.60) . "<br />");
echo(floor(0.40) . "<br />");
echo(floor(5) . "<br />");
echo(floor(5.1) . "<br />");
echo(floor(-5.1) . "<br />");
echo(floor(-5.9))
?>

```

خروجی :

```

0
0
5
5
-6
-6

```

۳-۳-۴- تابع fmod ()

این تابع دو یا چند پارامتر را دریافت و باقیمانده حاصل تقسیم پارامتر اول بر پارامتر دوم

را برمی گرداند.

مثال:

```
<?php
$r = fmod(15,2);
echo $r
?>
```

خروجی:

1

۴-۳-۴- تابع max ()

این تابع دو یا چند پارامتر را به عنوان ورودی دریافت می کند، سپس عدد بزرگ تر را به عنوان خروجی برمی گرداند.

مثال:

```
<?php
echo(max(5,7,2,0) . "<br />");
echo(max(_3,5,0) . "<br />");
echo(max(_3,_5) . "<br />");
echo(max(7.25,7.30))
?>
```

خروجی:

7

5

-3

7.3

۴-۳-۵- تابع min ()

این تابع دو یا چند پارامتر را به عنوان ورودی دریافت می کند، سپس عدد کوچک تر را به عنوان خروجی برمی گرداند.

مثال:

```
<?php
echo(min(5,7,2,0) . "<br />");
echo(min(_3,5,0) . "<br />");
```

```
echo(min(-3,-5) . "<br />");  
echo(min(7.25,7.30))  
?>
```

خروجی :

```
0  
-3  
-5  
7.25
```

۶-۳-۴- تابع rand ()

از تابع rand () برای تولید اعداد صحیح تصادفی استفاده می‌شود. اگر این تابع بدون پارامتر ورودی فراخوانی شود، عددی که تولید می‌شود مقداری بین صفر و بیشترین عدد صحیح روی سیستم خواهد داشت. اما چنانچه بخواهید محدوده مشخصی برای تولید اعداد تصادفی در نظر بگیرید باید دو پارامتر ابتدا و انتهای محدوده را به عنوان ورودی تابع قرار دهید.

به طور مثال حاصل فراخوانی تابع به صورت rand() عدد تصادفی صحیح و بزرگ‌تر از صفر خواهد بود. اما با فراخوانی تابع به شکل rand(۱۰۰,۱۰۰) عددی تصادفی در محدوده بین ۱۰ و ۱۰۰ تولید خواهد کرد.

مثال:

```
<?php  
echo(rand() . "<br />");  
echo(rand() . "<br />");  
echo(rand(10,100)) . "<br />");  
echo(rand(100,300))  
?>
```

خروجی :

```
2776  
37974  
35  
186
```

۷-۳-۴- تابع (round)

این تابع اعداد اعشاری را به نزدیک‌ترین اعداد صحیح تبدیل می‌کند.

مثال:

```
<?php
echo(round(0.60) . "<br />");
echo(round(0.50) . "<br />");
echo(round(0.49) . "<br />");
echo(round(-4.40) . "<br />");
echo(round(-4.60))
?>
```

خروجی:

```
1
1
0
-4
-5
```

۴-۴- تعریف توابع جدید

همان‌طور که گفته شد در زبان PHP توابع متعددی در نظر گرفته شده‌اند که در روند برنامه نویسی بسیاری از مسائل را تسهیل می‌کنند. اما ممکن است یک برنامه نویس در نظر داشته باشد یک سری عملیات مشخص را به دفعات در برنامه خود استفاده کند. در این گونه مواقع می‌توان بدون نیاز به درج کدهای تکراری، یک تابع طراحی نمود و هر جای برنامه که لازم باشد، از نام تابع برای فراخوانی آن استفاده نمود.

ساختار کلی تعریف تابع جدید به شکل زیر است:

```
function نام_تابع ( پارامترهای ورودی )
{
    بدنه تابع
}
```

نکته: اگر تابع پارامتر ورودی نداشته باشد، باید پیرانتزهای جلوی نام آن را خالی بگذارید. برنامه زیر یک تابع ساده با نام hello می‌سازد که عبارت hello, welcome! را در خروجی چاپ می‌کند.

```
function hello()  
{  
    echo ("hello , welcome!");  
}
```

سعی کنید در انتخاب نام تابع دقت کنید، و نام آن را از روی عملکردش انتخاب نمایید. پس از نوشتن تابع، در برنامه و هر جایی که نیاز به تابع داشته باشید کافی است نام تابع را بنویسید. به این عمل فراخوانی تابع گفته می‌شود.

```
<?php  
function hello()  
{  
    echo ("hello , welcome!");  
}  
  
hello();  
?>
```

خروجی:

hello, welcome!

هر تابع می‌تواند یک یا چند پارامتر ورودی داشته باشد. فرض کنید قصد داریم برنامه‌ای طراحی کنیم که در صفحه مرورگر کاربر، با استفاده از یک تابع پیام خوشامدگویی به همراه نام او را نشان دهد. در این صورت نام کاربر به عنوان پارامتر ورودی در اختیار تابع قرار می‌گیرد.

مثال:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://  
www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html><head>
```

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head><body>
<?php
function hello($name)
{
echo ("hello , welcome! " . $name);
}
$myname="mina";
hello($myname);
?>
</body>
</html>

```

hello , welcome! mina

خروجی

در برنامه فوق، پارامتر نام به عنوان یک مقدار ثابت در نظر گرفته شده و از قبل مقداردهی شده است اما شما می‌توانید هنگام باز شدن صفحه مرورگر، با استفاده از زبان‌هایی مانند جاوااسکریپت از کاربر بخواهید اسم خود را وارد کند.

ممکن است توابع طوری پیاده‌سازی شوند که مقدار مشخصی را محاسبه کنند و در برنامه از این مقدار به عنوان پارامتر خروجی تابع در برنامه استفاده شود. ساختار کلی پیاده‌سازی چنین توابعی به صورت زیر است:

```

function نام تابع ( پارامترهای ورودی)
{
بدنه تابع
return "متغیری که باید برگشت داده شود." ;
}

```

مثال:

می‌خواهیم تابعی را پیاده‌سازی کنیم که دو عدد را به عنوان پارامتر ورودی دریافت کرده، سپس حاصل ضرب آنها را محاسبه و به عنوان پارامتر خروجی برمی‌گرداند.

```

<?php

```

```

function calc($n,$m)
{
    $p = $n*$m;
    return $p;
}
$n1=10;
$n2=43;
$n3 = calc($n1,$n2);
echo $n3;
?>

```

۴۳.

خروجی :

در کد فوق دو متغیر با نام‌های $n1$ و $n2$ تعریف شده است که به ترتیب با اعداد 10 و 43 مقداردهی شده‌اند، سپس تابع $calc()$ با دو پارامتر ورودی (43 و 10) فراخوانی می‌شود. درون این تابع، حاصل ضرب پارامترهای ورودی محاسبه شده، توسط دستور $return$ برگردانده می‌شود.

درون برنامه اصلی، مقدار برگشت داده شده توسط دستور $return$ در متغیر $$n3$ قرار داده می‌شود و در سطر آخر برنامه توسط دستور $echo$ مقدار این متغیر در خروجی چاپ می‌شود.

تمرین: تابعی برای محاسبه مجذور یک عدد دلخواه ایجاد کنید.

نکته: تابع را می‌توان در هر جایی از برنامه php تعریف کرد ولی بهتر است کلیه توابع را در ابتدای برنامه تعریف کنیم و سپس در صورت لزوم در ادامه برنامه فراخوانی نماییم.

نکته: در هنگام تعریف توابع جدید باید به این نکته توجه داشت که متغیر در زبان PHP محدود کاملاً محلی دارند و فقط درون تابعی که تعریف شده‌اند قابل دسترسی هستند. به همین دلیل اغلب برنامه‌نویسان متغیرهای مهم را که ممکن است در جاهای مختلف برنامه مورد استفاده قرار گیرند، درون اسکریپت اصلی می‌نویسند.

روش دیگری به منظور تعریف حوزه سراسری برای متغیرها، استفاده از آرایه از پیش تعریف شده [] \$GLOBALS است.

استفاده از این آرایه به این صورت است که نام متغیر مورد نظر را به عنوان اندیس آرایه درج می‌کنید.

به این ترتیب با استفاده از آرایه مخصوص [] \$GLOBALS می‌توان برنامه فوق را به شکل زیر بازنویسی کرد :

\$PI=3.14;

\$r=5;

\$circumference=\$GLOBALS["PI"]*۲*\$GLOBALS["r"];

تابع `strlen` طول رشته‌ای را که به عنوان پارامتر ورودی به آن داده می‌شود برمی‌گرداند.
تابع `strtoupper` رشته‌ای را که به عنوان پارامتر ورودی می‌گیرد به حروف الفبای بزرگ انگلیسی تبدیل می‌کند.

تابع `strtolower` تمام حروف پارامترهای ورودی را به حروف کوچک الفبای انگلیسی تبدیل می‌کند.

تابع `date()` یک پارامتر را به منظور قالب درخواستی برای برگرداندن تاریخ به عنوان ورودی دریافت می‌کند، سپس یک رشته را به عنوان خروجی برمی‌گرداند.
تابع `time()` در PHP تعداد ثانیه‌های گذشته از اولین روز سال ۱۹۷۰ میلادی تا لحظه جاری را نشان می‌دهد.

تابع `abs()` به منظور محاسبه قدر مطلق یک پارامتر به کار می‌رود.
تابع `floor()` عددی را که به عنوان پارامتر ورودی دریافت می‌کند به عدد صحیح کوچک‌تر تبدیل می‌نماید و مقدار اعشار آن را از بین می‌برد.

تابع `fmod()` دو پارامتر را دریافت و باقیمانده حاصل تقسیم پارامتر اول بر پارامتر دوم را برمی‌گرداند.

تابع `max()` دو یا چند پارامتر را به عنوان ورودی دریافت می‌کند، سپس عدد بزرگ‌تر را به عنوان خروجی برمی‌گرداند.

تابع `min()` دو یا چند پارامتر را به عنوان ورودی دریافت می‌کند، سپس عدد کوچک‌تر را به عنوان خروجی برمی‌گرداند.

از تابع `rand()` برای تولید اعداد صحیح تصادفی استفاده می‌شود.
تابع `round()` اعداد اعشاری را به نزدیک‌ترین اعداد صحیح تبدیل می‌کند.
برنامه‌نویس می‌تواند تابع جدیدی را برای عملیات موردنظر خود ایجاد نماید. برای فراخوانی از نام تابع استفاده می‌شود.

اگر در تابع محاسبه خاصی انجام شود مقدار خروجی این محاسبه توسط دستور `return` به برنامه اصلی بازگردانده می‌شود.

خودآزمایی

- ۱- برنامه‌ای بنویسید که نام و نام خانوادگی شما را به عنوان پارامتر ورودی دریافت نموده، سپس تعداد حروف آن را چاپ کند.
 - ۲- یک تابع برای محاسبه مساحت مستطیلی به طول ۵ و عرض ۴ پیاده‌سازی کنید.
 - ۳- برنامه‌ای بنویسید که در آن بدون استفاده از تابع `date()` تاریخ جاری را به صورت سال، ماه و روز نمایش دهد.
-

فصل ۵

طراحی برنامه‌های تحت وب و کار با فرم‌ها

هدف‌های رفتاری:

پس از آموزش این فصل هنرجو می‌تواند:

- ۱- عملکرد متغیرهای درون ساخت `$_POST` ، `$_GET` و `$_REQUEST` را شرح دهد.
- ۲- از `$_POST` ، `$_GET` و `$_REQUEST` در برنامه استفاده کند.
- ۳- یک فرم برای بارگذاری فایل ایجاد کند.
- ۴- فایل‌ها را پس از بارگذاری در مسیر ریشه، در مسیر دیگری ذخیره کند.

در جلد اول این کتاب با فرم‌ها و بخش‌های آن آشنا شدید. همان‌طور که می‌دانید، یکی از مهم‌ترین مسائلی که هنگام طراحی وب سایت پویا لازم است در نظر گرفته شود، این است که اطلاعات باید به وسیله فرم‌ها از کاربر دریافت شوند و سپس به منظور تحلیل و پاسخگویی مناسب برای اسکریپت PHP ارسال شوند. دسترسی به اطلاعات ارسالی، توسط متغیرهای درون ساخت، به نام‌های GET-\$، post-\$ و REQUEST-\$ انجام می‌شود که در این فصل به بررسی آنها می‌پردازیم. معمولاً ساختار فرم و اجزای آن در فایل HTML ایجاد می‌شود. با ایجاد فرم و خصوصیت‌های مهم آن مثل action و rethed در جلد اول آشنا شدیم و در ادامه موارد کاربرد آنها را در این فصل فرامی‌گیریم. بهتر است برنامه‌ها را در محیط DreamWeaver نوشته و از امکانات آن استفاده نماییم.

۱-۵- POST_\$

نمونه کد زیر در فایل HTML به نام form.htm یک فرم با دو جعبه متن و یک دکمه submit برای ارسال اطلاعات ایجاد می‌کند.

مثال:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="fname" />
<p>
Age: <input type="text" name="age" />
</p>
<p>
```

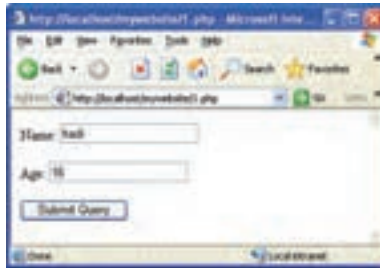
```

<input type="submit" Value = " Submit Query" />
</p>
</form>

</body>
</html>

```

این فایل نام و سن کاربر را دریافت، و توسط متد post آن را برای فایل welcome.php ارسال می‌کند. خروجی فایل form.htm به صورت زیر خواهد بود. فرض کنید کاربر، اطلاعات فرم را مانند شکل زیر پر کرده باشد.



شکل ۱-۵

در فایل welcome.php با استفاده از متغیر درون ساخته `$_POST` آرگومان‌های ارسال شده توسط کاربر در فرم برنامه form.htm، قابل دسترسی هستند.

```

Welcome <?php echo $_POST["fname"]; ?>!
<br />
You are <?php echo $_POST["age"]; ?> years old.

```

در این صورت خروجی مشابه عبارت زیر خواهد بود.
خروجی :

```

Welcome hadi!
You are 16 years old.

```

۲-۵-GET - \$

اگر متدی که فرم از آن برای ارسال اطلاعات استفاده می‌کند، get باشد، متغیر `$_GET`

می‌تواند عملکردی مشابه متغیر `$_POST` در مثال فوق داشته باشد و برای دسترسی به مقادیر ارسال شده توسط فرم به کار برده شود.

فرض کنید در فایل `form.htm` از متد `get` برای ارسال اطلاعات استفاده شده باشد، در این صورت فایل `welcome.php` به صورت زیر به عناصر فرم دسترسی خواهد داشت :

```
Welcome <?php echo $_GET["fname"]; ?>!  
<br />  
You are <?php echo $_GET["age"]; ?> years old!
```

۳-۵-\$_REQUEST

PHP یک متغیر درون ساخته مهم دارد که می‌تواند صرف نظر از متد استفاده شده برای ارسال اطلاعات توسط فرم به مقادیر `$_GET` و `$_POST` و همچنین مقدار `$_COOKIE` دسترسی داشته باشد.

این متغیر می‌تواند با هر دو متد ارسال `get` و `post` استفاده شود.

اگر فایل `welcome.php` را در مثال قبل، با استفاده از `$_REQUEST` بنویسیم، کد فایل عبارت است از :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://  
www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html><head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>  
</head><body>  
Welcome <?php echo $_REQUEST["fname"]; ?>!  
<br />  
You are <?php echo $_REQUEST["age"]; ?> years old.  
</body></html>
```

۴-۵- ایجاد فرم بارگذاری فایل

برای بارگذاری یک فایل در اینترنت، باید آن را از کاربر دریافت نمود. دریافت فایل از کاربر و ارسال آن از طریق فرم انجام می‌شود.

به منظور انتخاب فایل برای بارگذاری، باید نوع عنصر input را از نوع فایل (FileField)

قرار دهید :

```
<input type="file" name="file" id="file" enctype="form-data/multipart" />
```

در این صورت دکمه Browse نیز روی فرم ظاهر می‌شود که امکان انتخاب فایل از طریق کادر

محواره Choose file را فراهم می‌کند.

هنگام طراحی فرم بارگذاری فایل، باید در پنجره properties، خصوصیت enctype را از

نوع multipart/form-data قرار دهید، این نوع کدگذاری، برای ارسال اطلاعات باینری مانند

محتویات فایل مناسب است.

مثال :

کد زیر یک فرم انتخاب فایل برای بارگذاری را ایجاد می‌کند :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://  
www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
```

```
</head>
```

```
<body>
```

```
<form action="upload_file.php" method="post" enctype="multipart/form-data">
```

```
<label for="file">Filename: </label>
```

```
<input type="file" name="file" id="file" />
```

```
<br />
```

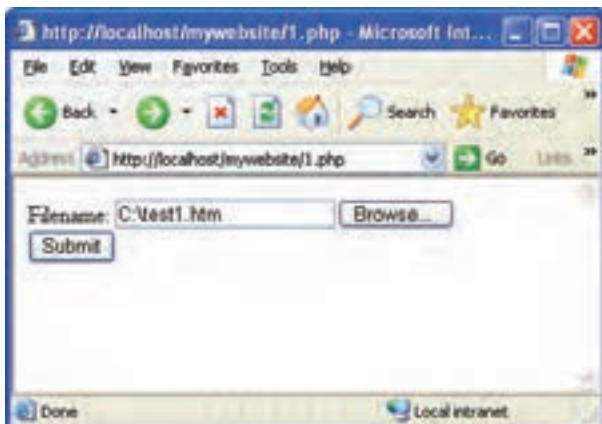
```
<input type="submit" name="submit" value="Submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

با اجرای کد فوق مرورگر مشابه شکل ۲-۵ یک فایل را از کاربر دریافت می‌کند.



شکل ۲-۵

در این کد، فایل دریافت شده، برای اسکریپت `upload_file.php` ارسال می‌شود. اطلاعات فایل بارگذاری شده، از طریق متغیر از پیش تعریف شده `$_FILES` به اسکریپت `upload_file.php` ارسال می‌شود که به شکل‌های زیر می‌توان آن را به کار برد.

- `$_FILES["file"]["error"]`: نتیجه خطای احتمالی، هنگام بارگذاری فایل را مشخص می‌کند. اگر هنگام بارگذاری فایل هیچ خطایی رخ نداده باشد، این تابع مقدار صفر را بر می‌گرداند.
- `$_FILES["file"]["name"]`: نام فایل بارگذاری شده را مشخص می‌کند.
- `$_FILES["file"]["type"]`: نوع و پسوند فایل بارگذاری شده را مشخص می‌کند. به عنوان مثال برای فایل‌های تصویری می‌تواند `image/gif`، `image/tiff`، `image/jpg`، `image/jpeg`... باشد.
- `$_FILES["file"]["size"]`: اندازه فایل بارگذاری شده را برحسب بایت مشخص می‌کند.
- `$_FILES["file"]["tmp_name"]`: مسیر موقتی فایل‌های ذخیره شده مربوط به سرویس دهنده را مشخص می‌کند. تا زمانی که فایل‌ها توسط سرویس دهنده WAMP اجرا می‌شوند، این فایل‌ها در مسیر نصب سرویس دهنده WAMP ذخیره می‌شوند.

به منظور پیاده‌سازی اسکریپت `upload_file.php` می‌توانید کد زیر را تایپ کنید و آن را با نام `upload_file.php` ذخیره نمایید.

```
<?php
if($_FILES["file"]["error"] > 0)
{
    echo "Error: " . $_FILES["file"]["error"] . "<br />";
}
```

```

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

```

اگر به طور مثال اسم فایل بارگذاری شده test1.htm باشد، نتیجه اجرای اسکریپت
upload_file.php به صورت زیر خواهد بود :

```

Upload: test1.htm
Type: text/html
Size: 0.14609375 Kb
Stored in: C:\wamp\tmp\php19.tmp

```

نکته : فراهم کردن امکان بارگذاری فایل توسط کاربر در وب سایت به دلیل احتمال وجود تهدیدات امنیتی می تواند خطرات جدی در پی داشته باشد، بنابراین فقط در صورت ضرورت از این قابلیت در صفحات وب استفاده کنید.

نکته : دقت کنید که آرگومان اول \$FILES مقدار خصوصیت name عنصر input از نوع فایل (File Field) است.

مثال فوق شکل بسیار ساده از بارگذاری فایل است، شما به عنوان طراح وب سایت می توانید، محدودیت هایی را برای فایل انتخاب شده از طرف کاربر تعیین کنید.

به طور مثال اگر در صفحه وب از کاربر خواسته باشید عکس خودش را بارگذاری کند، می توانید از او بخواهید تصویر ارسالی اش را از نوع gif، .bmp. و یا .jpg با اندازه کمتر در ۲۰ کیلوبایت (حدوداً ۲۰۰۰۰ بایت) را انتخاب کند. سپس در کد اسکریپت upload_file.php نوع و اندازه آن را بررسی کنید و در صورت تناقض به فایل اجازه بارگذاری بر روی سرور دهنده را ندهید.

در این صورت می توانید مشابه کد زیر را برای اسکریپت upload_file.php بنویسید :

```
<?php
```



```

if(((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpg")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/bmp")))
&& ($_FILES["file"]["size"] <= 20000))
{
if($_FILES["file"]["error"] > 0)
{
echo "Error ";
}
else
{
echo "name" . $_FILES["file"]["name"] . "<br />";
echo "Size" . $_FILES["file"]["size"] . "<br />";
echo "Type" . $_FILES["file"]["type"] . "<br />";
echo "Stored in" . $_FILES["file"]["tmp_name"];
}
}
else
{
echo "invalid type or size";
}
?>

```

تمرین: در یک صفحه وب از کاربر بخواهید رزومه کاری اش را در قالب فایل متنی با پسوند .doc یا .rtf و اندازه حداکثر ۱۰۰ کیلوبایت، ارسال کند و اگر فایل انتخاب شده کاربر این ویژگی‌ها را نداشت، پیغام مناسبی نمایش داده شود.

۵-۵- ذخیره فایل بارگذاری شده

مسیر ریشه یک مسیر موقتی برای مرورگر است و فایل‌های کپی شده در این مسیر، پس از

خاتمه اجرای فایل و بسته شدن مرورگر حذف می‌شوند. به همین دلیل اگر فایل بارگذاری شده از طرف سرویس گیرنده، را بخواهید در سرویس دهنده استفاده کنید، باید عمل انتقال فایل و ذخیره آن در مسیری غیر از مسیر ریشه سرویس دهنده انجام شود.

برای انتقال فایل‌ها از ریشه، تابع `move_uploaded_file` مورد استفاده قرار می‌گیرد. شکل کلی تابع:

`move_uploaded_file` (آدرس جدید و آدرس مسیر موقتی)

برای این کار می‌توانید کد زیر را به انتهای اسکریپت `upload_file.php` اضافه کنید:

```
move_uploaded_file($_FILES["file"]["tmp_name"],
"upload/" . $_FILES["file"]["name"]);
echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
```

هنگام کپی فایل بر روی سرویس گیرنده ممکن است فایلی هم نام و هم نوع با فایل کاربر وجود داشته باشد، در نتیجه کپی فایل کاربر باعث از بین رفتن فایل هم نامی که از قبل وجود داشته، می‌شود، برای پیشگیری از این مشکل می‌توانید از تابع `file_exists` به منظور بررسی تکراری نبودن فایل استفاده کنید. آرگومان ورودی این تابع مسیر کامل و نام فایل مورد نظر است و خروجی آن `boolean` می‌باشد. کاربرد آن را در مثال بعد خواهید دید.

توصیه: بهتر است قبل از انجام عمل انتقال و نوشتن اسکریپت مربوط به آن، در مسیر ذخیره وبسایت خود، پوشه‌ای را برای ذخیره فایل‌های بارگذاری شده، ایجاد کنید. (در مثال فوق پوشه `upload`)

مثال:

کد زیر در صورت عدم وجود خطا، مشخصات فایل ارسال شده توسط کاربر را در خروجی نشان می‌دهد. سپس اگر فایل تکراری بود، پیغام `already exists` به همراه نام فایل ظاهر می‌شود، در غیر این صورت عمل انتقال آن به پوشه `upload` که قبلاً آن را در مسیر ذخیره فایل‌های وبسایت، ایجاد کرده‌ایم، انجام می‌شود.

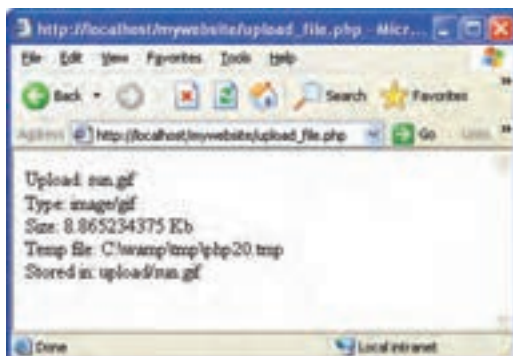
```
<?php
if(($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpg")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/bmp"))
&&($_FILES["file"]["size"] <= 20000)
```

```

{
if($_FILES["file"]["error"]> 0)
{
echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
}
else
{
echo "Upload: " . $_FILES["file"]["name"] . "<br />";
echo "Type: " . $_FILES["file"]["type"] . "<br />";
echo "Size: " . ($_FILES["file"]["size"] / 1024) . "Kb<br />";
echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";
if (file_exists("upload/" . $_FILES["file"]["name"]))
{
echo $_FILES["file"]["name"] . " already exists.";
}
else
{
move_uploaded_file($_FILES["file"]["tmp_name"],
"upload/" . $_FILES["file"]["name"]);
echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
}
}
}
else
{
echo "Invalid file";
}
?>

```

شکل زیر نتیجه اجرای کد فوق را هنگام بارگذاری فایل تصویری gif . sun نشان می دهد.



شکل ۳-۵

مطالعه آزاد

نامه‌های الکترونیکی

در یک وب سایت برای ارتباط دو طرفه و تعاملی با کاربر بهتر است امکان ارسال نامه الکترونیکی به طور مستقیم فراهم شود.

در زبان اسکریپت نویسی PHP این امکان توسط تابع `mail()` فراهم شده است. شکل کلی این تابع به صورت زیر است:

`mail(to,subject,message,headers,parameters);`

پارامترهای این تابع در جدول زیر توضیح داده شده اند.

عملکرد	پارامتر
توسط این پارامتر آدرس پست الکترونیکی دریافت کننده نامه تعیین می شود. این پارامتر می تواند آدرس پست الکترونیکی چند نفر را دریافت کند.	to
این پارامتر عنوان نامه را تعیین می کند.	subject
متن اصلی نامه را معین می کند و می تواند حداکثر تا ۷۰ کاراکتر را دریافت کند	message
این پارامتر اختیاری است و بخش های آدرس پست الکترونیکی فرستنده نامه، Cc و Bcc را به نامه الکترونیکی اضافه می کند.	headers
این پارامتر نیز اختیاری است و چنانچه بخواهید غیر از پارامترهای فوق، پارامتر دیگری به نامه الکترونیکی اضافه کنید، می توانید در این بخش درج نمایید.	parameters

نکته: برنامه ارسال نامه الکترونیکی بر روی سیستم به صورت محلی اجرا نمی‌شود و این کدها یا باید روی سرور دهنده وب واقعی اجرا شوند و یا اینکه یکی از نرم‌افزارهای سرور دهنده پست الکترونیکی مانند kerio mail server, MDaemon mail server و ... را روی سیستم فرد نصب و راه‌اندازی نمایید.

نکته: در اغلب سرورهای PHP امکان ارسال نامه الکترونیکی به طور رایگان امکان پذیر است اما برای اطمینان بیشتر قبل از اینکه این تابع را در برنامه استفاده کنید، از نصب mail server روی سرور دهنده خود مطمئن شوید. مثال زیر یک نامه الکترونیکی ساده را در صفحه وب ارسال می‌کند.

مثال:

```
<?php
$to = "someone@example.com";
$subject = "Test mail";
$message = "Hello! This is a simple email message.";
$from = "someoneelse@example.com";
$headers = "From: $from";
mail($to,$subject,$message,$headers);
echo "Mail Sent.";
?>
```

در یک صفحه وب باید فرم مخصوصی به منظور دریافت پارامترهای نامه الکترونیکی و سپس ارسال آن طراحی شود. این فرم و عملکرد آن در مثال بعد بررسی شده است.

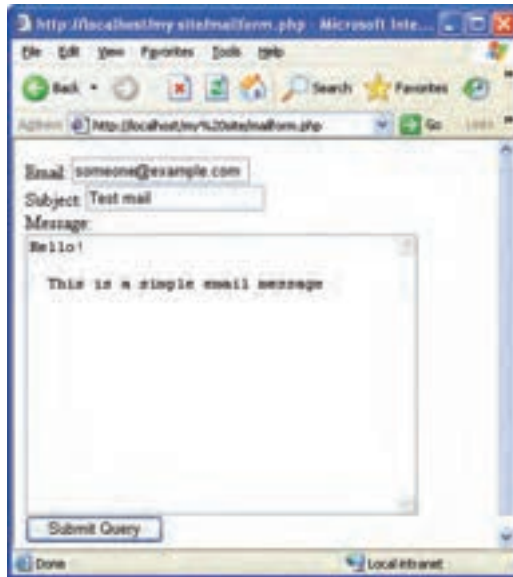
مثال:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
```

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
<?php
if (isset($_REQUEST["email"]))
//if "email" is filled out, send email
{
//send email
$email = $_REQUEST["email"];
$subject = $_REQUEST["subject"];
$message = $_REQUEST["message"];
mail( "someone@example.com", "Subject: $subject",
$message, "From: $email");
echo "Thank you for using our mail form";
}
else
//if "email" is not filled out, display the form
{
echo "<form method='post' action='mailform.php'>
Email: <input name='email' type='text' /><br />
Subject: <input name='subject' type='text' /><br />
Message: <br />
<textarea name='message' rows='15' cols='40'>
</textarea><br />
<input type='submit' />
</form>";
}
?>
</body>
</html>

```



شکل ۴-۵ - طراحی فرم ارسال نامه الکترونیکی

در PHP چند متغیر درون ساخته مهم وجود دارند که امکان دسترسی به مقادیر ارسالی در فرم را فراهم می‌کنند. این توابع عبارتند از:

`$_POST`

`$_GET`

`$_REQUEST`

اگر متدی که فرم از آن برای ارسال اطلاعات استفاده می‌کند، `get` باشد از `$_GET` برای دسترسی به مقادیر ارسالی توسط فرم استفاده می‌شود.

اگر متد مورد استفاده فرم، `post` باشد از `$_POST` استفاده می‌شود.

`$_REQUEST` می‌تواند صرف‌نظر از متد استفاده شده برای ارسال اطلاعات، به مقادیر

`$_GET` و `$_POST` و `$_COOKIE` دسترسی داشته باشد.

مهم و کاربردی فرم بارگذاری `$_FILES` نام دارد که حاوی اطلاعات زیر است:

• `$_FILES["file"]["error"]`: نتیجه خطای احتمالی، هنگام بارگذاری فایل را مشخص می‌کند.

اگر هنگام بارگذاری فایل هیچ خطایی رخ نداده باشد، این تابع مقدار صفر را بر می‌گرداند.

• `$_FILES["file"]["name"]`: نام فایل بارگذاری شده را مشخص می‌کند.

• `$_FILES["file"]["type"]`: نوع و پسوند فایل بارگذاری شده را مشخص می‌کند.

به عنوان مثال برای فایل‌های تصویری می‌تواند `image/jpeg`، `image/tiff`، `image/gif` باشد و ... باشد.

• `$_FILES["file"]["size"]`: اندازه فایل بارگذاری شده را برحسب بایت

مشخص می‌کند.

• `$_FILES["file"]["tmp_name"]`: مسیر ریشه فایل‌های ذخیره شده مربوط به

سرویس‌دهنده را مشخص می‌کند.

برای انتقال فایل از مسیر و نام موقتی به مسیر و نام جدید از تابع `move_uploaded_file`

استفاده می‌کنیم.

۱- یک فرم مطابق شکل زیر برای دریافت مشخصات کاربر طراحی کرده، اطلاعات آن را با استفاده از متد post برای صفحه‌ای با نام page2.php ارسال کنید، سپس در فایل page2.php با استفاده از تابع \$_POST اطلاعات وارد شده توسط کاربر را نمایش دهید.



شکل ۵-۵

- ۲- در سؤال اول، اطلاعات وارد شده کاربر را با استفاده از تابع \$_REQUEST نمایش دهید.
- ۳- به فرم مشخصات کاربر در شکل ۵-۵، امکان بارگذاری تصویری با قالب gif. را نیز اضافه کنید، سپس فایل بارگذاری شده را در پوشه ای با عنوان images ذخیره کنید.

فصل ۶

به کارگیری کوکی‌ها (Cookies) و جلسه‌ها (sessions)

هدف‌های رفتاری:

- ۱- مفهوم و کاربرد cookie را توضیح دهد.
- ۲- تنظیمات cookie را در برنامه انجام دهد.
- ۳- مفهوم و کاربرد session را شرح دهد.
- ۴- متغیرهای session را نام برده و از آنها در برنامه استفاده کند.
- ۵- توابع مهم برای کار با session ها را به کار ببرد.

در یک وب سایت اطلاعات مختلفی از طرف کاربران به سمت سرور ارسال می‌شود. تنها بخشی از این اطلاعات به طور مستقیم و با اطلاع کاربر، توسط سرور گیرنده ارسال می‌شوند. قسمت دیگری از آنها بدون اینکه کاربر به‌طور آگاهانه اقدام به ذخیره و ارسال آنها کرده باشد، ذخیره و ارسال می‌شوند.

اغلب این‌گونه داده‌ها توسط انواعی از امکانات زبان‌های برنامه‌نویسی به نام cookie ها و session ها ایجاد می‌شوند که در این فصل با ماهیت، کاربرد و نحوه مدیریت آنها آشنا خواهید شد.

۱-۶- cookie

cookie فایل متنی کوچکی است که از طرف سرور دهنده و توسط مرورگر روی سیستم کاربر ایجاد می‌شود. این فایل می‌تواند در موارد مختلفی به سرور دهنده وب و سرور دهنده برنامه کاربردی کمک کند. به‌عنوان مثال می‌تواند به سرور دهنده وب کمک کند تا کاربرانی را که قبلاً به سایت آنها مراجعه کرده‌اند بشناسد، یا اینکه به یادآوری رمز عبور کاربر در مراجعات بعدی کمک کند. یک کاربر می‌تواند سیستم خود را طوری تنظیم کند که سرور دهنده اجازه تولید cookie را نداشته باشد و یا اینکه کاربر را هنگام تولید cookie مطلع کند. cookie می‌تواند اطلاعات مهمی را درون خود نگهداری کند و می‌تواند توسط سرور دهنده‌گان غیر مجاز اطلاعات مهمی از هویت شما را فاش کند. به‌عنوان مثال اگر در یک سایت عضو شده باشید، چنانچه یک cookie برای آن روی هارد سیستم شما ایجاد شده باشد، یک سایت غیر مجاز می‌تواند با اهداف سوء از طریق خواندن cookie به رمز عبور شما دسترسی پیدا کند.

نکته: هر فایل cookie به تنهایی نمی‌تواند برای سیستم مشکلات امنیتی ایجاد کند اما می‌تواند در آسیب رساندن به امنیت سیستم شما دخیل باشد. به‌طور مثال هکرها و افرادی که قصد سوء استفاده از اطلاعات موجود بر روی سیستم شما را دارند، با خواندن محتویات cookie، به اطلاعات محرمانه شما دسترسی پیدا کنند.

حداکثر اندازه فایل cookie ، ۴KB (کیلوبایت) است.

یکی از ویژگی‌های فایل‌های cookie این است که پس از آنکه توسط یک مرورگر روی یک سیستم ایجاد شدند، فقط روی همان سیستم می‌توانند خوانده شوند. هر cookie یک آدرس URL منحصر به فرد دارد و سرور گیرنده از روی این آدرس تعیین می‌کند که cookie مربوط به هر کاربر کدام است.

استفاده از cookie در سرویس گیرنده‌ها و مرورگرهای مختلف محدودیت‌هایی دارد. به عنوان مثال حداکثر تعداد فایل‌های cookie مرورگر Internet explorer، ۳۰۰ فایل است. PHP قادر است عملیات ایجاد، مدیریت و حذف cookie را به خوبی انجام دهد.

۱-۱-۶- ایجاد cookie

ایجاد cookie در فایل‌ها و اسناد PHP توسط دستور setcookie انجام می‌شود. ساختار کلی این دستور به شکل زیر است:

setcookie (Name, Value, Expire, Path, Domain, Secure);

(سطح دسترسی و اطلاعات دامنه، مسیر، تاریخ انقضای cookie، مقدار cookie، نام cookie);
درج پارامترهای cookie اختیاری است و به دلخواه می‌توانید تعدادی از آنها را از لیست پارامترها حذف کنید.

- پارامتر Name، نام کوکی و value مقدار آن را مشخص می‌کند. این دو پارامتر اجباری هستند.
- در پارامترهای تابع فوق Path مسیری را در سمت سرویس دهنده تعیین می‌کند که در آن کوکی قابل دسترسی است و در حالت پیش فرض، کوکی می‌تواند توسط کل دامنه در سمت سرویس دهنده، مورد دستیابی قرار گیرد.

- Domain دامنه مجاز را تعریف می‌کند که به طور پیش فرض همان دامنه سرویس دهنده‌ای است که آن را ایجاد کرده است. اطلاعات دامنه مشخص می‌کند که کوکی توسط چه دامنه‌ای قابل دسترسی است.

- گزینه Secure سطح دسترسی به cookie را تعیین می‌کند و به صورت پیش فرض غیرفعال است. اگر این پارامتر فعال شده باشد، کوکی فقط مجاز است از طریق کانال‌های امن، مانند https منتقل شود. مقدار صفر برای غیرفعال و ۱ برای فعال کردن آن است.

- توسط تاریخ انقضا می‌توان اعتبار یا عدم اعتبار cookie را سنجید، اگر تاریخ انقضای یک cookie تمام شده باشد، مرورگر دیگر آن را برای سرویس دهنده ارسال نمی‌کند (تاریخ انقضا با قالب استاندارد GMT (گرنویچ) تعیین می‌شود). این پارامتر برحسب ثانیه نوشته می‌شود.

پارامترهای cookie توسط سرویس گیرنده تعیین می‌شود و می‌تواند علاوه بر پارامترهای ساختار فوق، پارامترهای دیگری را شامل شود، علاوه بر این ممکن است یک Cookie فقط شامل نام و یک پارامتر دیگر باشد.

در ادامه در مثال ۱ به محض اجرای صفحه در مرورگر، یک cookie با نام user برای کاربر

وبسایت ساخته می‌شود. این cookie با مقدار meisam مقداردهی شده و برای آن تاریخ انقضای یک ساعت در نظر گرفته شده است.

مثال ۱ :

```
<?php  
Setcookie ("user", "meisam", time () + 3600);  
?>
```

اگر با نسخه 6 Internet explorer یا قبل از آن کار می‌کنید برای مشاهده cookie در مسیر نصب ویندوز، Documents and Settings، نام User و سپس پوشه s cookie را باز کنید، مشاهده خواهید کرد که فایل مربوط به آن محتویاتی مشابه شکل ۱-۶ دارد.



شکل ۱-۶

برای اینکه cookie ایجاد شده را راحت‌تر پیدا کنید، بهتر است ابتدا تمام فایل‌های cookie موجود بر روی سیستم را پاک کنید.

در مثال ۲ یک cookie با زمان انقضای یک ماه ایجاد شده است.

نکته : چنانچه ۷ Internet Explorer و یا بالاتر از آن استفاده می‌کنید برای مشاهده کوکی‌ها از منوی Tools گزینه Internet Options را انتخاب نموده، در قسمت Browsing history روی دکمه Setting کلیک کرده و در کادر باز شده روی دکمه view files کلیک نمایید.

مثال ۲ :

```
<?php  
$expire=time()*60*60*24*30*;  
setcookie ("user", "meisam", $expire);  
?>
```

۲-۱-۶- دسترسی به اطلاعات cookie

Cookie های ارسال شده از طرف هر سرویس دهنده در یک آرایه به نام `$_COOKIE` ذخیره شده‌اند و دسترسی به آنها مانند سایر آرایه‌هاست.

به عنوان مثال اگر یک Cookie با نام `username` داشته باشیم می‌توانیم توسط `$_COOKIE ["user name"]` به محتوای آن دسترسی پیدا کنیم.

نکته: دستور `Set cookie` در برنامه باید قبل از هر خروجی (مثل `echo`) نوشته شود. علاوه بر این باید در ابتدای برنامه و قبل از برچسب `<html>` نوشته شود.

مثال:

می‌خواهیم یک cookie ایجاد کنیم که تعداد بازدیدهای کاربر را از سایت تعیین کند. نام cookie را `visit` می‌گذاریم.

همچنین یک متغیر به نام `$Count` تعریف می‌کنیم که با هر بار ملاقات توسط کاربر، به مقدار آن یک واحد افزوده شود.

```
<?php
if isset($_COOKIE ["visit"])
{
$count=$_COOKIE ["visit"] + 1;
setcookie ("visit",$count,time( )+3600);
echo $count. "<br/>";
}
else
{
$count=1;
setcookie ("visit",$count,time( )+3600);
echo $count. "<br/>";
}
?>
```

با اجرای فایل فوق در مرورگر مشاهده خواهید کرد که با هر بار اجرای صفحه (و حتی فشردن دکمه F5 به منظور بازنشانی مجدد صفحه) مقدار شمارنده یک واحد افزایش پیدا می‌کند.

به ساختار کد فوق دقت کنید :

متغیر `$Count` را در نظر بگیرید. در صورتی که اولین بار باشد که این برنامه اجرا می‌شود چون هنوز کوکی ایجاد نشده این متغیر با عدد یک مقداردهی می‌شود و سپس `cookie` ایجاد شده و مقدار متغیر `count` در آن ذخیره می‌شود.

حال اگر بار اول اجرای این برنامه نباشد، به مقدار قبلی `count` یک واحد افزوده شده و به همین ترتیب در هر بار ملاقات صفحه توسط کاربر، یک واحد به شمارنده افزوده می‌شود.

سؤال: تاریخ انقضای `cookie` در مثال فوق چه مدت است؟

تمرین: با استفاده از تابع `isset()` تعداد بازدیدهای کاربر از سایت را به دست آورید.

۳-۱-۶- حذف یک `cookie`

• برای حذف یک `cookie` کافی است زمان انقضای آن را در گذشته قرار دهیم. اغلب مرورگرها کوکی‌هایی را که تاریخ انقضای آنها گذشته باشد، نادیده می‌گیرند و پس از بستن پنجره مرورگر، آنها را حذف می‌کنند.

مثال:

در مثال زیر زمان انقضای یک `cookie` با نام `user`، یک ساعت قبل از زمان اجرای برنامه تنظیم شده است.

```
setcookie ("user", "", time()-3600);
```

• همچنین برای حذف `cookie` می‌توان از درج زمان انقضاء خودداری نمود. در این صورت پس از بستن پنجره فعلی مرورگر، `cookie` نیز حذف خواهد شد. در این صورت شکل کلی تابع به شکل زیر استفاده می‌شود :

```
setcookie ("نام cookie");
```

۲-۶- جلسه `session`

`session` یکی از مباحث اصلی شبکه و اینترنت است و توسط آن می‌توان اطلاعات کاربران را نگهداری کرد. تفاوت `session` با `cookie` در این است که اطلاعاتی که توسط `session` نگهداری می‌شوند همگی در سمت سرور بوده و هیچ اطلاعاتی در سمت سرویس‌گیرنده ذخیره نمی‌شود. (برخلاف کوکی).

هر کاربر که به سایتی وارد می‌شود، با اجرای اسکریپت PHP، یک جلسه ایجاد می‌شود و در صورتی که اولین بار باشد که وارد آن سایت می‌شود یک شناسه منحصر به فرد به او اختصاص داده می‌شود. این شناسه می‌تواند در تمام صفحات وب سایت، برای اهداف مختلف از قبیل شناسایی کاربر، تعداد دفعات مشاهده صفحات، زمان مشاهده و ... به کار برده شود. شناسه اختصاصی توسط دستور (session_id) تعیین می‌شود. این شناسه در اینترنت با شکستن آدرس IP و به صورت اعداد هگزا دسیمال تعیین می‌شود.

مثال زیر یک نمونه از این شناسه را نشان می‌دهد.

مثال:

88dced2a835ae4cf04f95ce2347fbcf

۱-۲-۶- ایجاد جلسه

نکته: یکی از مهم‌ترین کاربردهای جلسه زمانی است که یک کاربر پس از وارد کردن مشخصات کاربری اش، به صفحات دیگر وب سایت می‌رود، در این صورت با استفاده از جلسه، می‌توان برای دسترسی به مشخصات او، در همه جای برنامه اقدام نمود.

برای شروع جلسه دستور (session_start) به کار می‌رود. این دستور معمولاً در ابتدای صفحه و قبل از شروع سایر برچسب‌ها حتی DOCTYPE و html نوشته می‌شود.

مثال:

با ذخیره و اجرای کد زیر در هر بار اجرای صفحه توسط مرورگر، جلسه شروع می‌شود و مشاهده خواهید کرد که شناسه اختصاص داده شده در هر بار اجرا تغییر پیدا می‌کند.

```
<?php
```

```
session_start();
```

```
?>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org
```

```
TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html>
```

```
<head>
```



```

<title>start a session</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
<?php
echo"your session id is : " . session_id();
?>
</body>
</html>

```

در فایل `ini` PHP یک شناسه باعنوان `auto_start` وجود دارد. این شناسه به طور پیش فرض مقدار صفر دارد. اگر این مقدار را به یک تغییر دهید، برای هر سند PHP به طور خودکار یک جلسه ایجاد می‌شود. اما اگر مقدار آن صفر باشد باید با استفاده از دستورات برنامه نویسی برای ایجاد جلسه اقدام شود.

۲-۲-۶- متغیرهای جلسه

هر جلسه می‌تواند تعداد زیادی متغیر داشته باشد و توسط آنها می‌توان اطلاعات کاربران را در صفحات مختلف مدیریت کرد.

این متغیرها در یک آرایه فوق سراسری به نام `$_SESSION` نگهداری می‌شوند. تعریف متغیرهای جلسه به صورت زیر است:

مقدار متغیر = [نام متغیر] `$_SESSION`;

مثال: `$_SESSION['number'] = ۱۲۳۹;`

مثال: `$_SESSION['name'] = "zahra";`

پس از تعریف متغیرهای جلسه در یک صفحه، در صفحات دیگر می‌توان به آنها دسترسی پیدا کرد. دسترسی به متغیرهای جلسه با همان نامی صورت می‌گیرد که هنگام تعریف متغیرها به کار رفته بود.

مثال:

دستورات زیر را در یک فایل به نام `set-session.php` نوشته و ذخیره کنید.

```

<?php
session_start();

```

```
$_SESSION['username']="Ali";
echo ($_SESSION['username']);
?>
```

همان‌طور که اشاره شد، در صفحات و فایل‌های دیگر نیز می‌توان به محتوای جلسه دسترسی پیدا کرد، به‌منظور دسترسی به محتوای جلسه مثال قبل، کدهای زیر را در فایل دیگری به نام Access_session.php نوشته و آن را اجرا کنید :

```
<?php
session_start();
echo ($_SESSION['username']);
?>
```

پس از اجرای کد فوق، مشاهده خواهید کرد نام 'Ali' که در فایل قبل به عنوان محتوای جلسه مقداردهی شده بود، نمایش پیدا می‌کند.

مثال:

می‌خواهیم با استفاده از متغیرهای جلسه، تعداد دفعات مشاهده صفحه وب را توسط هر کاربر محاسبه کنیم :

```
<?php
session_start();
if(isset($_SESSION['views']))
$_SESSION['views']=$_SESSION['views']+1;
else
$_SESSION['views']=1;
echo "Views=". $_SESSION['views']. "<br/>";
?>
```

در مثال زیر شکل به کارگیری متغیرهای جلسه به‌طور کاربردی بررسی شده است.

مثال:

می‌خواهیم در فایلی به نام session1.php، یک جلسه ایجاد کنیم، سپس در فایل دیگری به نام session2.php آن را بخوانیم.

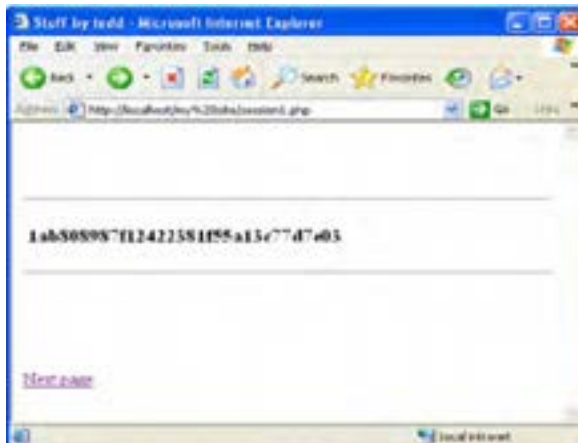
کد زیر را نوشته، سپس در فایل به نام session1.php ذخیره کنید :

```
<?php
session_start();
?>
<h3><?php echo session_id(); ?> </h3> <hr/>
<a href="session2.php">Next page</a>
```

کد زیر را نیز در فایل به نام session2.php ذخیره کنید :

```
<?php
session_start();
isset($_SESSION['count'])? $_SESSION['count']++ : $_SESSION['count']=1;
?>
<?php echo($_SESSION['count']); ?> times in this session
```

با اجرای فایل session1.php در مرورگر شناسه جلسه را مشاهده خواهید کرد.



شکل ۲-۶- فایل session1.php

چنانچه روی متن Next Page کلیک کنید، session2.php اجرا خواهد شد. حال با هر بار رفت و برگشت بین این دو صفحه و یا refresh کردن صفحه session2.php مشاهده خواهید کرد که شمارنده تعداد بازدیدهای صفحه، یک واحد افزایش می‌یابد.

۳-۲-۶- حذف جلسه و متغیرهای آن

حذف جلسه از طریق تابع `session_destroy()` صورت می‌گیرد. اگر این تابع در یک جلسه استفاده شود، تمام متغیرها و شناسه‌های جلسه نیز حذف می‌شوند.

```
<?php
session_destroy();
?>
```

چنانچه قصد داشته باشید بدون از بین بردن جلسه، تنها یک یا چند متغیر آن را از بین ببرید، می‌توانید از تابع `unset()` استفاده کنید.

مثال:

```
<?php
unset($_SESSION['views']);
?>
```

۴-۲-۶- نگهداری تنظیمات شخصی کاربر

همان‌طور که گفته شد جلسه‌ها می‌توانند متغیرهای زیادی را در صفحه ایجاد کنند، این متغیرها می‌توانند حتی شامل اطلاعات محرمانه کاربر باشند و به دلیل این که اطلاعات جلسه به صورت یک آرایه فوق سراسری ذخیره می‌شود، می‌تواند مورد استفاده هکرها قرار گیرد. به همین دلیل باید به نوعی به دنبال مخفی کردن این اطلاعات بود.

رمزگذاری اطلاعات جلسه روشی است که توسط PHP قابل پشتیبانی است، این عمل توسط دستور `session_encode()` صورت می‌گیرد، درمقابل اگر اطلاعات `session` بخواید مورد استفاده فایل PHP قرار گیرد باید از حالت رمز خارج شود و این عمل توسط دستور `session_decode()` صورت می‌گیرد.

مثال:

```
<?php
session_start();
$_SESSION["login_ok"] = true;
$_SESSION["name"] = "sina";
$_SESSION["age"] = 34;
```

```
$_SESSION["data"]= session_encode();  
echo $_SESSION["data"];  
?>
```

سپس در صفحات دیگر برای رمز گشایی و مشاهده متغیرهای جلسه می‌توان دستورات زیر را

به کار برد :

```
session_decode($_SESSION['data']);  
echo "Decoded session data for session"  
. session_id() . " : " . $_SESSION['login_ok'] . "&" . $_SESSION["name"];
```

Cookie فایل کوچک متنی است که از طرف سرویس دهنده و توسط مرورگر روی سیستم کاربر ایجاد و ذخیره می‌شود.

هر فایل cookie به تنهایی نمی‌تواند برای سیستم مشکلات امنیتی ایجاد کند اما می‌تواند در آسیب رساندن به امنیت سیستم شما دخیل باشد به‌طور مثال هکرها و افرادی که قصد سوءاستفاده از اطلاعات موجود بر روی سیستم شما را دارند، با خواندن محتویات cookie، به اطلاعات محرمانه شما دسترسی پیدا می‌کنند. حداکثر اندازه فایل cookie، ۴KB (کیلوبایت) است.

ایجاد cookie در PHP توسط دستور `set cookie` انجام می‌شود. session یکی از مباحث اصلی شبکه و اینترنت است و توسط آن می‌توان اطلاعات کاربران را نگهداری کرد. هر کاربر که به سایتی وارد می‌شود، با اجرای اسکریپت PHP یک جلسه ایجاد می‌شود و در صورتی که اولین بار باشد که وارد آن سایت می‌شود، یک شناسه منحصر به فرد به او اختصاص داده می‌شود. این شناسه می‌تواند در تمام صفحات وب سایت برای اهداف مختلف از قبیل شناسایی کاربر، تعداد دفعات مشاهده صفحات، زمان مشاهده و ... به کار برده شود.

حذف جلسه از طریق کد `session_destroy()` صورت می‌گیرد.

- ۱- مفهوم و کاربرد cookie را توضیح دهید.
 - ۲- مفهوم و کاربرد session را توضیح دهید.
 - ۳- در یک صفحه وب، فرمی برای دریافت مشخصات کاربر طراحی کنید و آن را برای صفحه‌ای با نام session_c.php ارسال کنید، سپس در صفحه session_c.php یک cookie برای ذخیره شناسه کاربر طراحی کنید که به مدت یک ماه انقضا داشته باشد.
 - ۴- در سؤال ۳، شناسه کاربر را در یک جلسه ذخیره کنید.
-

فصل ۷

MySQL

هدف‌های رفتاری:

پس از آموزش این فصل هنرجو می‌تواند:

- ۱- از نرم افزار WAMP برای طراحی و مدیریت پایگاه داده، استفاده کند.
- ۲- با سیستم مدیریت پایگاه داده MySQL در برنامه PHP، ارتباط برقرار کند.
- ۳- در برنامه PHP، یک پایگاه داده جدید بسازد.
- ۴- اجزای مختلف پایگاه داده را ایجاد کند.
- ۵- اطلاعات مورد نظر را در جدول درج کند.
- ۶- اطلاعات درج شده را ویرایش و به روزرسانی کند.
- ۷- اطلاعات درج شده را نمایش دهد.
- ۸- رکورد یا رکوردهای مورد نظر را از جدول پایگاه داده، حذف کند.

MySQL یک سیستم مدیریت پایگاه داده (بانک اطلاعاتی) است و مانند سایر پایگاه‌های داده، اطلاعات را به صورت جدول ذخیره می‌کند.

همانطور که می‌دانید، جدول مجموعه‌ای از داده‌های یک موجودیت است و از سطرها و ستون‌های مختلفی تشکیل شده است.

طراحی پایگاه داده، به دلیل اینکه اطلاعات را به صورت گروه‌بندی شده ذخیره می‌کند، بسیار مفید است و کار بر روی داده‌ها را راحت می‌کند.

به طور مثال، پایگاه داده یک مدرسه می‌تواند دارای جداول زیر باشد:

- دانش‌آموزان (Students)

- دبیران (Teachers)

- دروس (کتاب‌های درسی) (Books)

- کلاس‌ها (Classes)

شکل زیر یک مثال برای جدول دانش‌آموزان است که دارای چهار فیلد نام، نام خانوادگی، کد دانش‌آموز و شماره شناسنامه است.

جدول ۷-۱

First Name	Last Name	Student code	IDno
Ali	Alavi	124	13452
Hasan	Hasani	125	123456
Ahmad	Ahmadi	126	44562

در جدول ۷-۱ سه رکورد مربوط به سه دانش‌آموز درج شده است.

نکته: پرس و جو (Query) نیز یک پرسش یا درخواست از پایگاه داده است که برای برگرداندن اطلاعات مشخص از مجموعه‌ای از رکوردها استفاده می‌شود.

به طور مثال برای برگرداندن نام خانوادگی دانش‌آموزان در جدول ۷-۱ می‌توان پرس و جوی زیر را ایجاد نمود:

```
SELECT LastName FROM Students
```

با اجرای پرس و جوی فوق، نتیجه به صورت زیر برگردانده می شود :

جدول ۲-۷

LastName
Alavi
Hasani
Ahmadi

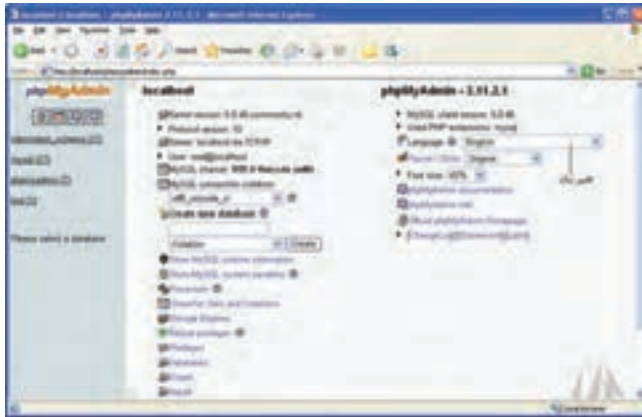
۷-۱ - ایجاد پایگاه داده

ایجاد پایگاه داده در سرویس دهنده WAMP بسیار ساده است و نیاز به دانش بالایی در زمینه کار با زبان SQL و بانک اطلاعاتی ندارد. اما اگر از رابط متنی و با اجرای پرس و جوها به طور مستقیم در MySQL استفاده می کنید، بهتر است بر تمام دستورات ایجاد و مدیریت پایگاه داده و جداول تسلط کافی داشته باشید. برای ایجاد بانک اطلاعاتی در سرویس دهنده WAMP روی آیکن این برنامه بر روی ناحیه تذکر ویندوز کلیک کرده، سپس گزینه phpmyadmin را انتخاب کنید (شکل ۷-۱).



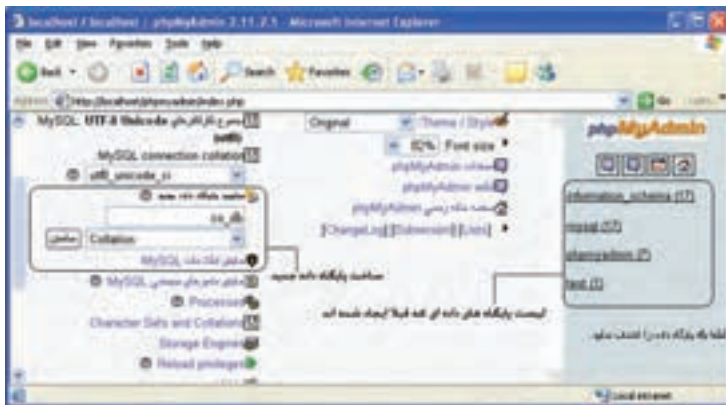
شکل ۷-۱

پنجره مربوط به ایجاد و مدیریت پایگاه داده به صورت شکل ۷-۲ باز می‌شود.



شکل ۷-۲

یکی از امکانات phpMyAdmin قابلیت پشتیبانی از زبان فارسی است و چنانچه در کادر باز شو Language که در شکل ۷-۲ با عنوان تغییر زبان مشخص شده است، گزینه Persian را انتخاب کنید، عبارات اصلی به زبان فارسی نشان داده می‌شوند.



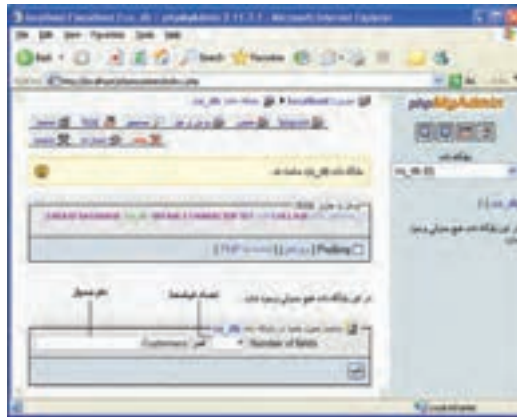
شکل ۷-۳

در شکل ۷-۳ بخش مربوط به ساخت پایگاه داده جدید مشخص شده است. در کادر متن «ساخت پایگاه داده جدید»، نام پایگاه داده‌ای که می‌خواهید ایجاد کنید را وارد نمایید، اگر قصد داشته باشید اطلاعات جداول را به صورت فارسی وارد کنید، باید در کادر باز شو Collation، کنار کادر متن نام پایگاه داده، گزینه `Utf8_general_ci` یا `Utf8_persian_ci` را انتخاب کنید.

سپس به منظور ایجاد پایگاه داده، روی دکمه "ساختن" کلیک کنید.

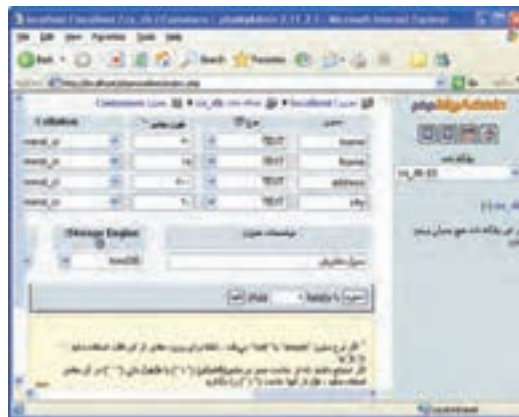
۷-۲- ساخت جداول پایگاه داده

پس از ساخت پایگاه داده، باید جدول‌های مورد نیاز را نیز طراحی کنید. برای ساخت جداول درون هر پایگاه داده مطابق شکل ۷-۴ و در بخش «ساختن جدول جدید» نام جدول و تعداد فیلدهای آن را در کادر متن مربوطه وارد کنید.



شکل ۷-۴

سپس برای ساخته شدن جدول روی دکمه تأیید کلیک کنید. پنجره phpMyAdmin نام فیلدها و نوع هر کدام به همراه سایر ویژگی‌ها را مطابق شکل ۷-۵ از شما دریافت خواهد کرد.



شکل ۷-۵

اگر بخش Collation را برای پایگاه داده تنظیم کرده باشید، قابلیت‌های زبان موردنظر شما در سراسر بانک در دسترس است و نیاز نیست در شکل ۷-۵ برای هر فیلد این قسمت را تنظیم کنید.

هنگام تعریف هر فیلد باید نوع داده آن مشخص شود. مهمترین انواع داده MySQL عبارتند از:

- int: برای تعریف اعداد صحیح بدون علامت به کار می‌رود.

- varchar (size): برای تعریف رشته‌ای با طول مشخص به کار می‌رود. حتماً باید مقدار Size را بنویسید. مثلاً برای فیلد Firstname طول ۱۵ حرف را مشخص کنید.

- text: متغیرهای رشته‌ای با حداکثر تعداد ۶۵۵۳۵ کاراکتر را می‌توان توسط این نوع داده تعریف کرد و لازم نیست طول متغیر مربوط به آن مشخص شود. پس از انجام تنظیمات، روی دکمه «ذخیره» کلیک کنید.

تمرین: بانک اطلاعاتی یک مدرسه که در ابتدای همین فصل جداول آن نام برده شد را بسازید. پس از ساختن بانک اطلاعاتی، علاوه بر جدول Students فیلدهای سایر جداول را نیز طراحی کرده، سپس تمام آنها را در phpMyAdmin ایجاد کنید.

۷-۳- ایجاد ارتباط با سیستم مدیریت پایگاه داده MySQL

همان‌طور که پیش از این اشاره شد، مهمترین ویژگی وب سایت‌های پویا، توانایی ارتباط آنها با پایگاه داده و کار با آن است.

با روش ساخت پایگاه داده و جداول آن در WAMP آشنا شدید. برخی از برنامه نویسان و طراحان وب سایت‌های پویا، تمام اعمال مدیریتی از جمله ساخت پایگاه داده و جداول مورد نیاز را توسط اسکریپت‌های برنامه انجام می‌دهند. به همین دلیل باید با تمام دستورات مربوط به این اعمال آشنا باشید.

برای ایجاد و مدیریت اجزای بانک اطلاعاتی در برنامه، از دستورات SQL استفاده می‌شود که در درس بانک اطلاعاتی با آنها به طور مفصل آشنا شده‌اید، به همین دلیل در این فصل به معرفی مختصر دستورات مورد نیاز می‌پردازیم. سایر دستورات SQL نیز در MySQL قابل دسترسی هستند و در صورت نیاز می‌توانید از آنها استفاده کنید.

برای ارتباط با MySQL در صفحات وب، باید قبل از هر چیز، اتصال با سرویس دهنده پایگاه داده ایجاد شود.

اتصال به پایگاه داده در PHP توسط تابع mysql_connect() انجام می‌شود.

ساختار کلی این تابع به صورت زیر است :

```
mysql_connect(servername,username,password);
```

تابع `mysql_connect` سه پارامتر دارد که همراه با توضیحات در جدول ۷-۳ بیان شده‌اند.

جدول ۷-۳

پارامتر	توضیحات
servername	این پارامتر اختیاری است و نام سرویس دهنده را مشخص می‌کند، چنانچه این پارامتر مقدار دهی نشود، به طور پیش فرض مقدار «localhost : ۳۳۰۶» برای آن تعیین می‌شود.
username	این پارامتر نیز اختیاری است و شناسه کاربری سرویس دهنده را تعیین می‌کند. در سرویس دهنده WAMP، این شناسه، همان کلمه کاربری است که برای کار با WAMP، تنظیم شده است. اگر شناسه کاربری را تنظیم نکرده باشید، مقدار آن را root در نظر بگیرید و اگر این پارامتر را مقداردهی نکنید، کاربری که پایگاه داده را ایجاد کرده است به طور پیش فرض تعیین می‌شود.
password	مقدار این پارامتر نیز اختیاری است و رمز ورود شناسه کاربری را دریافت می‌کند، چنانچه مقداردهی نشود، یک رشته تهی را به طور پیش فرض در نظر می‌گیرد.

تابع `mysql_close()` نیز برای قطع ارتباط با پایگاه داده به کار می‌رود.

بهرتر است پس از خاتمه کار با پایگاه داده، توسط این تابع در برنامه اتصال به آن قطع شود.

اگر بخواهیم در PHP یک پیغام نمایش داده شود، سپس از اسکریپت خارج شود، از تابع `die()`

استفاده می‌شود. این تابع می‌تواند برای اطمینان از بروز یا عدم بروز خطا هنگام اجرای کدها استفاده شود.

مثال :

در مثال زیر اتصال به پایگاه داده در متغیر `$con` ذخیره می‌شود که یک مقدار از نوع `boolean`

است سپس در برنامه از این متغیر استفاده می‌شود. تابع `die` یک پیغام را نمایش می‌دهد و در صورتی

اجرا می‌شود که اتصال به پایگاه داده انجام نشده باشد.

پس از خاتمه عملیات نیز ارتباط با پایگاه داده قطع می‌شود.

```
<?php
```

```
$con = mysql_connect("localhost","root","");
```

```

if (!$con)
{
die("Could not connect: " . mysql_error());
}
// some code
mysql_close($con);
?>

```

در مثال فوق تابع `mysql_error()` خطایی که منجر به عدم اتصال به پایگاه داده شده است را برمی گرداند.

نکته: شکل کلی تابع `mysql_close()` به صورت زیر است:

```
mysql_close(متغیر اتصال);
```

۷-۴- ایجاد یک پایگاه داده و اجزای آن

همان طور که می دانید در یک بانک اطلاعاتی یک یا چند جدول وجود دارد و هر جدول نیز از رکوردهای متعددی تشکیل شده است. هر رکورد هم شامل چند فیلد مرتبط با هم و مربوط به یک موجودیت خاص است.

در این بخش با نحوه ایجاد پایگاه داده و هر یک از اجزای آن توسط PHP آشنا خواهید شد.

۷-۴-۱- ایجاد یک پایگاه داده

برای ایجاد پایگاه داده از دستور `CREATE DATABASE` استفاده می شود.

ساختار کلی این دستور به شکل زیر است:

```
CREATE DATABASE نام پایگاه داده
```

نکته: در PHP برای اجرای دستورات مربوط به SQL که بر روی پایگاه داده صورت می گیرد،

تابع `mysql_query()` به کار می رود. این تابع یک `query` برای ایجاد پایگاه داده به MySQL که اتصال به آن از قبل برقرار شده است، ارسال می کند.

شکل کلی تابع `mysql_query()` به صورت زیر است:

```
mysql_query(متغیر اتصال, پرس و جوی مورد نظر);
```

مثال زیر پس از اتصال به MySQL، یک پایگاه داده با نام "my_db1" می سازد:

```
<?php
```

```
$con = mysql_connect("localhost","root","");
```

```

if (!$con)
{
die("Could not connect: " . mysql_error());
}
$sql="CREATE DATABASE my_db1";
if (mysql_query($sql,$con))
{echo "Database created"; }
else
{
echo "Error creating database: " . mysql_error();
}
mysql_close($con);
?>

```

نکته: در مثال فوق، اتصال به سیستم مدیریت پایگاه داده‌ای برقرار شده است که کلمه کاربری آن root و رمز ورودی برای آن تعیین نشده است. چنانچه روی سیستم شما تنظیماتی غیر از این انجام شده، این دو پارامتر را با توجه به آن تنظیمات وارد کنید.

۲-۴-۷- ایجاد یک جدول در پایگاه داده

ایجاد جدول در پایگاه داده‌ای که از قبل ایجاد شده است، توسط دستور CREATE TABLE انجام می‌شود.

این دستور نیز توسط تابع mysql_query () اجرا می‌شود.

ساختار کلی دستور CREATE TABLE به شکل زیر است:

```

CREATE TABLE نام جدول
(
    نوع داده آن نام فیلد(ستون) اول
    ,نوع داده آن نام فیلد(ستون) دوم
    ,نوع داده آن نام فیلد(ستون) سوم
    ....
)

```

نکته: از آنجا که در یک سیستم مدیریت پایگاه داده، ممکن است بانک‌های اطلاعاتی متعددی ایجاد شده باشند، لازم است پیش از ایجاد جدول، بانک اطلاعاتی موردنظرمان را مشخص کنیم.

یعنی همان بانکی که قرار است جدول جدید در آنجا ایجاد شود را انتخاب نمایم. به این منظور از تابع `mysql_Select_db`

استفاده می‌کنیم. شکل کلی آن به صورت زیر است:

```
mysql_select_db(متغیر اتصال, نام بانک);
```

مثال:

در مثال زیر یک جدول با نام "Persons" که دارای سه فیلد با عناوین "FirstName" "LastName" و "Age" است، ساخته می‌شود.

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
// Create database
if (mysql_query("CREATE DATABASE my_db", $con))
{
echo "Database created";
}
else
{
echo "Error creating database: " . mysql_error();
}
// select DataBase
mysql_select_db("my_db", $con);
// Create table
$sql = "CREATE TABLE Persons
(
FirstName varchar(15),
LastName varchar(15),
```

```

Age int
);
// Execute query
mysql_query($sql,$con);
mysql_close($con);
?>

```

۳-۴-۷- فیلد کلیدی و خاصیت افزایش خودکار

هر جدول می‌تواند یک فیلد کلیدی داشته باشد که برای اطمینان از تکراری نبودن رکورد در جدول استفاده می‌شود. عبارت PRIMARY KEY برای تعریف فیلد کلیدی هنگام ساخت جدول به کار می‌رود.

علاوه بر این فیلد کلیدی نمی‌تواند بدون مقدار باشد. جلوگیری از بدون مقدار بودن یک فیلد توسط خاصیت NOT NULL صورت می‌گیرد.

در بسیاری از جدول‌ها، به عنوان فیلد کلیدی، شناسه‌ای غیر از مشخصه‌های اصلی موجودیت، انتخاب می‌شود. در این گونه مواقع معمولاً لازم است خاصیت افزایش خودکار این فیلد نیز فعال شود. برای فعال کردن این خاصیت می‌توان از عبارت AUTO_INCREMENT استفاده نمود.

مثال:

در این مثال دستورات ساخت جدول مثال قبل طوری بازنویسی شده است که علاوه بر تعریف کلید، خاصیت افزایش خودکار و خاصیت جلوگیری از بی مقدار بودن آن نیز بررسی شود:

```

$sql = "CREATE TABLE Persons
(
personID int NOT NULL AUTO_INCREMENT PRIMARY KEY,
FirstName varchar(15),
LastName varchar(15),
Age int
);

```

```
mysql_query($sql,$con);
```

پس از اجرای موفقیت آمیز هر کدام از کدهای مربوط به پایگاه داده در مرورگر، با مراجعه به

محیط سرویس دهنده WAMP، پایگاه داده، جدول و فیلدهای مربوط به آن را مشاهده خواهید کرد که توسط کدهای PHP ساخته شده اند.

تمرین: بانک اطلاعاتی مربوط به یک مدرسه که در ابتدای این فصل معرفی شد را به همراه جداول و فیلدهای هر کدام با استفاده از دستورات PHP بسازید.

۴-۷-۴- درج اطلاعات در جدول

با استفاده از دستور INSERT INTO می توان اطلاعات یک رکورد را در یک جدول درج نمود. این دستور به یکی از دو شکل کلی زیر به کار می رود:

شکل اول:

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

نکته: از این به بعد در تمامی مثال ها فرض بر این است که پایگاه داده از قبل ساخته شده است. اما اگر از نام پایگاه داده ای که تاکنون ایجاد نشده است استفاده شود، عملیات کار با پایگاه داده ناموفق خواهد بود.

شکل دوم:

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

مثال:

این مثال اطلاعات مربوط به دو نفر را در جدولی که از قبل با نام Persons طراحی شده است، درج می کند:

```
<?php  
$con = mysql_connect("localhost","root","");  
if (!$con)  
{  
die('Could not connect: ' . mysql_error());  
}  
mysql_select_db("my_db", $con);  
$query1="INSERT INTO Persons (FirstName, LastName, Age) VALUES  
( 'Mahdi',
```

```
'Razavi', '15')"
mysql_query($query1);
$query2="INSERT INTO Persons(First Name, Last Name, Age)
VALUES ('Hadi', 'Salimi', '13'), ('Zahra', 'sarami', '15')";
mysql_query($query2);
mysql_close($con);
?>
```

همان‌طور که اشاره شد تابع () mysql_query برای اجرای دستورات مربوط به پایگاه داده به کار می‌رود. در مثال فوق، این تابع، query مربوط به درج اطلاعات فیلدهای LastName، FirstName و Age برای دو رکورد جدید را اجرا می‌کند.

نکته: دقت کنید که در کد php، مقادیر درج شده در جدول باید داخل کوتیشن تکی یا single quotation نوشته شوند.

۵-۴-۷- درج اطلاعات در جدول از طریق فرم

همان‌طور که می‌دانید، فرم‌ها به منظور تعامل با کاربران در صفحات وب سایت، استفاده می‌شوند. از فرم‌ها می‌توان برای دریافت اطلاعات از کاربر و درج در پایگاه داده، استفاده نمود. در مثال زیر یک فرم با سه فیلد برای ارتباط با جدول Persons طراحی شده است. با دریافت اطلاعات از کاربر و ارجاع به صفحه insert.php عمل درج در پایگاه داده، انجام می‌شود.

مثال:

برنامه زیر فرم دریافت اطلاعات را طراحی می‌کند:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" http://www.
w3.org/TR/xhtml1/DTD/xhtml1_strict.dtd">
```

```
<html>
```

```
<head>
```

```
<meta http_equiv="Content_Type" content="text/html; charset=utf_8"/>
```

```
</head>
```

```
<body>
```

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

```

Firstname: <input type="text" name="firstname" />
Lastname: <input type="text" name="lastname" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>

</body>
</html>

```

فایل insert.php که اطلاعات دریافت شده از فرم را در جدول درج می‌کند.

```

<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
die(" Could not connect: " . mysql_error());
}

mysql_select_db("my_db", $con);

$sql="INSERT INTO Persons (FirstName, LastName, Age)
VALUES
('$ _POST[firstname]', '$ _POST[lastname]', '$ _POST[age]')";

if (!mysql_query($sql,$con))
{
die(" Error: " . mysql_error());
}
echo "1 record added";
mysql_close($con);
?>

```

نکته: برای درج اطلاعات فارسی در جدول، حتماً باید از طریق فرم اقدام به ورود فیلدها نمایید و درج اطلاعات فارسی به طور مستقیم در جدول موجب می‌شود هنگام خواندن آنها، کدهای نامشخصی نمایش داده شود.

۵-۷- انتخاب و نمایش اطلاعات جدول

پس از ایجاد جدول و وارد کردن اطلاعات در آن می‌توانید با استفاده از دستور SELECT آنها را انتخاب نمایید.

ساختار کلی این دستور به صورت زیر است:

نام فیلد یا فیلدها SELECT
نام جدول FROM

مثال:

در این مثال تمام اطلاعات جدول Persons را نمایش می‌دهد.

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
die("Could not connect: " . mysql_error());
}

mysql_select_db("my_db", $con);

$result = mysql_query("SELECT * FROM Persons");

while($row = mysql_fetch_array($result))
{
echo $row['FirstName'] . " . " . $row['LastName'];
echo "<br />";
}
mysql_close($con);
?>
```

نکته: در php به عناصر آرایه با اندیس غیر عددی نیز می توان دسترسی داشت.

نکته: حلقه While در این برنامه تا هنگامی که سطری وجود داشته باشد تا در متغیر \$row قرار گیرد ادامه پیدا می کند. به عبارت دیگر تا اتمام تمامی سطرهای موجود در جدول Persons ادامه می یابد.

در این مثال تابع mysql_query() با اجرای کد SQL تمام اطلاعات جدول را در متغیر \$result می ریزد، سپس تابع mysql_fetch_array() اولین سطر از مجموعه رکوردها را که به صورت آرایه در نظر گرفته می شوند، درون متغیر \$row می ریزد، به همین دلیل \$row هم یک آرایه خواهد بود. با هر بار اجرای حلقه، و فراخوانی تابع mysql_fetch_array()، سطر بعدی فراخوانی می شود.

۱-۵-۷- نمایش اطلاعات پایگاه داده در جدول

با توجه به اینکه یکی از کاربردهای جدول ها در HTML، سازماندهی اطلاعات و نحوه نمایش آنهاست، بنابراین بهتراست اطلاعات فراخوانی شده از جداول بانک اطلاعاتی را در قالب جدول نمایش دهیم.

مثال:

مثال قبل را طوری بازنویسی می کنیم که اطلاعات را در جدول نشان دهد:

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);

$result = mysql_query("SELECT * FROM Persons");

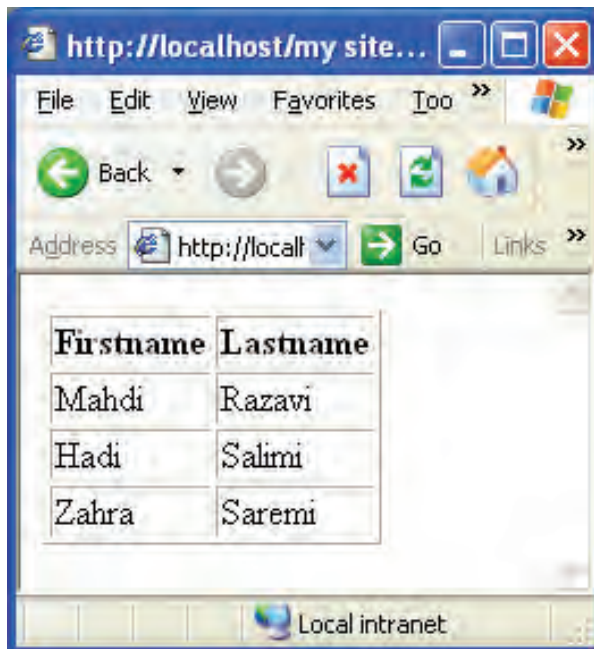
echo "<table border='1'>
<tr>
<th>Firstname</th>
<th>Lastname</th>
```

```

</tr>";
while($row = mysql_fetch_array($result))
{
echo "<tr>";
echo "<td> . $row['FirstName'] . "</td>";
echo "<td> . $row['LastName'] . "</td>";
echo "</tr>";
}
echo "</table>";
mysql_close($con);
?>

```

خروجی : شکل ۷-۶



شکل ۷-۶- نمایش اطلاعات در جدول

پژوهش: اسکریپتی بنویسید که اطلاعات جدول را به صورت مرتب شده (صعودی) بر اساس نام خانوادگی نمایش دهد. (راهنمایی: با استفاده از دستور مرتب سازی در SQL)

۶-۷- به روز رسانی اطلاعات جدول

یکی از مهمترین عملیات در پایگاه‌های داده روی اطلاعاتی که از قبل وارد شده‌اند، امکان ویرایش آنها است. به منظور تغییر اطلاعات موجود در جدول از دستور UPDATE استفاده می‌شود. ساختار کلی این دستور به شکل زیر است:

نام جدول UPDATE

... مقدار=فیلد دوم, مقدار=فیلد اول SET

شرط WHERE

مثال:

جدول Person مثال‌های قبل را در نظر بگیرید، فرض کنید فیلد مربوط به Age برای فردی به نام Mahdi Razavi اشتباه درج شده باشد و بخواهیم مقدار آن را به ۱۶ تغییر دهیم.

جدول ۴-۷

FirstName	LastName	Age
Mahdi	Razavi	15
Hadi	Salimi	13
Zahra	Saremi	15

در این صورت کد اسکریپت مورد نظر به شکل زیر نوشته می‌شود:

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}

mysql_select_db("my_db", $con);
```

```
mysql_query("UPDATE Persons SET Age = '16'
WHERE FirstName='Mahdi' AND LastName='Razavi'", $con);
mysql_close($con);
?>
```

جدول ۵-۷- به روز رسانی جدول Persons

FirstName	LastName	Age
Mahdi	Razavi	16
Hadi	Salimi	13
Zahra	Saremi	15

۷-۷- حذف رکورد

دستور DELETE FROM برای حذف رکورد از جدول به کار می‌رود.
ساختار کلی این دستور به شکل زیر است :

نام جدول DELETE FROM

شرط WHERE

توسط عبارت شرطی WHERE رکورد یا رکوردهایی که باید حذف شوند، تعیین می‌شود.

مثال :

فرض کنید در جدول Persons بخواهیم رکورد فردی را که نام خانوادگی او Razavi است را حذف کنیم. در این صورت اسکریپت اجرای رکورد مورد نظر به صورت زیر خواهد بود :

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
```

```
mysql_select_db("my_db", $con);
```

```
mysql_query("DELETE FROM Persons WHERE Last Name = 'Razavi' ");
```

```
mysql_close($con);
```

```
?>
```

پس از اجرای کد فوق، جدول Persons به شکل زیر خواهد بود :

جدول ۶-۷

FirstName	LastName	Age
Hadi	Salimi	13
Zahra	Saremi	15

جدول مجموعه‌ای از داده‌های مرتبط با هم است و از سطرها و ستون‌های مختلفی تشکیل شده است.

پرس و جو (Query) یک پرسش یا درخواست از پایگاه داده است که برای برگرداندن اطلاعات مشخص از مجموعه‌ای از رکوردها استفاده می‌شود.

اتصال به پایگاه داده در PHP توسط تابع `mysql_connect()` انجام می‌شود.

تابع `mysql_close()` نیز برای قطع ارتباط با پایگاه داده به کار می‌رود.

تابع `mysql_error()` خطایی که منجر به عدم اتصال به پایگاه داده شده است را برمی‌گرداند.

برای ایجاد پایگاه داده از دستور `CREATE DATABASE` استفاده می‌شود.

ایجاد جدول در پایگاه داده‌ای که از قبل ایجاد شده است، توسط دستور `CREATE TABLE`

انجام می‌شود و پیش از آن باید توسط تابع `mysql_select_db()` پایگاه داده موردنظر انتخاب شود.

پرس و جوها نیز توسط تابع `mysql_query()` اجرا می‌شوند.

هر جدول می‌تواند یک فیلد کلیدی داشته باشد که برای اطمینان از تکراری نبودن سطرها

در جدول استفاده می‌شود. عبارت `PRIMARY KEY` برای تعریف فیلد کلیدی هنگام

ساخت جدول به کار می‌رود.

با استفاده از دستور `INSERT INTO` می‌توان اطلاعات را در یک جدول درج نمود.

به منظور تغییر اطلاعات موجود در جدول از دستور `UPDATE` استفاده می‌شود.

دستور `DELETE FROM` برای حذف رکورد از جدول به کار می‌رود.

- ۱- توسط دستورات SQL، یک پایگاه داده برای کلاس خود ایجاد کنید.
 - ۲- توسط دستورات SQL، در پایگاه داده سؤال اول، جدولی برای نگهداری مشخصات همکلاسی‌های خود ایجاد کنید.
 - ۳- توسط دستورات SQL، مشخصات خودتان را به همراه چند نفر از دوستان خود درون جدول سؤال دوم، درج کنید.
 - ۴- توسط دستورات SQL، تمام رکوردهای موجود در جدول را نمایش دهید.
 - ۵- در جدولی که ایجاد نمودید، مشخصات خودتان را تغییر دهید.
 - ۶- عملیات سؤال ۳ را با استفاده از فرم انجام دهید.
 - ۷- در جدولی که ایجاد نمودید، مشخصات خودتان را حذف کنید.
-

فصل ۸

ارتباط با پایگاه داده در Dreamweaver

هدف‌های رفتاری:

پس از آموزش این فصل هنرجو می‌تواند:

- ۱- با سیستم مدیریت پایگاه داده MySQL توسط نرم افزار Dreamweaver، ارتباط برقرار کند.
- ۲- از رفتارهای سرویس دهنده در نرم افزار Dreamweaver، استفاده کند.
- ۳- رفتارهای جدیدی برای سرویس دهنده تعریف کند.
- ۴- توسط نرم افزار Dreamweaver، اطلاعات درج شده در جدول پایگاه داده را نمایش دهد.
- ۵- توسط نرم افزار Dreamweaver، اطلاعات مورد نظر را در جدول پایگاه داده درج کند.
- ۶- توسط نرم افزار Dreamweaver، اطلاعات درج شده را ویرایش و به روزرسانی کند.
- ۷- رکورد یا رکوردهای موردنظر را از جدول پایگاه داده، حذف کند.
- ۸- مفهوم فرم‌های Master و Detail را بیان کند.

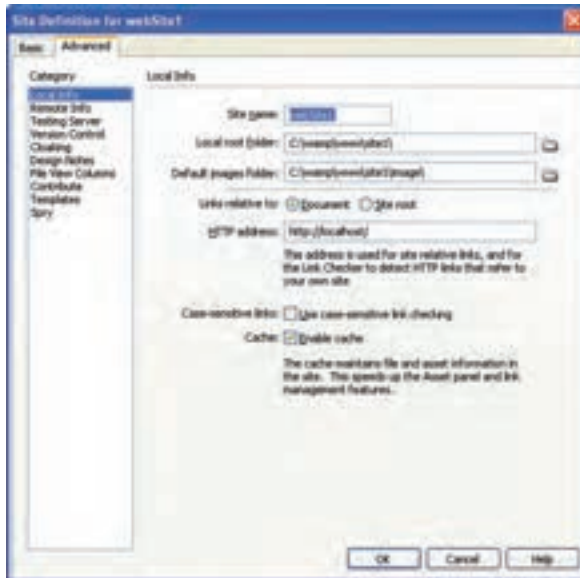
همان‌طور که می‌دانید، PHP یک زبان اسکریپت نویسی Server Side است که برای طراحی وب سایت‌های پویا به کار می‌رود.

در جلد اول این کتاب با نرم افزار Dreamweaver آشنا شدید.

یکی از قابلیت‌های مهم این نرم افزار ایجاد وب سایت‌های پویا و توانایی ایجاد ارتباط با پایگاه داده و مدیریت آن است. استفاده از این نرم افزار موجب صرفه جویی در زمان کدنویسی می‌شود.

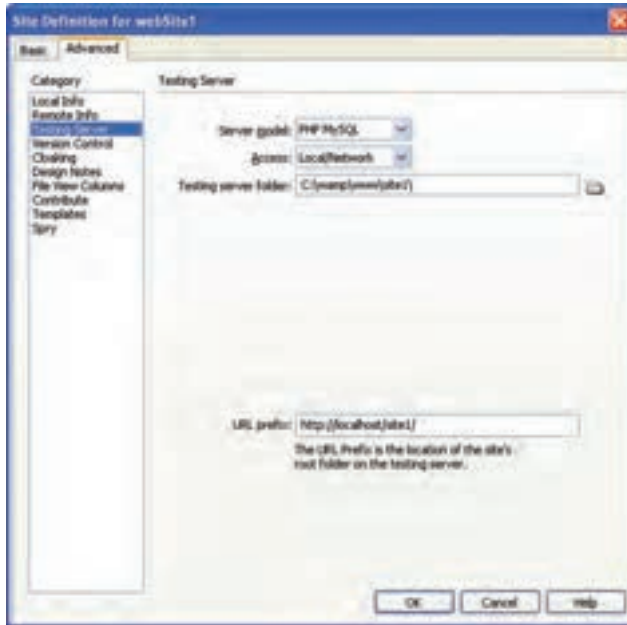
۸-۱- تنظیمات اولیه برای ارتباط با پایگاه داده

ابتدا یک وب سایت جدید در مسیری که برای سرویس دهنده WAMP، در نظر گرفته‌اید بسازید. و تنظیمات مربوط به نام و مسیر فایل‌های آن را مطابق دستورالعمل و با دقت انجام دهید. (شکل ۸-۱) با روش مقدماتی ساخت وب سایت آشنا شدید. روش پیشرفته ساخت وب سایت تشابه زیادی با روش مقدماتی دارد، اما در روش پیشرفته که از طریق زبانه Advanced شکل ۸-۱ امکان پذیر است، امکان انجام تنظیمات دقیق‌تر مربوط به اتصالات و وب سایت پویا نیز فراهم می‌شود. انجام تنظیمات گروه Local Info شامل تعیین نام وب سایت، مسیر ذخیره فایل‌های وب سایت، مسیر ذخیره تصاویر و ... است.



شکل ۸-۱- تنظیمات مسیر وب سایت

نکته: انجام تنظیمات این مراحل بسیار حساس است و در صورت عدم انجام صحیح این تنظیمات، قادر به برقراری ارتباط با پایگاه داده و یا سایر اجزای آن نخواهید بود. در گروه بندی Testing Server مدل سرویس دهنده (Server Model) و نحوه دسترسی به آن (Access) براساس شکل ۸-۲، تنظیم کنید.



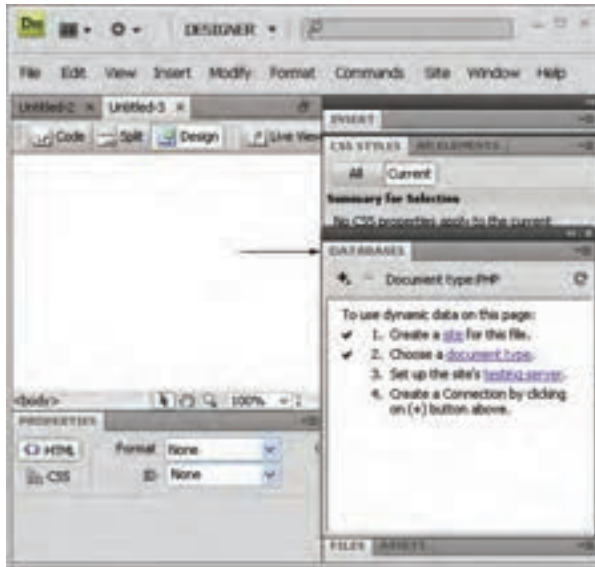
شکل ۸-۲- تنظیمات سرویس دهنده مورد استفاده

در گروه Testing Server مطابق شکل ۸-۲، پس از انتخاب PHP MySQL برای Server Model، نحوه دسترسی را از نوع Local/ Network قرار دهید. مسیر کامل ذخیره وب سایت در پوشه سرویس دهنده WAMP را مقابل Testing Server Folder وارد نمایید.

مسیر http://localhost/web1/ و در ادامه آن، پوشه ای که درون www (در مسیر WAMP) برای ذخیره وب سایت ایجاد کرده اید را در قسمت URL prefix برای دسترسی به مسیر سرویس دهنده وارد کنید.

پس از ایجاد وب سایت جدید با تنظیمات فوق، اتصال به پایگاه داده و ارائه سایر قابلیت های مربوط به آن توسط پنل DATABASES فراهم می شود. اگر هر کدام از پنل های مورد نیاز خود را

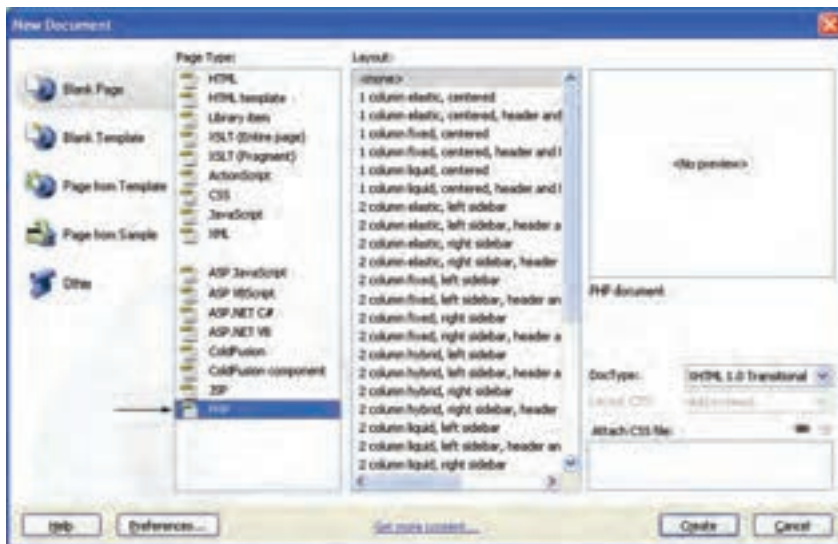
مشاهده نمی‌کنید، از منوی Window آن را روی محیط کاری نمایان سازید. (شکل ۸-۳)



شکل ۸-۳

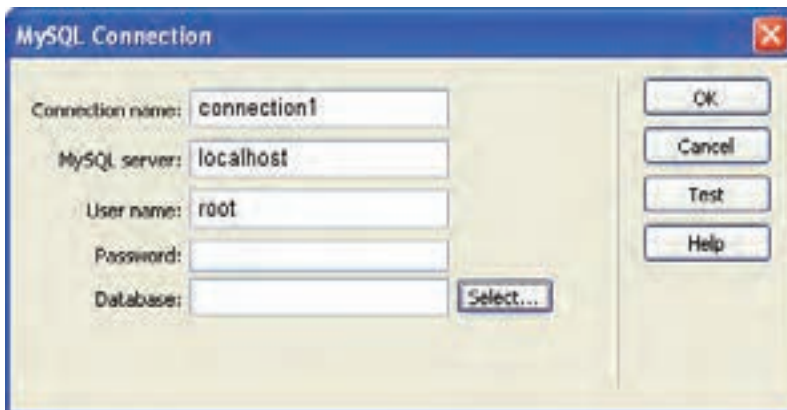
سپس یک صفحه وب جدید بسازید. هنگام ایجاد یک فایل جدید، انواع مختلف صفحه به طور

آماده وجود دارد، نوع PHP را انتخاب نمایید. (شکل ۸-۴)



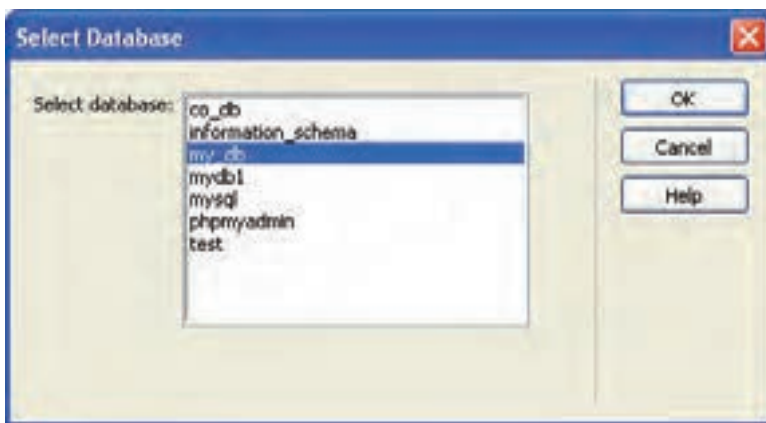
شکل ۸-۴

برای اتصال به پایگاه داده در پنل DATABASES روی علامت  کلیک کنید و گزینه MySQL Connection را انتخاب کنید.



شکل ۵-۸- ایجاد اتصال به پایگاه داده

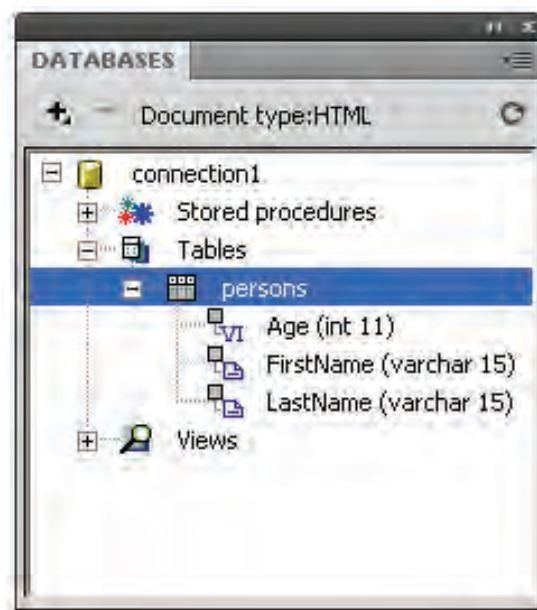
به منظور انتخاب پایگاه داده مورد نظر می‌توانید نام آن را مقابل Database وارد کنید و یا اینکه با کلیک روی دکمه Select آن را از لیست پایگاه داده‌های موجود، انتخاب نمایید.



شکل ۶-۸- انتخاب پایگاه داده

در این مرحله از کار و ایجاد ارتباط با پایگاه داده، ممکن است با پیغام خطاهای مختلفی مواجه شوید، در این صورت قبل از هر کاری با مطالعه پیغام خطا سعی کنید علت را پیدا کنید. این خطاها می‌توانند دلایل متفاوتی داشته باشند که متداول‌ترین آنها عدم اجرای WAMP و یا بخشی از سرویس‌های آن است.

پس از انجام اتصالات به پایگاه داده و برقراری ارتباط، پنل DATABASES جداول موجود در پایگاه داده‌ای که اتصال با آن برقرار شده است، به همراه فیلدهای هر کدام را نشان می‌دهد.



شکل ۷-۸- پنل DATABASES

علاوه بر این در صورتی که ارتباط با پایگاه داده به درستی برقرار شده باشد، در مسیر ذخیره وب سایت، دو پوشه با نام‌های Connections و _mmServerScripts ساخته می‌شود که تنظیمات مربوط به این ارتباط را نگهداری می‌کنند.

۱-۱-۸- ایجاد Recordset

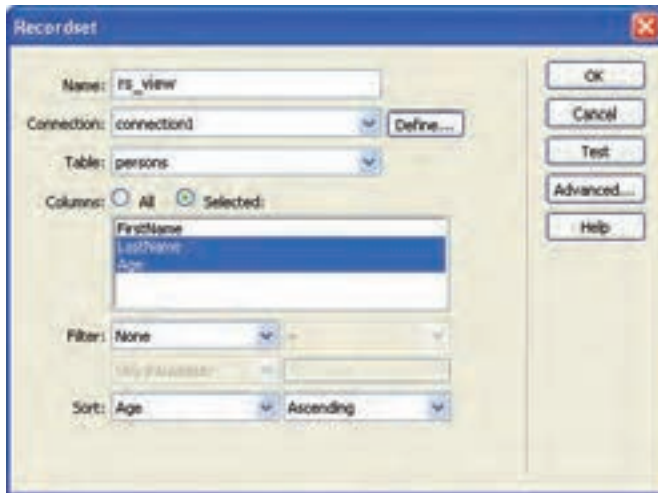
همان طور که می‌دانید، صفحات وب قادر نیستند به طور مستقیم به اطلاعات ذخیره شده در بانک‌های اطلاعاتی دسترسی پیدا کنند و برای برقراری این ارتباط از Recordset استفاده می‌نمایند. **نکته:** یک Recordset مجموعه‌ای از اطلاعات مربوط به رکوردها است که توسط Query ایجاد می‌شود.

Queryها ابزارهای اصلی کار با پایگاه داده و اجزای آن هستند و اغلب عملیات روی پایگاه داده نیازمند ایجاد یک Query است.

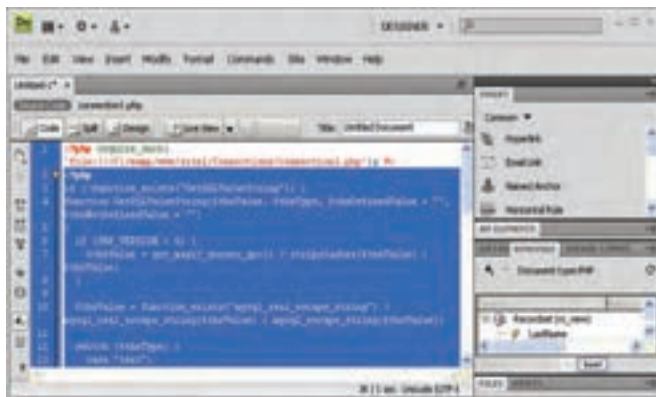
فرض کنید می‌خواهیم اطلاعات درج شده در جدول Persons از پایگاه داده‌ای که در مبحث قبل ایجاد کرده‌ایم را نمایش دهیم.

ابتدا باید برای فیلدهای مورد نظر در جدول، برای نمایش در صفحه وب، یک Recordset ایجاد شود.

برای ساخت یک Recordset جدید در پنل BINDINGS روی علامت **+** کلیک کنید و از منوی باز شده، گزینه (Query) Recordset را انتخاب نمایید.



شکل ۸-۸ - ایجاد Recordset

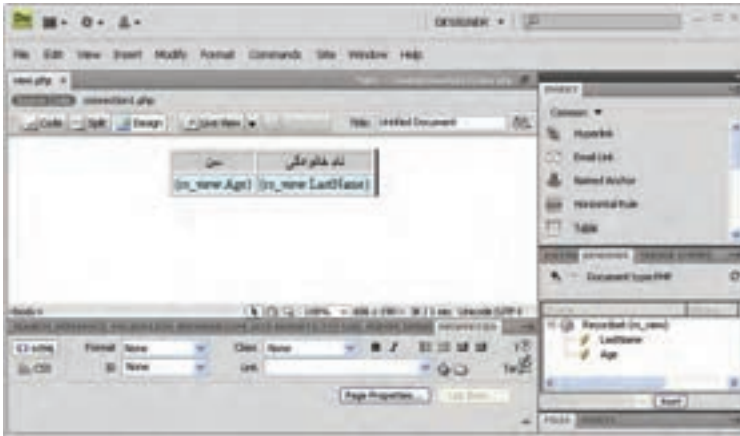


شکل ۸-۹ - کدهای درج شده به طور خودکار

۸-۲- مشاهده اطلاعات پایگاه داده

حال، برای استفاده از Recordset به منظور مشاهده رکوردهای موجود در پایگاه داده، در صفحه وب، باید هر کدام از فیلدها را به محل مورد نظر انتساب دهید.

جدولی به تعداد ستون فیلدهای مورد نظر درج کنید و در هر یک از سطرهاى آن، رکورد مربوطه را درگ کنید.



شکل ۸-۱۰- انتساب فیلدها به جدول

اگر این صفحه را از طریق localhost در مرورگر مشاهده کنید، خواهید دید که اولین رکورد جدول پایگاه داده را نشان می‌دهد.



شکل ۸-۱۱- مشاهده فیلدهای انتخاب شده از اولین رکورد جدول

برای مشاهده تمام رکوردهای جدول، باید از رفتارهای سرویس دهنده استفاده شود. این رفتارها از طریق پنل SERVER BEHAVIORS قابل استفاده هستند.

سطری که در نرم افزار Dreamweaver فیلدهای نام خانوادگی و سن را در آن درگ کردید،

انتخاب نمایید و در پنل SERVER BEHAVIORS روی علامت **+** کلیک کنید و در منوی مربوط به آن، گزینه Repeat Region را انتخاب نمایید.

چنانچه از انتخاب دقیق این سطر اطمینان ندارید، می‌توانید با مراجعه به نمای کد صفحه، بخشی را که درون برچسب <tr> قرار دارد را انتخاب کنید.

```
<tr>
  <td><?php echo $row_rs_view['Age']; ?></td>
  <td><?php echo $row_rs_view['LastName']; ?></td>
</tr>
```

پس از اعمال ویژگی Repeat Region، مشاهده خواهید کرد که کد فوق، درون یک حلقه while به صورت زیر قرار گرفته است :

```
<?php do { ?>
  <tr>
    <td><?php echo $row_rs_view['Age']; ?></td>
    <td><?php echo $row_rs_view['LastName']; ?></td>
  </tr>
  <?php } while ($row_rs_view = mysql_fetch_assoc($rs_view)); ?>
```

نکته : تابع () mysql_fetch_assoc یک سطر از جدولی را که توسط recordset مربوطه، تعیین شده است را به صورت آرایه برمی‌گرداند و به طور خودکار، در هر بار اجرای آن، سطر بعدی جدول برگردانده می‌شود.

اگر در جدول فوق، اطلاعات قابل نمایش بیش از یک صفحه باشد، باید توسط دکمه‌های مخصوصی، امکان پیمایش بین صفحات مختلف و مشاهده تمام رکوردها را فراهم نمایید. برای این کار کافی است از Recordset Navigation Bar استفاده کنید.

Recordset Navigation Bar می‌تواند شامل عبارات متنی پیوند داده شده و یا دکمه‌های متعدد باشد. شکل زیر دو نمونه از آن را نشان می‌دهد.

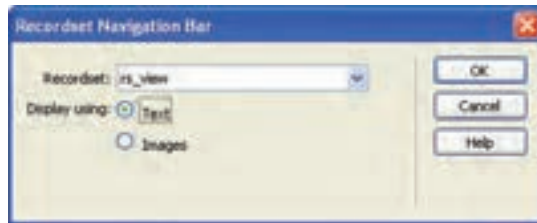


شکل ۸-۱۲

برای افزودن Recordset Navigation Bar در Dreamweaver رفتارهای سرویس دهنده مناسب وجود دارد و به منظور استفاده از آن در نمای طراحی صفحه نمایش اطلاعات پایگاه داده مراحل زیر را دنبال کنید:

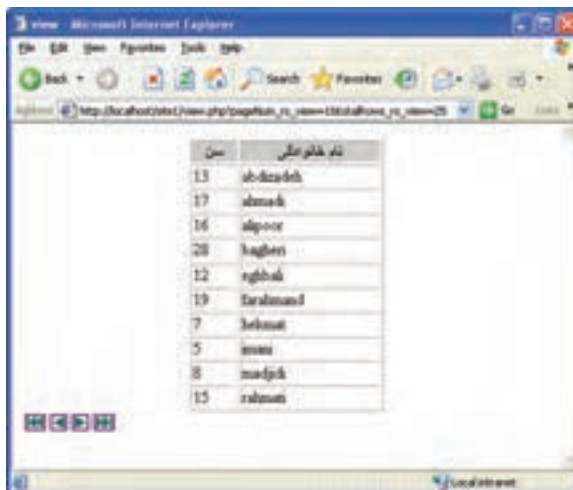
– از منوی Insert گزینه Data Object، سپس Recordset Paging و از زیر منوی مربوط به آن Recordset Navigation Bar را انتخاب نمایید.

– در کادر محاوره شکل ۸-۱۳ Recordset مربوط به نمایش اطلاعات و نوع Navigation Bar را انتخاب نمایید. (با انتخاب گزینه Text عبارات متنی و با انتخاب گزینه Images دکمه‌های تصویری نمایش پیدا خواهد کرد.)



شکل ۸-۱۳

پس از تنظیم و تأیید کادر Recordset Navigation Bar، با اجرای صفحه وب فوق در مرورگر مشاهده خواهید نمود که با کلیک روی گزینه‌های مربوط به رکوردهای بعدی، تمام رکوردهای موجود در صفحات دیگر نیز قابل مشاهده هستند.



شکل ۸-۱۴

۳-۸- درج اطلاعات در پایگاه داده

همانطور که می‌دانید برای درج اطلاعات در پایگاه داده، از فرم استفاده می‌شود. ابتدا فرم مورد نظر را برای ارتباط با پایگاه داده، طراحی کنید.



The image shows a simple web form for data entry. It consists of three text input fields stacked vertically. The first field is labeled 'نام:' (Name), the second 'نام خانوادگی:' (Last Name), and the third 'سن:' (Age). Below the input fields are two buttons: 'انصراف' (Cancel) on the left and 'ارسال' (Submit) on the right.

شکل ۸-۱۵

اگر بخواهید اطلاعات را در همان جدولی از پایگاه داده، که از قبل اتصال با آن را برقرار کردید، درج کنید، دیگر نیازی به ساختن مجدد اتصال نیست. پس از ساخت فرم از پنل SERVER BEHAVIORS روی علامت **+** کلیک کنید و در منوی مربوط به آن، گزینه Insert Record را انتخاب نمایید.



شکل ۸-۱۶- درج رکورد

در کادر محاوره شکل ۸-۱۶، هر کدام از ستون‌های جدول را به مقدار متناظر در فرم نسبت دهید. (با انتخاب هر گزینه در بخش Columns گزینه مناسب برای آن را در مقابل کادر بازشو Value انتخاب نمایید.)

در گزینه مقابل After inserting, go to انتخاب کنید که قصد دارید پس از درج اطلاعات در پایگاه داده، کاربر به آنجا هدایت شود.

۱-۳-۸- درج اطلاعات در پایگاه داده با استفاده از Wizard

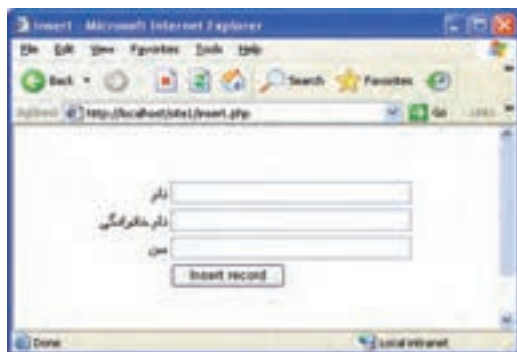
روش دیگر درج اطلاعات در پایگاه داده، استفاده از امکانات Server Behavior در Dreamweaver است.

برای استفاده از این قابلیت در نمای طراحی (Design) از صفحه PHP مورد نظر از منوی Insert گزینه Data Objects، سپس گزینه Insert Record و پس از آن Record Insertion Form Wizard را انتخاب نمایید.



شکل ۸-۱۷

برای تنظیم کادر محاوره شکل ۸-۱۷ در قسمت Label عبارت متنی مقابل هریک از فیلدهای جدول را وارد کنید، در کادر متن to go, After inserting, درج اطلاعات رکورد، کاربرد به آنجا منتقل شود. پس از انجام تمام تنظیمات مورد نظر و پس از تأیید کادر محاوره ۸-۱۷، مشاهده خواهید نمود که فرم درج اطلاعات در پایگاه داده به طور خودکار ساخته می‌شود.



شکل ۸-۱۸

آن را به صفحه deleteConfirm.php پیوند دهید و به انتهای آدرس URL آن عبارت زیر را اضافه کنید:

```
?recordID=<<?php echo $row_recordsetName["fieldName"];?>
```

در عبارت فوق، به جای recordsetName نام Recordset و به جای fieldName نام فیلدی که می خواهید حذف رکورد بر اساس آن انجام شود را وارد کنید.

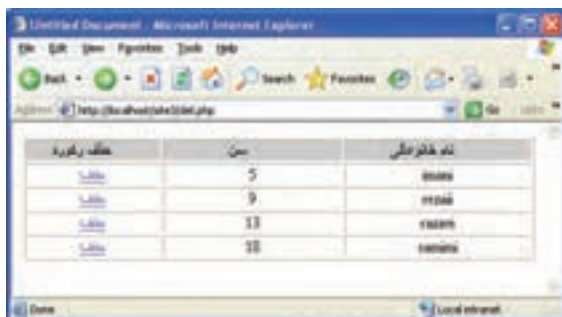
recordID نیز نام یک متغیر است که به دلخواه می توانید آن را تغییر دهید.

علامت ؟ در ابتدای عبارت فوق، یک پارامتر را برای URL تعیین می کند.

با توجه به توضیحات فوق، اگر بخواهید در جدول Person حذف اطلاعات بر اساس فیلد

نام خانوادگی انجام شود، و در متغیر record_ID ریخته شود، باید کد زیر را درج نمایید:

```
deleteConfirm.php?record_ID=<<?php echo $row_Recordset1["LastName"];?>
```



شکل ۲۰-۸

در Dreamweaver حالت Live Data را برای نمایش اطلاعات جدول، فعال کنید. برای

فعال کردن این حالت، در منوی View، گزینه Live Data را در حالت انتخاب شده قرار دهید.


در مرحله بعد، باید ابتدا رکورد مورد نظر برای حذف، نمایش داده شود. سپس از کاربر خواسته

شود که در صورت اطمینان روی دکمه حذف کلیک کند.

بنابراین، در صفحه deleteConfirm.php علاوه بر فیلد مخفی و دکمه، کادرهای متنی برای

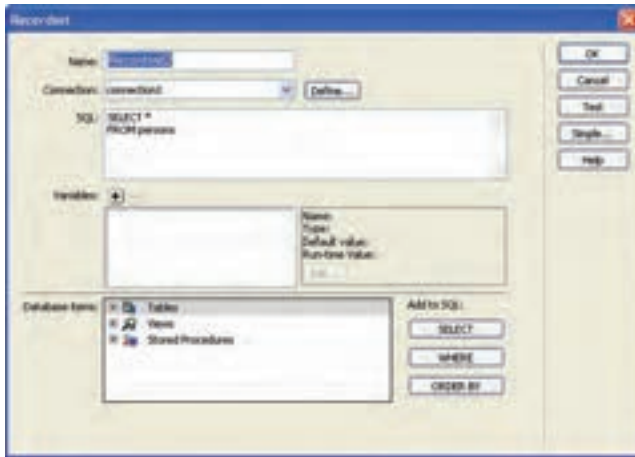
نمایش اطلاعات مربوط به رکورد انتخاب شده برای حذف، طراحی کرده، سپس یک Recordset

جدید ایجاد نمایید.

برای ایجاد Recordset مانند مباحث قبل در پنل BINDINGS روی علامت  کلیک کنید

و از منوی باز شده، گزینه Recordset (Query) را انتخاب نمایید.

اگر کادر مربوط به ایجاد Recordset در حالت Advanced (مانند شکل ۸-۲۱) باز شده است، بر روی دکمه Simple کلیک کنید تا کادر آن مطابق شکل ۸-۲۲ باز شود.



شکل ۸-۲۱





شکل ۸-۲۲

در کادر محاوره ۸-۲۲، تنظیمات را مطابق شکل انجام دهید. در این کادر، در کادر بازشو مقابل Filter، فیلدی را که حذف بر اساس آن انجام می‌شود و توسط صفحه قبل ارسال شده است را انتخاب نمایید و در مقابل URL Parameter نام متغیری را که هنگام ارسال اطلاعات توسط فایل deleteConfirm.php تعیین کرده بودید را وارد کنید (به کوچک و بزرگ بودن حروف دقت کنید).

به هر کدام از کادرهای متنی فرم، فیلد مربوطه را از Recordset ای که ساخته اید، انتساب دهید.



شکل ۸-۲۳

برای فیلد مخفی نیز، پس از انتخاب آیکن زرد رنگ، در پنل Properties با کلیک روی علامت  فیلدی که برای حذف در نظر گرفته شده بود (در اینجا LastName) را انتساب دهید. در این مرحله از کار، می توانید از رفتارهای سرویس دهنده استفاده نمایید. در پنل SERVER BEHAVIORS روی علامت  کلیک کنید و در منوی مربوط به آن، گزینه Delete Record را انتخاب نمایید.



شکل ۸-۲۴

این کادر را نیز مطابق شکل ۸-۲۴ تنظیم کنید.

در مقابل Primary Key column فیلد LastName که برای حذف انتخاب شده بود را تعیین کنید، و در کادر باز شو Primary Key Value، گزینه Form Variable را انتخاب و در مقابل آن، عنوان فیلد مخفی را وارد کنید. (به کوچک و بزرگ بودن حروف دقت کنید یعنی حروف (hiddenField)

در کادر متن After deleting, go to، صفحه وبی را وارد کنید که می خواهید پس از حذف اطلاعات رکورد انتخابی، کاربر به آنجا منتقل شود.

حال با اجرای فایل اولیه حذف اطلاعات، (در اینجا فایل del.php) می توانید با مشاهده

بخشی از اطلاعات مربوط به رکوردهای جدول، پس از انتخاب ستون حذف مربوط به آن، و تأیید عملیات، رکورد مورد نظر خود را از بانک اطلاعاتی حذف نمایید.

۵-۸- ویرایش اطلاعات پایگاه داده

برای ویرایش اطلاعات موجود در پایگاه داده، بهتر است ابتدا رکورد مورد نظر برای ویرایش انتخاب شود. سپس در یک فرم، اطلاعات موجود نمایش یابد و کاربر بتواند اطلاعات جدید را وارد کرده، پس از کلیک روی دکمه مخصوص، عمل ویرایش اطلاعات در بانک اطلاعاتی انجام شود. روش انتخاب رکورد به سلیقه طراح وابسته است. در اینجا به طور نمونه در کنار دکمه حذف که پیش از این برای حذف رکورد درج کردیم، یک پیوند برای ویرایش اطلاعات رکورد اضافه می‌نماییم. برای طراحی عملیات ویرایش، ابتدا فرمی را برای نمایش اطلاعات مربوط به رکورد مورد نظر کاربر طراحی کنید و آن را با نام مشخصی ذخیره کنید (به طور مثال با نام update.php). همانند مرحله حذف اطلاعات، بر روی این فرم، یک فیلد مخفی، کادرهای متنی نمایش اطلاعات رکورد و یک دکمه نیز درج نمایید.

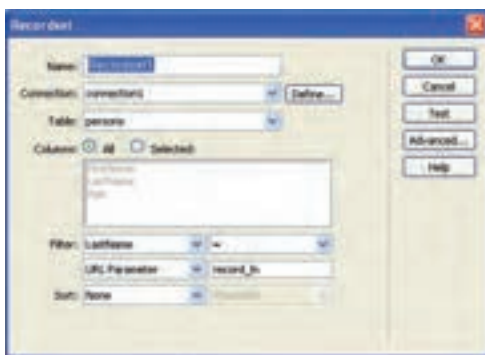
برای نمایش رکوردها، می‌توانید مانند بخش حذف، در جدول نمایش اطلاعات، یک ستون اضافی برای ویرایش نیز درج کنید و آن را به صفحه update.php پیوند دهید و با توجه به توضیحات بخش حذف اطلاعات، اگر بخواهید در جدول Persons حذف اطلاعات بر اساس فیلد نام خانوادگی انجام شود، باید به جای آدرس URL آن کد زیر را درج نمایید:

```
update.php?record_ID=<?php echo $row_recordset1['LastName']; ?>
```



شکل ۲۵-۸

در Dreamweaver حالت Live Data را برای نمایش اطلاعات جدول، فعال کنید. در مرحله بعد، باید ابتدا اطلاعات رکورد مورد نظر برای ویرایش، نمایش داده شود. بنابراین، در صفحه update. php علاوه بر فیلد مخفی و دکمه، کادرهای متنی برای نمایش اطلاعات مربوط به رکورد انتخاب شده برای ویرایش طراحی کرده، سپس یک Recordset جدید ایجاد نمایید. مانند مرحله حذف، در حالت Simple اطلاعات Recordset را مطابق شکل زیر وارد نمایید.





شکل ۸-۲۶

در صفحه update. php به هر کدام از کادرهای متنی فرم، فیلد مربوطه را از Recordset ای که ساخته‌اید، انتساب دهید. و بخش Filter را مطابق توضیحات ارائه شده در بخش حذف اطلاعات، تنظیم کنید.



شکل ۸-۲۷

برای فیلد مخفی نیز، پس از انتخاب آیکن زرد رنگ، در پنل Properties با کلیک روی علامت  فیلدی که برای ویرایش در نظر گرفته شده بود را انتساب دهید. در این مرحله از کار، می‌توانید از رفتارهای سرویس دهنده استفاده نمایید. در پنل SERVER BEHAVIORS روی علامت  کلیک کنید و در منوی مربوط به آن،

گزینه Update Record را انتخاب نمایید.



شکل ۸-۲۸

این کادر را نیز مطابق شکل ۸-۲۸ تنظیم کنید. در کادر متن After Updating, go to صفحه وبی را وارد کنید که می‌خواهید پس از ویرایش اطلاعات رکورد، کاربر به آنجا منتقل شود.

۸-۶- فرم‌های Master و Detail

در برخی از جدول‌های پایگاه داده ممکن است فیلدهای مربوط به رکوردها زیاد باشند و یا به هر دلیل نمایش تمام آنها غیر ضروری به نظر برسد. به طور مثال در یک فروشگاه الکترونیکی ابتدا در صفحه وب، اطلاعات کلی مربوط به اجناس از قبیل نام جنس و قیمت آن نمایش داده می‌شود و کاربر با کلیک روی عنوان جنس می‌تواند سایر ویژگی‌های آن را مشاهده نماید. در این صورت به فرم اولیه که شامل اطلاعات کلی است، Master و به فرمی که اطلاعات دقیق‌تر را نشان می‌دهد، Detail گفته می‌شود.

برای طراحی چنین فرم‌هایی به ترتیب مراحل زیر را دنبال کنید:

– یک صفحه PHP جدید ایجاد کنید و برای آن Recordset مربوطه را ایجاد کنید. (این صفحه Master خواهد بود، دقت کنید که در این صفحه هنگام ساخت Recordset تمام فیلدهایی که قصد نمایش آنها در فرم Detail را دارید انتخاب شوند.)

– در نمای طراحی صفحه Master از منوی Insert گزینه Data Objects و پس از آن Master Detail Page Set را انتخاب نمایید.



شکل ۸-۲۹

– کادر محاوره شکل ۸-۲۹ را مطابق توضیحات روی آن تنظیم، و سپس روی Ok کلیک کنید.
 – پس از تأیید کادر محاوره شکل ۸-۲۹ مشاهده خواهید نمود که علاوه بر صفحه Master یک صفحه جدید با عنوان Detail ساخته شده است و با کلیک روی اطلاعات نمایش داده شده در صفحه Master می‌توانید اطلاعات کامل آن را در صفحه Detail مشاهده نمایید.

مهم‌ترین ویژگی نرم افزار Dreamweaver ایجاد وب سایت‌های پویا و توانایی برقراری ارتباط با پایگاه داده و مدیریت آن است. استفاده از این نرم افزار موجب صرفه جویی در زمان کدنویسی می‌شود.

اتصال به پایگاه داده و ارائه سایر قابلیت‌های مربوط به آن توسط پنل DATABASES فراهم می‌شود.

در صورتی که ارتباط با پایگاه داده به درستی برقرار شده باشد، در مسیر ذخیره وب سایت، دو پوشه با نام‌های Connections و _mmServerScripts ساخته می‌شود که تنظیمات مربوط به این ارتباط را نگهداری می‌کنند.

صفحات وب قادر نیستند به طور مستقیم به اطلاعات ذخیره شده در بانک‌های اطلاعاتی دسترسی پیدا کنند و برای برقراری این ارتباط از Recordset استفاده می‌نمایند.

یک Recordset مجموعه‌ای از اطلاعات مربوط به رکوردها است که توسط Query ایجاد می‌شود.

تابع () mysql_fetch_assoc یک سطر از جدولی را که توسط recordset مربوطه، تعیین شده است را به صورت آرایه برمی‌گرداند و به طور خودکار، در هر بار اجرای آن، سطر بعدی جدول برگردانده می‌شود.

در صورتی که تمامی فیلدهای مربوط به رکوردها، در صفحه ضروری نباشد به فرم اولیه که شامل اطلاعات کلی است، Master و به فرمی که اطلاعات دقیق تر را نشان می‌دهد، Detail گفته می‌شود.

- ۱- یک پایگاه داده برای نمایش اطلاعات همکلاسی‌های کلاس خود ایجاد کنید.
 - ۲- در نرم افزار Dreamweaver با پایگاه داده سؤال اول، ارتباط برقرار کنید.
 - ۳- یک فرم برای دریافت اطلاعات هم کلاسی‌هایتان طراحی کنید.
 - ۴- تمام اطلاعات مندرج در پایگاه داده را به وسیله دکمه‌هایی برای نمایش رکوردهای صفحات دیگر، نمایش دهید.
 - ۵- صفحه‌ای برای ویرایش رکورد مورد نظر طراحی کنید.
 - ۶- صفحه‌ای برای حذف رکورد مورد نظر طراحی کنید.
 - ۷- فرم‌های Master و Detail را برای نمایش رکوردهای جدول، به دلخواه خود طراحی کنید.
-

فصل ۹

پروژه

هدف‌های رفتاری:

پس از آموزش این فصل هنرجو می‌تواند:

- ۱- مراحل ساخت وب‌سایت را به‌صورت عملی شرح دهد.
- ۲- با اجرای قدم به قدم تحلیل و طراحی سیستم فروشگاه الکترونیکی به‌طور عملی یک وب‌سایت پویا را طراحی کند.

در این فصل به منظور آشنایی عملی با طراحی وب سایت پویا، مراحل ساخت آن را برای پیاده‌سازی فروشگاه الکترونیکی بررسی می‌کنیم.

۱-۹- نیازسنجی

قبل از شروع طراحی وب سایت، باید اهدافی که قرار است سایت برای سفارش دهنده محقق نماید، مشخص شود.

فرض کنید صاحب یک فروشگاه، هدف از ساخت وب سایت را، خرید راحت‌تر و بدون نیاز به مراجعه حضوری مشتریان و در نتیجه فروش بیشتر کالاهايش، ذکر می‌کند.

پس از مشخص شدن هدف، فروشگاه‌های را که قرار است برای آن یک وب سایت طراحی کنید را مورد تحلیل قرار دهید و نیازهای آن‌را به‌طور دقیق بررسی کنید، و نتایج بررسی خود را در قالب یک سند آماده نمایید. و سند را در اختیار صاحب فروشگاه قرار دهید تا در صورت نیاز، اصلاحات اولیه صورت گیرد.

فرض کنید بخشی از نتیجه بررسی این مرحله به صورت زیر تهیه شده است :

- امکان درج کالا به همراه مشخصات اصلی و کاربردی
- گروه‌بندی کالاها بر اساس کاربرد آنها
- امکان عضویت در وب سایت توسط کاربران
- مشاهده کالاهاى موجود توسط کاربران
- سفارش کالا توسط کاربر
- ...

۲-۹- معماری اطلاعات

• مهم‌ترین عملیاتی که مدیر وب سایت باید بتواند انجام دهد عبارتند از :

- ۱- ورود به سیستم و صفحه کاربری خود
- ۲- انتخاب یک کالا برای ویرایش مشخصات
- ۳- ایجاد کالای جدید و درج مشخصات آن
- ۴- درج گروه‌بندی جدید برای مجموعه‌ای از کالاها

- ۵- کلیک روی دکمه‌های مخصوص حذف، اضافه و ویرایش کالا
- ۶- بازگشت به مرحله ۲ و انجام نامحدود عملیات تغییرات کالاها
- ۷- بازگشت به صفحه اصلی کاربری
- ۸- خروج از سیستم

● مهم‌ترین عملیاتی که مشتری باید بتواند انجام دهد عبارتند از :

- ۱- ورود به سیستم و صفحه کاربری مشتری
- ۲- انتخاب محصولات از روی مشخصات درج شده در وب سایت
- ۳- مشاهده لیست محصولاتی که تاکنون برای خرید انتخاب کرده است
- ۴- انتخاب صفحه یا کالاهای مورد نظر خود برای درج در لیست علاقمندی‌ها
- ۵- کلیک روی دکمه خرید برای اتمام عمل خرید کالاها
- ۶- ذخیره سفارش‌ها و بازگشت به صفحه کاربری خود
- ۷- گرفتن صورت حساب
- ۸- امکان حذف سفارشات از لیست
- ۹- تکرار مراحل ۲ تا ۸ بدون محدودیت
- ۱۰- خروج از صفحه کاربری

با در نظر گرفتن اطلاعات بالا، باید ساختار سایت را ترسیم و ارتباط بین صفحات و اطلاعات را مشخص نمایید.

اگر بخواهید سیستمی که طراحی می‌کنید، ایده آل و مورد پسند کاربران باشد، باید سعی کنید تمام سرویس‌های فوق را برای کاربر و مدیر سیستم پیاده‌سازی کنید.

۳-۹- طراحی

در این مرحله براساس اطلاعات به دست آمده در مراحل قبلی، ساختار کلی سایت و ارتباط بین اطلاعات و صفحات طراحی می‌گردد.

مدیریت سایت

- تنظیمات سایت
- مدیریت کاربران
- مشتری‌ها
- سفارش کالا

محصولات

• لوازم خانگی

• ورزشی

• لوازم تحریر

• اسباب بازی

• پوشاک

• اتومبیل

درباره ما

• معرفی فروشگاه

• فهرست کالاها و حوزه فعالیتها

ارتباط با ما

• آدرس فروشگاه

• تلفن فروشگاه

• آدرس پست الکترونیک فروشگاه

۹-۴- ثبت Domain و خرید فضا (Host)

در این مرحله از کار بهتر است آدرس Domain و فضای Host مربوط به وب سایت انتخاب و خریداری شود.

فرض کنید با بررسی آدرسهای Domain موجود در اینترنت توسط وب سایتی مانند www.nic.ir و نام فروشگاه و حوزه کاری آن، نام www.saminshop.ir را انتخاب کرده باشید. پس از انتخاب نام، باید از طریق شرکت‌های ارائه دهنده خدمات میزبانی وب سایت، پس از پرداخت هزینه، Domain مورد نظر را ثبت و میزان فضای host مورد نیاز را نیز خریداری کنید.

۹-۵- طراحی گرافیکی و قالب وب سایت

در فصل دوم از جلد اول این کتاب، با تأثیر روانی رنگ‌ها بر روی مخاطب آشنا شدید. با استفاده از این تأثیرات و در نظر گرفتن سلیقه اکثر کاربران برای طراحی ظاهر صفحات، توسط نرم افزارهای گرافیکی از قبیل Photoshop اقدام می‌نمایید.

فرض کنید با در نظر گرفتن تمام استانداردها و موارد مطرح شده در جلد اول، برای صفحه اصلی وب سایت، طرحی مشابه شکل زیر را طراحی کرده باشید :




شکل ۱-۹- قالب سایت

در صورت تمایل می‌توانید بخش‌هایی از صفحه و عناصر موجود در آن را برای افزایش جذابیت و تأثیر روی مخاطب، با استفاده از نرم افزار Flash، متحرک کنید.

مطالعه آزاد

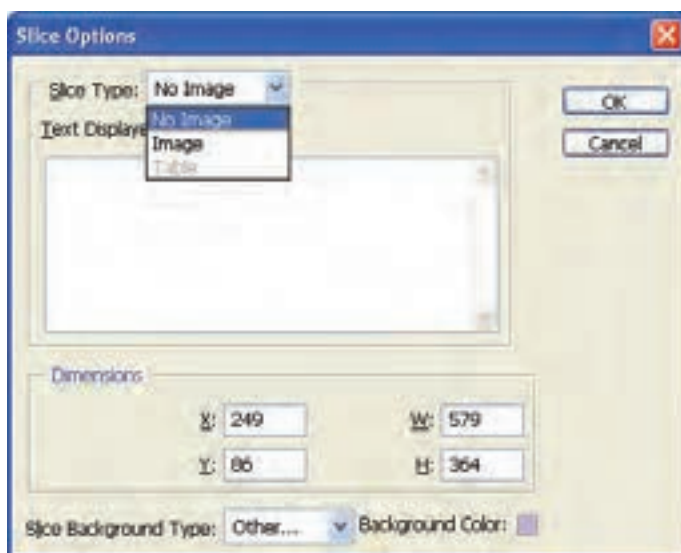
۱-۵-۹- ایجاد صفحه وب توسط نرم افزار Photoshop

طرحی را که برای سایت در نظر گرفته‌اید در نرم افزار Photoshop باز کنید. توسط ابزار  Slice تصویر را با توجه به رنگ و عناصر موجود، به تکه‌های منظم و مختلفی تقسیم کنید.



شکل ۲-۹

پس از تکه تکه کردن تصاویر به این روش، می‌توانید صفحه را به صورت وب ذخیره کنید. نرم‌افزار Photoshop قادر است تصاویر را با استفاده از جدول و تکنیک‌های استاندارد وب، به صورت سازمان‌دهی شده، تولید و ذخیره کند، به این صورت که هر تکه‌ای که توسط ابزار Slice ایجاد شده است، با یک فایل تصویر مجزا ذخیره می‌شود. هنگام تکه تکه کردن تصویر، بهتر است به منظور کاهش حجم فایل و افزایش سرعت بارگذاری آن، بخش‌هایی را که رنگ یکنواختی دارند، به صورت تصویر ذخیره نکنید. برای خارج کردن تکه‌های مورد نظر از حالت تصویر، کافی است روی ناحیه مورد نظر راست کلیک کرده، گزینه Edit Slice Options را انتخاب نمایید. (شکل ۹-۳)



شکل ۹-۳

از کادر باز شو Slice Type گزینه No Image را انتخاب نمایید. به منظور انتخاب رنگ برای ناحیه انتخاب شده، در قسمت Slice Background Type گزینه Other و مقابل Background Color رنگ مورد نظر را برای ناحیه، انتخاب کنید.

تصاویر کوچک به نسبت تصاویر بزرگ، سرعت بارگذاری بالاتری دارند و به همین دلیل سرعت باز شدن صفحات وب با تصاویر کوچک، به مراتب سریع‌تر است.

برای ذخیره صفحه با قالب HTML و ایجاد صفحه وب آماده، از منوی File، گزینه Save For Web را انتخاب کنید.

در پنل سمت چپ، ویژگی‌های مختلف فایل‌های خروجی، از جمله تعداد رنگ و عمق بیتی را

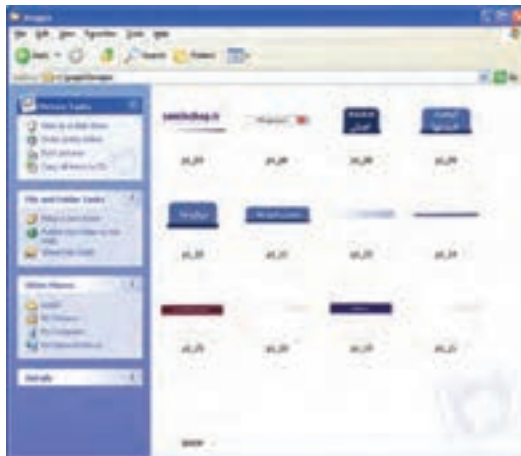
تعیین کنید. (شکل ۹-۴)



شکل ۹-۴

هر چه تعداد رنگ و عمق بیت‌ها کمتر باشد، حجم فایل‌ها کم‌تر می‌شود، اما باید دقت داشته باشید که این کاهش را تا حدودی انجام دهید که کیفیت تصاویر اولیه، از بین نرود. پس از ذخیره صفحه وب در مسیر مورد نظر، خواهید دید که صفحه شما در قالب یک فایل وب ذخیره شده است که تصاویر مربوط به آن نیز در یک پوشه کنار فایل قرار داده شده‌اند. در مسیر فایل‌های تصویر ایجاد شده، برای قطعه‌هایی از صفحه که بدون تصویر ذخیره شده‌اند، یک تصویر به نام spacer با ابعاد 1×1 پیکسل به صورت شفاف ایجاد می‌شود و در جاهای مورد نیاز، بسته به طول و عرض ناحیه، ابعاد این تصویر تغییر داده می‌شود. (شکل ۹-۵)

برای قرار دادن محتویات مورد نظر، از قبیل متن، تصویر و ... باید تصویر spacer.gif را توسط ویرایش‌گر و یا به وسیله نرم افزار Dreamweaver از این ناحیه حذف نمایید.



شکل ۹-۵

اکنون به راحتی می‌توانید این صفحه وب را توسط نرم افزار Dreamweaver باز کرده، و بخش‌های مورد نظر خود را ویرایش کنید. سپس برای استفاده در وب سایت، در مسیر سرویس دهنده WAMP، آن را ذخیره نمایید.

۹-۶- مدیریت کاربران

در تمام وب سایت‌های پویا صفحات کاربری مختلفی برای انواع کاربران در نظر گرفته می‌شود و برای هر کدام از انواع کاربران، فقط امکان ارتباط با بخش‌های مجاز فراهم می‌شود. یعنی باید با در نظر گرفتن تمهیدات لازم از قبیل استفاده از Sessionها، Cookieها و... این امکان را در صفحات مختلف فراهم نمایید.

ممکن است سه نوع کاربر وب سایت را مشاهده کنند :

- ۱- کاربری که مدیر سیستم است.
- ۲- کاربری که به عنوان مشتری قبلاً فرم ثبت نام را پر کرده و کد اشتراک دریافت نموده است.
- ۳- کاربری که تاکنون در وب سایت فروشگاه عضو نشده و به عنوان میهمان وب سایت را مشاهده می‌کند.

بنابراین لازم است در مراحل طراحی، این سه نوع کاربر به طور دقیق متمایز شوند. اگر عمل تشخیص و تفکیک کاربران به خوبی انجام شود، امنیت وب سایت نیز تا حد زیادی فراهم خواهد شد.

۹-۷- بانک اطلاعاتی

با توجه به نیازهایی که در بخش قبل برای انواع کاربران بررسی شد، باید برای طراحی بانک اطلاعاتی اقدام کنید. به طور مثال جدول مشتری و کالا به صورت روبه‌رو طراحی شده است.

مشتری customer :

فیلد	توضیحات
c_id	شناسه اختصاصی به هر مشتری (فیلد کلیدی)
c_name	نام مشتری
c_gender	جنسیت مشتری
c_tel	تلفن تماس با مشتری
c_addr	آدرس مشتری
c_mail	پست الکترونیکی مشتری
c_pass	رمز ورود مشتری
c_detail	جزئیات و توضیحات ضروری

مدیر فروشگاه manager :

فیلد	توضیحات
m_id	شناسه اختصاصی به هر مدیر (فیلد کلیدی)
m_name	نام مدیر
m_gender	جنسیت مدیر
m_tel	تلفن تماس با مدیر
m_addr	آدرس مدیر
m_mail	پست الکترونیکی مدیر فروشگاه
m_pass	رمز ورود مدیر فروشگاه
m_detail	جزئیات و توضیحات ضروری

تمرین : سایر جداول مربوط به موجودیت‌های این سیستم و ارتباط بین این جداول را طراحی نمایید.

نکته : معمولاً برای پیاده‌سازی سیستم‌ها و بررسی موجودیت‌ها، یک یا چند فیلد اضافی برای توسعه احتمالی سیستم در آینده در نظر گرفته می‌شود.

جداول فوق را در MySQL توسط سرویس دهنده WAMP ایجاد کنید.

اگر بخواهید توسط کدهای SQL برای ایجاد بانک اطلاعاتی و جداول فروشگاه، اقدام کنید دستورات مورد نیاز به صورت زیر خواهد بود :

کد ایجاد بانک اطلاعاتی :

```
<?php
$c1 = mysql_connect("localhost","root","");
if (!$c1)
{
    die('Could not connect: ' . mysql_error());
}
// Create database
if (mysql_query("CREATE DATABASE samishop",$c1))
{
    echo "Database created";
}
else
{
    echo "Error creating database: " . mysql_error();
}
mysql_close($c1);
?>
```

کد ایجاد جدول مشتری :

```
<?php
$c1 = mysql_connect("localhost","root","");
if (!$c1)
{
    die('Could not connect: ' . mysql_error());
```

```

}
// Create table
mysql_select_db("saminshop", $c1);
$sql = "CREATE TABLE customer
(
c_id int NOT NULL,
PRIMARY KEY(c_id),
c_name varchar(15),
c_gender varchar(6),
c_tel varchar(11),
c_mail varchar(11),
c_pass varchar(11),
c_addr varchar(200),
c_detail varchar(300),
)";

// Execute query
mysql_query($sql,$con);

mysql_close($c1);
?>

```

تمرین: کدهای لازم برای ایجاد سایر جداول را بنویسید.

۹-۸- طراحی بخش مدیریت سیستم

مدیر سیستم کاربری است که قابلیت‌ها و امکانات مدیریت پایگاه داده فروشگاه را در اختیار دارد. بنابراین باید قابلیت‌های متعددی را برای او در نظر بگیرید. بخشی از قابلیت‌هایی که بهتر است

برای مدیر سیستم در وب سایت در نظر گرفته شود عبارتند از :

- ورود به سیستم (Login)
- درج اطلاعات اجناس
- حذف اجناس
- ویرایش اطلاعات اجناس
- گزارش گیری از اطلاعات اجناس موجود در انبار فروشگاه
- گزارش گیری از اطلاعات مشتریان
- ...

در برخی از وب سایت‌ها برای بخش مدیریت سیستم، یک پوشه جدا ساخته می‌شود و تمام صفحات وب مربوط به مدیر سیستم درون آن قرار می‌گیرند. علاوه بر این می‌توانید به طور مستقیم آنها را در همان پوشه اصلی وب سایت قرار دهید، که در این صورت نام فایل‌های مربوطه را باید طوری انتخاب نمایید که در مراجعات بعدی دچار سردرگمی نشوید.

نکته: به منظور سهولت دسترسی مشتریان، نام index یا default را برای صفحه خانگی بخش مربوط به آنان در نظر بگیرید.

۱-۸-۹- طراحی صفحه ورود مدیر فروشگاه

در مرحله اول برای جلوگیری از ورود افراد غیر مجاز باید کاربری که به عنوان مدیر فروشگاه قصد ورود به سیستم را دارد، کلمه کاربری و رمز ورود خود را وارد کند. می‌توانید صفحه ورود را مشابه شکل زیر طراحی کنید.



شکل ۹-۶

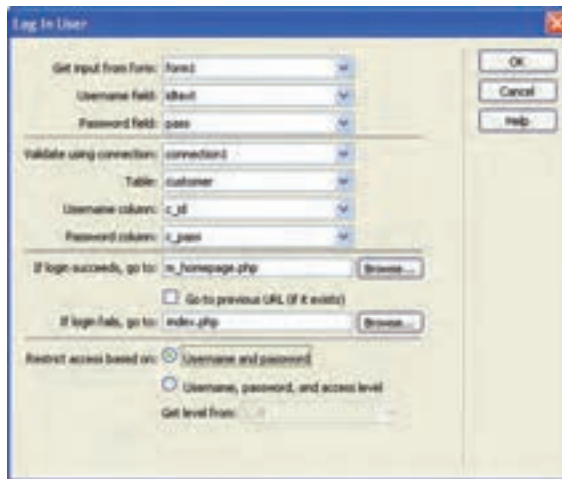
در صورتی که کاربر کلمه کاربری و رمز ورود خود را درست وارد کرده باشد، باید یک جلسه ایجاد کنید که محتوای وارد شده را ذخیره کند تا برای مشاهده سایر صفحات مربوط به مدیر سیستم به کار برده شوند.

به طور مثال اگر متغیرهای مربوط به کلمه کاربری و رمز ورود به ترتیب با عنوانین ADMIN_ USER و ADMIN_PASS باشند کد مربوط به ایجاد و ذخیره جلسه به صورت زیر است :

```
$_SESSION['log'] = ADMIN_USER ;
$_SESSION['pass'] = ADMIN_PASS;
```

در این صفحه فرض می‌کنیم در پایگاه داده، کلمه کاربری، همان فیلد m_id و رمز عبور m_pass باشد. باید اتصال با پایگاه داده برقرار شود، سپس یک Recordset برای ارتباط با جدول Manager ساخته شود.

پس از ساخت فرم شکل ۹-۶، برای درج کدها، می‌توانید به طور مستقیم اقدام به کدنویسی نمایید و یا اینکه از رفتارهای سرویس دهنده استفاده کنید، به این ترتیب که از منوی Insert گزینه Data Object، سپس User Authentication و پس از آن گزینه Log In User را انتخاب نمایید.



شکل ۹-۷

در شکل ۹-۷، مقابل Username field عنوان کادر متنی که بر روی فرم برای دریافت کلمه کاربری قرار داده اید و در قسمت Password field کادر متنی مربوط به درج رمز ورود را وارد کنید. در بخش دوم شکل ۹-۷ نیز عنوان اتصالی که برای پایگاه داده ساخته شده است، پس از آن در قسمت

Table عنوان جدول حاوی کلمه کاربری و رمز ورود، و در قسمت های Username column و Password column به ترتیب فیلدهای کلمه کاربری و رمز ورود کاربر در پایگاه داده را معین نمایید. در کادر متن If login succeeds go to عنوان صفحه‌ای را وارد کنید که در صورت صحت کلمه کاربری و رمز ورود، کاربر باید به آنجا منتقل گردد. در کادر متن If login fails go to عنوان صفحه‌ای را وارد کنید که در صورت عدم صحت کلمه کاربری و رمز ورود، کاربر باید به آنجا منتقل شود. تنظیمات کادر محاوره فوق را انجام دهید و آن را تأیید نمایید.

۲-۸-۹- طراحی صفحه اصلی مدیر فروشگاه

کاربری که به عنوان مدیر فروشگاه کلمه کاربری و رمز ورود به سیستم را به درستی وارد کرده باشد، باید بتواند تمام قابلیت‌هایی که برای مدیر فروشگاه در نظر گرفتیم را مشاهده کند. به همین منظور می‌توانید ابتدا صفحه‌ای طراحی نمایید که به تمام صفحات مربوط به درج کالای جدید، ویرایش یا حذف کالا، ویرایش مشخصات کاربری و ... پیوند داشته باشد.



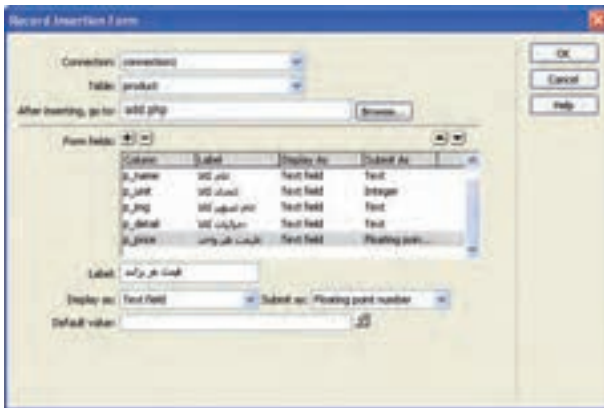
شکل ۸-۹

مدیر فروشگاه با کلیک روی هر یک از عناوین موجود در شکل ۸-۹ باید به صفحات متناظر منتقل شود.

۳-۸-۹- طراحی صفحه درج کالای جدید

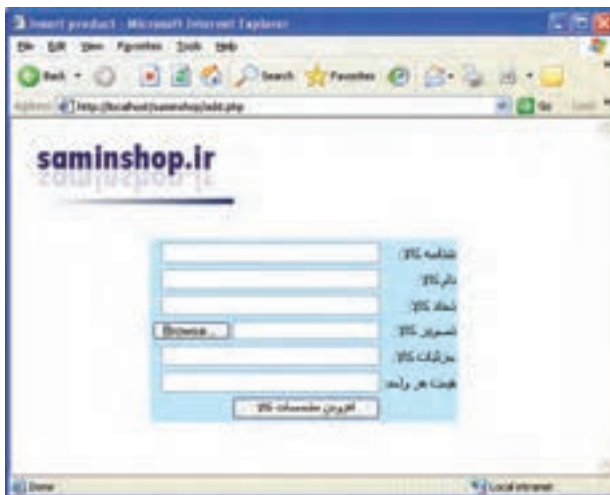
برای درج اطلاعات مربوط به کالای جدید نیز باید در یک صفحه جدید اتصال به پایگاه داده و Recordset مربوط به جدول کالاها ساخته شود.

سپس برای دریافت اطلاعات از مدیر فروشگاه، باید فرم مخصوصی طراحی گردد. به منظور طراحی این فرم، علاوه بر طراحی دستی، می‌توانید از رفتارهای سرویس دهنده نیز بهره بگیرید. از منوی Insert گزینه Data object، سپس Insert record و در زیر منوی مربوط به آن گزینه Record Insertion Form Wizard را انتخاب نمایید.



شکل ۹-۹

در کادر محاوره ۹-۹ تنظیمات مربوط به هر فیلد را انجام دهید و آن را تأیید کنید.



شکل ۹-۱۰

مدیر فروشگاه باید یک پوشه در مسیر وب سایت برای تصویر کالاها موجود قرار دهد، سپس با افزودن کالای جدید و درج تصویر مربوط به آن، نام آن را نیز در فیلد مربوطه در پایگاه داده وارد کند.

۹-۸-۴- طراحی صفحه ویرایش مشخصات کالا

همان گونه که در فصل هشتم اشاره شد، روش های مختلفی را برای انتخاب رکورد مورد نظر کاربر به منظور ویرایش، می توانید در نظر بگیرید، روش انتخاب رکورد به سلیقه طراح وابسته است، در اینجا به طور نمونه یک جدول برای نمایش تمام اطلاعات موجود در بانک اطلاعاتی طراحی می کنیم که در کنار هر فیلد آن پیوندی برای ویرایش اطلاعات رکورد اضافه می نمایم.

برای طراحی عملیات ویرایش، یک صفحه وب جدید درج کرده و آن را با نام مشخصی ذخیره کنید (به طور مثال با نام update.php).

در صفحه update.php یک فرم، یک فیلد مخفی، کادرهای متنی نمایش اطلاعات رکورد انتخاب شده و یک دکمه نیز درج نمایید.



شکل ۹-۱۱

در یک صفحه وب دیگر، جدولی برای نمایش اطلاعات کالاها طراحی کنید و یک ستون اضافی برای ویرایش در نظر بگیرید و آن را به صفحه update.php پیوند دهید و با توجه به توضیحات فصل ۸، اگر بخواهید در جدول product حذف اطلاعات بر اساس فیلد شناسه کالا انجام شود، باید به جای آدرس URL آن کد زیر را درج نمایید :

```
update.php?record_in=<?php echo $row_recordset1['p_id']; >
```

کلاس جزئیات	کلاس نام	کلاس شناسه
Recordset1.p_detail	Recordset1.p_name	Recordset1.p_id

شکل ۹-۱۲

در Dreamweaver حالت Live Data را برای نمایش اطلاعات جدول، فعال کنید. در مرحله بعد، باید ابتدا اطلاعات رکورد مورد نظر برای ویرایش، نمایش داده شود. بنابراین، در صفحه update.php علاوه بر فیلد مخفی و دکمه، کادرهای متنی برای نمایش اطلاعات مربوط به رکورد انتخاب شده برای ویرایش طراحی کرده، سپس یک Recordset جدید ایجاد نمایید.

در صفحه update.php به هر کدام از کادرهای متنی فرم، فیلد مربوطه را از Recordset بی که ساخته‌اید، انتساب دهید.

سایر تنظیمات مربوط به ویرایش رکوردها و رفتار سرویس دهنده Update Record را مطابق توضیحات ارائه شده در فصل هشتم، تنظیم کنید.

برای امکان مشاهده لیست کالاها موجود در صفحات دیگر نیز از Recordset Navigation Bar استفاده نمایید.

از منوی Insert گزینه Data Object، سپس Recordset Paging و از زیر منوی مربوط به آن Recordset Navigation Bar را انتخاب نمایید.

پس از تنظیم و تأیید کادر Recordset Navigation Bar، با اجرای صفحه وب فوق در مرورگر مشاهده خواهید نمود که با کلیک روی گزینه‌های مربوط به رکوردهای بعدی، تمام رکوردهای موجود در صفحات دیگر نیز قابل مشاهده هستند.

۹-۸-۵- طراحی صفحه حذف کالا

برای فراهم کردن امکان حذف کالا نیز، می‌توانید با توجه به توضیحات ارائه شده در فصل هشتم، پس از فراهم کردن امکان انتخاب رکورد مورد نظر کاربر، برای حذف آن از پایگاه داده اقدام نمایید.

۹-۸-۶- طراحی صفحه گزارشات فروشگاه

در هر فروشگاه، برای بررسی وضعیت موجودی انبار، وضعیت فروش، وضعیت مشتریان و ... لازم است امکان گزارش‌گیری از اطلاعات موجود در پایگاه داده فراهم شود.

نحوه گزارش‌گیری نیز به سلیقه طراح وابسته است و باید حداکثر رضایت کاربر را به دنبال داشته باشد.

به‌طور ساده برای طراحی این صفحات می‌توانید جدول‌های نمایش اطلاعات را در صفحات وب مربوط به هر یک، طراحی کنید.

۹-۹- طراحی بخش مشتریان

مشتریانی که از وب سایت فروشگاه دیدن می‌کنند، ممکن است خود دو دسته باشند، دسته اول کاربرانی هستند که هنوز اقدام به پر کردن فرم عضویت ننموده‌اند. اما دسته دوم کاربرانی هستند که قبلاً با ثبت عضویت خود، شناسه کاربری دریافت کرده‌اند.

برخی از قابلیت‌هایی که بهتر است برای مشتری میهمان در نظر گرفته شود عبارتند از :

– درخواست عضویت و ثبت اطلاعات کاربری

– مشاهده لیست کالاها موجود در فروشگاه

– ...

امکانات قابل طراحی برای کاربران عضو نیز به صورت زیر است :

– ورود به سیستم (Login)

– مشاهده لیست کالاها موجود در فروشگاه

– سفارش خرید

– حذف اجناس سفارش داده شده (قبل از تأیید نهایی)

– ویرایش سفارشات خرید (قبل از تأیید نهایی)

– تغییر رمز ورود به سیستم

– ویرایش اطلاعات کاربری

– ...

۹-۹-۱- ورود کاربر

همان‌طور که قبلاً اشاره شد در اغلب وب سایت‌ها، صفحه‌ای که به عنوان صفحه خانگی کاربر در نظر گرفته شده است، با عنوان index و یا default ذخیره می‌شود. این امر موجب می‌شود کاربر بلافاصله پس از درج نام وب سایت، بتواند صفحه کاربری خود را مشاهده نماید.

بهرتر است در صفحه اول، کاربر عضو به راحتی بتواند محل وارد کردن کلمه کاربری به همراه رمز ورود را پیدا کند. همچنین کاربرانی که هنوز عضو فروشگاه نشده‌اند، بدون دردسر با کلیک روی بخش مربوط به عضویت، نسبت به پر کردن فرم مربوطه اقدام نمایند.

طراحی بخش ورود به سیستم برای مشتریان همانند ورود به سیستم برای مدیر فروشگاه است. پس از ورود کاربر، دیگر لازم نیست بخش مربوط به ورود کلمه کاربری و رمز، نشان داده شود و به جای آن بهتر است لیست سفارشات کاربر جایگزین گردد.



شکل ۱۳-۹

پس از اینکه صحت کلمه کاربری و رمز تأیید شد، یک جلسه برای نگهداری مشخصات کاربر طراحی کنید. و آن را در ابتدای تمام صفحات مربوط به مشتری درج کنید.

شرط بررسی غیر مجاز بودن دسترسی کاربر به صورت زیر است:

```
if (!isset($_SESSION['log']) || !isset($_SESSION['pass']))
```

```
{
```

دستورات یا پیغامی که در صورت غیر مجاز بودن دسترسی باید نمایش داده شوند، در این قسمت قرار می‌گیرند.

```
}
```

۹-۹-۲- عضویت کاربر

به منظور عضویت در وب سایت فروشگاه، می‌توانید فرمی مشابه شکل ۹-۱۴ طراحی کنید.



شکل ۹-۱۴

۹-۹-۳- مشاهده لیست کالاها

برای طراحی امکان مشاهده لیست کالاهای موجود در فروشگاه نیز مانند بخش گزارش گیری کالاها برای مدیر فروشگاه اقدام نمایید.

۹-۹-۴- سفارش خرید

برای سازماندهی بهتر عملیات سفارش و خرید کالا، می‌توانید یک جدول مجزا در پایگاه داده در نظر بگیرید و با انتخاب هر کالا توسط کاربر، شناسه آن را در پایگاه داده درج کنید و در بخش لیست سفارشات، عنوان تمام کالاهایی که توسط کاربر انتخاب شده‌اند را درج کرده، در کنار هر یک، امکان حذف از لیست را فراهم نمایید.

لیست سفارشات من	
عنوان	قیمت (تومان)
ماشین لباسشویی	۵۰۰ حذف
عروسک	۱۰ حذف

شکل ۹-۱۵

۱۰-۹- عضویت کاربر

اگر فروشگاه الکترونیکی که برای طراحی آن اقدام نموده‌اید، بزرگ باشد، بهتر است فروشندگان و مدیریت فروشگاه متمایز گردند. این امر علاوه بر قابلیت مدیریت متمرکز، امکان اشتباه در گزارشات و اطلاعات اجناس را کاهش می‌دهد. در این صورت، برای فروشندگان می‌توانید بخش تحویل کالا به مشتری را طراحی نمایید، به این منظور فروشندگان باید بتوانند از اطلاعات مربوط به سفارش کالا گزارش تهیه کنند.

منابع

www.w3school.com

www.php.net

www.academictutorials.com

