

سرشناسه	اسپانجارز ایمار - ۱۹۷۱ م - Spainjaars, Imar
عنوان و نام پدیدآور	راهنمای جامع مایکروسافت ASP.NET 4.5.1
نویسنده: ایمار اسپانجارز؛ مترجم: امیر احسان رفائی	
مختصات نشر	تهران: مهرگان قلم
مختصات ظاهری	۷۷۰ صفحه: مصور
شابک	۹۷۸-۶۰۰-۶۲۳۶-۵۹-۹
وضعیت فهرست‌نویسی: قیاً	
عنوان اصلی	Beginning asp.net 4.5.1: in c# and vb, 2014
موضوع	صفحه‌های مرور فعال - وب سایت‌ها طراحی توسعه مایکروسافت دانش فنی ویرگ
شماره افزوده	رفائی: امیر احسان ۱۳۵۹، مترجم
ردیف‌های کنگره	۱۳۹۳ کف ۵ پ ۶۴ ص ۸۸۸۵ / ۵ (۵) ۲۸
ردیف‌های دیوبی	۰۰۵/۲۷۶
شماره کتاب‌شناسی ملی:	۲۵۳۵۳۳۵

نام کتاب: راهنمای جامع مایکروسافت ASP.NET 4.5.1

نویسنده: ایمار اسپانجارز

ترجمه: امیر احسان رفائی

ناشر: مهرگان قلم

نوبت چاپ دوم: ۱۴۰۲

شابک: ۹۷۸-۶۰۰-۶۲۳۶-۵۹-۹

تیراژ: ۵۰ جلد

قیمت: ۵۳۰۰۰ تومان

تکثیر نام با قلمنی از اثر بدون
مجاز کسری نشر ممنوع است و
مخالفتان به موجب قانون حمایت
حقوق مولفان، مصنفان و
هم‌مندان تحت پیگرد قانونی قرار
خواهند گرفت.



Mehreganeh Ghalam

مرکز یحیی

انتشارات مهرگان قلم

تهران - میدان انقلاب - خیابان کارگر جنوبی - بعد از وحیدمظفری - کوچه گشتاسب - پلاک ۶ - طبقه ۲

تلفن: ۰۹۱۳۳۱۳۶۰۵۸ - ۶۶۲۷۶۲۱۷ - ۶۶۲۷۷۲۲۴-۵

وبسایت: www.mehreganeghalam.com

۱۳	فصل ۱: شروع به کار با ASP.NET 4.5.1
۱۵	Microsoft Visual Studio Express For Web
۱۵	دریافت ویژوال استودیو
۱۵	نصب ویژوال استودیو اکسپرس برای وب
۱۷	ایجاد اولین وب سایت ASP.NET 4.5.1
۲۲	آشنایی با ASP.NET 4.5.1
۲۴	آشنایی با HTML
۲۴	عناصر HTML و تگ‌ها
۲۸	خاصیت‌های HTML
۲۸	توضیحات HTML
۲۸	قوانین HTML5
۲۸	بستن عناصر
۲۹	استفاده از خاصیت‌ها
۲۹	تودرتو کردن صحیح عناصر
۲۹	اضافه کردن معرفی DOCTYPE به صفحه
۳۰	نگاهی به علائم ASP.NET
۳۰	نگاهی به IDE
۳۰	ناحیه توسعه اصلی
۳۱	انتخاب پروفایل برنامه‌نویسی
۳۱	منوی اصلی
۳۲	پنجره Notification
۳۲	قسمت نوار ابزار
۳۲	Toolbox
۳۲	Solution Explorer
۳۳	Server Explorer
۳۳	Team Explorer
۳۳	شبکه Properties
۳۴	پنجره سند
۳۴	صفحه استارت
۳۹	پنجره‌های حاوی اطلاعات
۳۹	Error List
۳۹	پنجره Output
۳۹	پنجره Find Results
۴۰	سفارشی کردن IDE
۴۰	بازچینی پنجره‌ها



۴۱	 Toolbox تغییر
۴۳	 شخصی کردن نوارهای ابزار
۴۵	 تغییر میانبرهای صفحه کلید
۴۶	 برگرداندن تغییرات اعمال شده
۴۷	 یک برنامه مثال

فصل ۲: ایجاد یک وب سایت ASP.NET ۵۱

۵۲	 ایجاد وب سایت‌ها با ویژوال استودیو ۲۰۱۳
۵۲	 انواع پروژه مختلف
۵۳	 Web Application Projects
۵۳	 Web Site Projects
۵۳	 انتخاب بین Web Application Projects و Web Site Projects
۵۳	 انتخاب الگوی سایت مناسب
۵۴	 ASP.NET Web Site (Razor v2 / Razor v3)
۵۴	 ASP.NET Empty Web Site
۵۵	 ASP.NET Dynamic Data Entities Web Site
۵۵	 WCF Service
۵۵	 Creating and Opening a New Website
۵۵	 ایجاد وب‌سایت‌های جدید
۵۸	 باز کردن وب‌سایت‌های موجود
۵۹	 کار کردن با فایل‌ها در وب سایت
۵۹	 انواع فایل زیاد یک وب سایت ASP.NET
۶۰	 فایل‌های وب
۶۲	 فایل‌های کد
۶۳	 فایل‌های داده‌ای
۶۳	 اضافه کردن فایل‌های موجود
۶۴	 سازماندهی سایت
۶۵	 انواع فایل خاص
۶۶	 کار کردن با فرم‌های وب (Web Forms)
۶۶	 نماهای مختلف روی وب‌فرم‌ها
۶۸	 انتخاب بین Code Behind و صفحه‌ها با کد درون خطی
۷۴	 اضافه کردن کد مارک‌آپ به صفحه
۷۵	 قرار دادن متن و اعمال فرمت روی آن
۷۸	 اضافه کردن جدول‌ها و دیگر مارک‌آپ‌ها
۸۳	 وصل کردن صفحات

فصل ۳: طراحی صفحات وب ۸۷

۸۸	 چرا به CSS نیاز است
۸۸	 مشکل اعمال فرمت با استفاده از HTML



۸۹	چگونگی حل مشکل توسط CSS
۹۰	مقدمه‌ای بر CSS
۹۴	CSS - زبان
۹۴	برگ شیوه
۹۵	انتخاب‌گرها
۹۷	انتخاب‌گر کلاس
۹۹	مشخصات
۱۰۰	مقادیر
۱۰۱	مختصرنویسی
۱۰۲	مدل جعبه‌ای CSS
۱۱۰	اضافه کردن CSS به صفحه وب
۱۱۱	انتخاب بین برگ‌های شیوه بیرونی، جاسازی شده و درون خطی
۱۱۲	کارکردن با CSS در ویژوال استودیو
۱۱۳	استفاده از ویرایشگر CSS
۱۱۹	برگ‌های شیوه جاسازی شده و درون خطی
۱۲۶	اعمال شیوه‌ها
۱۲۸	مدیریت شیوه‌ها

فصل ۴: کار کردن با کنترل‌های سرور ASP.NET ۱۳۳

۱۳۴	آشنایی با کنترل‌های سرور
۱۳۹	نگاهی نزدیک‌تر به کنترل‌های سرور ASP.NET
۱۳۹	تعریف کنترل‌ها درون صفحه
۱۳۹	مشخصات معمول برای همه کنترل‌ها
۱۴۲	انواع کنترل‌ها
۱۴۳	کنترل‌های Standard
۱۵۱	کنترل‌های محفظه‌ای
۱۵۵	استفاده از کنترل Wizard
۱۶۰	دیگر کنترل‌های استاندارد
۱۶۱	کنترل‌های HTML
۱۶۲	کنترل‌های Data
۱۶۲	کنترل‌های Validation
۱۶۲	کنترل‌های Navigation
۱۶۳	کنترل‌های Login
۱۶۳	Ajax Extensions
۱۶۳	WebParts
۱۶۳	Dynamic Data
۱۶۳	موتور حالت ASP.NET
۱۶۴	چگونگی عملکرد موتور حالت
۱۶۹	همه کنترل‌ها دارای حالت نمایش نیستند



نکته‌ای درباره حالت نمایش و عملکرد..... ۱۷۰

فصل ۵: برنامه‌نویسی صفحات وب ASP.NET ۱۷۳

مقدمه‌ای بر برنامه‌نویسی ۱۷۴

انواع داده و متغیرها ۱۷۵

تبدیل و کست کردن انواع داده ۱۷۹

استفاده از آرایه‌ها و مجموعه‌ها ۱۸۲

دستورات ۱۸۷

عملگرها ۱۸۸

تصمیم‌گیری ۱۹۸

ساختارهای If Else If و Else If ۱۹۸

ساختارهای Select Case/ switch ۲۰۱

حلقه‌ها ۲۰۸

حلقه For ۲۰۸

حلقه For Each/foreach ۲۱۰

حلقه While ۲۱۰

خارج شدن دائمی از حلقه‌ها ۲۱۲

سازماندهی کد ۲۱۳

متدها: توابع و ساب‌روتین‌ها ۲۱۳

پوشه App_Code ۲۱۶

سازماندهی کد با فضاها نام ۲۲۱

نوشتن توضیحات ۲۲۴

اصول شی‌گرایی ۲۲۷

عبارت‌های مهم شی‌گرایی ۲۲۸

اشیا ۲۲۸

کلاس‌ها ۲۲۹

متدها ۲۳۴

سازنده‌ها ۲۳۵

وراثت ۲۳۸

تعیین کننده‌های دسترسی ۲۴۱

رویدادها ۲۴۳

فصل ۶: ایجاد وب‌سایت‌هایی با ظاهر یکدست ۲۴۷

ایجاد ظاهر یکدست با استفاده از صفحه‌های مستر ۲۴۸

ایجاد صفحه‌های مستر ۲۵۱

ایجاد صفحه‌های محتویات ۲۵۳

نگاهی نزدیک‌تر به صفحات مستر ۲۵۷

تودرتو کردن صفحات مستر ۲۵۸



۲۵۸		خطار درباره صفحات مستر
۲۵۹		استفاده از یک صفحه مبنای مرکزی
۲۶۱		مقدمه‌ای بر چرخه عمر صفحه ASP.NET
۲۶۲		پیاده‌سازی صفحه مبنا
۲۶۸		ایجاد الگوهای صفحه قابل استفاده مجدد
۲۷۲		تم‌ها
۲۷۳		انواع تم
۲۷۴		انتخاب بین تم و StyleSheetTheme
۲۷۴		اعمال تم‌ها
۲۷۹		بسط دادن تم‌ها
۲۸۲		تغییر دینامیک تم‌ها
۲۹۱		پوسته‌ها
۲۹۲		ایجاد یک فایل پوسته
۲۹۴		پوسته‌های دارای نام
۲۹۵		غیرفعال کردن تم برای کنترل‌های خاص

فصل ۷: هدایت درون صفحات ۲۹۷

۲۹۸		روش‌های مختلف برای حرکت درون سایت
۲۹۹		آدرس‌های نسبی و مطلق
۳۰۲		اسناد پیش‌فرض
۳۰۲		استفاده از کنترل‌های هدایتی
۳۰۳		معماری کنترل‌های هدایتی
۳۰۴		بررسی فایل Web.sitemap
۳۰۴		عناصر اصلی فایل Web.sitemap
۳۰۶		استفاده از کنترل Menu
۳۰۷		استفاده از حالت رندر
۳۰۸		ایجاد یک نسخه ساده از کنترل منو
۳۱۱		اعمال شیوه روی کنترل منو
۳۱۶		استفاده از کنترل TreeView
۳۲۰		استفاده از کنترل SiteMapPath
۳۲۳		مسیریابی
۳۲۳		تنظیم URLهای بدون پسوند
۳۲۶		مواردی که برای URLهای بدون پسوند باید در نظر بگیرید
۳۲۶		هدایت با کدنویسی
۳۲۶		هدایت کلاینت به یک صفحه متفاوت
۳۳۰		هدایت‌های مبتنی بر سرور

فصل ۸: کنترل‌های کاربری ۳۳۳

۳۳۴		آشنایی با کنترل‌های کاربری
-----	-------	----------------------------



۳۳۵	ایجاد کنترل‌های کاربری
۳۳۸	اضافه کردن کنترل‌های کاربری به یک صفحه محتویات یا صفحه مستر
۳۴۲	ثبت سطح سایت کنترل‌های کاربری
۳۴۳	چند اخطار درباره کنترل‌های کاربری
۳۴۳	آشنایی با IDهای کلاینت و مدیریت آن‌ها
۳۴۴	آشنایی با ClientIDMode
۳۴۵	اضافه کردن منطق به کنترل‌های کاربری
۳۴۵	ایجاد انواع داده برای مشخصات
۳۵۲	پیاده‌سازی مشخصات حالت نمایش
۳۵۹	موارد مهم در حالت نمایش

فصل ۹: تأیید اعتبار ورودی کاربر ۳۶۱

۳۶۲	دریافت داده‌ها از کاربر
۳۶۴	تأیید اعتبار ورودی کاربر در وب‌فرم‌ها
۳۶۴	کنترل‌های تأیید اعتبار ASP.NET
۳۶۵	یک هشدار برای تأیید اعتبار روی کلاینت
۳۶۵	استفاده از کنترل‌های تأیید اعتبار
۳۶۸	استفاده از انواع داده HTML5
۳۷۱	کنترل‌های تأیید اعتبار استاندارد
۳۷۲	تفاوت میان مشخصات Text و ErrorMessage
۳۷۳	کنترل RangeValidator
۳۷۳	کنترل RegularExpressionValidator
۳۷۴	کنترل CompareValidator
۳۷۸	کنترل‌های CustomValidator و ValidationSummary
۳۸۵	آشنایی با تأیید اعتبار درخواست
۳۸۶	پردازش داده‌ها در سرور
۳۸۶	ارسال ایمیل از وب سایت
۳۸۷	پیکربندی وب سایت برای ارسال ایمیل
۳۸۹	ایجاد پیغام‌های ایمیل
۳۹۳	خواندن از فایل‌های متنی

فصل ۱۰: AJAX در ASP.NET ۴۰۱

۴۰۲	آشنایی با Ajax
۴۰۴	استفاده از Ajax در ASP.NET برای پروژه‌ها
۴۰۴	ایجاد صفحات بدون چشم‌زدن
۴۰۵	کنترل UpdatePanel
۴۰۸	نگاهی نزدیک‌تر به UpdatePanel
۴۰۸	مشخصات معمول UpdatePanel



۴۰۹	مشکلات UpdatePanel
۴۰۹	کنترل ScriptManager
۴۱۱	فراهم آوردن فیدبک برای کاربران
۴۱۱	کنترل UpdateProgress
۴۱۶	کنترل Timer
۴۱۷	استفاده از سرویس‌های وب با وب‌سایت‌های Ajax
۴۱۷	سرویس‌های وب
۴۱۸	آشنایی با WCF
۴۱۹	فراخواندن سرویس‌ها از کد مبتنی بر کلاینت
۴۲۱	مبادله اشیا پیچیده با WCF
۴۲۴	ایجاد سرویس‌های وب
۴۲۸	پیگر بندی ScriptManager

فصل ۱۱: استفاده از JQuery ۴۳۵

۴۳۶	مقدمه‌ای بر JQuery
۴۳۷	آشنایی با NuGet
۴۳۸	استفاده از پنجره Manage Nuget Packages
۴۳۹	استفاده از Package Manager Console
۴۴۱	انتخاب مکان برای ارجاع JQuery
۴۴۲	روش‌های مختلف برای شامل کردن کتابخانه JQuery
۴۴۵	دستورالعمل نگارش JQuery
۴۴۶	عملکردهای اصلی JQuery
۴۴۷	انتخاب موارد با JQuery
۴۴۷	انتخاب گره‌های ساده
۴۴۷	انتخاب گره‌های جهانی
۴۴۷	انتخاب گر عنصر
۴۴۸	انتخاب گر ID
۴۴۸	انتخاب گر کلاس
۴۴۸	انتخاب گره‌های گروهی و ترکیب شده
۴۵۰	فیلترهای ساده
۴۵۳	فیلترهای پیشرفته
۴۵۴	تغییر DOM با JQuery
۴۵۴	متدهای CSS
۴۵۶	اداره کردن رویدادها
۴۵۷	عملکردهای مختلف JQuery
۴۵۸	ایجاد جلوه با JQuery
۴۶۴	تأیید اعتبار با JQuery

**فصل ۱۲: آشنایی با پایگاه داده ۴۶۹**

۴۷۰	پایگاه داده چیست.....
۴۷۲	انواع مختلف پایگاه‌های داده رابطه‌ای.....
۴۷۲	نصب SQL Server 2012 Express.....
۴۷۲	استفاده از SQL برای کار کردن با داده‌های پایگاه داده.....
۴۷۶	دریافت و اداره داده‌ها با SQL.....
۴۷۷	خواندن داده‌ها.....
۴۷۷	انتخاب داده‌ها.....
۴۷۸	فیلتر کردن داده‌ها.....
۴۷۹	مرتب کردن داده‌ها.....
۴۸۴	وصل کردن داده‌ها.....
۴۸۸	ایجاد داده‌ها.....
۴۸۸	به روز کردن داده‌ها.....
۴۸۹	حذف داده‌ها.....
۴۹۲	ایجاد جدول.....
۴۹۲	انواع داده در SQL Server.....
۴۹۴	آشنایی با کلید اصلی و هویت‌ها.....
۴۹۹	ایجاد رابطه بین جدول‌ها.....

فصل ۱۳: نمایش و به روزرسانی داده‌ها..... ۵۰۵

۵۰۶	کنترل‌های داده‌ای.....
۵۰۶	کنترل‌های لیست.....
۵۰۸	کنترل‌های تک موردی.....
۵۰۹	کنترل‌های ایجاد کنند صفحه.....
۵۰۹	کنترل‌های منبع داده.....
۵۱۰	دیگر کنترل‌های داده‌ای.....
۵۱۰	کار کردن کنترل‌های منبع داده و اتصال داده‌ای با یکدیگر.....
۵۱۰	نمایش و ویرایش داده‌ها با GridView.....
۵۱۶	قرار دادن داده‌ها با DetailsView.....
۵۱۸	ذخیره کردن رشته‌های اتصال در Web.config.....
۵۲۰	فیلتر کردن داده‌ها.....
۵۲۷	تغییر ظاهر کنترل‌های داده‌ای.....
۵۲۷	پیکربندی ستون‌ها یا فیلدهای کنترل‌های اتصال داده‌ای.....
۵۳۴	به روز کردن و قرار دادن داده‌ها.....
۵۳۴	استفاده از DetailsView برای قرار دادن و به روزرسانی داده‌ها.....

فصل ۱۴: فریم‌ورک ADO.NET Entity و LINQ ۵۵۳

۵۵۴	آشنایی با LINQ.....
-----	---------------------



۵۵۵	 LINQ برای اشیا
۵۵۶	 LINQ برای XML
۵۵۶	 LINQ به ADO.NET
۵۵۶	 آشنایی با فریم‌ورک ADO.NET Entity
۵۵۷	 نگاشت یک مدل داده‌ای به یک مدل شی
۵۶۵	 آشنایی با روش استفاده از کوئری
۵۶۵	 عملگرهای کوئری استاندارد
۵۶۵	 عملگر Select
۵۶۶	 عملگر From
۵۶۶	 عملگر Order By
۵۶۶	 عملگر Where
۵۶۷	 عملگرهای Count و Average, Max, Min, Sum
۵۶۷	 عملگرهای SkipWhile و TakeWhile, Skip, Take
۵۶۸	 عملگرهای SingleOrDefault و Single
۵۶۹	 عملگرهای LastOrDefault و Last, FirstOrDefault, First
۵۷۰	 شکل دادن به داده‌ها با انواع ناشناس
۵۷۷	 استفاده از اتصال مدل با کوئری‌های LINQ
۵۷۷	 آشنایی با اتصال مدل
۵۸۵	 آشنایی با کنترل ListView
۵۹۹	 استفاده از کنترل‌های اتصال داده دارای نوع قوی
۶۰۸	 آشنایی با کنترل DataPager

فصل ۱۵: کار کردن با داده‌ها – مباحث پیشرفته ۶۱۷

۶۱۸	 اعمال فرمت روی کنترل‌ها با استفاده از شیوه‌ها
۶۲۰	 آشنایی با شیوه‌ها
۶۲۴	 ترکیب شیوه‌ها، تم‌ها و پوسته‌ها
۶۲۹	 اداره رویدادها
۶۲۹	 نگاهی دوباره به چرخه عمر صفحه و کنترل
۶۳۶	 چرخه زندگی و رویدادها در کنترل‌های داده‌ای
۶۴۳	 اداره خطاهایی که در کنترل‌های منبع داده رخ می‌دهند
۶۴۸	 کدنویسی دسترسی به داده‌ها

فصل ۱۶: امنیت در وب سایت ASP.NET ۶۶۱

۶۶۲	 آشنایی با سرویس‌های برنامه ASP.NET
۶۶۳	 آشنایی با کنترل‌های ورود
۶۷۰	 کنترل‌های Login
۶۷۰	 کنترل Login
۶۷۲	 کنترل LoginView
۶۷۳	 کنترل LoginStatus



۶۷۴		LoginName	کنترل
۶۷۷		CreateUserWizard	کنترل
۶۸۰		PasswordRecovery	کنترل
۶۸۱		ChangePassword	کنترل
۶۸۳			پیکربندی برنامه وب
۶۸۷		Role Manager	سرویس
۶۸۸		Role Manager	پیکربندی
۶۸۸		WSAT	مدیریت کاربران با
۶۹۳			پیکربندی برنامه وب برای کار کردن با نقش‌ها
۶۹۷			بررسی نقش‌ها با کدنویسی

فصل ۱۷: شخصی کردن وب سایت‌ها ۷۰۳

۷۰۴			پیکربندی یک پروفایل
۷۰۴			ایجاد مشخصات پروفایل ساده
۷۰۵			ایجاد گروه‌های پروفایل
۷۰۶			استفاده از انواع داده غیراستاندارد
۷۱۱			استفاده از پروفایل

فصل ۱۸: اداره استثناها ۷۳۳

۷۳۴			انواع خطاها
۷۳۴			خطاهای گرامری
۷۳۶			خطاهای منطقی
۷۳۶			خطاهای زمان اجرا
۷۳۶			دریافت و اداره استثناها
۷۴۷			اداره جهانی خطا و صفحه‌های خطای دلخواه

فصل ۱۹: آنلاین کردن سایت ۷۵۵

۷۵۶			آماده کردن وب سایت برای انتشار
۷۵۶			از تنظیمات وارد شده با دست دوری کنید
۷۵۷		Web.config	فایل
۷۵۷			گرامر عبارتی
۷۵۹		WebConfigurationManager	کلاس
۷۶۵			آماده شدن برای انتشار وب سایت
۷۶۵			انتشار سایت

PDF Compressor Free Version

فصل ۱

شروع به کار با ASP.NET 4.5.1





از اولین توزیع داتنت فریم‌ورک نسخه 1.0 در اوایل ۲۰۰۲، مایکروسافت تلاش زیادی روی ASP.NET کرده است که بخشی از فریم‌ورک داتنت می‌اشد که از آن برای ایجاد برنامه‌های وب غنی استفاده می‌کنید. اولین انتشار به معنی تغییرات بسیار بزرگی نسبت به تکنولوژی قدیمی مایکروسافت برای ایجاد وب سایت‌ها بود که با نام Active Server Pages (ASP) شناخته می‌شد که امروزه از آن با نام ASP کلاسیک یاد می‌شود. با معرفی ASP.NET نسخه 1.0 و ویژوال استودیو داتنت توسعه‌دهندگان مزایای زیادی نسبت به ASP کلاسیک به دست آوردند:

- ◀ یک جداسازی مناسب بین ارائه و کد فراهم آمد. در ASP کلاسیک، منطق برنامه شما اکثراً با HTML صفحه به هم می‌ریخت و کار تغییر دادن صفحه در آینده را سخت می‌کرد.
- ◀ یک مدل توسعه که بسیار به روش برنامه‌نویسی برنامه‌های دسکتاپ نزدیک است، در اختیار توسعه دهندگان قرار گرفته است. این باعث شده تا بسیاری از برنامه‌نویسان ویژوال بیسیک دسکتاپ بتوانند به سراغ برنامه‌های وب بروند.
- ◀ یک ابزار توسعه غنی که ویژوال استودیو داتنت نامیده می‌شود که به توسعه‌دهندگان امکان ایجاد و کدنویسی برنامه‌های وب را به صورت بصری می‌دهد.
- ◀ انتخاب بین تعدادی زبان شی‌گرا (OOP).
- ◀ دسترسی به فریم‌ورک داتنت که برای اولین بار به این معنی بود که توسعه دهندگان دارای یک روش آسان و یکدست برای دسترسی به بسیاری از امکانات پیشرفته برای کار کردن با پایگاه‌های داده، فایل‌ها، ایمیل، ابزارهای شبه و موارد دیگر شدند.

بعد از اولین توزیع در سال ۲۰۰۲، مایکروسافت نسخه دیگری از فریم‌ورک داتنت (1.1 .NET) را به همراه IDE توسعه (که ویژوال استودیوی ۲۰۰۳ نامیده می‌شد) ارائه کرد. بسیاری از کاربران به این به عنوان یک سرویس‌پک برای نسخه اولیه نگاه می‌کردند با این که بهبودهای بسیار زیادی در فریم‌ورک و ابزارهای طراحی فراهم آورده بود. در نوامبر ۲۰۰۵، ویژوال استودیوی ۲۰۰۵ و ASP.NET نسخه 2.0 ارائه شد. مایکروسافت این محصول را به مقدار زیادی بهبود داده و امکانات و ابزارهای بسیاری را به آن اضافه کرده تا پیچیدگی نسخه اول ASP.NET را کاهش دهد. ویزاردها و کنترل‌های هوشمند جدید امکان کاهش نیازمندی‌های کد برای ایجاد یک برنامه، کاهش منحنی یادگیری برای بسیاری از توسعه‌دهندگان تازه‌کار و افزایش خلاقیت را فراهم آوردند.

در نوامبر ۲۰۰۷، مایکروسافت ویژوال استودیوی ۲۰۰۸ و ASP.NET نسخه 3.5 را ارائه کرد. بعد از آن نیز ویژوال استودیوی ۲۰۱۲ و ASP.NET نسخه 4 در مارس ۲۰۱۲ و سپس ویژوال استودیوی ۲۰۱۲ و ASP.NET نسخه 4.5 در سال ۲۰۱۲ ارائه شد. هر نسخه بهبودهای بسیار زیادی را ارائه کردند.

نسخه کنونی یعنی ویژوال استودیوی ۲۰۱۳ و ASP.NET 4.5.1 بر مبنای ویژوال استودیوی موفق ۲۰۱۲ و ASP.NET 4.5 توسعه پیدا کرده و امکانات بسیاری را به آن اضافه کرده است.

سایت ساده همراه این کتاب و همه مثال‌ها با ویژوال استودیو اکسپرس ۲۰۱۳ برای وب (VSEW) طراحی شده و بنابراین مهم است که آن را روی کامپیوتر خود نصب کرده باشید. قسمت بعدی چگونگی دریافت و نصب VSEW را

نشان می‌دهد. وقتی آن را دریافت کرده و اجرا کردید، می‌بینید چگونه اولین وب سایت خود را ایجاد کرده و سپس امکانات زیادی از VSEW را خواهید دید.

Microsoft Visual Studio Express For Web

مایکروسافت ویژوال استودیو شامل ابزارهای بسیاری است که در ایجاد برنامه‌های وب پیچیده ASP.NET به شما کمک می‌کنند.

ویژوال استودیو برای وب در دو نسخه است: یک نسخه جداگانه و رایگان که ویژوال استودیو اکسپرس برای وب نامیده شده و به صورت بخشی از یک بسته بزرگ‌تر با نام ویژوال استودیو ۲۰۱۳ که در نسخه‌های مختلف با قیمت‌های متفاوت موجود است.

با این که نسخه اکسپرس ویژوال استودیو رایگان است، اما شامل همه امکانات و ابزارهای مورد نیاز برای ایجاد برنامه‌های پیچیده و غنی وب می‌باشد. تمام مثال‌های این کتاب با نسخه رایگان اکسپرس ایجاد شده‌اند بنابراین نیازی به خرید نسخه گران‌قیمت ویژوال استودیو نیست.

من از عبارت ویژوال استودیو برای مراجعه به هردو نسخه رایگان و تجاری ویژوال استودیو استفاده می‌کنم. وقتی بخواهم به صورت مخصوص درباره نسخه رایگان صحبت کنم، از عبارت اکسپرس یا ویژوال استودیو اکسپرس برای وب استفاده می‌کنم.

دریافت ویژوال استودیو بسیار آسان است. می‌توانید آن را از وب سایت مایکروسافت دانلود کنید.

دریافت ویژوال استودیو

می‌توانید نسخه رایگان ویژوال استودیو اکسپرس برای وب را از وب سایت مایکروسافت در آدرس www.microsoft.com/express/ دریافت کنید. در صفحه Express از لینک‌های دانلود استفاده کرده تا به صفحه‌ای برسید که دانلود برای محصولات اکسپرس را پیشنهاد می‌دهد که شامل ویژوال استودیو اکسپرس برای وب نیز می‌باشد. در این صفحه ویژوال استودیو برای وب را به صورت نسخه نصبی وب دریافت می‌کنید که تنها نصب کننده را دانلود کرده و سپس فایل‌های لازم برای نصب دانلود می‌شوند.

نصب ویژوال استودیو اکسپرس برای وب

نصب VSEW سراسر است و راحت است. بسته به روش نصب، کامپیوتر شما، سرعت اتصال اینترنت، نصب VSEW می‌تواند بین ۲۰ دقیقه و یک ساعت یا حتی بیشتر زمان بگیرد.

تمرین: نصب ویژوال استودیو اکسپرس ۲۰۱۳ برای وب

اکنون چگونگی نصب VSEW را روی کامپیوتر خود می‌بینید. این تمرین فرض می‌کند که از نسخه نصب تحت وب استفاده می‌کنید.

۱- به آدرس www.microsoft.com/express/ رفته و دستورالعمل‌ها برای دانلود VSEW را دنبال کنید. می‌توانید به صورت مستقیم با انتخاب Download و سپس کلیک روی Install Now کار دانلود را شروع کرده یا یک فایل ISO را دانلود کنید که می‌توانید آن را به صورت یک درایو مجازی در ویندوز باز کرده و کار نصب را از آن شروع می‌کنید. تنها مطمئن شوید که Visual Studio Express 2013 for Web را دانلود می‌کند. اگر



لینک تغییر کرده یا هیچ دسترسی مستقیمی به دانلود VSEW وجود نداشت، در وب به دنبال " install Visual Studio Express 2013 for Web " جستجو کرده تا به صفحه دانلود برای شروع نصب بروید.

۲- وقتی دانلود کامل شد، نصب را شروع کنید. باید یک صفحه مشابه با شکل ۱-۱ را ببینید.



شکل ۱-۱

گزینه برای توافق نامه را انتخاب کنید.

اگر نسخه کامل ویژوال استودیو را نصب می کنید، روی دکمه Next کلیک کنید که بعد از موافقت با توافق نامه ظاهر می شود. اکنون یک صفحه مشابه با شکل ۱-۲ را می بینید. مطمئن شوید که حداقل Microsoft Web Developer Tools را انتخاب کرده اید.

۳- در هر دو مورد، روی دکمه Install کلیک کرده تا نصب شروع شود.

۴- بعد از مدتی باید پیغامی ببینید که نشان می دهد ویژوال استودیو با موفقیت نصب شده است. برنامه نصب کننده، نیاز به ری بوت کامپیوتر بعد از نصب دارد. وقتی نصب کامل شد، ویژوال استودیو آماده استفاده است.



شکل ۱-۲

ایجاد اولین وب سایت ASP.NET 4.5.1

احتمالاً نمی‌توانید منتظر شروع برای اولین وب سایت ASP.NET خود بمانید و بنابراین به جای بحث درباره موارد نظری وبسایت‌ها در ویژوال استودیو، تمرین بعدی چگونگی شروع به کار و ایجاد اولین پروژه وب سایت شما را نشان می‌دهد. سپس در توضیح آن چگونگی عملکرد این تمرین را می‌بینید که دید خوبی درباره مسائل پشت پرده دیدن صفحه ASP.NET در مرورگر وب به شما می‌دهد.

تمرین: ایجاد اولین وب سایت ASP.NET شما

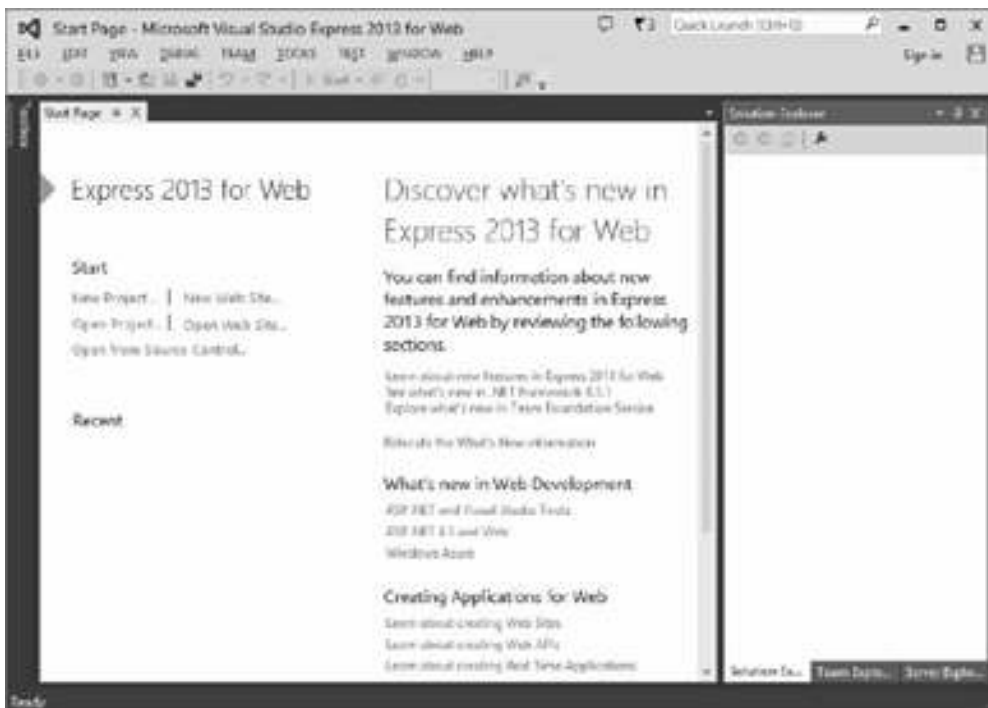
۱- ویژوال استودیو را از منو یا صفحه استارت اجرا کنید.

یک صفحه به شما نشان داده شده که اجازه وارد شدن به ویژوال استودیو با یک اشتراک مایکروسافت را به شما می‌دهد. اگر می‌خواهید تنظیمات روی چند کامپیوتر با یکدیگر یکسان باشند، این کار مفید خواهد بود. روی Sign In کلیک کنید تا با نام کاربری و کلمه عبور خود وارد شوید یا روی Sign Up کلیک کرده تا یک اشتراک جدید را ایجاد کنید. اگر نمی‌خواهید وارد شوید، روی Not Now, may be later کلیک کنید. اولین باری که ویژوال استودیو را اجرا می‌کنید، اجرای آن اندکی طول می‌کشد زیرا در حال پیکربندی خود است. اجراهای بعدی به مراتب سریع‌تر خواهد بود.



۲- اگر از نسخه تجاری ویژوال استودیو استفاده می‌کنید، یک پنجره ظاهر شده که در آن مجموعه تنظیمات مختلف برای اولین اجرای ویژوال استودیو را تعیین می‌کنید. انتخاب‌ها در این پنجره روی چیدمان پنجره‌ها، جعبه‌ابزارها، منوها و میانبرها تأثیر می‌گذارد. تنظیمات Web Development را انتخاب کنید که برای توسعه دهندگان ASP.NET طراحی شده است. همیشه می‌توانید یک پروفایل دیگر را نیز انتخاب کنید.

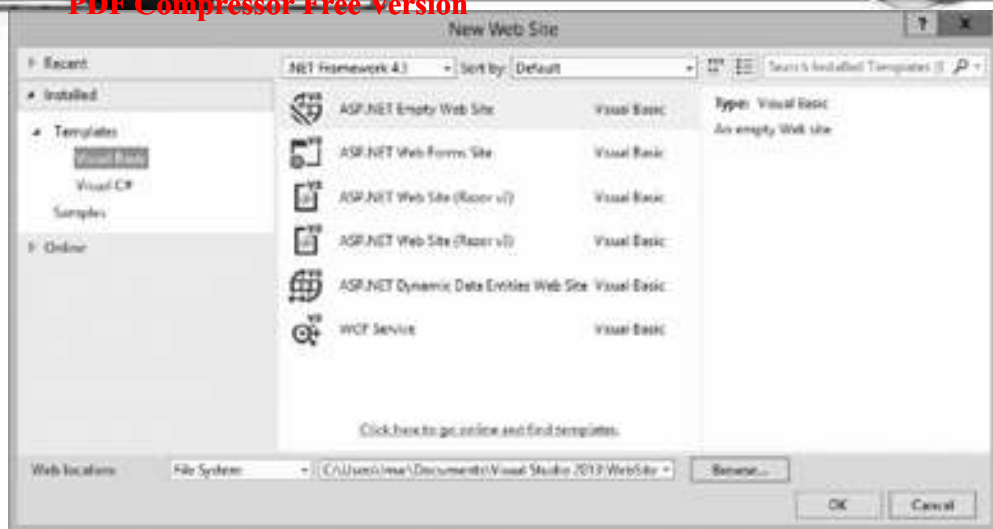
۳- وقتی ویژوال استودیو به طور کامل پیکربندی شد، صفحه اصلی آن را می‌بینید که در شکل ۱-۳ نشان داده شده است.



شکل ۱-۳

در قسمت بعدی توضیح کاملی درباره پنجره‌ها، نوارهای ابزار، پنل‌ها و منوها خواهید دید. برای اکنون کافی است روی ایجاد یک سایت جدید متمرکز می‌شوید. روی منوی File در گوشه بالا سمت چپ کلیک کرده و دستور New Web Site را انتخاب می‌کنید. اگر از نسخه تجاری ویژوال استودیو استفاده می‌کنید، بسته به تنظیمات انتخاب شده هنگام اجرای اول ویژوال استودیو ممکن است لازم باشد ابتدا زیرمنوی New را باز کنید. مطمئن شوید که به صورت تصادفی از منوی New Project استفاده نمی‌کنید زیرا برای ایجاد انواع دیگر برنامه‌های دات‌نت استفاده می‌شود.

پنجره New Website همانند شکل ۱-۴ ظاهر می‌شود.



شکل ۴-۱

۴- در وسط این پنجره، در بالای لیست الگوهای موجود، ASP.NET Framework 4.5.1 را از لیست پایین افتادنی انتخاب کنید. سپس در قسمت Installed Templates در سمت چپ، یک زبان برنامه‌نویسی برای سایت خود انتخاب کنید. این کتاب از ویژوال بیسیک و ویژوال C# استفاده می‌کند و شما نیز می‌توانید یکی از این دو را انتخاب کنید.

۵- در لیست الگوها در وسط روی ASP.NET Web Forms Site کلیک کنید. مطمئن شوید که File System گزینه انتخاب شده در لیست پایین افتادنی Web Location در پایین سمت چپ است. اگر می‌خواهید می‌توانید مکان روی دیسک که سایت روی آن ذخیره می‌شود را با کلیک روی دکمه Browse و انتخاب یک مکان جدید روی هارد دیسک خود تغییر دهید. برای اکنون مکان پیش‌فرض که یک پوشه در پوشه Documents شما است مفید می‌باشد.

۶- روی OK کلیک کنید. ویژوال استودیو یک وب سایت جدید برای شما ایجاد کرده که شامل تعدادی فایل و پوشه است (شکل ۵-۱ را ببینید) تا بتوانید به سراغ وب سایت خود بروید.

۷- دو بار روی فایل Default.aspx کلیک کرده تا باز شود. همه کد درون بلاک <asp:Content> را که دارای ID برابر با BodyContent است (با <div> در خط ۵ شروع شده و با </div> خاتمه پیدا می‌کند) را حذف کنید. آن را با کد توپر نشان داده شده در این جا جایگزین کنید:

```
<asp:Content runat="server" ID="BodyContent" ContentPlaceHolderID="MainContent">
    <h2>Hello World</h2>
    <p>Welcome to Beginning ASP.NET 4.5.1 on <%= DateTime.Now.ToString() %></p>
</asp:Content>
```

کدهایی با فرمت مشابه با این کد را در کتاب زیاد خواهید دید. وقتی گفته می‌شود که کدی مانند این را تایپ کنید که قسمتی از آن توپر است، تنها باید کد توپر را تایپ کنید. بقیه کد باید درون فایل باقی بماند.



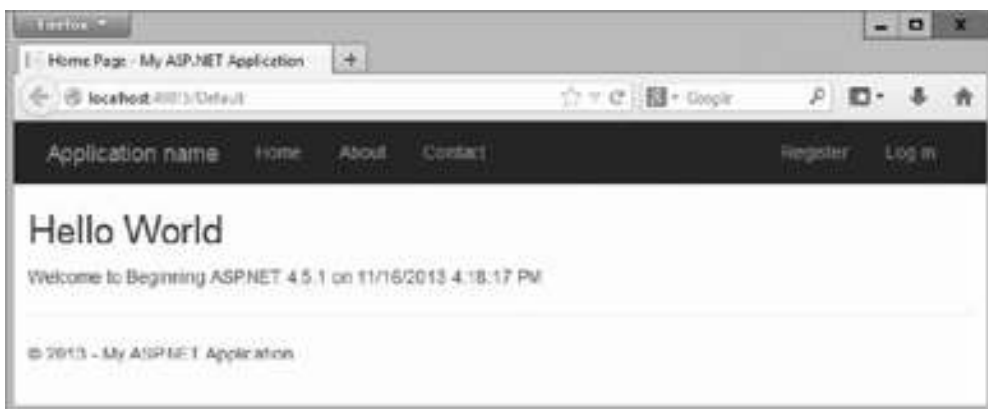
PDF Compressor Free Version

نگران کد با براکت‌های زاویه‌دار (<>) و علامت درصد در پیغام خوش‌آمد نباشید. به این‌ها تگ (Tag) گفته شده و بعداً در مورد آن‌ها بیشتر یاد می‌گیرید. با این که این کد برای شما آشنا نیست، اما می‌توانید عملکرد آن را حدس بزنید: تاریخ و زمان امروز را نمایش می‌دهد.



شکل ۵-۱

۸- کلید **Ctrl + F5** را روی صفحه کلید بزنید تا مرورگر وب پیش‌فرض شما صفحه را باز کند. صفحه‌ای مشابه با شکل ۶-۱ خواهید دید.



شکل ۶-۱

اگر هم اکنون تجربه کار با ویژوال استودیو را دارید احتمالاً قبلاً از F5 استفاده کرده بودید. اگر از این گزینه استفاده کنید، سایت در حالت دیباگ باز شده و پنجره‌ای برای فعال کردن دیباگ ظاهر می‌شود (که می‌توانید با خیال راحت این کار را انجام دهید). دیباگ کردن با F5 در فصل ۱۸ توضیح داده می‌شود. برای اکنون باید از Ctrl + F5 استفاده کنید.

اگر نوار اطلاعاتی درباره تنظیمات اینترنت اکسپلورر ظاهر شد، روی Turn on Internet Settings کلیک کنید.

اگر تاریخ و زمان درون صفحه را نمی‌بینید، یا اگر خطایی دریافت کرده‌اید، دوباره نگاهی به کد در صفحه خوش‌آمد بیندازید. این با یک براکت (<) و سپس یک علامت درصد و یک دو نقطه شروع می‌شود. سپس با یک علامت درصد و یک براکت زاویه‌دار دیگر (>) بسته می‌شود. همچنین مطمئن شوید که کد را همانند متن بالا تایپ کرده‌اید (کوچکی و بزرگی حروف را نیز باید در نظر بگیرید). این به خصوص در زمانی که از C# استفاده می‌کنید بسیار مهم است زیرا به بزرگی و کوچکی حروف حساس است.

۹- دقت کنید که چگونه یک آیکن کوچک نشان‌دهنده IIS Express در نوار تذکر و پندوز ظاهر می‌شود که در شکل ۷-۱ نیز نشان داده شده است.



شکل ۷-۱

آیکن به سرور وب داخلی با نام IIS Express متعلق است. این سرور وبی است که به صورت خودکار توسط ویژوال استودیو اجرا شده تا به درخواست‌ها برای صفحه شما پاسخ دهد. درباره چگونگی پردازش صفحه در همین فصل بیشتر یاد خواهید گرفت.

اگر این آیکن را در کنار ساعت نمی‌بینید، روی فلش کوچک به سمت بالا کلیک کرده و سپس روی Customize کلیک کنید. سپس IIS Express System Tray را برابر با Show Icon and Notification قرار دهید.

تمام شد. شما اولین سایت ASP.NET خود را با ویژوال استودیو ایجاد کردید.

چگونگی عملکرد

با این که وب سایت ایجاد شده بسیار ساده است، اما فرآیند آن یک صفحه Default.aspx ایجاد کرده که اصلاً ساده نیست. یک صفحه ASP.NET (که به عنوان یک صفحه ASPX یا یک فرم وب نیز شناخته می‌شود) به خودی خود کار زیادی انجام نمی‌دهد. این صفحه باید توسط یک سرور وب پردازش شده و به آن سرویس داده شود تا مرورگر شما بتواند آن را نمایش دهد. به همین خاطر است که ویژوال استودیو IIS Express را اجرا کرده تا درخواست برای صفحه را اداره کند. سپس مرورگر وب پیش‌فرض شما را اجرا کرده و به آن می‌گوید که به آدرس وب سرور (در این مثال http://localhost:49815/Default) برود با این که شماره پورت واقعی در آدرس در هر باری که سرور وب را اجرا می‌کنید می‌تواند تغییر کند زیرا به صورت تصادفی توسط ویژوال استودیو انتخاب می‌شود.



مهم است که تشخیص دهید فایل ASPX که در ویژوال استودیو تغییر دادید مشابه با چیزی نیست که سرانجام در مرورگر نمایش داده می‌شود.

وقتی یک صفحه را در ویژوال استودیو ایجاد می‌کنید، علامت‌هایی را به آن اضافه می‌کنید. این علامت‌ها ترکیبی از HTML، کد برای کنترل‌های سرور ASP.NET که در فصل ۴ بیشتر درباره آن‌ها یاد می‌گیرید، کد نوشته شده در Visual Basic.NET یا C# و موارد دیگر هستند.

وقتی در مرورگر خود برای یک صفحه ASPX درخواست می‌دهید، سرور وب صفحه را پردازش کرده و هر کد مبتنی بر سروری که در فایل پیدا می‌کند را اجرا کرده و علامت‌ها و کد ASP.NET را به HTML ساده تبدیل کرده و سپس آن را به مرورگر ارسال کرده که آن نیز صفحه را نشان می‌دهد. HTML نهایی این تمرین تاریخ و زمان کنونی را نمایش می‌دهد.

برای دیدن تفاوت HTML نهایی از صفحه ASPX، منبع صفحه را در مرورگر خود باز کنید. در اغلب مرورگرها با کلیک راست روی صفحه در مرورگر و انتخاب View Source یا View Page Source می‌توانید کد منبع صفحه را ببینید. با این کار ویرایشگر متنی پیش‌فرض شما باز شده و کد HTML برای صفحه را نشان می‌دهد.

اگر مرورگر را بسته‌اید، دوباره Ctrl + F5 را در ویژوال استودیو بزنید تا صفحه باز شده و سپس View Source را انتخاب کنید.

در فایل منبع به پایین رفته تا به خطی با متن Welcome برسید. دقت کنید که چگونه به جای کد بین تگ‌ها، تاریخ و زمان واقعی را می‌بینید:

```
<h2>Hello World</h2>
```

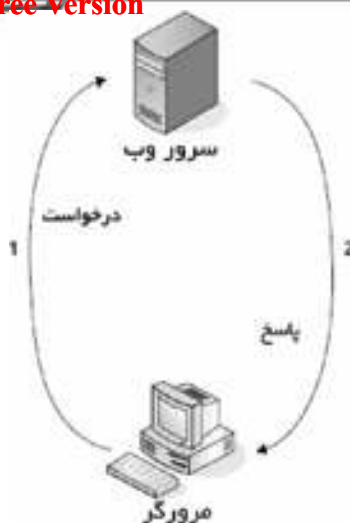
```
<p>Welcome to Beginning ASP.NET 4.5.1 on 11/16/2013 4:18:17 PM</p>
```

وقتی سرور وب صفحه را پردازش می‌کند، تاریخ و زمان کنونی را از سرور دریافت کرده و آن را درون کد HTML ارسال شده به مرورگر قرار می‌دهد. بسته به تنظیمات زبان روی نصب ویندوز شما، ممکن است تاریخ و زمان با فرمت متفاوتی را ببینید.

آشنایی با ASP.NET 4.5.1

وقتی یک URL مانند www.wrox.com را در مرورگر خود وارد کرده و Enter را می‌زنید، مرورگر درخواستی را به سرور وب در آن آدرس ارسال می‌کند. این کار از طریق HTTP انجام می‌شود که مخفف HyperText Transfer Protocol است. این پروتکل توسط مرورگرهای وب برای برقراری ارتباط با سرورهای وب استفاده می‌شود. وقتی برای یک URL درخواست می‌دهید، این درخواست به مرورگر وب ارسال می‌شود. وقتی سرور فعال درخواست معتبر باشد، سرور درخواست را پذیرفته، آن را پردازش کرده و سپس پاسخ را به مرورگر کلاینت ارسال می‌کند. رابطه میان درخواست و پاسخ در شکل ۱-۸ نشان داده شده است.

به دلیل این که از IIS Express استفاده می‌کنید، سرور و کلاینت واقعاً ماشین یکسانی هستند. با این حال در دنیای واقعی، وب سایت روی یک سرور وب بیرونی قرار داشته که کلاینت‌های زیادی می‌توانند به آن دسترسی داشته باشند.



شکل ۸-۱

برای فایل‌های استاتیک ساده مانند فایل‌های HTML یا تصاویر، سرور وب فایل را از هارددیسک محلی خود خوانده و آن را به مرورگر ارسال می‌کند. با این حال برای فایل‌های دینامیک مانند صفحه‌های ASPX این کافی نیست. اگر سرور وب فایل ASPX را به صورت مستقیم به مرورگر ارسال کند، تاریخ و زمان کنونی را در مرورگر نخواهید دید و به جای آن کد واقعی را می‌بینید:

```
<%: DateTime.Now.ToString() %>
```

بنابراین به جای ارسال مستقیم فایل، سرور درخواست را به یک نرم‌افزار دیگر که قادر به پردازش صفحه است ارسال می‌کند. این کار با یک مفهوم با نام Application Mapping یا Handler Mapping انجام می‌شود که در آن یک پسوند فایل (در این مثال .aspx) به یک برنامه که قادر به اداره آن است نگاشت می‌شود. در مورد یک صفحه .aspx، درخواست سرانجام توسط موتور اجرای ASP.NET اداره و پردازش می‌شود (که قسمتی از فریم‌ورک دات‌نت مایکروسافت است که برای اداره درخواست‌های وب طراحی شده است).

نکته

در شکل ۷-۱ پسوند .aspx را در نام صفحه نمی‌بینید. با این حال در پشت صحنه درخواست توسط Default.aspx انجام می‌شود.

در زمان پردازش صفحه، سه قسمت اصلی می‌توانند روی روش رسیدن صفحه نهایی به مرورگر تأثیر بگذارند:

- متن استاتیک هر متن استاتیکی مانند کد HTML، CSS یا جاوااسکریپتی که درون یک صفحه قرار می‌دهید، به صورت مستقیم به مرورگر ارسال می‌شود.
- کنترل‌های سرور ASP.NET: این کنترل‌ها درون صفحه APSX شما قرار گرفته و وقتی پردازش می‌شوند کد HTML را که درون صفحه قرار داده شده است نشان می‌دهند. درباره این کنترل‌ها در این فصل و فصل ۴ یاد می‌گیرید.



کد برنامه‌نویسی: کد مانند ویژوال بیسیک یا C# را به صورت مستقیم درون صفحه قرار می‌دهید که در این تمرین نیز دیدید. به علاوه می‌توانید کد را در یک فایل جداگانه نیز قرار دهید. عبارت رسمی برای این فایل کد، Code Beside است. اما برنامه‌نویس‌ها به آن فایل Code Behind می‌گویند که من نیز در این کتاب از آن استفاده می‌کنم. این کد می‌تواند توسط موتور اجرا به صورت خودکار اجرا شده یا بر مبنای یک عمل کاربر اجرا شود. در هر صورت اجرای کد روی روش نمایش صفحه تأثیر می‌گذارد که برای این کار به پایگاه داده دسترسی پیدا کرده، محاسبات را انجام داده، کنترل‌های خاصی را پنهان کرده یا نمایش می‌دهد و بسیاری کارهای دیگر انجام می‌شود. درباره فایل Code Behind در فصل بعدی یاد گرفته و برنامه‌نویسی صفحه‌های ASP.NET را با جزئیات بیشتر در فصل ۵ خواهید دید.

وقتی کار پردازش صفحه به اتمام رسید، و تمام HTML برای صفحه جمع‌آوری شد، HTML به مرورگر ارسال می‌شود. مرورگر نیز آن را خوانده، پردازش کرده و صفحه را برای شما نمایش می‌دهد.

آشنایی با HTML

HTML زبان اصلی برای ایجاد صفحه‌ها وب است که توسط مرورگرهای وب قابل درک و تفسیر است. آخرین نسخه HTML، HTML5 است. با این که مشخصات HTML5 هنوز در حال توسعه است، اما اکثر مرورگرهای مدرن بخش‌هایی از این مشخصات را وارد کرده‌اند و این پشتیبانی در نسخه‌های جدیدتر مرورگرها افزایش می‌یابد. من نیز در این کتاب از این نسخه HTML استفاده می‌کنم.

عناصر HTML و تگ‌ها

برای مشخص کردن چگونگی رندر (Render یا نمایش) محتویات در مرورگر، HTML از متن‌هایی درون براکت‌های زاویه‌دار استفاده می‌کند. به این متن، یک تگ (Tag) گفته می‌شود؛ یک جفت تگ متن یا محتویات دیگری را درون خود نگه می‌دارند که به آن یک عنصر (Element) گفته می‌شود. نگاهی دیگر به کد HTML بیندازید که در تمرین این فصل دیدید:

```
<h2>Hello World</h2>
```

```
<p>Welcome to Beginning ASP.NET 4.5.1 on 11/16/2013 4:18:17 PM</p>
```

اولین خط این مثال حاوی یک عنصر <h2> با یک تگ باز (<h2>) و تگ بسته (</h2>) است. این عنصر برای علامت‌گذاری یک عنوان در سطح دوم استفاده می‌شود (اگر اندکی در کد منبع نهایی درون مرورگر به بالا بروید یک عنصر <h1> را نیز خواهید دید). دقت کنید که عنصر توسط یک تگ مشابه بسته شده که دارای یک اسلش (/) است: </h2>. هر متنی بین این تگ‌های باز و بسته به عنوان بخشی از عنصر در نظر گرفته شده و بنابراین به عنوان یک عنوان رندر می‌شود. در اغلب مرورگرها این به معنی رندر شدن متن با اندازه بزرگ‌تر است. مشابه با تگ <h2>، تگ‌های <h1> تا <h6> برای عنوان‌ها وجود دارند.

در ادامه عنصر عنوان، یک عنصر <p> را می‌بینید که برای علامت‌گذاری یک پاراگراف استفاده می‌شود. تمام متن درون یک جفت تگ <p> به عنوان بخشی از پاراگراف در نظر گرفته می‌شود. به صورت پیش‌فرض، یک مرورگر یک پاراگراف را با فاصله حاشیه بیشتر در پایین رندر می‌کند (با این که می‌توانید آن را تغییر دهید).

تگ‌های زیادی در HTML وجود دارد که نمی‌توان همه آن‌ها را در این جا پوشش داد. جدول بعدی بعضی از تگ‌های معمول‌تر را به همراه توضیح چگونگی استفاده از آن‌ها نشان می‌دهد. برای یک لیست کامل از عناصر HTML نگاهی به وب سایت زیر بیندازید:

www.w3.org/TR/html5/index.html

تگ	توضیح	مثال
<html>	برای مشخص کردن شروع و انتهای کل صفحه استفاده می‌شود.	<pre><html> ...All other content goes here </html></pre>
<head>	برای مشخص کردن یک بخش خاص از صفحه استفاده می‌شود که حاوی اطلاعاتی درباره صفحه، شامل عنوان و ارجاع‌ها به منابع خارجی می‌باشد. محتویات این عنصر به صورت مستقیم روی صفحه نمایش داده نشده اما روی ظاهر و رفتار صفحه تأثیر می‌گذارد. این عنصر درون عنصر <html> قرار می‌گیرد.	<pre><head> ...Content goes here </head></pre>
<title>	برای تعریف عنوان صفحه استفاده می‌شود. این عنوان در نوار عنوان مرورگر نمایش داده می‌شود. این عنصر درون عنصر <head> قرار می‌گیرد.	<pre><title> Welcome to Planet Wrox 4.5.1 </title></pre>
<body>	برای مشخص کردن شروع و انتهای بدنه صفحه استفاده می‌شود. محتویات آن چیزی است که درون مرورگر خود می‌بینید. این عنصر درون عنصر <html> قرار می‌گیرد.	<pre><body> ...Page body goes here </body></pre>
<header>	برای مشخص کردن سرصفحه یک صفحه استفاده می‌شود. این عنصر و همه عناصر باقیمانده درون این جدول درون عنصر <body> قرار می‌گیرند.	<pre><header> ... </header></pre>
<section>	برای مشخص کردن بخش‌های مختلف درون صفحه شما استفاده می‌شود.	<pre><section></pre>



مثال	توضیح	تگ
<pre>...Content goes here </section></pre>	می‌توانید چند بخش درون یک صفحه وب داشته باشید.	
<pre><aside> ... </aside></pre>	برای مشخص کردن این که این بخشی از صفحه اصلی نیست استفاده می‌شود.	<aside>
<pre><article> ...Main page content goes here </article></pre>	برای مشخص کردن بخش اصلی محتویات درون یک صفحه استفاده می‌شود.	<article>
<pre><footer> ...Content for footer goes here </footer></pre>	بخش پایفحه یک صفحه را مشخص می‌کند.	<footer>
<pre> Visit the Wrox site </pre>	برای وصل کردن یک صفحه وب به صفحه دیگر (لینک دادن به آن) یا ایجاد لینک‌هایی درون یک صفحه استفاده می‌شود.	<a>
<pre></pre>	برای جاسازی تصاویر درون یک صفحه استفاده می‌شود.	
<pre>This is bold text while this text is in italic</pre>	برای اعمال فرمت روی متن به صورت توپر یا ایتالیک استفاده می‌شوند.	
<pre><input type="text" value="Some Text" /></pre>	برای فرم‌های ورودی استفاده شده که به کاربران اجازه ارسال اطلاعات به سرور را می‌دهند.	<form> <input> <textarea> <select>
<pre><table> <tr></pre>	این تگ‌ها برای ایجاد یک طرح دارای جدول استفاده می‌شوند.	<table> <tr>

مثال	توضیح	تگ
<pre><td>This is a Cell in Column 1</td> <td>This is a Cell in Column 2</td> </tr> </table></pre>	<table> کل جدول را تعریف کرده و <th> برای مشخص کردن خانه‌های سرستون و <tr> و <td> برای مشخص کردن (به ترتیب) ردیف‌ها و خانه‌ها استفاده می‌شوند.	<th> <td>
<pre> First item with a bullet Second item with a bullet First item with a number Second item with a number </pre>	این تگ‌ها برای ایجاد لیست‌های دارای شماره یا گلوله استفاده می‌شوند. تگ‌های و ظاهر لیست را مشخص می‌کنند (بدون ترتیب که دارای گلوله بوده و دارای ترتیب که با شماره مشخص می‌شوند) و تگ برای مشخص کردن موارد درون لیست استفاده می‌شود.	
<pre><p>This is some normal text while this text appears in red </p></pre>	این تگ برای پوشاندن و تأثیر روی بخش‌های دیگر سند استفاده می‌شود. این عنصر به صورت درون خطی ظاهر می‌شود و بنابراین هیچ شکست خط اضافی به صفحه اضافه نمی‌کند.	
<pre><div>This is some text on 1 line</div> <div> This text is put directly under</pre>	همانند تگ تگ <div> به عنوان یک محفظه برای عناصر دیگر استفاده می‌شود. با این حال <div> به عنوان یک عنصر بلاک عمل کرده که (به صورت پیش‌فرض) باعث ایجاد	<div>



تگ	توضیح	مثال
	یک شکست خط صریح قبل و بعد از عنصر <div> می‌شود.	the previous text on a new line. </div>
<audio>	برای جاسازی فایل‌های ویدئو و صدا	<video src="Somevideo.mpg" />
<video>	درون صفحه وب استفاده می‌شوند.	
<source>	عنصر <source> برای تعریف چند نوع منبع صدا و ویدئو استفاده می‌شود.	

خاصیت‌های HTML

به غیر از عناصر HTML، مثال‌های درون جدول قبلی خاصیت‌های HTML را نیز نشان می‌دهند. خاصیت‌ها (Attributes) حاوی اطلاعات بیشتری هستند که رفتار یک عنصر خاص را کنترل می‌کند. برای مثال تگ که برای نمایش یک تصویر استفاده می‌شود دارای یک خاصیت src است که منبع تصویر را مشخص می‌کند. به صورت مشابه یک تگ نیز حاوی یک خاصیت style است که رنگ متن را به قرمز تغییر داده است. برای دیدن لیست کامل خاصیت‌ها، به آدرس زیر مراجعه کنید:

www.w3.org/TR/html5/index.html#attributes-1

توضیحات HTML

برای اضافه کردن توضیح به چیزی در HTML، آن را درون تگ‌های توضیح قرار می‌دهید:

```
<!-- This is a comment -->
```

کدی که به توضیح تبدیل می‌شود، توسط مرورگر پردازش نمی‌شود (و بنابراین قابل دیدن نخواهد بود) اما هنوز هم به مرورگر ارسال می‌شود (و قابل دیدن در فایل منبع خواهد بود). به دلیل این که به مرورگر اضافه می‌شود باعث افزایش اندازه صفحه شده و بنابراین باید در مواقع لازم از آن‌ها استفاده کرد. در فصل‌های بعدی چگونگی اضافه کردن توضیحات به کد در سرور را می‌بینید که به کلاینت ارسال نمی‌شوند.

قوانین HTML5

قوانین HTML5 بسیار ساده هستند. بسیاری از مواقع ویرال استودیو آن را برای شما تصحیح کرده یا لیستی از خطاها و پیشنهادهای برای حل آن‌ها را ارائه می‌دهد. HTML5 نسبت به نسخه‌های HTML قبلی منعطف‌تر است.

بستن عناصر

اغلب عناصر در HTML باید بسته شوند. بنابراین وقتی با یک تگ <div> شروع می‌کنید، باید جایی درون صفحه از تگ بسته </div> استفاده شود. استثنای وجود دارند (مانند عنصر <p> در زمانی که صورت مستقیم بعد از آن عناصر دیگری می‌آیند) اما من ترجیح می‌دهم تا به صورت صریح تگ‌های خود را ببندم. این برای عناصری نیز می‌باشد که دارای تگ‌های بسته مخصوص به خود نیستند مانند یا
 (یک شکست خط اضافه می‌کند).

در HTML5 این تگ‌ها را به صورت تگ‌های خود بسته (Self Closing) می‌نویسند که در آن‌ها اسلش بسته درون خود تگ قرار می‌گیرد مانند `<img src = "Logo.gif" />` یا `<br />`.

استفاده از خاصیت‌ها

هر وقت یک خاصیت را درون یک تگ می‌نویسید، می‌توانید مقدار آن را در بین کوتیشن‌های دوتایی، کوتیشن یا بدون کوتیشن وارد کنید. برای مثال هنگام استفاده از خاصیت `src` تگ `<img>` می‌توانید آن را به صورت زیر وارد کنید:

```

```

همچنین می‌توانید از کوتیشن تکی برای مقدار خاصیت استفاده کنید مانند:

```
<img src='Logo.gif' />
```

هر دو روش به شرطی کار می‌کنند که هردو کوتیشن انتهایی از یک نوع باشند. برای مقادیری که حاوی فاصله نیستند، می‌توانید کوتیشن‌ها را حذف کنید:

```
<input value=yes>
```

همچنین بعضی وقت‌ها لازم است که کوتیشن‌های دوتایی و تکی را در یکدیگر قرار دهید. برای مثال وقتی برای بعضی از دستورات ASP.NET باید از کوتیشن دوتایی استفاده کرد، باید از کوتیشن منفرد برای جدا کردن مقادیر خاصیت استفاده کنید:

```
<asp:Label ID="TitleLabel" runat="server" Text='<%= Eval("Title") %>' />
```

از این روش نگارش زیاد در این کتاب خواهید دید.

این کتاب در هر جایی که امکان‌پذیر باشد در همه HTMLها از کوتیشن دوتایی استفاده می‌کند.

تودرتو کردن صحیح عناصر

وقتی عناصر تودرتو را ایجاد می‌کنید باید مطمئن شوید که ابتدا عنصر درونی‌تری که آخر از بقیه باز کرده‌اید را بسته و سپس عنصر بیرونی‌تر را ببندید. این مثال را در نظر بگیرید که یک متن را توپر و ایتالیک می‌کند:

```
<strong><em>This is some formatted text</em></strong>
```

دقت کنید که چگونه تگ `<em>` قبل از تگ `<strong>` بسته شده است. با تغییر ترتیب تگ‌های بسته، یک HTML نامعتبر ایجاد می‌شود:

```
<strong><em>This is some formatted text</strong></em>
```

اضافه کردن معرفی DOCTYPE به صفحه

یک DOCTYPE اطلاعاتی درباره نوع HTML که مرورگر باید انتظار داشته باشد به آن می‌دهد. به صورت پیش‌فرض ویژوال استودیو DOCTYPE برای HTML5 را به صفحه اضافه می‌کند:

```
<!DOCTYPE html>
```

این روی روش رندر صفحه توسط مرورگرهایی مانند اینترنت اکسپلورر تأثیر زیادی دارد و بنابراین اگر رفتار غیرعادی روی صفحه دیدید، حتماً DOCTYPE را چک کنید.



می‌توانید قوانین دستوری HTML5 را در آدرس زیر ببینید:

www.w3.org/TR/html-markup/syntax.html

نگاهی به علائم ASP.NET

زبان علامت‌گذاری برای کنترل‌های سرور ASP.NET مشابه با HTML هستند. همچنین این زبان دارای نگارش تگ‌ها، عناصر و خاصیت‌ها است و از براکت‌های زاویه‌دار و تگ‌های بسته یکسانی با HTML استفاده می‌کند. یک تفاوت بزرگ در این است که تگ‌های ASP.NET با یک پیشوند `asp:` شروع می‌شوند. برای مثال یک دکمه در ASP.NET دارای ظاهر زیر است:

```
<asp:Button ID="Button1" runat="server" Text="Click Me" />
```

ببینید که چگونه تگ با کاراکتر اسلش، خود بسته شده است و نیاز به تایپ تگ بسته جداگانه را از بین برده است. اگر بخواهید می‌توانید تگ بسته جداگانه را به جای آن استفاده کنید.

وقتی یک کنترل سرور پردازش می‌شود، HTML را بر می‌گرداند. بنابراین کد برای همین دکمه هنگام رندر در مرورگر به شکل زیر در می‌آید:

```
<input type="submit" name="Button1" value="Click Me" id="Button1" />
```

فرآیند تبدیل کنترل سرور به معادل HTML آن را قبلاً برای تاریخ کنونی سیستم دیدید. کنترل سرور در سرور با اداره کننده ASP.NET پردازش می‌شود. نتیجه این پردازش به شکل HTML است که به مرورگر ارسال و سپس نمایش داده می‌شود.

نگاهی به IDE

ویژوال استودیو یکی از غنی‌ترین و پرامکانات‌ترین محیط‌های توسعه یکپارچه (*Integrated Development Enviroment* یا IDE) برای ایجاد صفحات وب ASP.NET است. IDE اشاره به روشی دارد که همه ابزارهای مورد نیاز برای ایجاد برنامه‌های وب پیچیده در یک محیط یکپارچه شده‌اند. به جای نوشتن کد در یک ویرایشگر متنی، کامپایل کد در خط فرمان، نوشتن HTML و CSS در یک برنامه جداگانه، مدیریت پایگاه داده در یک برنامه دیگر، و ویژوال استودیو به شما اجازه داده تا همه این کارها را در یک محیط انجام دهید. به غیر از بهره مناسب عملکرد که این کار برای شما فراهم می‌آورد، یادگیری موارد جدید ویژوال استودیو نیز آسان‌تر می‌شود.

ناحیه توسعه اصلی

برای آشناتر شدن با ابزارهای بسیاری که در اینترفیس ویژوال استودیو در اختیار هستند، نگاهی به شکل ۱-۹ بیندازید. این صفحه‌ای است که بعد از ایجاد اولین وب سایت خود در ویژوال استودیو می‌بینید اما اکنون بعضی از عناصر آن هایلایت شده‌اند. اگر هم اکنون با نسخه قبلی ویژوال استودیو آشنا هستید می‌توانید از این قسمت چشم‌پوشی کنید.



شکل ۱-۹

انتخاب پروفایل برنامه‌نویسی

به دلیل این که VSE کاربران تازه‌کار برای توسعه ASP.NET و برنامه‌نویس‌های موقت را هدف قرار داده است، می‌توانید یکی از پروفایل‌های برنامه‌نویسی را انتخاب کنید: Basic Settings، Code Only و Expert Settings. در حالت Basic Settings، موارد منویی که زیاد از آن‌ها استفاده نمی‌شوند پنهان شده یا درون زیرمنوهای دیگری قرار می‌گیرند. پروفایل Code Only برای کسانی مناسب است که فقط کدنویسی می‌کنند و به امکانات طراحی VSEW علاقه ندارند مانند Design View یا Toolbox. حالت Expert Settings امکان دسترسی به عملکرد کامل VSEW را به شما می‌دهد. می‌توانید با انتخاب Tools>>Settings به این حالت‌ها دسترسی پیدا کنید. این کتاب فرض می‌کند که شما از حالت Expert Settings استفاده می‌کنید. شاید در ابتدا فکر کنید به بسیاری از این امکانات نیاز ندارید اما مطمئن باشید که تا انتهای کتاب از بسیاری از آن‌ها استفاده خواهید کرد.

منوی اصلی

در بالای برنامه، و در پایین نوار عنوان پنجره، منوی اصلی را می‌بینید. این منو حاوی موارد آشنایی است که در بسیاری از برنامه‌ها می‌بینید مانند File، Edit و Help. به علاوه منوهای خاصی برای ویژوال استودیو را نیز می‌بینید مانند Website و Debug. این منوها بسته به وظیفه‌ای که در حال انجام آن هستید به صورت دینامیک تغییر می‌کنند و بنابراین هنگام کار کردن موارد منو ظاهر و ناپدید می‌شوند. می‌توانید از Help>>Set Help Preference استفاده کرده تا راهنمای آنلاین و آفلاین را تنظیم کنید. راهنمای آفلاین باید ابتدا نصب شده و راهنمای آنلاین نیازمند اتصال به اینترنت است. در انتهای سمت راست یک لینک Sign In اجازه وارد شدن به اشتراک مایکروسافت شما را داده تا بتوانید تنظیمات خود را بین چند ماشین مختلف همزمان کنید.



پنجره Notification

در نوار عنوان برنامه در سمت چپ دکمه‌های Close، Minimize و Maximize، یک آیکون با پرچم کوچک می‌بینید. کلیک روی این آیکون پنجره Notification را نمایش می‌دهد که در آن اطلاعات مهمی ذکر می‌شوند. برای مثال داندلدهای موجود برای محتویات اسناد، لایسنس‌های منقضی شده، به‌روزرهای برنامه، و موارد بیشتر ذکر می‌شوند. بسته به اهمیت تذکرهای موجود، رنگ آیکون پرچم در نوار عنوان تغییر رنگ می‌دهد. این پنجره از طریق منوی View نیز قابل دسترسی است. اگر نسخه RTM ویژوال استودیو را نصب کرده باشید، احتمالاً چند به‌روزرسانی را می‌بینید که بهتر است آن‌ها را نصب کنید.

قسمت نوار ابزار

در پایین منو، قسمت نوار ابزار را می‌بینید که می‌تواند نوارهای ابزار مختلفی را نمایش داده تا دسترسی سریعی به اغلب عملکردهای ویژوال استودیو داشته باشید. در شکل ۱-۹ تنها دو نوار ابزار فعال هستند اما ویژوال استودیو دارای نوارهای ابزار مختلفی است که برای کاربردهای متفاوت طراحی شده‌اند. برای فعال یا غیرفعال کردن یک نوار ابزار، روی یک نوار ابزار موجود کلیک راست کرده و از منوی ظاهر شده، نوار ابزار مورد نظر خود را انتخاب کنید.

Toolbox

در سمت چپ صفحه اصلی، سربرگی را برای Toolbox می‌بینید. اگر روی این سربرگ کلیک کنید، Toolbox ظاهر شده که می‌توانید محتویات آن را ببینید. اگر روی آیکون سنجاق کوچک در گوشه بالا سمت راست Toolbox کلیک کنید (یا هر پل دیگری که دارای این آیکون است)، به IDE سنجاق شده و بنابراین همیشه نمایش داده می‌شود.

همانند نوار منو و نوارهای ابزار، Toolbox به صورت خودکار خود را به‌روز کرده تا محتویاتی را نشان دهد که به وظیفه کنونی که روی آن کار می‌کنید مربوط هستند. به عنوان مثال وقتی یک صفحه ASPX استاندارد را ویرایش می‌کنید، Toolbox کنترل‌های زیادی را که برای صفحه شما موجود هستند نمایش می‌دهد. می‌توانید به سادگی یک مورد را از این Toolbox کشیده و آن را در مکانی روی صفحه خود قرار دهید تا در آنجا ظاهر شود. این کنترل‌ها با جزئیات بیشتر در فصل ۴ توضیح داده می‌شوند. دقت کنید که هر دسته Toolbox حاوی یک آیکون Pointer نیز می‌باشند. این خودش یک کنترل نیست. در طراحی‌های دیگر ویژوال استودیو (مانند Win Forms)، این آیکون برای خارج شدن از حالت ترسیم کنترل استفاده شده اما دارای کاربرد کمی در ASP.NET است. همچنین می‌توانید ترتیب موارد را تغییر داده، آن‌ها را به Toolbox اضافه یا حذف کنید و حتی ابزارهای مخصوص خود را به آن‌ها اضافه نمایید. شخصی کردن IDE در همین فصل توضیح داده خواهد شد.

اگر Toolbox روی صفحه قابل دیدن نیست، از **Ctrl + Alt + X** یا دستور **View >> Toolbox** برای دیدن آن استفاده کنید.

Solution Explorer

در سمت راست صفحه، Solution Explorer را می‌بینید. Solution Explorer پنجره مهمی است زیرا اجازه دیدن سریع فایل‌های تشکیل دهنده وب سایت شما را می‌دهد. به جای قرار دادن همه فایل‌ها درون یک



پوشه بزرگ، Solution Explorer به شما اجازه داد تا فایل‌ها را در پوشه‌های مختلفی قرار داده و یک ساختار سایت منطقی و سازماندهی شده ایجاد کنید. می‌توانید از Solution Explorer برای اضافه کردن فایل‌های جدید به سایت، انتقال فایل‌ها با کشیدن و انداختن، تغییر نام فایل‌ها و حذف آن‌ها از پروژه استفاده کنید. اغلب این عملکرد در پشت منوی کلیک راست پنهان است که بسته به موردی که روی آن کلیک راست می‌کنید، تغییر می‌کند.

در بالای Solution Explorer یک نوار ابزار کوچک می‌بینید که دسترسی سریعی به تعدادی عملکرد مربوط به وب سایت شما می‌دهد مانند بارگذاری مجدد پنجره Solution Explorer، یک گزینه برای تودرتو کردن فایل‌های مربوط، و یک دکمه به شما اجازه پیکربندی وب سایت را می‌دهد. اغلب این عملکردها در این کتاب توضیح داده می‌شوند.

می‌توانید با زدن `Ctrl + Alt + L` یا انتخاب Solution Explorer >> View از منوی اصلی به Solution Explorer دسترسی پیدا کنید.

Server Explorer

این پنجره، که در پشت Solution Explorer در شکل ۱-۹ مخفی است به شما اجازه کار کردن با پایگاه‌های داده و دیگر سرویس‌ها را می‌دهد.

می‌توانید با انتخاب Server Explorer >> View یا `Ctrl + Alt + S` به این پنجره دسترسی پیدا کنید. این مورد در فصل‌های مربوط به پایگاه‌های داده توضیح داده شده که از فصل ۱۲ شروع می‌شوند.

Team Explorer

این پنجره که در شکل ۱-۹ در پشت Solution Explorer است، برای وصل شدن به و کار کردن با Team Foundation Server (TFS)، Application Lifecycle Management، میکروسافت که برای کنترل منبع استفاده می‌شود، به کار می‌رود. برای جزئیات بیشتر و یک اشتراک رایگان از TFS به آدرس

<http://tfs.visualstudio.com>

مراجعه کنید.

TFS و Team Explorer در این کتاب توضیح داده نمی‌شوند.

شبکه Properties

در شبکه Properties می‌توانید مشخصات بسیاری از موارد درون ویژوال استودیو شامل فایل‌های درون Solution Explorer، کنترل‌های روی صفحه وب، مشخصات خود صفحه و موارد دیگر را دیده و ویرایش کنید. این پنجره همیشه در حال تغییر بوده تا مورد انتخاب شده کنونی را منعکس کند. می‌توانید با زدن کلید `F4` این پنجره را باز کنید. از میانبر مشابهی برای نمایش جزئیات مورد انتخاب شده توسط شبکه Properties استفاده می‌کنید.



پنجره سند

پنجره سند بخش اصلی پنجره است. این همان جایی است که اغلب کارهای خود را انجام می‌دهید. می‌توانید از این پنجره برای کار کردن با فرمت‌های سند مختلف استفاده کرده، فایل‌های ویژوال بیسیک و C# را کد نویسی کرده و حتی تصاویر را مورد استفاده قرار دهید. از این پنجره برای مدیریت پایگاه‌های داده، ایجاد نسخه‌هایی از سایت خود، دیدن صفحات درون سایت در یک مرورگر کوچک و موارد دیگر استفاده می‌نمایید.

صفحه استارت

هر وقت ویژوال استودیو را اجرا می‌کنید، صفحه استارت (Start Page) در پنجره سند نمایش داده می‌شود. در این صفحه می‌توانید وبسایت‌های جدیدی را به همراه پروژه‌های دیگر ایجاد و یا باز کنید. این صفحه لینک‌هایی برای اخبار و اطلاعات مربوط به توسعه سایت نیز فراهم آورده است. برای وصل کردن پروژه‌های خود به صفحه استارت، روی آیکن سنجاق در سمت چپ نام پروژه درون لیست پروژه‌های اخیر کلیک کنید. برای جدا کردن یک پروژه روی آن کلیک راست کرده و Remove From List را انتخاب کنید. برای باز کردن مجدد صفحه استارت، دستور View >> Start Page را انتخاب کنید.

تمرین

این تمرین چگونگی ایجاد یک وب سایت جدید با یک صفحه را نشان می‌دهد که حاوی تعدادی کنترل سرور ASP.NET است. می‌توانید ببینید چگونه از پنجره‌های ویژوال استودیو و Toolbox و شبکه Properties و کنترل‌های سرور ASP.NET استفاده کنید.

۱- ویژوال استودیو را اجرا کنید.

۲- اگر از نسخه Express استفاده می‌کنید، دستور Settings >> Tools را انتخاب کرده و سپس Expert Settings را انتخاب کنید تا پروفایل توسعه‌ای که امکان دسترسی به تمام امکانات ویژوال استودیو اکسپرس را می‌دهد فعال شود.

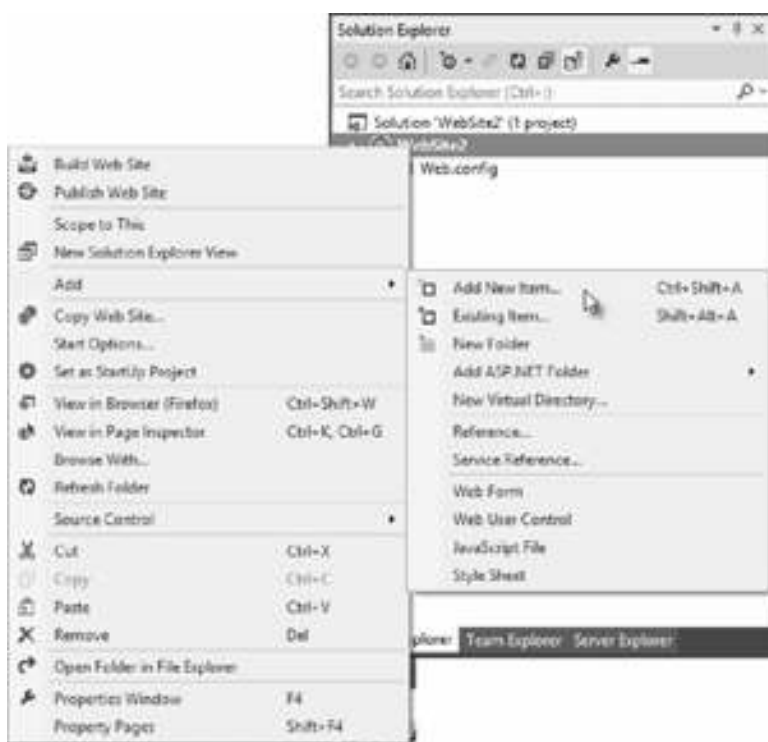
۳- در منوی File دستور New Site را انتخاب کنید. بسته به پیکربندی، ممکن است لازم باشد دستور File >> Web Site >> New را به جای آن انتخاب کنید. با این کار پنجره New Website ظاهر می‌شود.

۴- در این پنجره مطمئن شوید که ASP.NET Empty Web Site انتخاب شده است (شکل ۱-۱۰). مطمئن شوید که File System در لیست پایین افتادنی Web Location انتخاب شده است. مهم نیست که Visual Basic یا C# را برای این تمرین انتخاب کنید. روی OK کلیک کرده تا سایت جدید ایجاد شود.

۵- روی وب سایت جدید در Solution Explorer کلیک راست کنید. مطمئن شوید که روی وب سایت کلیک می‌کنید (که دارای برچسب Website2 است) و روی عنصر Solution پدر آن کلیک نمی‌کنید. این عنصر های‌لایت شده در شکل ۱-۱۱ است. در منوی ظاهر شده دستور Add >> Add New Item را انتخاب کنید.



شکل ۱۰-۱



شکل ۱۱-۱

۶- در پنجره‌ای که ظاهر می‌شود، روی Web Form کلیک کرده و سپس ControlsDemo را به عنوان نام وارد کنید. پسوند ASPX به صورت خودکار بعد از کلیک روی دکمه Add اضافه می‌شود (شکل ۱-۱۲). می‌توانید تنظیمات



دیگر این پنجره را در مقدار پیش‌فرض قرار دهید. این صفحه باید در Markup View ظاهر شده که HTML پیش‌فرض را به شما نشان دهد.



شکل ۱۲-۱

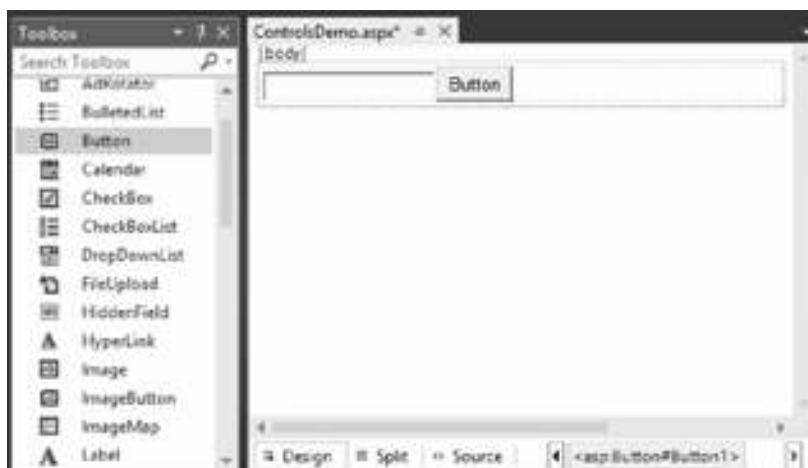
۷- روی دکمه Design در پایین پنجره Document کلیک کرده تا وارد حالت Design شوید (شکل ۱۳-۱).



شکل ۱۳-۱

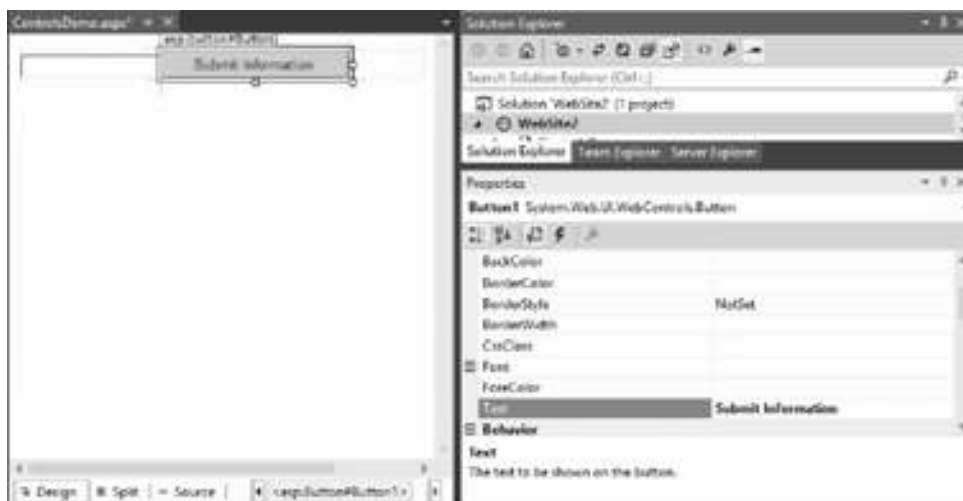
۸- اگر Toolbox باز نیست، از کلید **Ctrl + Alt + X** برای باز کردن آن استفاده کنید یا روی سربرگ Toolbox کلیک کنید تا باز شود. سپس روی آیکون سنجاق در گوشه بالا سمت راست آن کلیک کرده تا Toolbox

همیشه باز باشد. یک TextBox و یک Button را از دسته Standard به ناحیه نقطه‌چین در نمای Design صفحه ببرید. باید نمای Design شما مانند شکل ۱-۱۴ باشد.



شکل ۱-۱۴

۹- روی دکمه در نمای Design کلیک راست کرده و Properties را انتخاب کنید. در شبکه Properties مشخصه Text را در پایین دسته Appearance پیدا کنید (شکل ۱-۱۵) و آن را از Button به Submit تغییر دهید. به محض این که کلید Tab را بزنید یا جایی خارج از شبکه Properties کلیک کنید، نمای Design صفحه به‌روز شده و متن جدید را روی دکمه نشان می‌دهد.



شکل ۱-۱۵

۱۰- کلید **Ctrl + F5** را بزنید تا صفحه درون مرورگر پیش‌فرض شما باز شود (شکل ۱-۱۶). دقت کنید که لازم نیست تغییرات را به صورت صریح روی صفحه ذخیره کنید (با این که بهتر است این کار را مرتباً با کلید **Ctrl + S** انجام دهید). به محض زدن **Ctrl + F5**، ویژوال استودیو تغییرات را ذخیره کرده تا سند را باز کند.



۱۱- متنی را در کادر متنی تایپ کرده و روی دکمه کلیک کنید. دقت کنید که بعد از بارگذاری صفحه، متن هنوز هم در کادر متنی نمایش داده می‌شود. به غیر از آن هیچ تغییری ظاهر نمی‌شود زیرا کدی برای دکمه ننوشته‌اید.



شکل ۱-۱۶

چگونگی عملکرد

وقتی کنترل‌های Button و TextBox را از Toolbox به روی صفحه می‌کشید، ویژوال استودیو کد لازم را اضافه می‌کند. به صورت مشابه وقتی مشخصه Text دکمه را در شبکه Properties تغییر می‌دهید، ویژوال استودیو به صورت خودکار کد برای آن کنترل را تغییر می‌دهد. به جای استفاده از شبکه Properties می‌توانید متن را به صورت مستقیم بین علامت‌های کوتیشن مارک مشخصه Text در نمای Markup وارد کنید.

بعد از تغییر مشخصه Text، صفحه شما باید حاوی این کد در نمای Markup باشد:

```
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<asp:Button ID="Button1" runat="server" Text="Submit Information" />
```

وقتی Ctrl + F5 را می‌زنید تا صفحه را در مرورگر ببینید، سرور وب درخواست را دریافت کرده، صفحه توسط موتور ASP.NET پردازش شده و HTML برای صفحه به مرورگر ارسال می‌شود.

بعد از تایپ مقداری متن و کلیک روی دکمه، فرآیند مشابهی تکرار می‌شود: سرور وب درخواست را دریافت کرده، صفحه پردازش شده و نتیجه به مرورگر ارسال می‌شود. وقتی روی این دکمه کلیک می‌کنید، باید اتفاق افتادن یک پست‌بک (Postback) می‌شوید که در آن هر اطلاعات حاوی درون صفحه – مانند متنی که در کادر متنی تایپ کرده‌اید – به سرور ارسال می‌شود. ASP.NET با رندر مجدد صفحه به پست‌بک پاسخ می‌دهد. با این حال این بار کنترل‌هایی مانند TextBox با مقادیری که ارسال شده بودند نمایش داده می‌شوند.

از دستور View Source مرورگر خود استفاده کرده تا نگاهی به کد HTML صفحه بیندازید. باید کدی مشابه با این را درون صفحه ببینید:

```
<input name="TextBox1" type="text" value="Hello World" id="TextBox1" />
<input type="submit" name="Button1" value="Submit Information" id="Button1" />
```

همانند مثال قبلی، HTML نتیجه از کد ASPX اولیه متفاوت است.

پست‌بک‌ها مفهوم مهمی در ASP.NET هستند و جزئیات بیشتری از آن‌ها را در فصل‌هایی مانند فصل ۴ خواهید دید.

پنجره‌های حاوی اطلاعات

به غیر از پنجره‌هایی که به صورت پیش‌فرض هنگام اجرای ویژوال استودیو قابل دیدن هستند، پنجره‌های زیاد دیگری نیز وجود دارند. اغلب آن‌ها را در هنگام استفاده از این کتاب خواهید دید، اما بعضی از آن‌ها ارزش دارند که اکنون به آن‌ها اشاره شود. برای دسترسی به هر پنجره‌ای که در ادامه درباره آن صحبت می‌شود، از منوی اصلی View استفاده می‌کنید.

Error List

پنجره Error List لیستی از چیزهایی را به شما می‌دهد که هم اکنون در سایت شما شکسته شده‌اند که این شامل کدهای اشتباه در فایل‌های ASPX یا HTML، خطاهای برنامه‌نویسی در فایل‌های ویژوال بیسیک و C# نیز می‌شوند. این پنجره حتی می‌تواند خطاها در فایل‌های XML و CSS را نیز نشان دهد. پنجره Error List پیغام‌های خود را در سه دسته نشان می‌دهد: Errors، Warnings و Messages که برحسب شدت مشکل دسته بندی شده‌اند. شکل ۱-۱۷ پنجره Error List را برای صفحه‌ای نشان می‌دهد که دارای تعدادی مشکل در CSS و XHTML خود است.

	Description	File	Line	Column	Project
1	Validation (XHTML): Element 'div' is missing its closing tag.	ControlsDemo.aspx	18	6	WebSite2
2	Validation (CSS 3.0): "disply" is not a known CSS property name.	ControlsDemo.aspx	11	10	WebSite2
3	The block is unclosed, ']' expected.	ControlsDemo.aspx	11	24	WebSite2

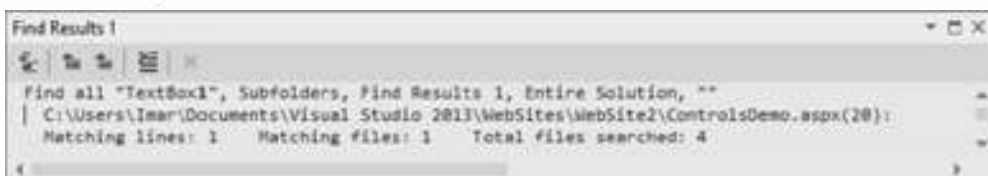
شکل ۱-۱۷

پنجره Output

وقتی سعی می‌کنید تا سایت خود را با منوی Build ایجاد کنید، پنجره Output به شما می‌گوید که آیا این کار موفق یا غیر موفق بوده است. اگر ایجاد خطا بدهد (برای مثال به دلیل وجود یک مشکل برنامه‌نویسی)، به شما می‌گوید که چرا ایجاد سایت مشکل پیدا کرده است. در نسخه‌های تجاری ویژوال استودیو، این پنجره برای اطلاعات دیگری نیز استفاده می‌شود مانند وضعیت برنامه‌های پلاگ‌این اکسترنال. ایجاد- یا به عبارت دیگر کامپایل- وب سایت‌ها بعداً در این کتاب توضیح داده می‌شود.

پنجره Find Results

امکانات Find و Replace و ویژوال استودیو هنگام مدیریت محتویات سایت ارزش بسیاری دارند. خیلی مواقع باید متنی را در سند یا کل سایت کنونی تغییر دهید. دستورات Find in File (Ctrl + Shift + F) و Replace in Files (Ctrl + Shift + H) هردو نتایج خود را در پنجره Find Results نمایش می‌دهند که در شکل ۱-۱۸ نشان داده شده است.



شکل ۱-۱۸

سفارشی کردن IDE

با این که طراحی استاندارد ویژوال استودیو و ابزارهای آن مفید هستند اما خیلی از مواقع لازم است که IDE را تغییر داده تا به شکل مناسب شما در بیاید. به عنوان مثال می‌توانید بعضی از پنجره‌ها را به مکانی انتقال دهید که دسترسی به آن‌ها آسان‌تر باشد یا می‌توانید پنجره‌های پرکاربرد بیشتری را باز نگه دارید. ویژوال استودیو قابلیت سفارشی شدن زیادی را در اختیار شما قرار می‌دهد. در بخش بعدی چگونگی انجام بسیاری از این کارها را یاد می‌گیرید.

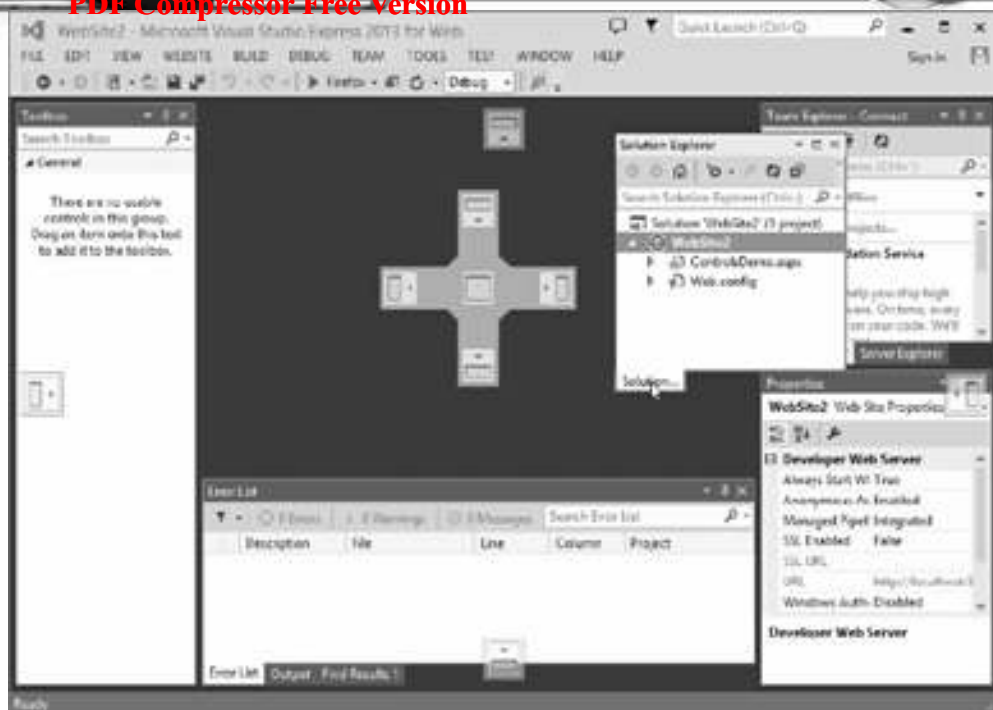
باز چینی پنجره‌ها

برای این که هر پنجره در مکان مناسب قرار بگیرد، می‌توانید آن را با ماوس در IDE جابجا کنید. کافی است نوار عنوان یا سربرگ پایین پنجره را با ماوس گرفته و آن را در مسیر مکان جدید بکشید. وقتی کشیدن را شروع کردید، می‌بینید که ویژوال استودیو سرخ‌های بصری را به شما نشان داده تا بدانید که پنجره در کجا خاتمه می‌یابد (شکل ۱-۱۹).

اگر پنجره را به روی یکی از چهار نشان‌دهنده وسط بکشید، پنجره به کنار پنجره موجود چسبانده می‌شود. وقتی پنجره را رها کنید، این پنجره در مکان جدید قرار می‌گیرد. اگر پنجره را روی مربع در وسط نشان‌دهنده بزرگ بکشید، این پنجره به آن پنجره چسبیده و فضای صفحه یکسانی را به اشتراک می‌گذارد. هر پنجره دارای سربرگ مخصوص به خود است که می‌توانید در پایین هر پنجره ببینید.

به غیر از چسباندن پنجره‌ها به پنجره‌های دیگر درون IDE، می‌توانید پنجره‌ها را شناور کنید. برای تغییر یک پنجره چسبیده به یک پنجره شناور، آن را از مکان کنونی آن کشیده و جایی درون IDE بیندازید بدون این که به یکی از سرخ‌های بصری روی صفحه برخورد کند. همچنین می‌توانید دستور `Window>>Float` را نیز از منوی اصلی انتخاب کرده یا روی نوار عنوان پنجره کلیک راست کرده و `Float` را انتخاب کنید.

برای بازایی یک پنل شناور به مکان چسبیده شده قبلی، روی نوار عنوان آن کلیک راست کرده و `Window>>Dock` را انتخاب کنید.



شکل ۱-۱۹

تغییر Toolbox

می‌توانید Toolbox را نیز تغییر دهید. به صورت پیش‌فرض موارد درون Toolbox به صورت الفبایی ذخیره می‌شوند اما می‌توانید با استفاده از کشیدن و انداختن مکان آن‌ها را تغییر دهید. برای این کار Toolbox را باز کرده (Ctrl + Alt + X) و یک مورد (مانند TextBox در پایین دسته Standard) را به یک مکان متفاوت بکشید. همچنین می‌توانید با کلیک راست روی موارد و انتخاب Delete از منوی میانبر آن‌ها را حذف کنید. نگران از دست رفتن همیشگی موارد نباشید؛ همیشه می‌توانید با انتخاب Reset Toolbox از منوی میانبر، Toolbox را به حالت اول خود برگردانید.

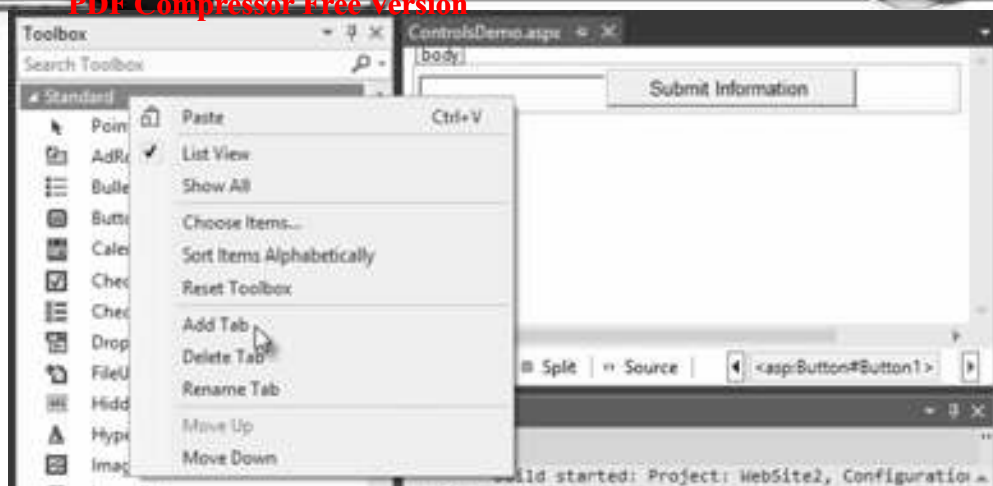
به علاوه می‌توانید مواردی را به Toolbox اضافه کنید. معمول‌ترین کاربرد، برای اسنپت‌های کد (Code Snippets) است. کافی است متن یا کدی را درون پنجره سند انتخاب کرده و سپس آن را به Toolbox بکشید. سپس می‌توانید روی موارد کلیک راست کرده و Rename Item را انتخاب کنید تا نامی مناسب به آن‌ها بدهید. برای جلوگیری از به هم ریختگی Toolbox با اسنپت‌های کد، ایجاد دسته جداگانه‌ای برای هر کدام را در نظر بگیرید. می‌توانید این کار را با انتخاب Add Tab از منوی کلیک راست Toolbox انجام دهید. یک نام وارد کرده و Enter را بزنید تا سربرگ Toolbox شما آماده استفاده شود.

در تمرین بعدی، می‌توانید IDE را به شکل مورد نظر خود تغییر دهید.

تمرین: شخصی کردن IDE

- در این تمرین ابتدا پنجره‌های درون IDE ویژوال استودیو را باز و سپس ترتیب آن‌ها را تغییر می‌دهید. نگران به هم ریختن چیزها نباشید. اندکی جلوتر در این فصل، دستورالعمل‌هایی برای چگونگی بازنشانی IDE را خواهید دید.
- ۱- اگر وب سایت تمرین قبلی را بسته‌اید، آن را دوباره باز کنید یا یک سایت جدید را از منوی File ایجاد کنید.
 - ۲- در منوی View، روی Error List کلیک کنید تا پنجره آن باز شود. اگر مورد Error List را به صورت مستقیم نمی‌بینید، دستور Expert Settings >> Settings >> Tools را انتخاب کرده و دوباره دستور قبلی را انتخاب نمایید. دقت کنید که Error List به صورت پیش‌فرض در پایین پنجره سند قرار می‌گیرد.
 - ۳- از منوی View، دستور Task List را انتخاب کنید. به صورت پیش‌فرض، این پنجره در مکان یکسانی با Error List قرار می‌گیرد که سربرگ هردو پنجره کنار یکدیگر قرار می‌گیرند.
 - ۴- روی سربرگ Task List کلیک کرده و دکمه ماوس را نگه داشته و این پنجره را از مکان آن به سمت پنجره سند بکشید. وقتی پنجره را رها می‌کنید، به صورت یک پنجره شناور درون IDE ظاهر می‌شود. برای برگرداندن پنجره، آن را به مربع وسط Error List بکشید. برای تغییر ترتیب ظاهر شدن سربرگ‌ها، یک سربرگ را به روی سربرگ‌های دیگر بکشید.
 - ۵- می‌توانید این مراحل را برای دیگر پنجره‌های قابل دیدن درون IDE یا پنجره‌های دیگری که از طریق منوی View باز می‌کنید، تکرار نمایید. مقداری زمان صف کرده تا با پنجره‌های مختلف و چگونگی بازچینی مکان آن‌ها روی صفحه آشنا شوید.
 - ۶- صفحه ControlsDemo.aspx را باز کنید (یا اگر یک سایت جدید ایجاد می‌کنید یک صفحه ASPX اضافه نمایید). برای این کار در Solution Explorer دو بار روی آن کلیک کنید. وقتی صفحه باز شد، Toolbox به صورت خودکار ظاهر می‌شود. اگر این طور نشد، Ctrl + Alt + X را مورد استفاده قرار دهید.
 - ۷- روی Toolbox کلیک راست کرده و Add Tab را انتخاب کنید (شکل ۱-۲۰). HTML Fragments را به عنوان نام جدید وارد کرده و Enter را بزنید. این یک دسته جدید به Toolbox اضافه کرده که با آن مانند دسته‌های دیگر رفتار می‌شود.
 - ۸- در حالی که پنجره سند صفحه ASPX شما را در نمای Markup نمایش می‌دهد، به صورت مستقیم <h1> را بعد از تگ باز <div> تایپ کنید. دقت کنید ویژوال استودیو تگ بسته </h1> را به صورت خودکار برای شما اضافه می‌کند. یعنی باید به کد زیر برسید:

```
<form id="form1" runat="server">
    <div>
        <h1></h1>
    </div>
```



شکل ۱-۲۰

۹- تگ باز و بسته <h1> را انتخاب کرده و سپس انتخاب را از نمای Source به روی سربرگ Toolbox جدید ایجاد شده در مرحله ۷ بکشید. انتخاب به صورت <h1></h1> Text: نمایش داده می‌شود.

۱۰- روی مورد Toolbox ایجاد شده کلیک راست کرده و Rename را انتخاب کرده و 1 Heading را به عنوان نام وارد کنید.

۱۱- نشانگر را دوباره در پنجره سند قرار داده و سپس Ctrl + K و سپس Ctrl + D را بزنید تا سند فرمت شود. یا می‌توانید دستور Format Document >> Edit را از منوی اصلی انتخاب کنید.

این کار باعث اعمال فرمت روی سند با توجه به قوانینی است که در پنجره Text Editor Options مشخص کرده‌اید. این اعمال فرمت روی اسناد دیگر مانند C#، ویژوال بیسیک و C#، XML و CSS نیز قابل انجام است.

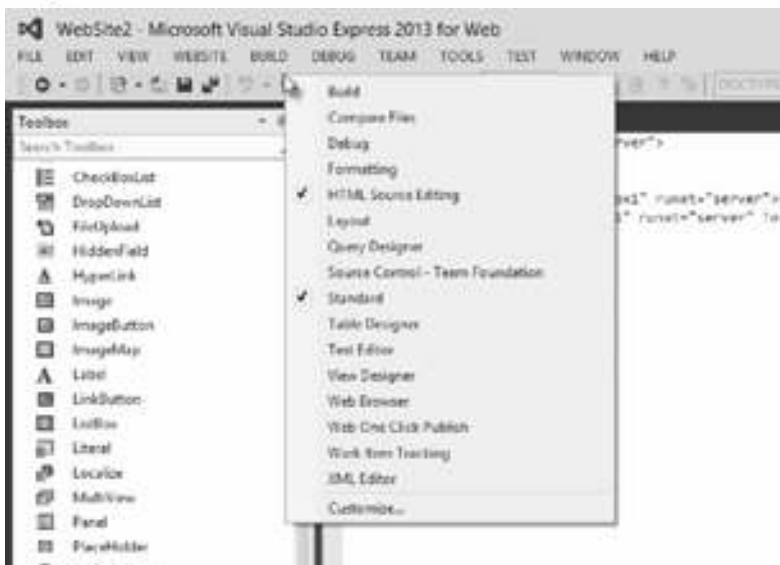
از این به بعد، هر وقت به یک عنوان در سند خود نیاز داشتید، کافی است نشانگر را در مکان مورد نظر در پنجره سند قرار داده و دو بار کلیک کنید تا عنوان مناسب در Toolbox در آن مکان قرار بگیرد.

شخصی کردن نوارهای ابزار

می‌توانید با سه روش نوارهای ابزار را تغییر دهید: می‌توانید نوارهای ابزار داخلی را نمایش داده یا پنهان کنید؛ می‌توانید دکمه‌هایی را به نوارهای ابزار موجود اضافه یا از آن‌ها حذف کنید یا می‌توانید نوارهای ابزار خود را با دکمه‌هایی که اغلب استفاده می‌کنید، ایجاد نمایید.

فعال یا غیرفعال کردن نوارهای ابزار

برای فعال یا غیرفعال کردن یک نوار ابزار، روی یک نوار ابزار موجود یا نوار منو کلیک راست کرده و سپس مورد مناسب را از لیست انتخاب می‌کنید (شکل ۱-۲۱). وقتی نوار ابزار نمایش داده شد، می‌توانید از دستگیره آن در سمت چپ نوار ابزار استفاده کرده تا آن را به مکان جدیدی بکشید.



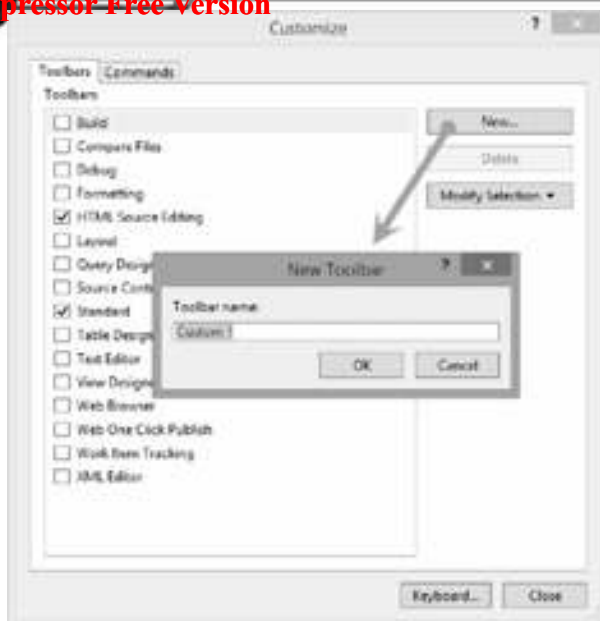
شکل ۲۱-۱

ویرایش نوارهای ابزار موجود

اگر فکر می‌کنید که نوار ابزار موجود دارای ابزار مورد نظر شما نیست یا حاوی دکمه‌های بدون کاربردی است می‌توانید دکمه‌های روی آن را تغییر دهید. برای این کار روی یک نوار ابزار کلیک راست کرده و سپس CUSTmoize را انتخاب کنید (به پایین منوی شکل ۲۱-۱ مراجعه کنید). سپس سربرگ Commands را انتخاب کرده و نوار ابزار مورد نظر را از لیست Toolbars انتخاب کنید. از دکمه‌های سمت راست می‌توانید دستورات جدید را اضافه یا دستورات قبلی را حذف کنید تا ترتیب آن‌ها را تغییر دهید (شکل ۲۲-۱).



شکل ۲۲-۱



شکل ۱-۲۳

ایجاد نوار ابزار جدید

ایجاد نوار ابزار جدید وقتی مفید است که بخواهید عملکردهای خاصی را که زیاد استفاده می‌کنید، در یک جا قرار دهید. برای ایجاد یک نوار ابزار جدید، پنجره Customize را باز کنید (که در قسمت‌های قبلی توضیح داده شده است). روی دکمه New کلیک کرده و سپس یک نام برای نوار ابزار خود وارد کنید (شکل ۱-۲۳). سپس به سربرگ Commands رفته و نوار ابزار را همانند نوارهای ابزار موجود ویرایش کنید.

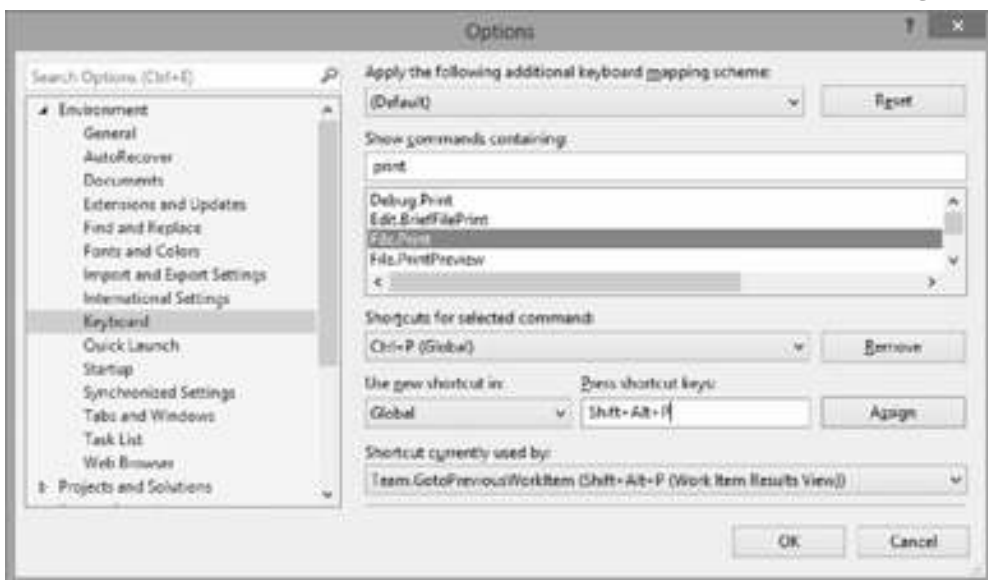
تغییر میانبرهای صفحه کلید

تنظیم دیگری که بسیاری از توسعه‌دهندگان تغییر می‌دهند، میانبرهای صفحه کلید است. میانبرهای صفحه کلید روشی عالی برای صرفه‌جویی در زمان هستند زیرا امکان انجام یک وظیفه را با یک میانبر صفحه کلید ساده می‌دهند. برای تغییر میانبرهای صفحه کلید، دستور Options >> Tools را انتخاب کرده، Environment را باز کرده و روی Keyboard کلیک کنید (شکل ۱-۲۴). یا Ctrl + Q را زده، keyboard را تایپ کرده و Enter را بزنید. دستوری که به دنبال آن هستید را از لیست دستورات پیدا کنید. به دلیل طولانی بودن این لیست، می‌توانید با تایپ چند حرف از آن دستور، لیست را کوتاه‌تر کنید. برای مثال print را در فیلد Show Commands Containing وارد کرده تا لیستی از دستورات مربوط به چاپ ظاهر شوند.

سپس در فیلد Press Shortcut Keys، یک میانبر جدید وارد کرده و روی Assign کلیک کنید. ویدیو به شما اجازه داده تا یک کلید دوتایی میانبر برای یک دستور را وارد کنید. به عنوان مثال می‌توانید برای دستور Close All Documents، از میانبرهای صفحه کلید Ctrl + K، Ctrl + O استفاده کنید. برای



استفاده از این دستور باید هردو ترکیب کلید را پشت سرهم بزنید. با این که یک میانبر دوتایی به نظر وقت تلف کردن می‌رسد اما باعث شده تا ترکیب‌های صفحه کلید بیشتری داشته باشید.



شکل ۱-۲۴

برگرداندن تغییرات اعمال شده

نگران به هم ریختن ویژوال استودیو با سفارشی کردن آن نباشید. همیشه روش‌هایی برای برگرداندن ویژوال استودیو به حالت قبلی وجود دارد.

بازنشانی چیدمان پنجره‌ها

دستور Reset Window Layout در منوی Window همه پنجره‌ها را به مکان اصلی آن‌ها بر می‌گرداند. این دستور وقتی مفید است که چند پنجره را در مکان‌های نامناسب قرار داده و IDE به هم ریخته شده است.

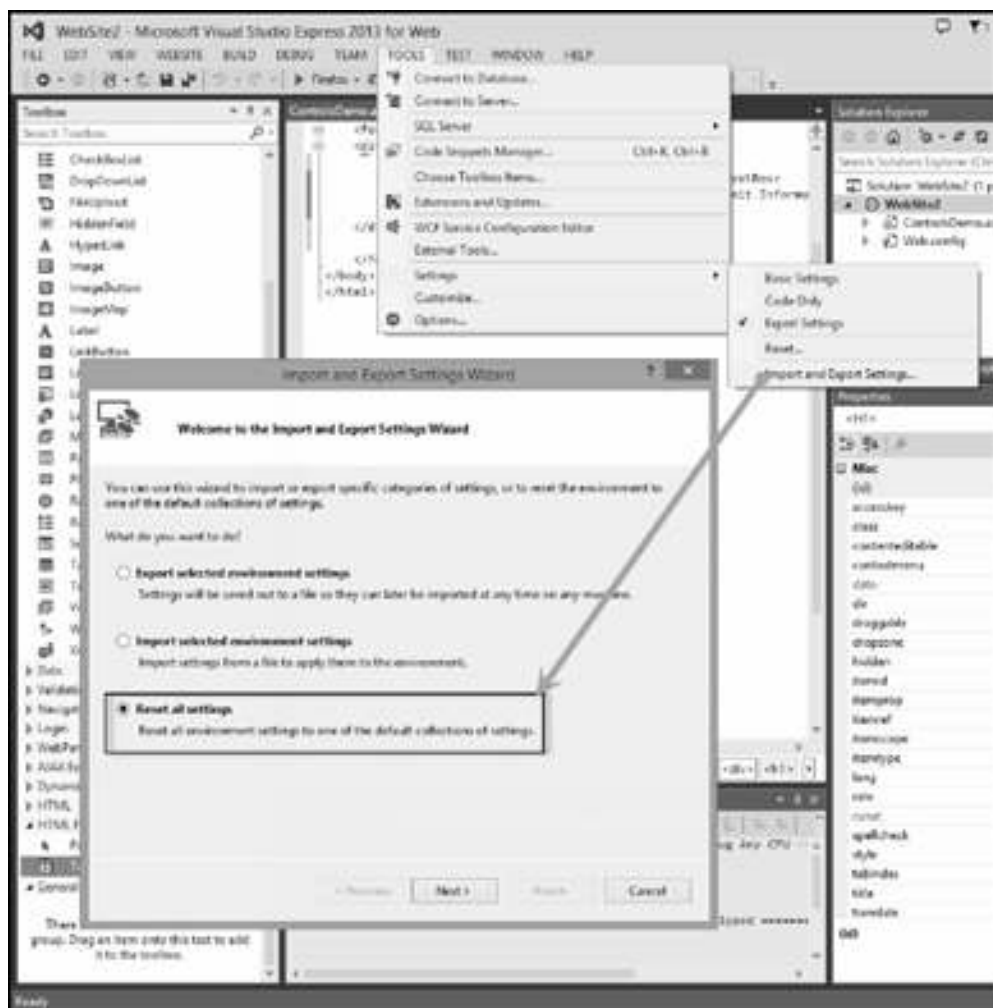
بازنشانی Toolbox

برای بازنشانی Toolbox، روی آن کلیک راست کرده و Reset Toolbox را انتخاب کنید. قبل از این کار دقت کنید که همه اسنیپت‌های کد شما حذف خواهند شد.

بازنشانی همه تنظیمات

برای این که همه تنظیمات ویژوال استودیو را به حالت اولیه آن‌ها باز گردانید، دستور Tools» Settings» Import and Export Settings را انتخاب کنید (یا بسته به نسخه ویژوال استودیو شما ممکن است Tools» Import and Export Settings باشد). سپس گزینه Reset All Settings را انتخاب کرده و روی Next کلیک کنید (شکل ۱-۲۵). اگر بخواهید یک پشتیبان از تنظیمات کنونی ایجاد کنید، می‌توانید از

گزینه Export برای صادر کردن تنظیمات استفاده کرده و سپس برای برگرداندن آن به این مکان باز گشته و از گزینه Import استفاده کنید.



شکل ۱-۲۵

یک برنامه مثال

این کتاب درباره ایجاد وب سایت است و بنابراین منطقی است که این کتاب دارای یک سایت نمونه کامل و مثال باشد که برای نشان دادن بسیاری از قابلیت‌های ASP.NET استفاده شود.

این سایت مثال، دارای نام Planet Wrox است که به عنوان یک انجمن آنلاین برای کسانی که به موسیقی علاقه دارند عمل می‌کند. این سایت امکانات زیر را به بازدیدکنندگان می‌دهد:

◀ دیدن آلبوم‌ها و کنسرت‌هایی که توسط ناظر روی سایت ارسال شده‌اند.

**PDF Compressor Free Version**

- ◀ بخش Gig Pics که یک آلبوم عکس آنلاین است که کاربران می‌توانند تصاویر گرفته شده در کنسرت را به اشتراک بگذارند.
 - ◀ توانایی سوییچ بین تم‌های گرافیکی مختلف که سایت پیشنهاد می‌دهد، امکان تغییر ظاهر و حس سایت را بدون تغییر محتویات آن می‌دهد.
 - ◀ امکان ذخیره تنظیمات موسیقی که روی اطلاعاتی که کاربران روی سایت می‌بینند تأثیر می‌گذارد.
 - ◀ دسترسی به امکانات بیشتر برای کاربران ثبت نام شده
 - این سایت به ناظران (یعنی دارنده سایت) اجازه انجام کارهای زیر را می‌دهد:
 - ◀ اضافه و نگهداری از نظرات
 - ◀ مدیریت سبک‌های موسیقی مختلف درون سیستم
 - ◀ مدیریت آلبوم‌های تصاویر ایجاد شده توسط بازدیدکنندگان سایت
- شکل ۱-۲۶ صفحه خانه Planet Wrox را نشان می‌دهد.



شکل ۱-۲۶

شکل ۱-۲۷ صفحه دیگری از Planet Wrox را نشان می‌دهد که این بار تم دیگری روی آن اعمال شده است. این صفحه به کاربران اجازه وارد کردن اطلاعات شخصی و تنظیمات خود را با توجه به سبک موسیقی مورد علاقه آن‌ها را می‌دهد.



شکل ۱-۲۷

می‌توانید یک مثال آنلاین از این سایت را در www.planetwrox.com ببینید. در این جا می‌توانید مشابه با یک بازدید کننده از سایت بازدید کنید.

می‌توانید منبع برای برنامه مثال و دیگر مثال‌های این کتاب را نیز از وب سایت Wrox در آدرس زیر پیدا کنید:
www.wrox.com/go/begaspnet451

در انتهای این کتاب قادر به ایجاد همه عملکردهای این سایت (و احتمالاً عملکردهای به مراتب بیشتر) درون سایت خود خواهید بود.

PDF Compressor Free Version