

مرجع آموزشی برنامه‌نویسی
ASP.NET 4.5 In C#

از مبتدی تا پیشرفته

ترجمه:

سید منصور عمرانی، مهدی کرامتی

انتشارات پندار پارس

انتشارات پنا

دفتر فروش: انا
تلفن: 572335



نام کتاب : مرجع آموزشی برنامه‌نویسی ASP.NET 4.5 in C# از مبتدی تا پیشرفته

ناشر : انتشارات پندار پارس

ترجمه و تالیف : سید منصور عمرانی، مهدی کرامتی

چاپ نخست : اردیبهشت 92

شمارگان : 1000 نسخه

طرح جلد : فرزانه روزبهرانی

لیتوگرافی، چاپ، صحافی : ترام‌سنج، فرشویه، خیام

قیمت : 45000 تومان به همراه DVD شابک : 978-600-6529-37-0



*هرگونه کپی برداری، تکثیر و چاپ کاغذی یا الکترونیکی از این کتاب بدون اجازه ناشر تخلف بوده و پیگرد قانونی دارد *

پیش‌گفتار

کتابی که در دست دارید ترجمه‌ی کتاب ASP.NET 4.5 in C# from Novice to Professional از انتشارات Apress است که در سال 2012 منتشر شده است. البته کتاب لاتین یاد شده برای نخستین بار تالیف نشده و در واقع ویرایش نسخه‌ی پیشین است که برای NET 4.5. به‌روز شده است. نسخه‌های قبلی این کتاب تا اولین نسخه‌ی آن که در سال 2006 برای ASP.NET 2.0 نوشته شد نیز همین داستان ویرایش مجدد و به‌روز رسانی را داشتند. طبعاً این ماجرا هم ادامه خواهد داشت. یعنی با انتشار هر نسخه‌ی جدید NET. می‌توانید خودتان را آماده‌ی مشاهده‌ی نسخه‌ی جدید کتاب ASP.NET in C# from Novice to Professional کنید.

نگاهی بر کتاب لاتین

هرچه باشد این کتاب واقعاً ارزشمند است و حقیقتاً باید به مولف کتاب متیو مک‌دونالد تبریک گفت. زیرا توانسته تمام آنچه را که یک نفر برای یادگیری ASP.NET نیاز دارد در یک کتاب جمع کند. سطح کتاب همان طور که در عنوان آن بیان شده از مبتدی تا پیشرفته است و مولف نیز حقیقتاً به این عنوان وفا کرده است. به طوری که برای استفاده از کتاب نیازی نیست قبلاً برنامه‌نویسی کرده یا با C# آشنا باشید. زیرا دو فصل از کتاب (فصل 2 و 3) به همین مبحث اختصاص دارد و با مطالعه‌ی آن می‌توانید دانش لازم برای برنامه‌نویسی به زبان C# را فرا بگیرید. اما اگر با اصول کلی وب، زبان C# و NET. تا حدی آشنا هستید می‌توانید خواندن کتاب را از فصل 4 شروع کنید و 100 صفحه در خواندن کتاب صرفه‌جویی کنید.

سبک آموزش و بیان مطالب توسط مولف بسیار شیوا است و ابهامی در آن وجود ندارد. فصل‌بندی کتاب نیز مناسب است و در هر فصل حق مطلب نسبت به موضوع فصل ادا می‌گردد. در حقیقت بدون اغراق این کتاب از نظر پوشش دادن مطالب فنی و تخصصی، بی‌نقص است و تمام نکات و مسائلی را که یک برنامه‌نویس ASP.NET باید در خصوص ASP.NET Web Forms بداند، از نکات ساده گرفته تا نکات تخصصی و پیشرفته، پوشش می‌دهد و خواننده را قدم به قدم با قابلیت‌های مختلف ASP.NET آشنا می‌کند. سبک بیان مطالب نیز به طور عملی است و در سرتاسر کتاب مثال‌های بیشماری از کوچک گرفته تا بزرگ برای آشنایی بیشتر خواننده بیان شده است.

به همین دلیل پس از پایان کتاب دیدی کاملاً جامع نسبت به ASP.NET، جزییات و تمام قابلیت‌های آن پیدا می‌کنید و تمام آنچه را که برای ساخت یک سایت حرفه‌ای ASP.NET نیاز دارید به دست می‌آورید. لذا می‌توانید مطمئن باشید به راحتی می‌توانید یک سایت ASP.NET برای خودتان درست کنید و به یک برنامه‌نویس حرفه‌ای ASP.NET که با ریزه‌کاری و اصول برنامه‌نویسی ASP.NET Web Forms آشنا است بدل شوید.

با توجه به این نکات بود که تصمیم گرفتیم این کتاب را برای شما ترجمه کنیم. در ترجمه‌ی کتاب نیز سعی کردیم تمام توانایی خود را برای روان‌تر شدن و خواناتر شدن متن به کار ببریم و کتابی برای شما فراهم کنیم که خواندن آن راحت باشد. به همین دلیل در برخی موارد مجبور شدیم متن اصلی کتاب را تغییر بدهیم و آن را به گونه‌ای که برای خواننده‌ی ایرانی روشن‌تر است بیان کنیم. با این وجود در برخی موارد نکات مهمی را مشاهده کردیم که نیازمند توضیح بیشتر بود. این توضیحات را نیز به شکل پاورقی گنجانده‌ایم. لذا مطالب و نکته‌های مفیدی در پاورقی‌های کتاب مشاهده خواهید کرد.

ویرایش‌های قبلی کتاب لاتین و سیر تغییرات آن

نکته‌ی دیگری که لازم است در مورد آن توضیح بدهیم این است که مولف کتاب لاتین برای به‌روز کردن آن در هر نسخه از NET. معمولاً قسمت‌هایی از ویرایش قبلی کتاب را حذف نموده تا بتواند مطالب دیگری مربوط به نسخه‌ی جدید NET. در آن بگنجانند. به عنوان مثال در اولین نسخه‌ی کتاب یعنی نسخه‌ی ASP.NET 2.0 سه فصل به وب‌سرویس‌ها و یک فصل به وب‌پارته‌ها اختصاص داشت که در نسخه‌ی ASP.NET 3.5 حذف شد. زیرا در NET. 3.5 بحث وب‌سرویس‌ها به سکوی WCF منتقل شد. وب‌پارته‌ها نیز به دلیل استقبال کم حذف شد. در عوض مولف فصل دیگری برای ASP.NET AJAX و همچنین برنامه‌نویسی گرافیک به کتاب اضافه کرد. همچنین در ASP.NET 2.0 مبحثی به کنترل‌های اشتقاقی (Derived Controls) اختصاص داشت که حذف شد. اما از آن سو مولف توضیح مفصلی در مورد CSS به کتاب اضافه کرد و تغییرات زیادی به فصل «مقدمات ADO.NET» اعمال کرد که البته به عقیده‌ی ما کارایی آن را کاهش داد.

در نسخه‌ی ASP.NET 4.0 کتاب چندان فرقی نکرد و تنها بحث مسیریابی آدرس‌های URL، Machine Key و SSL به آن اضافه شد. البته یک فصل مهم در مورد LINQ و Entity Framework نیز به آن اضافه شد که ارزش کتاب را بیشتر کرد. اما متأسفانه باز هم فصل «ADO.NET» کوتاه‌تر شد و این کوتاه شدن به فصل «XML» نیز سرایت کرد! ضمن این که قابلیت جعل هویت (Impersonation) نیز از کتاب حذف شد. در نسخه‌ی ASP.NET 4.5 نیز تقریباً می‌توان گفت کتاب نسبت به نسخه‌ی ASP.NET 4.0 دستخوش تغییر جدی نشد و تنها بحث «HTML5» و کنترل Chart به کتاب اضافه شد. در عوض قابلیت استفاده از لاگ رویدادهای ویندوز از کتاب حذف شد.

اما یکی از اختلاف‌های بزرگ ویرایش‌های مختلف کتاب لاتین، به فصل «انتشار» یا Publish کردن سایت‌ها مربوط می‌شود. در ASP.NET 2.0 مولف نحوه‌ی این کار را در IIS 6.0 و Windows Server 2003 توضیح داده بود. در ASP.NET 3.5 نحوه‌ی این کار در IIS 7.0 و Windows Server 2008 را نیز به کتاب افزود. اما در ASP.NET 4.0 انتشار سایت‌ها در IIS 6.0 و Windows Server 2003 را از کتاب حذف کرد و فقط به IIS 7.0 و Windows Server 2008 بسنده کرد. در ASP.NET 4.5 نیز این مطالب را به IIS 7.5 و Windows Server 2012 ارتقا داد.

ما چطور ترجمه کردیم؟

اگر می‌خواستیم محتوای حذف شده در ویرایش‌های قبلی کتاب را نگه داریم طبیعی است که تعداد صفحات کتاب به جای 1000 صفحه به بیش از 1500 صفحه افزایش پیدا می‌کرد. در نتیجه کتاب دیگر در یک جلد قابل چاپ نبود. ضمن این که قیمت آن را به میزان چشمگیری افزایش می‌داد. لذا ما نیز هنگام ترجمه ناگزیر شدیم از مطالب قدیمی‌تر چشم‌پوشی کنیم. اما در عین حال سعی کردیم برخی از تغییرات را با پاورقی توضیح بدهیم. در صورت علاقمندی به مطالبی که از ویرایش‌های قبلی کتاب لاتین حذف شده به آن نسخه‌ها مراجعه کنید.

ارزیابی کلی

اما به طور کلی محتوای کتاب بین ویرایش‌های مختلف آن چندان فرقی ندارد. می‌توانید با مقایسه‌ی فهرست کتاب‌ها این را ببینید. این مساله چیز عجیبی هم نیست و از یک واقعیت مهم نشات می‌گیرد: سکوی ASP.NET Web Forms از ابتدا در اولین نسخه‌ی خود (ASP.NET 1.1) بر اساس چهارچوب بسیار محکمی طراحی شد که نیاز چندین سال در آن دیده شده بود. لذا ASP.NET Web Forms تا امروز آزمون خود را به طور کامل و البته به خوبی پس داده است و آنقدرها چیز جدیدی وجود ندارد که دیگر بشود به آن اضافه کرد. ممکن است بعضی‌ها از این مطلب خوششان نیاید؛ اما این واقعیت محض است.

اما آیا این چیز بدی است؟ به هیچ وجه. اتفاقاً ثابت بودن و تغییر نکردن چیز بسیار خوبی است. زیرا نیازی نیست مدام خود را به‌روز نگه دارید. در واقع بدون اغراق باید بگوییم اگر ASP.NET 2.0 را خوب بلد باشید می‌توانید مطمئن باشید ASP.NET 3.5، ASP.NET 4.0 و ASP.NET 4.5 را هم از قبل بلدید! زیرا فرق زیادی با هم ندارند. چیزی که بین این نسخه‌ها تغییر می‌کند سکوی NET. و قابلیت‌های زبان‌های آن است.

خاتمه

ما از خواندن این کتاب لذت بردیم و مطالب آن برای خود ما مفید بود. امیدواریم شما هم از خواندن کتاب لذت ببرید و همان چیزی باشد که دنبالش بودید. در صورتی که هر انتقاد، نظر یا سوالی در رابطه با محتوای کتاب داشتید می‌توانید از طریق آدرس‌های زیر با مترجم و ویراستار فنی کتاب در تماس باشید:

mansoor.omrani@yahoo.com,

mehdi.keramati@yahoo.com

همچنین در سایت برنامه‌نویس به آدرس <http://barnamenevis.org> یک تالار گفتگوی اختصاصی به ASP.NET وجود دارد که می‌توانید سوالات خود را در آن مطرح کنید. ضمن این که می‌توانید با شرکت در دوره‌های حضوری آموزش ASP.NET که تبلیغات آن در سایت برنامه‌نویس درج شده است، ASP.NET را به طور حضوری فرا بگیرید. در انتها برای شما عزیزان آرزوی موفقیت و سربلندی داریم و امیدواریم این کتاب مورد توجه جامعه‌ی برنامه‌نویسان ASP.NET قرار بگیرد.

سید منصور عمرانی

مهدی کرامتی

1	فصل 1. چهارچوب کاری .NET.
1	سیر تکامل برنامه‌نویسی وب
2	HTML و فرم‌های HTML
4	برنامه‌نویسی سمت سرویس‌دهنده
5	برنامه‌نویسی سمت سرویس‌گیرنده
8	چهارچوب کاری .NET.
9	.NET، VB.NET و زبان‌های .NET.
10	زبان میانی
12	ماشین اجرای زبان مشترک یا CLR
14	کتابخانه‌ی کلاس‌های .NET.
15	Visual Studio
16	کلام آخر
17	فصل 2. زبان C#
17	زبان‌های .NET.
18	اصول زبان C#
18	حساسیت به حروف
19	کامنت‌ها
20	انتهای دستورات
21	بلاک‌ها
21	متغیرها و انواع داده
24	دستورات نسبت‌دهی و مقداردهی اولیه‌ی متغیرها
25	رشته‌ها و کاراکترهای گریز
26	آرایه‌ها
28	ArrayList
28	نوع داده‌ی شمارش‌پذیر (Enumeration)
30	عملگرها
31	ریاضیات پیشرفته
32	تبدیل نوع
34	دستکاری شیء‌گرا
35	نوع رشته یا String
37	کلاس DateTime و TimeSpan
39	نوع آرایه یا Array
41	دستورات شرطی
42	دستور if
43	دستور switch
44	حلقه‌ها
44	حلقه‌ی for
45	حلقه‌ی foreach
46	حلقه‌ی while
47	متدها
49	پارامترها
49	سر بارگذاری متدها
50	پارامترهای اختیاری و پارامترهای نام‌دار
52	نماینده‌ها
54	کلام آخر
55	فصل 3. نوع داده‌ها، اشیاء و فضاهای نام
55	میانی کلاس‌ها

57	اعضای استاتیک
57	تعریف یک کلاس نمونه
58	تعریف یک کلاس ساده
59	ایجاد نمونه‌ای از یک کلاس
61	خصوصیت‌ها
63	خصوصیت‌های خودکار
64	متدها
65	سازنده‌ها
67	رویدادها
69	آزمایش کلاس Product
71	نوع داده‌های مقداری و نوع داده‌های ارجاعی
71	نسبت‌دهی
72	مقایسه
73	ارسال آرگومان‌ها بر اساس مقدار و بر اساس ارجاع
75	مروری بر نوع‌ها در .NET
76	فضاهای نام و اسمبلی‌ها
78	نحوه‌ی استفاده از فضاهای نام
79	Import کردن فضاهای نام
80	اسمبلی‌ها
82	برنامه‌نویسی پیشرفته‌ی کلاس‌ها
82	ارث‌بری
84	اعضای استاتیک
85	تبدیل اشیاء به یکدیگر
89	کلاس‌های تک تکه
90	ژنریک‌ها
92	کلام آخر
93	فصل 4. Visual Studio
93	وعده‌ی Visual Studio
95	ایجاد سایت‌های وب
96	ایجاد یک برنامه‌ی وب خالی
100	سایت‌های وب و پروژه‌های وب
101	فایل‌های Solution
102	پنجره‌ی Solution Explorer
104	ایجاد یک فرم یا صفحه‌ی وب
106	پشتیبانی از سکوها‌ی مختلف
107	ارتقاء سایتی که توسط نسخه‌ی قدیمی‌تری از Visual Studio ساخته شده است
108	طراحی یک صفحه‌ی وب
109	اضافه کردن کنترل‌های وب
111	پنجره‌ی مشخصات
114	اسکلت یک فرم وب
114	تگ‌های فرم وب
115	دایرکتیو Page
116	نوع سند یا DocType
117	اصول و قوانین HTML
118	المان‌ها
121	ویژگی‌ها
122	قالب‌بندی

122	یک صفحه‌ی وب کامل
124	نوشتن کد صفحه
124	کلاس Code-Behind
125	اضافه کردن اداره‌گر رویداد
127	Outlining
127	IntelliSense
128	لیست اعضای کلاس
129	علامت زدن خطاها به طور خودکار
130	خطاهای HTML
132	الحاق فضاهای نام به طور خودکار
133	قالب‌بندی و رنگ‌بندی خودکار
133	قابلیت اشکال‌زدایی در Visual Studio
134	وب‌سرور IIS Express
136	اشکال‌زدایی خط به خط
139	بررسی مقدار متغیرها
141	کلام آخر
143	فصل 5. اصول و مبانی فرم‌های وب
143	ساختار کلی یک برنامه‌ی ASP.NET
145	انواع فایل‌های ASP.NET
146	دایرکتوری‌های اصلی یک برنامه‌ی ASP.NET
147	معرفی کنترل‌های سمت سرور
149	کنترل‌های HTML سمت سرور
149	تبدیل یک صفحه‌ی HTML به یک صفحه‌ی ASP.NET
152	View State
153	کلاس کنترل‌های HTML
156	نوشتن برنامه‌ی میدل رز
160	اداره کردن رویداد
162	پشت صحنه‌ی برنامه‌ی میدل ارز
164	اداره کردن خطا
165	بهبود برنامه‌ی میدل ارز
165	اضافه کردن چندین نوع ارز دیگر
166	کالبدشکافی کد
167	گنجاندن اطلاعات در لیست
169	اضافه کردن تصویر
171	اعمال شیوه‌های CSS
172	نگاهی عمقی به کنترل‌های HTML سرویس‌دهنده
173	رویدادهای کنترل‌های HTML
174	رویدادهای پیشرفته و بررسی کنترل HtmlInputImage
176	کلاس پایه‌ی HtmlControl
177	کلاس HtmlContainerControl
178	کلاس HtmlInputControl
178	کلاس Page
180	هدایت کاربر به یک صفحه‌ی دیگر
182	رمز کردن رشته‌های HTML
185	رویدادهای یک برنامه‌ی ASP.NET
185	فایل Global.asax
187	سایر رویدادهای کلی یک برنامه‌ی وب
187	پیگر بندی برنامه‌های ASP.NET
188	فایل web.config

190	پیگردینی تو در تو
192	ذخیره کردن تنظیمات شخصی در فایل web.config
195	ابزار مدیریت وب سایت یا WAT
196	کلام آخر
197	فصل 6. کنترل های وب
197	پیش به سوی کنترل های وب
198	کنترل های پایه ای وب
200	نحوه ی تعریف کنترل های وب در صفحات.aspx
201	درختواره ی کلاس های کنترل های وب
202	کلاس پایه ای WebControl
204	واحد اندازه گیری
205	خصوصیت های شمارش پذیر
205	رنگ ها
206	فونت ها
209	تمرکز
210	دکمه ی پیش فرض
211	نحوه ی تعریف اداره گر رویداد
212	کنترل های لیستی
213	کنترل های لیستی چند انتخابی
214	کنترل BulletedListControl
217	کنترل جدول
221	رویدادهای کنترل های وب و قابلیت پس فرستاده شدن خودکار صفحه یا AutoPostBack
224	اساسا PostBack در ASP.NET چگونه کار می کند؟
225	چرخه ی حیات یک صفحه ی ASP.NET
228	کالبدشکافی برنامه
228	یک برنامه ی ساده
232	بهبود برنامه ی کارت تبریک
235	تولید خودکار کارت های تبریک
236	کلام آخر
237	فصل 7. اداره کردن خطا، ردگیری و لاگ گیری
237	خطاهای رایج
240	اداره کردن استثناء
240	کلاس Exception
243	زنجیره ی استثناء
243	اداره کردن استثناء
245	به دام اندازی استثناء های خاص
246	اداره گرهای استثناء تو در تو
248	کالبدشکافی برنامه ...
248	مثالی عملی از اداره ی خطا
249	مهارت در کار با استثناء
250	تولید استثناء به طور عمدی
254	لاگ گیری از خطاها
255	مشاهده ی لاگ های رویدادهای ویندوز
258	نحوه ی ثبت لاگ در Windows Events Logs
259	ایجاد لاگ های شخصی در Event Logs
260	نوشتن یک کلاس لاگ گیری شخصی
262	نحوه ی خواندن لاگ های ثبت شده در Event Logs
265	ردگیری روند پردازش و اجرای صفحات
265	فعال کردن ردگیری در صفحات ASP.NET

266	اطلاعات ردگیری
267	قسمت جزئیات درخواست (Request Details)
267	قسمت اطلاعات ردگیری (Trace Information)
268	قسمت درختواره‌ی کنترل‌ها (Control Tree)
269	قسمت وضعیت جلسه و برنامه (Session State/Application State)
269	قسمت Request Cookies and Response Cookies
270	قسمت کلکسیون هدرها (Headers Collection)
270	قسمت کلکسیون اطلاعات فرم (Form Collection)
271	قسمت کلکسیون رشته‌ی پرس و جو (Querystring Collection)
271	قسمت متغیرهای سرور (Server Variables)
271	اضافه کردن اطلاعاتی به لیست اطلاعات ردگیری
274	ردگیری در سطح برنامه
276	کلام آخر
277	فصل 8. مدیریت وضعیت
277	مشکل وضعیت
278	View State
278	کلکسیون ViewState
279	مثالی از View State
280	ایمن کردن View State
280	جلوگیری از دستکاری View State توسط دیگران
281	خصوصی کردن View State
282	حفظ مقادیر متغیرهای عضو
284	ذخیره کردن اشیاء در View State
285	منتقل کردن اطلاعات از یک صفحه به صفحه‌ی دیگر
285	ارسال اطلاعات به یک صفحه‌ی دیگر یا Cross-Page Postback
287	به دست آوردن اطلاعات بیشتری درباره‌ی صفحه‌ی مبدا
291	رشته‌ی QueryString
293	مثالی از استفاده از QueryString
294	رمز کردن آدرس‌های URL
296	کوکی‌ها
297	مثالی از به کارگیری کوکی در یک صفحه‌ی وب
298	وضعیت جلسه
299	ردگیری Session
300	نحوه‌ی استفاده از Session
302	مثالی از استفاده از Session در یک صفحه‌ی وب
304	تنظیمات Session
305	Cookieless
309	Timeout
310	Mode
314	فشرده‌سازی
315	وضعیت برنامه و شیء Application
317	مروری بر شیوه‌های مختلف مدیریت وضعیت
319	کلام آخر
321	فصل 9. اعتبارسنجی
321	درک مفهوم اعتبارسنجی
323	کنترل‌های اعتبارسنج
324	اعتبارسنجی سمت سرویس‌دهنده
325	اعتبارسنجی سمت سرویس‌گیرنده
325	اعتبارسنجی در HTML5

327	کلاس کنترل‌های اعتبارسنج
329	یک مثال ساده از اعتبارسنجی
331	حالات مختلف خصوصیت Display در کنترل‌های اعتبارسنج
334	اعتبارسنجی دستی
336	معتبرسازی بر اساس عبارات با قاعده
337	لفظها و فرا کاراکترها
338	تکرار منتهای
338	تکرار نامنتهای
339	تعریف بازهای از کاراکترها
339	معتبرسازی بر اساس عبارات با قاعده
342	صفحه‌ی عضویت یک سایت تجارت الکترونیک با قابلیت اعتبارسنجی
346	گروه‌های اعتبارسنجی
348	کلام آخر
349	فصل 10. کنترل‌های پیشرفته
349	کنترل تقویم (Calendar)
351	تنظیم شکل ظاهری کنترل تقویم
353	روزهای استثناء
358	کنترل تبلیغ‌گردان (AdRotator)
359	نحوه‌ی تعریف تبلیغ‌ها
360	کلاس AdRotator
362	صفحاتی با چندین نما
364	کنترل MultiView
365	ایجاد نماها
366	نمایش یکی از نماها
369	کنترل Wizard
369	مراحل ویزارد
373	رویدادهای Wizard
374	تنظیم ظاهر Wizard
376	کنترل Wizard و اعتبارسنجی
377	کلام آخر
379	فصل 11. کنترل‌های کاربر و برنامه‌نویسی گرافیک
379	کنترل‌های کاربر
380	نحوه‌ی ساخت یک کنترل کاربر
383	کنترل‌های کاربر مستقل
385	کنترل‌های کاربر مجتمع
387	رویدادهای کنترل‌های کاربر
391	پاس دادن اطلاعات اضافی به اداره‌گر رویدادهای کنترل‌های کاربر
394	برنامه‌نویسی گرافیک
395	اصول و مبانی نقاشی
398	ترسیم متن
400	نمایش تصاویر ترسیم شده درون صفحات وب
402	قالب‌های تصویری و کیفیت تصویر
405	کنترل Chart
405	ایجاد یک نمودار ساده
407	روش‌های دیگر مشخص کردن داده‌های نمودار
408	کلام آخر
409	فصل 12. شیوه‌ها، تم‌ها و صفحات الگو
410	شیوه‌ها
410	انواع شیوه‌ها
411	ایجاد یک شیوه‌ی درون خطی ساده

412	ابزار شیوه‌ساز
416	پنجره‌ی CSS Properties
417	وراثت شیوه‌ها
418	ایجاد شیوه‌نامه
421	پنجره‌ی CSS Outline
422	نحوه‌ی استفاده از شیوه‌نامه
423	پنجره‌ی Apply Style
424	ایجاد شیوه‌های بیشتر
424	تم‌ها
425	نحوه‌ی کار تم‌ها
427	نحوه‌ی ایجاد تم و پوسته در Visual Studio
428	مدیریت تداخل تم‌ها
430	ایجاد چندین پوسته برای یک کنترل
432	پوسته‌های پیشرفته
434	صفحات الگو
435	یک صفحه‌ی الگو و صفحه‌ی محتوای ساده
438	نحوه‌ی ارتباط صفحات الگو و محتوا با یکدیگر
440	الگوی یا چند ناحیه‌ی محتوا
443	محتوای پیش فرض
444	الگوها و مسیرهای نسبی
446	صفحات الگوی پیشرفته
446	طرح‌بندی شیوه‌ای
452	صفحات الگو و code-behind
452	تعامل با الگو از طریق کُد
454	کلام آخر
455	فصل 13. پیمایش سایت
455	نقشه‌ی سایت
456	نحوه‌ی تعریف نقشه‌ی سایت
461	مشاهده‌ی یک نقشه‌ی سایت در عمل
461	نحوه‌ی استفاده از نقشه‌ی سایت
463	قرار دادن نقشه‌ی سایت در صفحات الگو
465	نمایش قسمتی از نقشه‌ی سایت
465	نمایش زیر درخت‌ها
470	استفاده از نقشه سایت‌های مختلف در یک فایل
471	کلاس SiteMap
473	نگاشت آدرس‌های URL و مسیریابی
474	نگاشت آدرس‌های URL
475	مسیریابی
477	کنترل SiteMapPath
478	شخصی‌سازی کنترل SiteMapPath
479	استفاده از الگوها و شیوه‌ها در کنترل SiteMapPath
481	اضافه کردن اطلاعات شخصی به نقشه‌ی سایت
483	کنترل TreeView
483	خصوصیت‌های کنترل TreeView
486	شیوه‌های کنترل TreeView
487	اعمال شیوه‌ها به انواع گره‌های مختلف در TreeView
488	اعمال شیوه‌ها به گره‌ها
490	کنترل Menu
491	شیوه‌های مورد استفاده در کنترل Menu

493	استفاده از الگوها در کنترل Menu
496	سخن آخر
497	فصل 14. مبانی ADO.NET
497	درک مفهوم پایگاه داده
499	تنظیم کردن پایگاه داده
500	SQL Server Express
500	مشاهده‌ی پایگاه داده و تغییر دادن اطلاعات آن توسط Visual Studio
503	ابزار خط فرمان sqlcmd
505	اصول و مبانی SQL
506	اجرای پرس و جوهای SQL در محیط Visual Studio
507	دستور SELECT
508	نمونه‌ای از یک دستور SELECT
508	بهبود دادن دستور SELECT
509	یک دستور SELECT دیگر
509	عبارت WHERE
509	مطابقت دادن مقادیر رشته‌ای توسط عملگر LIKE
510	پرس و جوهای انجمنی
511	دستور UPDATE
512	دستور INSERT
513	دستور DELETE
514	مدل فراهم آورنده‌ی داده
515	دسترسی مستقیم به داده‌ها
517	ایجاد اتصال به دیتابیس
518	رشته‌ی اتصال
519	Windows Authentication
520	اتصال‌های User Instance
522	نحوه‌ی تعریف رشته‌ی اتصال
522	برقراری اتصال
525	بازیابی یا انتخاب رکوردها
525	شیء DataReader
526	سر هم کردن تمام جزئیات
527	پر کردن DropDownList
529	بازیابی مشخصات کامل مولف
530	به‌روز رسانی رکوردها
531	نمایش مقادیر در جعبه‌های متن
532	ایجاد رکورد جدید
535	ایجاد فرمان‌های قدرتمندتر
537	به‌روز رسانی رکورد
538	حذف رکورد
540	دسترسی منفصل به داده‌ها
542	بازیابی رکوردها
543	بازیابی رکوردهای چندین جدول
545	تعریف روابط بین جداول
547	کلام آخر
549	فصل 15. اتصال به داده‌ها
549	معرفی اتصال به داده‌ها
550	انواع اتصال در ASP.NET
550	اتصال تک مقداری یا اتصال ساده
550	اتصال چند مقداری یا اتصال لیستی
551	اتصال به داده‌ها چگونه کار می‌کند؟

551	اتصال تک مقداری
552	مثال ساده‌ای از اتصال
554	اتصال خصوصیت‌ها
555	مشکلات اتصال تک مقداری
556	استفاده از کُد نویسی به جای اتصال تک مقداری
557	اتصال چند مقداری
558	اتصال با استفاده از کنترل‌های لیستی ساده
558	مثالی از اتصال به یک لیست ساده
560	کلکسیون‌های نوع‌دار
560	اتصال چندین کنترل به یک منبع داده
562	اتصال به دیکشنری
563	استفاده از خصوصیت DataValueField
565	اتصال به داده‌های ADO.NET
567	ساخت صفحه‌ی ویرایشگر رکورد
571	کنترل‌های منبع داده
573	چرخه‌ی حیات صفحه هنگام اتصال
574	کنترل SqlDataSource
576	بازیابی رکوردها
578	کنترل‌های منبع داده چگونه کار می‌کنند؟
579	فرمان‌های پارامتری
581	انواع پارامترها
582	تنظیم پارامترها از طریق کُد
585	اداره کردن خطا
585	به‌روز رسانی رکوردها
588	بررسی اثرات همروندی
591	کلام آخر
593	فصل 16. کنترل‌های داده‌ای
594	کنترل GridView
594	تولید خودکار ستون‌ها
596	تعریف ستون‌ها
598	تنظیمات ستون‌ها
599	ایجاد ستون‌ها به کمک Visual Studio
600	قالب‌بندی در GridView
601	قالب‌بندی فیلدها
603	به‌کارگیری شیوه‌ها
605	تنظیم شیوه‌ها از طریق Visual Studio
606	سطرهای یا قالب‌بندی خاص
608	انتخاب سطرها در GridView
608	اضافه کردن دکمه‌ی Select
610	استفاده از محتوای ستون‌های داده‌ای به عنوان دکمه‌ی انتخاب سطرها
610	استفاده از قابلیت انتخاب رکورد در GridView برای ایجاد یک فرم کل به جزء
613	ویرایش رکوردها توسط GridView
615	مرتب‌سازی و صفحه‌بندی
616	مرتب‌سازی
617	مرتب‌سازی و انتخاب رکوردها
618	صفحه‌بندی
620	صفحه‌بندی و انتخاب رکوردها
620	به‌کارگیری الگوها در GridView
622	استفاده از چندین الگو

623	تعریف الگوها توسط Visual Studio
624	اداره کردن رویدادها در الگوهای GridView
625	ویرایش رکوردها توسط الگوها در GridView
627	ویرایش و معتبرسازی
628	ویرایش بدون استفاده از ستون فرمان
629	کنترل FormView و DetailsView
630	کنترل DetailsView
631	تعریف فیلدهای قابل نمایش
633	کنترل FormView
635	کلام آخر
637	فصل 17. فایل‌ها و جریان‌ها
637	فایل‌ها و برنامه‌های وب
638	اطلاعات سیستمی فایل‌ها
639	کلاس Path
641	کلاس‌های File و Directory
645	کالبدشکافی برنامه
646	کلاس DirectoryInfo و FileInfo
648	کلاس DriveInfo
649	یک برنامه‌ی ساده‌ی مرورگر فایل
652	کالبدشکافی برنامه
653	خواندن و نوشتن فایل‌ها با استفاده از جریان‌ها
653	فایل‌های متنی
655	فایل‌های دودویی
657	میانبرهایی برای خواندن و نوشتن فایل‌ها
658	یک صفحه‌ی مهمان ساده
661	کالبدشکافی برنامه
662	آپلود فایل‌ها از کامپیوتر کاربر به سرور
662	کنترل FileUpload
665	کالبدشکافی برنامه
665	کلام آخر
667	فصل 18. XML
667	توضیح XML
669	توسعه‌ی لیست محصولات با استفاده از XML
671	اصول و مبانی XML
672	ویژگی‌ها
673	کامنت
673	کلاس‌های استفاده از XML در .NET
674	کلاس XmlTextWriter
676	کالبدشکافی برنامه
677	کلاس XmlTextReader
680	کالبدشکافی برنامه
682	کلاس XmlDocument و کار با اسناد XML در حافظه
686	کالبدشکافی برنامه
686	خواندن یک سند XML
689	جستجو در یک سند XML
690	اعتبارسنجی اسناد XML
691	فضاهای نام XML
694	نوشتن محتوای XML حاوی فضای نام
695	تعریف اسکیمای XML
696	کالبدشکافی اسکیمای
697	معتبرسازی یک فایل XML
699	نمایش اسناد XML و تبدیل آنها

702	کنترل Xml
703	کلام آخر
705	فصل 19. مبانی امنیت
705	تعیین نیازمندی‌های امنیتی
707	آزمایش و انتشار تنظیمات امنیتی
708	تصدیق هویت و تفویض اختیار
709	تصدیق هویت فرم
710	تنظیمات Web.config
711	قوانین تفویض اختیار
713	محدود کردن دسترسی به یک دایرکتوری خاص
714	کنترل دسترسی به فایل‌های به خصوص
715	محدود کردن دسترسی برای کاربران خاص
716	ابزار WAT
718	صفحه‌ی Login
721	بازبانی هویت کاربر
722	خروج از حساب کاربری
723	کوکی‌های ماندگار
724	تصدیق هویت ویندوز
725	تنظیمات Web.config
728	یک برنامه‌ی آزمایشی ساده در خصوص تصدیق هویت ویندوز
729	کلام آخر
731	فصل 20. عضویت
732	انبار داده‌های عضویت
733	استفاده از قابلیت عضویت در SQL Server Express
736	استفاده از نسخه‌ی کامل SQL Server
739	تنظیم فراهم آورنده‌ی عضویت
743	ایجاد کاربران توسط WAT
745	کلاس‌های Membership و MembershipUser
749	تصدیق هویت با استفاده از کلاس Membership
750	حساب‌های غیر فعال
751	کنترل‌های امنیتی
752	کنترل Login
758	کنترل CreateUserWizard
763	کنترل PasswordRecovery
765	امنیت مبتنی بر نقش
766	ایجاد و نسبت دادن نقش‌ها
768	محدود کردن دسترسی بر اساس نقش‌های مختلف
769	کنترل LoginView
771	کلام آخر
773	فصل 21. قابلیت پروفایل
774	مفهوم پروفایل
774	کارایی پروفایل‌ها
775	اطلاعات پروفایل چطور ذخیره می‌شود
777	نحوه‌ی استفاده از SqlProfileProvider
778	فعال کردن تصدیق هویت
779	استفاده از SQL Server Express
779	استفاده از نسخه‌ی کامل SQL Server
781	پایگاه داده‌ی پروفایل
782	تعریف خصوصیت‌های داده‌ای پروفایل

784	نحوه‌ی استفاده از خصوصیت‌های پروفایل
786	سریال‌سازی پروفایل
789	گروه‌بندی خصوصیت‌های پروفایل
789	پروفایل و نوع داده‌های شخصی
791	کالبدشکافی برنامه:
791	سریال‌سازی نوع داده‌های شخصی
792	ذخیره‌سازی خودکار پروفایل‌ها
793	لیست متدهای پروفایل
796	پروفایل‌های ناشناس
798	منتقل کردن پروفایل‌های ناشناس
799	کلام آخر
801	فصل 22. برنامه‌نویسی مولفه‌ای
801	چرا باید از مولفه استفاده کنیم؟
803	اصولا مولفه یعنی چه؟
803	طراحی سه لایه
804	کیسوله‌سازی
805	اشیاء تجاری یا کسب و کار
805	اشیاء داده‌ای
805	مولفه‌ها و کلاس‌ها
806	نحوه‌ی ساخت یک مولفه
807	کلاس‌ها و فضاها‌ی نام
809	اعضای کلاس
809	اضافه کردن ارجاعی به یک مولفه
812	نحوه‌ی استفاده از مولفه
814	خصوصیت‌ها و وضعیت
815	طراحی یک کلاس وضعیت‌دار برای حساب بانکی
816	طراحی یک کلاس بدون وضعیت برای انجام کلیه‌ی امور حساب بانکی
817	مولفه‌های دسترسی به داده
817	یک مولفه‌ی ساده‌ی دسترسی به داده
820	کالبدشکافی برنامه
821	نحوه‌ی استفاده از مولفه‌ی دسترسی به داده
823	کالبدشکافی برنامه
824	بهبود دادن مولفه با مدیریت بهتر خطاها
825	بهبود دادن مولفه با برگرداندن اطلاعات محاسباتی
826	منبع داده‌ی ObjectDataSource
827	ایجاد کلاس‌هایی که شیء ObjectDataSource آنها را درک کند
827	بازیابی رکوردها
828	استفاده از متدهای پارامتردار
830	به‌روز رسانی رکوردها
833	کلام آخر
835	فصل 23. قابلیت کش
835	درک مفهوم کش کردن اطلاعات
836	چه موقع از کش استفاده کنیم؟
837	کش کردن اطلاعات در ASP.NET
839	کش کردن خروجی
840	کش سمت کلاینت
841	کش کردن صفحه و Query String
843	کش کردن صفحه بر اساس پارامترهای به خصوصی در QueryString
844	مثالی از صفحه‌ای با چندین نسخه‌ی کش شده
845	کش کردن قطعه‌ای

846	پروفایل‌های کش
847	کش کردن داده‌ها
848	اضافه کردن اقلام داده‌ای به Cache
849	مثال ساده‌ای از کش داده‌ای
850	استفاده از کش برای فراهم کردن نماهای مختلف
853	استفاده از قابلیت کش در خصوص کنترل‌های منبع داده
854	استفاده از قابلیت کش در کنترل SqlDataReader
857	استفاده از قابلیت کش در کنترل ObjectDataSource
858	تعیین وابستگی برای آیتم‌های کش شده
858	وابستگی فایلی
859	وابستگی‌های به داده‌های درون کش
859	وابستگی کش به SQL Server
861	فعال کردن Service Broker
862	راه‌اندازی سرویس کش
862	ایجاد وابستگی کش دینامیکی
863	کلام آخر
865	فصل 24. LINQ و Entity Framework
865	درک مفهوم LINQ
866	مبانی LINQ
869	کالبدشکافی برنامه
870	عبارت‌های LINQ
871	انتخاب ستون‌های داده‌ای
873	فیلترسازی و مرتب‌سازی
874	Entity Framework
876	نحوه‌ی ایجاد یک Entity Data Model
879	دیاگرام مدل
883	به‌روز رسانی مدل
883	کد تولید شده برای مدل
884	موجودیت‌ها
885	Context
886	بازیابی داده‌ها از مدل
886	اداره کردن خطاها
889	پیمایش روابط
891	انجام عملیات پیشرفته‌تر با استفاده از Entity Framework
892	بازیابی داده‌ها با استفاده از LINQ to Entities
895	کنترل کردن زمان بارگذاری داده‌ها
897	درج، حذف، به‌روز رسانی
900	مدیریت همروندی
901	کنترل EntityDataSource
901	نمایش اطلاعات
905	وبرایش رکوردها
905	کلام آخر
907	فصل 25. ASP.NET AJAX
908	درک مفهوم Ajax
908	Ajax: مزایا
909	Ajax: معایب
910	ASP.NET AJAX Toolkit
911	کنترل ScriptManager
913	بازتازه کردن جزئی
914	مثال ساده‌ای از UpdatePanel

918	اداره کردن خطاها
920	به روز رسانی شرطی
922	تریگرها
926	آگاه‌سازی از روند پیشرفت عملیات غیر همزمان
927	نمایش یک نوار پیشرفت شبیه‌سازی شده
929	لغو عملیات
931	بازتازه کردن به شکل زمان‌بندی شده
932	جعبه ابزار کنترل‌های ASP.NET AJAX
933	نصب ASP.NET AJAX Control Toolkit
935	کنترل Accordion
938	کنترل AutoCompleteExtender
942	به دست آوردن کنترل‌های بیشتر
947	کلام آخر
949	فصل 26. انتشار برنامه‌های ASP.NET
949	برنامه‌های ASP.NET و وب‌سرور
950	وب‌سرور چگونه کار می‌کند؟
951	دایرکتوری مجازی
952	آدرس‌های URL برنامه‌های وب
954	مزارع وب
955	(IIS) Internet Information Services
955	نسخه‌های مختلف IIS
956	نصب IIS در Windows Vista یا Windows 7
957	نصب IIS 7 در Windows Server 2008
959	مدیریت سایت‌ها توسط IIS Manager
960	نحوه‌ی ایجاد دایرکتوری مجازی
962	مفهوم Application Pools
965	حساب کاربری ASP.NET
967	تغییر حساب کاربری ASP.NET
968	اعطای دسترسی بیشتری به حساب کاربری ASP.NET
969	پیکربندی یک سایت وب
969	ایکون‌های قسمت ASP.NET
971	ایکون‌های قسمت IIS
973	ایکون‌های قسمت Management
973	صفحه‌ی پیش فرض
974	صفحات خطای شخصی
975	Machine Key
977	تصدیق هویت ویندوز
980	امنیت ارتباط با استفاده از SSL و گواهینامه‌ها
980	نحوه‌ی ایجاد یک درخواست گواهینامه‌ی SSL
981	پایه‌سازی SSL
983	انتشار یک سایت ساده
984	برنامه‌های وب و مولفه‌ها
985	سایر مراحل پیکربندی
985	کامپایل‌گد
987	انتقال سایت به سرور نهایی با استفاده از Visual Studio
987	ایجاد یک دایرکتوری مجازی برای یک پروژه‌ی جدید
990	کی‌ی کردن سایت وب
993	انتقال دادن سایت به سرور
994	کلام آخر

مایکروسافت در ابداع فناوری‌های جدید و لفافه‌پیچ کردن آنها در قالب اصطلاحات عجیب و غریب سابقه و شهرت زیادی دارد. چهارچوب کاری .NET نیز یکی از آنها است که اکنون بیش از یک دهه از عمر آن می‌گذرد. بعضی‌ها .NET را کپی ضعیفی از جاوا، یک اصطلاح تجاری بی‌معنی، تلاشی برای در اختیار گرفتن اینترنت و یا یک تکنولوژی شخصی می‌دانند که مایکروسافت برای خودش ابداع کرده است. اما هیچ یک از این توصیف‌ها به طور کامل درست نیست.

.NET در واقع مجموعه‌ای از تکنولوژی‌ها است که برخی از آنها واقعا انقلابی است و به این دلیل ساخته شده که برنامه‌نویسان و تولیدکنندگان نرم‌افزار بر اساس آن بتوانند برنامه‌ها و نرم‌افزارهای متنوعی تولید کنند. با استفاده از .NET برنامه‌نویسان می‌توانند برنامه‌های تحت ویندوز پیشرفته و قدرتمندی بنویسند، سرویس‌هایی ایجاد کنند که بی‌سروصدا در پشت صحنه مشغول کار هستند و حتی ابزارهایی درست کنند که تحت خط فرمان اجرا می‌شوند. البته با توجه به این که شما در حال خواندن این کتاب هستید طبیعتاً می‌خواهید از .NET برای ساخت برنامه‌های تحت وب استفاده کنید. در این حالت از زیر مجموعه‌ی خاصی از .NET به نام ASP.NET استفاده کرده و با یکی از زبان‌های اصلی .NET به نام C# کار خواهید کرد.

در این فصل با فناوری‌هایی که زیر بنای .NET را تشکیل می‌دهد آشنا خواهید شد. ابتدا مرور سریعی بر تاریخچه‌ی برنامه‌نویسی وب خواهید داشت و متوجه خواهید شد چهارچوب کاری .NET اساساً به چه دلیلی شکل گرفت. سپس مرور سطح بالایی بر قسمت‌های مختلف .NET خواهید داشت و خواهید دید ASP.NET 4.0 وارد صحنه می‌شود.

سیر تکامل برنامه‌نویسی وب

اینترنت در اواخر دهه‌ی 1960 به عنوان یک پروژه‌ی آزمایشی شروع به کار کرد. هدف اصلی از ابداع اینترنت ایجاد یک شبکه‌ی اطلاعاتی انعطاف‌پذیر بود که بتواند در برابر از دست رفتن تعداد زیادی از کامپیوترهایش دوام بیاورد و به کار خود ادامه بدهد، بدون آن که ارتباط سایر کامپیوترهای آن با یکدیگر مختل شود. بودجه‌ی اولیه‌ی این پروژه توسط وزارت دفاع آمریکا با در نظر گرفتن احتمال وقوع شرایط اضطراری خاص (نظیر حمله اتمی) فراهم شد.

در ابتدای تولد اینترنت فعالیت آن بیشتر به موسسات آموزشی و پیمان‌کاران وزارت دفاع محدود بود. اما با سرعتی باورنکردنی به شکل ابزاری برای همکاری آکادمیک بین دانشگاه‌ها رشد پیدا کرد. زیرا به محققین سراسر جهان امکان می‌داد اطلاعات خود را با یکدیگر به اشتراک بگذارند. در اوایل دهه‌ی 1990 مودم‌هایی ساخته شد که می‌توانستند از طریق خطوط تلفن کار کنند و پس از آن بود که دروازه‌ی اینترنت به روی کاربران عمومی نیز باز شد. در سال 1993 اولین مرورگر HTML ایجاد شد و انقلاب اینترنت به وقوع پیوست.

HTML و فرم‌های HTML

به سختی می‌توان سایت‌های اولیه‌ی دنیای وب را برنامه‌های تحت وب محسوب کرد. در واقع نسل اول سایت‌های وب بیشتر به بروشورهای تبلیغاتی شبیه بودند و عمدتاً از صفحات ثابت HTML تشکیل می‌شدند، صفحاتی که به‌روز رسانی آنها نیز باید به شکل دستی انجام می‌شد.

در حالت کلی یک صفحه‌ی HTML چیزی شبیه یک سند واژه‌پردازی است و محتوایی در آن وجود دارد که خودش کاری انجام نمی‌دهد اما می‌تواند در کامپیوتر به نمایش در بیاید. در مثال زیر یک صفحه‌ی HTML در ساده‌ترین حالت خود نشان داده شده که از یک عنوان و یک تکه متن ساده تشکیل شده است:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Sample Web Page</title>
  </head>
  <body>
    <h1>Sample Web Page Heading</h1>
    <p>This is a sample web page.</p>
  </body>
</html>
```

به طور کلی یک سند HTML از دو نوع محتوا تشکیل می‌شود: 1. متن‌ها 2. المان‌هایی که به مرورگر می‌گویند متن‌ها را چگونه نمایش بدهد. تشخیص المان‌ها بسیار ساده است، زیرا تمام آنها بین کاراکتر کوچکتر از < و بزرگتر از > قرار می‌گیرند. در HTML برای انواع و اقسام محتوا مانند عنوان، پاراگراف، لینک‌ها، متن ضخیم یا خوابیده، خطوط افقی و نظایر آن تگ‌های مختلفی وجود دارد. به عنوان مثال در <h1>Some Text</h1> از تگ <h1> استفاده شده است. این تگ به مرورگر اعلام می‌کند متن Some Text را بر اساس شیوه‌ی 1 Heading نمایش بدهد و مرورگر نیز برای آن از فونتی بزرگ و ضخیم استفاده می‌کند. به طور مشابه <p>This is a sample web page.</p> نیز یک پاراگراف ایجاد می‌کند. تگ <head> نیز اطلاعات کلی صفحه را در بر می‌گیرد و شامل تگی نظیر <title> است که برای مشخص کردن عنوان سند به کار می‌رود و در میله‌ی عنوان مرورگر نمایش داده می‌شود. از آن سو تگ <body> نیز محتوای واقعی صفحه را که در مرورگر نمایش داده می‌شود در بر می‌گیرد.

شکل 1-1 صفحه‌ی ساده‌ی HTML بالا را نمایش می‌دهد. در حال حاضر این صفحه فقط یک فایل استاتیک و ثابت (به نام SampleWebPage.htm) است که فقط از HTML تشکیل می‌شود، هیچ کاری انجام نمی‌دهد (نمی‌تواند با کاربر تعامل داشته باشد)، برای نمایش نیازی به وب سرور ندارد و یک برنامه‌ی وب نیز محسوب نمی‌شود.

■ **نکته** برای برنامه‌نویسی صفحات ASP.NET نیازی نیست در HTML خبره باشید (اگرچه چنین چیزی اغلب مواقع مفید است). برای آشنایی سریع با HTML به یکی از سایت‌های خودآموز HTML مانند <http://www.w3schools.com/html> مراجعه کنید. البته در فصل 4 نیز معرفی مختصری بر المان‌های HTML ارائه می‌شود.



شکل 1-1. یک صفحه‌ی HTML عادی

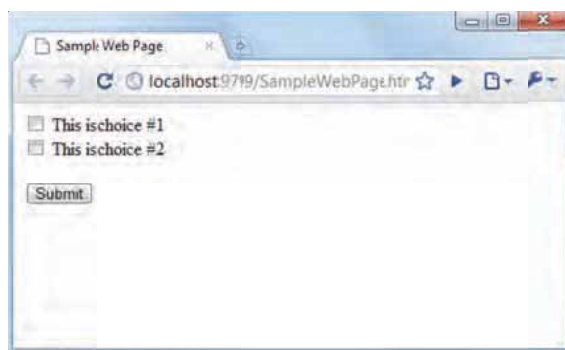
در HTML 2.0 با معرفی چیزی به نام فرم اولین نطفه‌ی برنامه‌نویسی وب شکل گرفت. فرم‌های HTML زبان HTML را طوری توسعه دادند که یک سند HTML نه فقط از تگ‌های قالب‌بندی بلکه از تگ‌هایی برای المان‌های گرافیکی یا «کنترل» نیز تشکیل می‌شد مانند لیست‌های پایین افتادنی¹، جعبه‌های متن و دکمه‌ها. در زیر یک صفحه‌ی وب نمونه نشان داده شده که با استفاده از کنترل‌های HTML ساخته شده است:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Sample Web Page</title>
  </head>
  <body>
    <form>
      <input type="checkbox"/>
      This is choice #1<br/>
      <input type="checkbox"/>
      This is choice #2<br/><br/>
      <input type="submit" value="Submit"/>
    </form>
  </body>
</html>
```

در یک فرم HTML تمام کنترل‌ها بین تگ‌های `<form>` و `</form>` قرار می‌گیرند. در مثال قبل دو چک‌باکس (تگ `<input type="checkbox"/>`) و یک دکمه (تگ `<input type="submit"/>`) تعریف شده است. تگ `
` نیز برای شکستن خطوط یا محتوا و منتقل کردن ادامه‌ی محتوا به خط بعد به کار می‌رود. این صفحه پس از نمایش در یک مرورگر شکلی شبیه شکل 1-2 دارد.

برنامه‌نویسان وب توسط فرم‌های HTML می‌توانستند صفحاتی برای ورود اطلاعات طراحی کنند. وقتی کاربر دکمه‌ی ارسال یا Submit فرم شکل 1-2 را کلیک می‌کند، مقدار تمام کنترل‌های ورودی (در اینجا مقدار دو چک‌باکس) به هم چسبانده شده و به صورت یک رشته به سمت سرور ارسال می‌شود. در سمت سرور نیز برنامه‌ی از پیش طراحی شده‌ای، اطلاعات را دریافت کرده و آن را پردازش می‌کند.

¹ Drop-Down List



شکل 2-1. یک فرم وب

نکته‌ی شگفت‌انگیزی که وجود دارد این است که این کنترل‌هایی که بیش از 15 سال پیش برای فرم‌های HTML طراحی شدند امروزه هنوز هم زیر بنای صفحات وب حتی صفحات پویا و پیشرفته‌ی ASP.NET را تشکیل می‌دهند. تنها چیزی که فرق کرده نوع برنامه‌های سمت سرور است. در گذشته وقتی کاربر دکمه‌ی ارسال یک فرم وب را کلیک می‌کرد ممکن بود اطلاعاتی که وی وارد کرده به یک آدرس پست الکترونیک فرستاده شده یا برای برنامه‌ای در سمت سرور ارسال شود که با استفاده از استاندارد بحث‌برانگیز CGI (Common Gateway Interface) نوشته شده بود. اما امروزه برای این کار می‌توانید از فناوری بسیار قدرتمندتری به نام ASP.NET استفاده کنید.

■ **نکته** در جدیدترین نسخه‌ی زبان HTML یعنی HTML5 برای اولین بار در تاریخ این زبان شکل جدیدی از المان‌های واسط کاربر معرفی شده است. بیشتر مواقع ASP.NET از این المان‌ها استفاده نمی‌کند، زیرا این المان‌ها هنوز توسط تمام مرورگرها پشتیبانی نمی‌شوند (حتی مرورگرهایی که از آنها پشتیبانی می‌کنند نیز به یک شکل با آنها برخورد نمی‌کنند). با این حال ASP.NET در صورتی که میسر باشد از تزئیناتی استفاده می‌کند که بر مبنای HTML5 کار می‌کنند مانند ویژگی‌های اعتبارسنجی. این بدین خاطر است که مرورگرهایی که از این قابلیت‌ها استفاده نمی‌کنند بتوانند از قابلیت‌های جدید صرف‌نظر کنند و در نتیجه صفحه باز هم بتواند به درستی کار کند.

برنامه‌نویسی سمت سرور دهنده

سکوه‌های اولیه‌ی برنامه‌نویسی وب دو مشکل کلیدی داشتند. اول این که برنامه‌ها و سایت‌هایی که بر اساس آنها نوشته می‌شد مقیاس‌پذیری خوبی نداشتند. در نتیجه سایت‌های محبوبی که مخاطب زیادی داشتند با زیاد شدن تعداد کاربران همزمان‌شان به نفس زدن افتاده و در نهایت یا در هم می‌شکستند و یا آنقدر کند می‌شدند که دیگر قابل استفاده نبودند. مشکل فناوری‌های برنامه‌نویسی تحت وب اولیه این بود که چیزی جز یک بستر برنامه‌نویسی ساده فراهم نمی‌کردند. اگر به قابلیت‌های سطح بالاتری نظیر تصدیق هویت، ذخیره‌سازی اطلاعات شخصی کاربران یا نمایش دادن اطلاعات بازبایی شده از دیتابیس نیاز پیدا می‌کردید مجبور بودید تمام این کارها را خودتان انجام بدهید و گداهای بسیار زیادی بنویسید. تردیدی وجود ندارد که نوشتن یک برنامه‌ی تحت وب بدین صورت بسیار خسته‌کننده و مستعد خطا است.

برای رفع این مشکلات، میکروسافت سکوهاى برنامه‌نویسى سطح بالاترى ایجاد کرد که اولین آن ASP و سپس ASP.NET بود. توسط این فناوری‌ها برنامه‌نویسان توانستند برنامه‌نویسى صفحات پویای وب را بدون نگرانی بابت جزئیات سطح پایین برنامه‌نویسى انجام بدهند. به همین دلیل هر دوی این فناوری‌ها از موفقیت چشمگیری برخوردار شدند.

سکوی اولیه‌ی ASP (یا ASP Classic) میلیون‌ها نفر از برنامه‌نویسان را به خود جلب کرد و بسیار فراتر از حدی که حتی میکروسافت انتظارش را داشت محبوبیت پیدا کرد. طولی نکشید که این سکو در انواع و اقسام برنامه‌های عجیب و غریب و نامعمول به کار گرفته شد، حتی برنامه‌های کاربردی‌ای که از نظر تجاری حساس بودند یا سایت‌های تجارت الکترونیکی با ترافیک بسیار بالا. اما از آنجایی که ASP با در نظر داشتن چنین کاربردهایی طراحی نشده بود، کمی بعد سر و کله‌ی مشکلات متعددی در زمینه‌های مختلفی مانند کارایی، امنیت و پیکربندی پیدا شد.

این همان جایی است که ASP.NET وارد صحنه شد. ASP.NET به عنوان یک چهارچوب محکم و با کیفیت برای برنامه‌های وب طراحی شد و مشکلات، کمبودها و محدودیت‌های ASP را بر طرف کرد. به علاوه ASP.NET در مقایسه با ASP از کارایی بسیار بهتری برخوردار است، ابزارهای طراحی بهتری دارد و حاوی مجموعه‌ای غنی از قابلیت‌های از پیش آماده است. در واقع از همان لحظه‌ای که ASP.NET معرفی شد به سرعت محبوبیت پیدا کرد. به طوری که حتی زمانی که هنوز نسخه‌ی بتای آن منتشر شده بود در بسیاری از سایت‌های بزرگ تجاری نیز به کار گرفته شد.

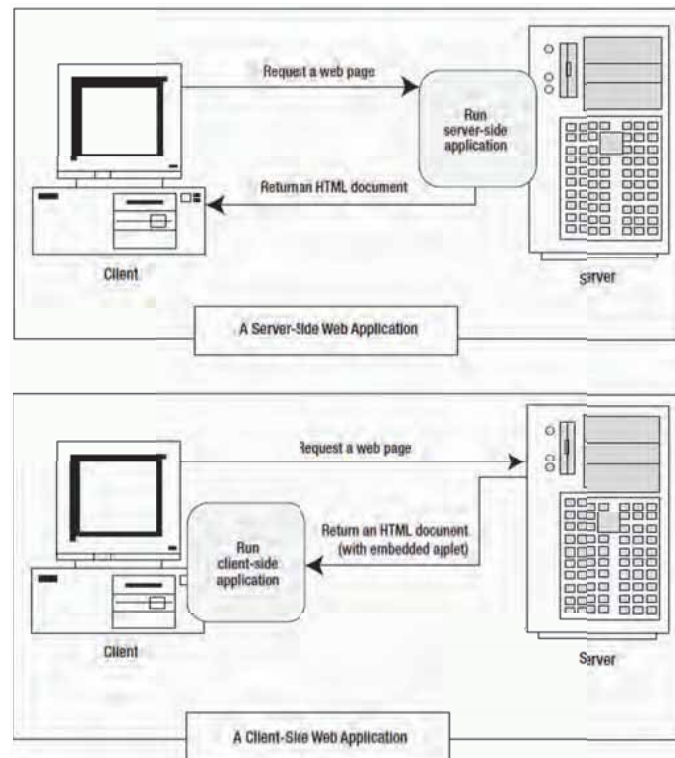
سادگی ASP.NET تا حدی است که حتی می‌توانید یک صفحه‌ی وب ASP.NET را بدون هیچ گونه اطلاعی درباره‌ی HTML نیز ایجاد کنید. با این حال اگر آشنایی مختصری با HTML داشته باشید می‌توانید صفحات وب را سریع‌تر و به شکل کاراتری ایجاد کنید.

■ **توجه** اگرچه پایه و اساس ASP و ASP.NET مشابه یکدیگر است اما این دو فناوری از لحاظ ساختاری کاملاً با یکدیگر فرق دارند. فناوری ASP مبتنی بر مفسرها و زبان‌های اسکریپتی است و مهارت در آن به تجربه‌ی عمیقی در HTML نیاز دارد، ضمن این که برنامه‌نویسی آن نیز پر مشقت و پر دردسر است. از آن سو ASP.NET یک فناوری کامپایلری با برنامه‌نویسی شیء‌گرا است و توسط آن می‌توانید صفحات وب را درست به سادگی یک برنامه‌ی تحت ویندوز ایجاد کنید.

برنامه‌نویسی سمت سرور

در همان زمانی که برنامه‌نویسی سمت سرور در حال پشت سر گذاشتن فناوری‌های مختلف بود، یک نوع برنامه‌نویسی دیگر نیز محبوبیت پیدا کرد. برنامه‌نویسان با گنجاندن برنامه‌های کوچکی در صفحات وب که با استفاده از جاوااسکریپت، Java، ActiveX و فلش نوشته می‌شد به دنبال روش‌های دیگری بودند که بتوانند پویایی صفحات را بیشتر بهبود بدهند. این فناوری‌ها مستلزم اجرا در سمت سرور نیستند، بلکه به طور کامل توسط مرورگر به صورت محلی در سمت کاربر اجرا می‌شوند.

بزرگترین مشکل فناوری‌های سمت سرویس‌گیرنده این است که به طور کامل توسط تمام مرورگرها و سیستم‌های عامل پشتیبانی نمی‌شوند. یکی از دلایل محبوبیت زیاد برنامه‌نویسی وب این بوده که نیازی به سی‌دی نصب، دانلود، پیکربندی و سایر مراحل خسته‌کننده‌ی انتشار (که معمولا با اشکال هم همراه است) ندارند. در حقیقت یک برنامه‌ی وب می‌تواند در هر کامپیوتری که به اینترنت متصل باشد به راحتی استفاده شود. اما وقتی برنامه‌نویسان از فناوری‌های سمت سرویس‌گیرنده استفاده کردند با مشکلات دیگری روبرو شدند. ناگهان ناسازگاری برنامه با مرورگرهای مختلف به یک مشکل بزرگ تبدیل شد. برنامه‌نویسان مجبور شدند سایت‌های خود را تحت مرورگرها و سیستم‌عامل‌های مختلف آزمایش کنند و با دامنه‌ی وسیعی از ویژگی‌های خاص مرورگرها، باگ‌ها، مشکلات و سازگاری با رفتارها و خصوصیت‌های نسخه‌های قبلی سر و کله بزنند. در واقع مدل برنامه‌نویسی سمت سرویس‌گیرنده برخی از مهمترین مزایای برنامه‌نویسی تحت وب را قربانی کرد.



شکل 3-1. برنامه‌های تحت سرویس‌گیرنده و تحت سرویس‌دهنده

به همین دلیل ASP.NET در درجه‌ی نخست به عنوان یک فناوری تحت سرویس‌دهنده طراحی شد نه یک فناوری سمت سرویس‌گیرنده. تمام کدهای ASP.NET در سمت سرویس‌دهنده اجرا می‌شود. اما وقتی اجرای کدها خاتمه می‌یابد، کاربر یک صفحه‌ی معمولی HTML دریافت می‌کند که در هر مرورگری به طور یکسان کار می‌کند. در شکل 3-1 تفاوت بین مدل سمت سرویس‌دهنده و سرویس‌گیرنده نشان داده شده است.

موارد زیر دلایل دیگری برای اجتناب از برنامه‌نویسی سمت سرویس‌گیرنده است:

- ✓ *انزوا*²: در کدنویسی سمت سرویس گیرنده نمی توان به منابعی که در سمت سرور وجود دارد دسترسی داشت. به عنوان مثال یک برنامه‌ی تحت سرویس گیرنده هیچ راهی برای خواندن یک فایل یا تعامل با پایگاه داده‌ی سمت سرور ندارد (حداقل بدون این که درگیر مشکلات امنیت و سازگاری مرورگرها شود).
- ✓ *امنیت*: کاربران می توانند کدهای سمت سرویس گیرنده را ببینند و وقتی کاربران مغرض و بدخواه از نحوه‌ی کار یک برنامه سر در بیاورند، معمولاً می توانند آن را دستکاری کرده و خرابکاری به بار بیاورند.
- ✓ *سرویس گیرندگان سبک*³: در حال حاضر ادوات مجهز به وب مانند تلفن های همراه، کامپیوترهای دستی، تبلت ها و PDA ها بسیار رایج هستند. این ادوات نیز معمولاً به صورت از قبل دارای یک جور توانایی مرور وب هستند، اما در عین حال از تمام قابلیت های مرورگرهای معمولی پشتیبانی نمی کنند. به عنوان مثال سرویس گیرنده های سبک ممکن است از قابلیتی نظیر جاوا اسکریپت یا فلش پشتیبانی نکنند.

با این حال برنامه نویسی سمت سرویس گیرنده به طور کامل نمرده و در سال های اخیر قیامی رنسانسی در برنامه نویسی سمت سرویس گیرنده به خصوص در مورد جاوا اسکریپت رخ داده است. به طوری که می توان برنامه های سمت کلاینتی درست کرد که بدون نیاز به رفت و برگشت صفحه اطلاعات را از یک سرور راه دور بازیابی کرده و صفحه را به روز کنند. خوشبختانه ASP.NET نیز از این تغییرات به دو شکل بهره می برد:

- ✓ *زرق و برق های جاوا اسکریپتی*: در برخی موارد توسط ASP.NET می توانید تکنیک های برنامه نویسی سمت کلاینت را با برنامه نویسی سمت سرور ترکیب کنید. به عنوان مثال کنترل های ASP.NET طوری طراحی شده اند که می توانند هوشمندانه قابلیت های مرورگر را کشف کرده و بر اساس آن، عملکرد خود را تغییر بدهند. اگر مرورگر کاربر از جاوا اسکریپت پشتیبانی کند، این کنترل ها کدهایی به زبان جاوا اسکریپت به صفحه اضافه کرده و واسط کاربر غنی تری فراهم می کنند که پاسخ پذیری بیشتری دارد. در فصل 9 مثال بسیار خوبی از این تکنیک را در خصوص اعتبارسنجی اطلاعات نشان می دهیم.
- ✓ *قابلیت های Ajax: Ajax* مجموعه ای از تکنیک های جاوا اسکریپتی است که برای ساخت صفحات وب بسیار سریع و پاسخ پذیر با قابلیت محتوای پویا به کار می رود. در فصل 25 نشان می دهیم چطور می توانید صفحات معمولی ASP.NET را با استفاده از تکنیک Ajax به نهایت پویایی خود برسانید. تفاوتی که روش ASP.NET با روش برنامه نویسی Ajax به صورت دستی دارد این است که ASP.NET خودش پیچیدگی ها و ملاحظات سازگاری مرورگرها را بر عهده گرفته و به شما اطمینان می دهد صفحه ی وب شما در تمام مرورگرها به درستی کار کند.

با این حال یک نکته ی مهم را فراموش نکنید. صرف نظر از این که مرورگر چه قابلیت هایی دارد و از چه تکنیک هایی در سمت کلاینت استفاده می کنید، کدهای C# صفحات شما همیشه در سمت سرور اجرا می شود.

² Isolation

³ Thin Clients

■ **نکته** با تمام خوبی‌هایی که در مورد ASP.NET گفتیم بد نیست بدانید ASP.NET (حداقل به تنهایی) بهترین سکوی برنامه‌های سمت کلاینت محسوب نمی‌شود. به عنوان مثال ASP.NET نمی‌تواند کمک چندانی به برنامه‌نویسانی که می‌خواهند یک بازی تحت وب بلادرنگ (real-time) یا برنامه‌ای شبیه Google Maps بنویسند بکند. اگر دنبال چنین برنامه‌هایی هستید خودتان باید انبوه گداهای جاوااسکریپتی مورد نیاز صفحات را بنویسید. اما اگر می‌خواهید یک فروشگاه تجارت الکترونیک یا یک سایت تجاری درست کنید، یا سایتی که حجم انبوهی از اطلاعات را نمایش می‌دهد، ASP.NET گزینه‌ی بسیار خوبی برای شما است و تمام نیازهای شما را پوشش می‌دهد.

چهارچوب کاری .NET

همان طور که گفته شد چهارچوب کاری .NET ترکیبی از چندین فناوری مختلف است. این فناوری‌ها عبارتند از:

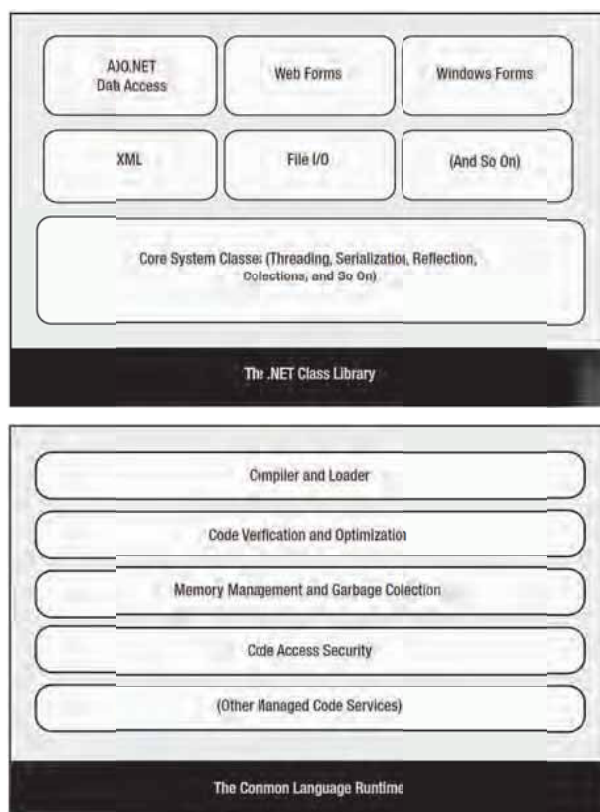
- ✓ زبان‌های .NET: این زبان‌ها عبارتند از C#, Visual Basic، F# و C++.
- ✓ CLR (Common Language Runtime): CLR یا «ماشین اجرای زبان مشترک» موتوری است که تمام برنامه‌های .NET را اجرا کرده و سرویس‌هایی را برای این برنامه‌ها به طور خودکار فراهم می‌کند مانند بررسی امنیت، مدیریت حافظه و بهینه‌سازی.
- ✓ کتابخانه‌ی کلاس‌های چهارچوب کاری .NET⁴: کتابخانه‌ی کلاس‌های .NET مجموعه‌ای است از هزاران قابلیت از پیش ساخته که می‌توانید به سادگی و با کمترین دردسر از آنها در برنامه‌های خود استفاده کنید. در برخی موارد این قابلیت‌ها در قالب یک فناوری مانند ADO.NET (فناوری ایجاد برنامه‌های پایگاه داده) و Windows Presentation Foundation یا WPF (فناوری ایجاد برنامه‌های رومیزی) دسته‌بندی می‌شوند.
- ✓ ASP.NET: این همان موتوری است که برنامه‌های وب .NET را میزبانی کرده و تقریباً از تمام قابلیت‌هایی که در کتابخانه‌ی کلاس‌های .NET وجود دارد پشتیبانی می‌کند. علاوه بر این ASP.NET خودش برخی کارها را نیز به طور خودکار برای شما انجام می‌دهد و می‌توانید از آنها در برنامه‌های تحت وب استفاده کنید مانند تصدیق هویت به شکل ایمن و ذخیره‌سازی اطلاعات.
- ✓ Visual Studio: این برنامه یک محیط برنامه‌نویسی تمام عیار است که قابلیت‌های فراوانی در تولید اشکال‌زدایی برنامه‌های .NET دارد اما استفاده از آن اختیاری است. سی‌دی‌ها یا دی‌وی‌دی‌های نصب Visual Studio به همراه خود نسخه‌ی کامل چهارچوب کاری .NET را نیز دارند. لذا در صورت نصب Visual Studio نیازی نیست چهارچوب کاری .NET را به طور جداگانه دانلود و نصب کنید.

برخی مواقع تقسیم‌بندی بین این اجزاء به طور دقیق روشن نیست. به عنوان مثال لفظ ASP.NET برخی مواقع برای ارجاع به قسمتی از کتابخانه‌ی کلاس‌های .NET به کار می‌رود که تنها برای طراحی صفحات وب استفاده می‌شوند. اما از سوی دیگر لفظ مزبور به کل برنامه‌های کاربردی وب در .NET نیز اطلاق می‌شود که شامل زبان‌های .NET و بسیاری از کلاس‌های زیر بنایی .NET است که به طور اختصاصی به وب تعلق ندارند (در این کتاب در حالت کلی از این تعریف استفاده می‌کنیم. به همین دلیل آموزش جامع و گسترده‌ی ما در خصوص ASP.NET شامل آموزش اصول

⁴ .NET Framework Class Library

و مبانی .NET، زبان C# و همچنین موضوعاتی است که هر برنامه‌نویس .NET می‌تواند از آنها استفاده کند مانند برنامه‌نویسی مولفه‌ها و دسترسی به پایگاه داده).

در شکل 1-4 کتابخانه‌ی کلاس‌های .NET و CLR، دو جزء کلیدی و بنیادین .NET نشان داده شده است.



شکل 1-4. چهارچوب کاری .NET.

در ادامه‌ی این فصل مرور سریعی بر اجزاء مختلفی که چهارچوب کاری .NET را تشکیل می‌دهند خواهیم داشت.

C#، VB.NET و زبان‌های .NET.

در این کتاب برای برنامه‌نویسی .NET، از زبان C#، زبانی که خود مایکروسافت استفاده از آن را ترجیح می‌دهد استفاده می‌کنیم. C# از لحاظ قواعد نحوی به زبان ++C و Java شباهت دارد، با این وجود هیچ مسیر مهاجرت مستقیمی از جاوا یا ++C به زبان C# وجود ندارد. زبان دیگری که بین برنامه‌نویسان .NET محبوبیت دارد زبان VB است که نسخه‌ی دات‌نتی زبان قدیمی‌تر Visual Basic 6 است و قوانین نحوی آن به این زبان و زبان VBA (Visual Basic for Applications) که برای نوشتن ماکرو در برنامه‌های Microsoft Office طراحی شده شباهت دارد.

با این حال همان طور که نمی‌توانید کدهای C# و C++ را به یکدیگر تبدیل کنید نمی‌توانید کدهای VB 6 را به VB یا بالعکس تبدیل کنید.

نکته‌ی جالبی که در خصوص زبان‌های C# و VB وجود دارد این است که علی‌رغم تفاوت قوانین نحوی آنها، این زبان‌ها بسیار شبیه یکدیگر هستند. به طوری که هر دو از کتابخانه‌ی کلاس‌های NET استفاده می‌کنند و CLR از هر دو به یک شکل پشتیبانی می‌کند.

این شباهت تا جایی است که تقریباً می‌توان هر تکه کد C# را خط به خط به کد معادل آن در VB تبدیل کرد (و بالعکس). البته در موارد بسیار کمی بین این دو زبان اختلاف وجود دارد، اما بیشتر مواقع فردی که یکی از زبان‌های NET را یاد گرفته باشد می‌تواند به سرعت و بدون دردسر به یک زبان NET دیگر مهاجرت کند. حتی ابزارهایی وجود دارد که می‌توانند این تبدیل را برای شما به طور خودکار انجام بدهند (مانند ابزار <http://converter.telerik.com> یا <http://tangiblesoftware.com>).

به طور خلاصه زبان C# و VB هر دو زبان‌هایی زیبا و پیشرفته بوده و برای تولید نسل بعدی برنامه‌های تحت وب بسیار مناسب هستند.

■ **توجه** وقتی NET 1.0 منتشر شد به همراه خود زبان‌های کاملاً جدیدی معرفی کرد و تغییرات این زبان‌ها در نسخه‌های بعدی NET نیز بسیار جزئی بود. اگرچه در NET 4.5 قابلیت‌های جدیدی به زبان C# اضافه شده است اما قسمت اعظم این زبان تغییری نکرده است. در فصل 2 و 3 با اصول و قواعد نحوی زبان C# آشنا خواهید شد و مبانی برنامه‌نویسی شیء‌گرا را یاد خواهید گرفت.

زبان میانی⁵

تمام کدهایی که به زبان‌های NET نوشته می‌شود پیش از اجرا ابتدا به یک زبان سطح پایین‌تر کامپایل می‌شود. این زبان سطح پایین‌تر «زبان میانی مشترک»، CIL⁶ یا به اختصار IL نام دارد. ماشین اجرایی NET (CLR) نیز فقط کدهای IL را درک می‌کند. از آنجایی که تمام زبان‌های NET بر اساس IL طراحی شده‌اند، شباهت‌های عمیقی بین آنها وجود دارد. به همین دلیل کارایی و قابلیت‌های زبان‌های C# و VB دقیقاً با یکدیگر یکسان است. در واقع این دو زبان آنقدر با یکدیگر سازگار هستند که یک صفحه‌ی وب C# می‌تواند از مولفه‌هایی که به یک زبان NET دیگر مانند VB نوشته شده استفاده کند و بالعکس.

این سازگاری زبان‌ها را با استفاده از چیزی تحت عنوان «مشخصات زبان مشترک» یا CLS⁷ به رسمیت بخشیده است. در واقع CLS قراردادی است که اگر رعایت شود تضمین می‌کند مولفه‌ای که به یک زبان NET نوشته

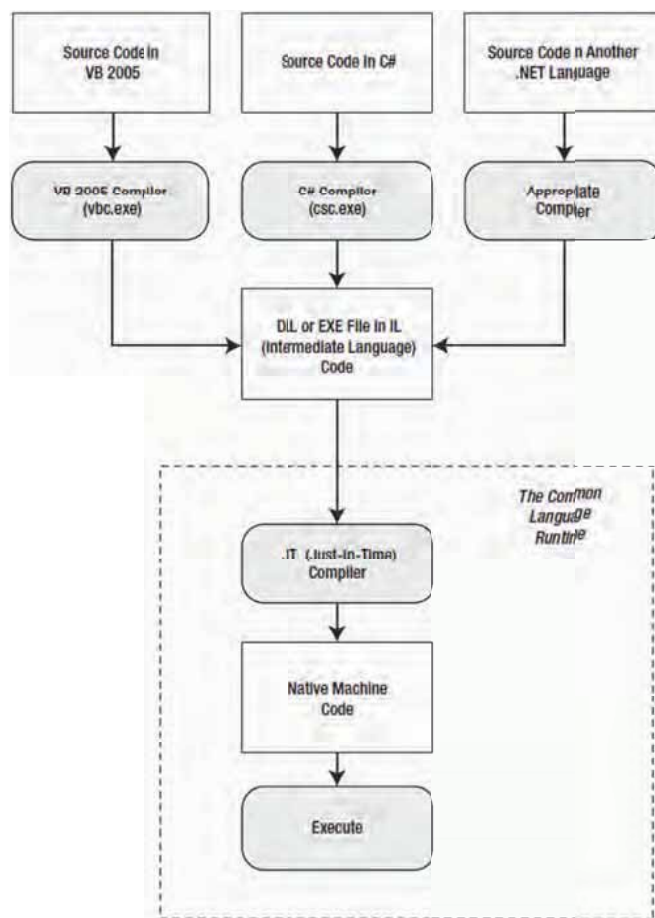
⁵ Intermediate Language

⁶ Common Intermediate Language

⁷ Common Language Specification

شده می‌تواند در تمام دیگر زبان‌های .NET. نیز مورد استفاده قرار بگیرد. یکی از قسمت‌های CLS «سیستم نوع داده‌های مشترک» یا CTS⁸ است که نوع داده‌هایی چون رشته‌ها، اعداد و آرایه‌ها در آن تعریف شده و بین تمام زبان‌های .NET. مشترک است. علاوه بر این CLS ساختار کلی اجزاء برنامه‌نویسی شیء‌گرا یعنی کلاس‌ها، متدها، رویدادها و برخی موارد دیگر را نیز تعریف می‌کند. البته بیشتر مواقع برنامه‌نویسان .NET. نیازی ندارند از نحوه‌ی کار CLS سر در بیاورند، حتی با وجودی که تمام برنامه‌های آنها متکی به CLS و CTS است.

در شکل 1-5 نحوه‌ی کامپایل زبان‌های .NET. به IL نشان داده شده است. هر فایل EXE یا DLL ای که با استفاده از یکی از زبان‌های .NET. ساخته شده باشد حاوی کدهای IL است. این فایل‌ها همان فایل‌هایی هستند که در نهایت به سایر کامپیوترها منتشر می‌کنید. به عنوان مثال در برنامه‌های وب، کد کامپایل شده‌ی سایت را به سرویس‌دهنده‌ی وب کپی می‌کنید.



شکل 1-5. نحوه‌ی کامپایل و اجرای برنامه‌های .NET. در زبان‌های مختلف

⁸ Common Type Specification

همان طور که گفته شد CLR فقط کُد IL اجرا می‌کند، این یعنی CLR هیچ ایده‌ای نسبت به زبانی که برنامه‌ی خود را با آن نوشته‌اید ندارد. همچنین توجه کنید که CLR برای اجرای برنامه یک مرحله‌ی کامپایل دیگر نیز انجام می‌دهد و کُد های IL را به زبان بومی ماشین که با سکوی سخت‌افزاری فعلی تناسب دارد تبدیل می‌کند.

این مرحله موقع بارگذاری برنامه درست پیش از اجرای آن رخ می‌دهد. در برنامه‌های ASP.NET فایلی که از تبدیل کُد IL به زبان ماشین به دست می‌آید در سرویس‌دهنده کش می‌شود تا در آینده باز هم بتوان از آن استفاده کرد و CLR مجبور نباشد هر بار دوباره این کار را تکرار کند. بدین ترتیب کارایی مناسبی فراهم می‌شود.

■ **توجه** ممکن است بپرسید چرا کامپایلرهای NET. برنامه‌های NET. را به طور مستقیم به به زبان ماشین ترجمه نمی‌کنند. علتش این است که تولید کُد زبان ماشین به عوامل متعددی بستگی دارد که مهمترین آن پردازنده است. اگر برنامه را به زبان ماشین یک کامپیوتر کامپایل کنید تضمینی وجود ندارد که در یک کامپیوتر دیگر با پردازنده‌ای متفاوت به درستی کار کند.

ماشین اجرای زبان مشترک یا CLR

CLR موتوری است که از تمام زبان‌های NET. پشتیبانی می‌کند. در بسیاری از زبان‌های پیشرفته از این ماشین‌های اجرا یا runtime نیز استفاده می‌شود. به عنوان مثال در VB 6 منطق ماشین اجرا درون فایل DLL ای به اسم msvbvm60.dll قرار دارد. بسیاری از برنامه‌های ++C نیز با فایلی به اسم mscrt40.dll لینک می‌شوند تا از قابلیت‌های این کتابخانه به طور مشترک استفاده کنند. این ماشین‌های اجرا ممکن است کتابخانه‌هایی فراهم کنند که توسط زبان برنامه‌نویسی استفاده می‌شود و همچنین ممکن است مسوولیت اجرای برنامه را نیز بر عهده داشته باشند (مانند زبان جاوا).

ماشین‌های اجرا چیز جدیدی نیستند و از قبل هم وجود داشته‌اند، اما CLR جاه‌طلبانه‌ترین ماشین اجرای میکروسافت از گذشته تاکنون بوده است. CLR نه تنها کُد های IL را اجرا می‌کند بلکه مجموعه‌ای از خدمات را مانند ارزیابی کُد⁹، بهینه‌سازی¹⁰ و مدیریت اشیاء¹¹ به طور خودکار برای برنامه‌ها فراهم می‌کند.

تمام برنامه‌های NET. درون CLR اجرا می‌شوند، چه در حال اجرای یک برنامه‌ی تحت ویندوز باشید و چه در حال استفاده از یک وب‌سرویس باشید. به عنوان مثال وقتی کلاینتی یک صفحه‌ی ASP.NET را درخواست می‌کند، موتور ASP.NET درون CLR به اجرا در می‌آید، کُد های صفحه را اجرا می‌کند و یک صفحه‌ی HTML تولید می‌کند که باید به سمت کلاینت ارسال شود.

⁹ Code Verification

¹⁰ Optimization

¹¹ Object management