

برنامه نویسی مبتنی بر وب

مهدی فرج پور

[m.farajpour.courses\(at\)gmail.com](mailto:m.farajpour.courses@gmail.com)

ASP.net

- ASP یک تکنولوژی مبتنی بر سرویس دهنده بوده که امکان اجرای اسکریپت ها ی موجود در یک صفحه وب را از طریق یک سرویس دهنده اینترنت فراهم می نماید
- ASP تکنولوژی متعلق به شرکت مایکروسافت است .
- ASP از کلمات **Active Server Page** مشتق شده است .
- ASP برنامه ای است که با مدیریت IIS اجراء می گردد.
- یک فایل ASP مشابه فایل Html است .
- محتویات یک فایل ASP شامل : متن ، Html ، XML و اسکریپت است .
- اسکریپت های موجود در یک فایل ASP بر روی سرویس دهنده اجراء می گردند.
- فایل های ASP دارای پسوند asp می باشند.
- پس از درخواست فایل های ASP توسط کاربران ، در ابتدا محتویات (اسکریپت ها) مربوطه بر روی سرویس دهنده اجراء و در ادامه نتایج بصورت تگ های Html برای کاربر ارسال خواهد شد.

ASP.net

قابلیتهای ASP.net

- مستقل از کلاینت (از مرورگر کاملاً مستقل است زیرا تمامی دستورات عملها بر روی سرور اجرا می گردند و مرورگر تنها صفحات **HTML** معمولی را بعنوان حاصل اجرای دستورات عملها دریافت می کند.)
- استفاده از زبان برنامه نویسی
- دسترسی آسان و سریع به انواع بانکهای اطلاعاتی

ASP.net

دات نت فریمورک

- قسمت اعظم کار ایجاد برنامه های بزرگ را بطور خودکار انجام می دهد .
- انواع مختلف نرم افزار (آنهایی که روی ماشین اجرا می شوند و آنهایی که روی وب اجرا می شوند) را یکپارچه می کند.
- برنامه نویس را از نوشتن ماژولهای پیچیده بی نیاز می کند
- به برنامه نویسانی که در زبانهای مختلف تخصص دارند اجازه می دهد تا روی یک پروژه مشترک کار کنند
- چرخه تولید نرم افزار را سرعت می دهد
- تمام انواع سخت افزار (از سوپر کامپیوتر گرفته تا PDA و تلفن همراه) را هدف گرفته است

ASP.net

معرفی ASP.NET

ASP.NET نسل بعدی Active Server Pages یا ASP است که توسط شرکت مایکروسافت ارائه شده است. این محصول توسط مایکروسافت بعنوان شاخص اصلی فناوری در ساخت سایتهای وب در نظر گرفته شده است. با استفاده از ASP.NET می توان هم اینترنت کوچک یک شرکت را ساخت و هم یک سایت وب تجاری خیلی بزرگ را طراحی و پیاده سازی نمود. مهمترین نکاتی که در طراحی این محصول در نظر گرفته شده است راحتی استفاده و بالا بودن کارایی و قابلیت آن می باشد.

ASP.net

ویژگیهای ASP.NET

- ASP.NET یک تکنولوژی سمت - سرور برای ایجاد صفحات دینامیک وب است
- صفحات ASP.NET روی سرور تنها یکبار کامپایل می شوند و نتیجه بصورت صفحات HTML به کاربر ارسال می شود.
- سازگاری با تمام مرورگرها
- جدا کردن کد از محتوا (المان های طراحی صفحه)
- مجموعه ASP.NET عضوی از بدنه .NET است. (بدنه .NET دارای بیش از ۴۵۰۰ کلاس آماده جهت استفاده در ASP.NET است.)

ASP.NET پلتفرمی است برای اجرای کدهای پویا (Dynamic) در سرویس دهنده وب ،

نه یک زبان برنامه نویسی

ASP.net

مقایسه ASP.NET با PHP

● پشتیبانی از زبان های مختلف :

Asp.NET را با زبان ها VB.NET, Delphi, C Sharp , J Sharp و چند زبان دیگر میتوان توسعه داد در حالی که Php را فقط با زبان Php می توان نوشت

● شی گرایي:

PHP از بدو تولد به صورت یک زبان اسکریپتی ارائه شد در حالی که Asp.NET با مفهوم شی گرایي (کلاس ، خاصیت ، رویداد و ...) پا به عرصه گذاشت.

● امنیت در وب :

خوشبختانه DotNET تدابیر امنیتی بسیار جالبی برای جلوگیری از نفوذ هکر ها به کار برده است ، این امر یکی از مهمترین عوامل برتری Asp.NET بر Php است.

● محیط توسعه نرم افزار

شرکت مایکروسافت چند نرم افزار قدرتمند برای توسعه Asp.NET فراهم نموده است : Visual Studio.NET , WebMatrix , Visual Web Developer این نرم افزار های قدرتمند شما را از هر نرم افزار دیگری بی نیاز می کند

در صورتی که Php هیچ نرم افزار اختصاصی نداشته و برخی نرم افزار ها این زبان را در کنار زبان های دیگر ارائه نموده اند مانند : Zend , Dreamweaver, NetBeans , Adobe

ASP.net

مقایسه PHP با ASP.NET

● سرعت در توسعه و پیاده سازی

ASP.NET با ارائه کنترل های قدرتمند وب و ارائه راهکارهای بسیار بهینه جهت ارتباط با بانک ها اطلاعاتی تحولی عظیم در توسعه نرم افزار های وب به وجود آورده

● سرعت بارگزاری صفحات

Asp.NET یک زبان کامپایل شده است در صورتی که Php یک زبان تفسیری است ، تمامی کد های Asp.NET پس از توسعه کامپایل شده و به Dll تبدیل می شوند ولی Php به صورت کد بارگزاری شده و در هنگام اجرا تفسیر می شود .

● سورس باز

یکی از بزرگترین مزایای Php سورس باز بودن آن است سورس باز نبودن Asp.NET یکی از نقاط ضعف آن محسوب می شود

● هزینه های جانبی

هزینه راه اندازی یک وب سایت Php کمتر از یک وب سایت Asp.NET به نظر می رسد (هاست ارزان و دیتا بیس «MySQL» رایگان)

ASP.net

مقایسه PHP با ASP.NET

● سیستم عامل

Php هم بر روی ویندوز و هم بر روی لینوکس نصب می شود ، و این یکی دیگر از مزیت های Php می باشد. اخیرا توسعه Asp.Net نیز روی لینوکس به سادگی انجام می پذیرد.

● سهولت یادگیری

یادگیری Asp.NET از هر زبان دیگری ساده تر است.

● کاربرد و بازار کار

اگر یک نگاه گذرا به صفحات آگهی روزنامه ها در بخش استخدام برنامه نویس بیاندازید متوجه خواهید شد که "استخدام برنامه نویس Asp.NET" سهم عظیمی از این آگهی ها را در بر گرفته است

همچنین شرکت مایکروسافت با برگزاری امتحانات متعدد و ارائه مدارک معتبر مانند MCSD و MCTS شما را برای ادامه تحصیل و یافتن شغل مناسب در کشورهای خارجی یاری خواهد داد

● پیشرفت و توسعه نرم افزار و به روز شدن

از زمان ظهور Asp.NET در سال ۲۰۰۲-۲۰۰۳ تکنولوژی Asp.NET پیشرفت چشم گیری کرده است ، هماهنگی روز افزون Asp.NET و SQL Server --- اضافه شدن Ajax و LINQ ، MVC ، Wcp ، Wpf ، SilverLight در Asp.NET یکی از نقاط قوت Asp.NET به شمار می رود

ASP.net

ساختار یک صفحه ASP.NET

- دایرکتیوها
- بلوک های اعلان کد
- کنترل های ASP.NET
- توضیحات (مستند)
- Server Include Directive
- متون و تگ های HTML

ASP.net

دایرکتیوها

بر چگونگی کامپایل شدن یک صفحه ASP.NET تاثیر می گذارد و با علامت `<%@ -----%>` مشخص می شود.

پر استفاده ترین دایرکتیوها :

1. PAGE

- تعیین زبان برنامه نویسی در صفحه
- فعال کردن قابلیت رد گیری (Tracing) و اشکال زدایی صفحه (Debugging) برای برنامه نویسی

```
<%@ Page language="C#" Trace="True" Debug="True" %>
```

۲- IMPORT

- برای معرفی کلاس های آماده NET . در صفحه جاری

```
<%@ Import Namespace="System.Web.Mail" %>
```


ASP.net

بلوکهای اعلان کد

حاوی بخش کد نویسی به زبان برنامه نویسی دلخواه می باشد
طرز استفاده :

```
<Script language="C#" Runat="Server">  
    بخش کد  
</ Script >
```

ASP.net

کنترل های ASP.net

- این کنترل ها همانند کنترل های HTML می باشند با این تفاوت که کنترل های ASP.NET در سمت سرور اجرا می شوند و می توان برای آنها کد نویسی کرد
- این کنترل ها در داخل تگ `<Form> ----- </Form>` قرار می گیرند

کاربرد	کنترل HTML	کنترل Asp.Net
نمایش متن	<code> , Div</code>	<code><asp:label></code>
اجرای دستورالعمل	<code><input type="submit"></code>	<code><asp:button></code>
گرفتن متن از کاربر	<code><input type="text"></code>	<code><asp:textbox></code>

ASP.net

توضیحات (مستند)

- برای مستند سازی و شرح کدها می توان از تگ توضیحات استفاده نمود که در زمان کامپایل این تگها کامپایل نمی شود

```
<%--
```

توضیحات

```
--%>
```

Server Include Directive

- با استفاده از این ویژگی می توان فایلی را وارد صفحه خود نمود

```
<!-- #include file="filepath" -->
```

ASP.net

متون و تگ های HTML

بخش استاتیک و ثابت صفحه وب خود را با همان تگ های آشنای HTML و نوشته ها می توان ساخت.

```
<html>
<head>
<title>عنوان صفحه وب</title>
</head>
<body>
  Welcome to my site
</body>
</html>
```

آشنایی با انواع فایل و پسوندهای آنها در برنامه های کاربردی

- .asax**: این پسوند برای فایل خاصی بنام Global.asax استفاده می گردد.
- .ascx**: این نوع فایل نمایانگر یک کنترل تعریف شده توسط برنامه نویس ASP.NET است (User Control).
- .asmx**: این پسوند برای سرویس های XML وب استفاده می گردد.
- .aspx**: از این پسوند که اصلی ترین پسوند ASP.NET است برای فرم های وب و صفحات معمولی ASP.NET استفاده می گردد.
- .axd**: این نوع فایل مربوط به tracing برنامه کاربردی ASP.NET است.
- .vsdisco**: این نوع فایل XML، لینک ها را در معرض منابع دیگری قرار می دهد که سرویس وب را توصیف می کنند.
- .htm**: فایل های معمولی از نوع HTML را در برمی گیرند.
- .xml**: این نوع سند XML مخصوص استفاده در برنامه های کاربردی ASP.NET است
- .vb**: این فایل شامل کد ویژوال بیسیک است
- .cs**: همانند پسوند vb است اما بجای کد ویژوال بیسیک حاوی کد زبان C# است.
- .config**: پسوند فایل web.config است. این نوع فایل بر یک فایل پیکربندی دلالت دارد.

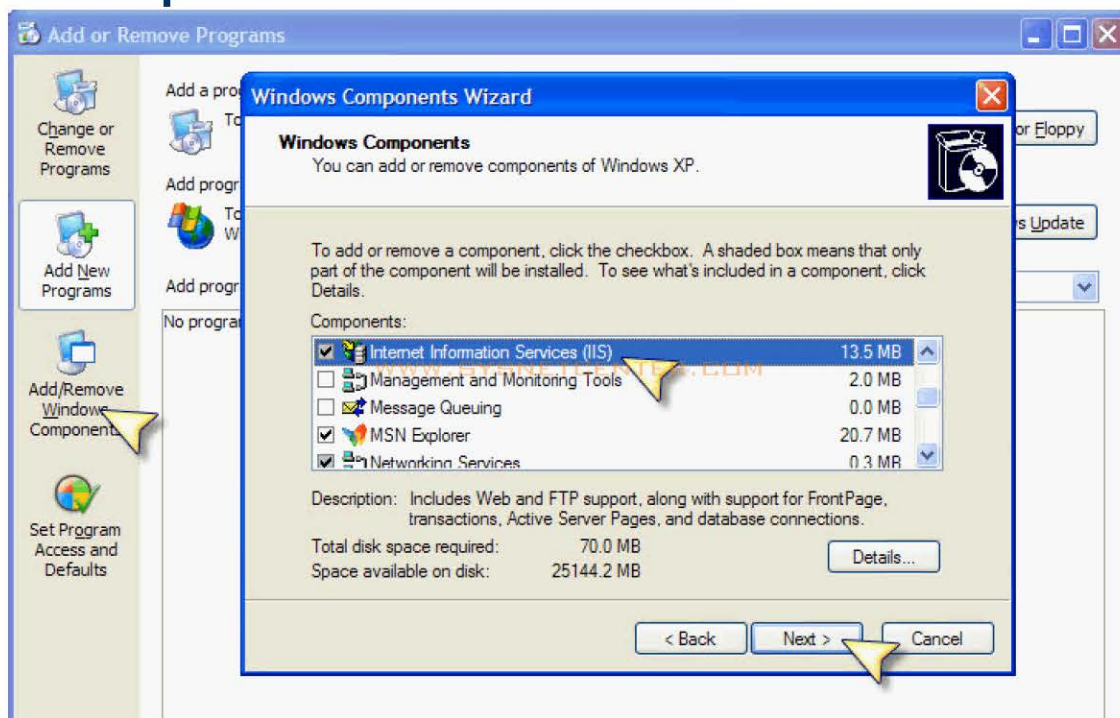
برای کار با ASP.NET به چه چیزهایی نیاز داریم؟

- ✓ سیستم عامل ویندوز (2000 – XP – 2003 و...)
- ✓ وب سرور IIS (سرور)
- ✓ بانک اطلاعاتی (SQL SERVER, ORACLE, MS-ACCESS)
- ✓ مرورگر اینترنت (Web Browser) (کلاینت)
- ✓ محیط طراحی (Visual Studio – Notepad – WEB Matrix)

برای کار با ASP.NET به چه چیزهایی نیاز داریم؟

- برای نصب IIS مراحل زیر را به ترتیب انجام می دهیم :

**Xp: Start → Control Panel → Add Or Remove Programs →
Add / Remove Windows Component**

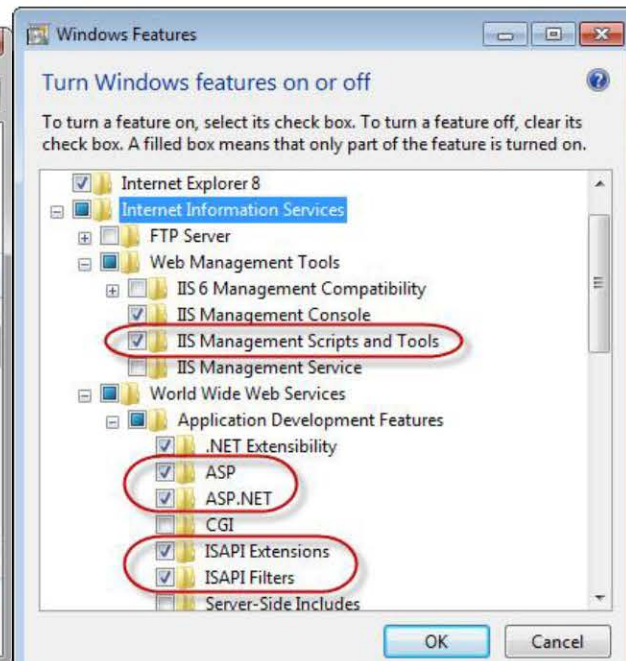
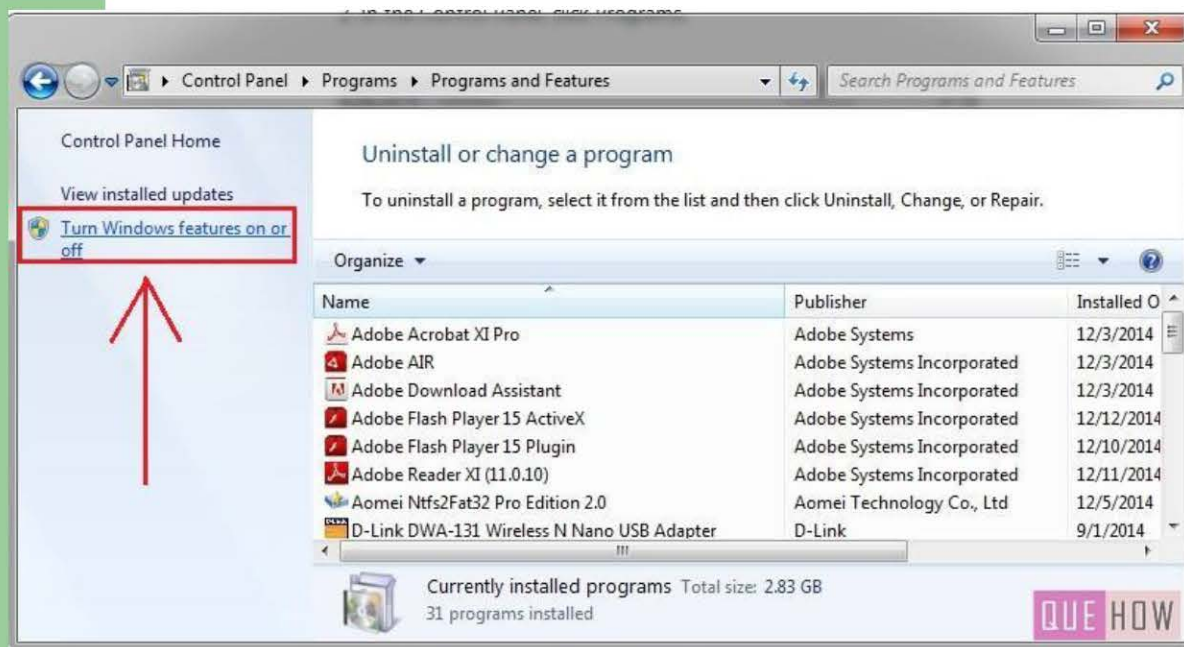


برای کار با ASP.NET به چه چیزهایی نیاز داریم؟

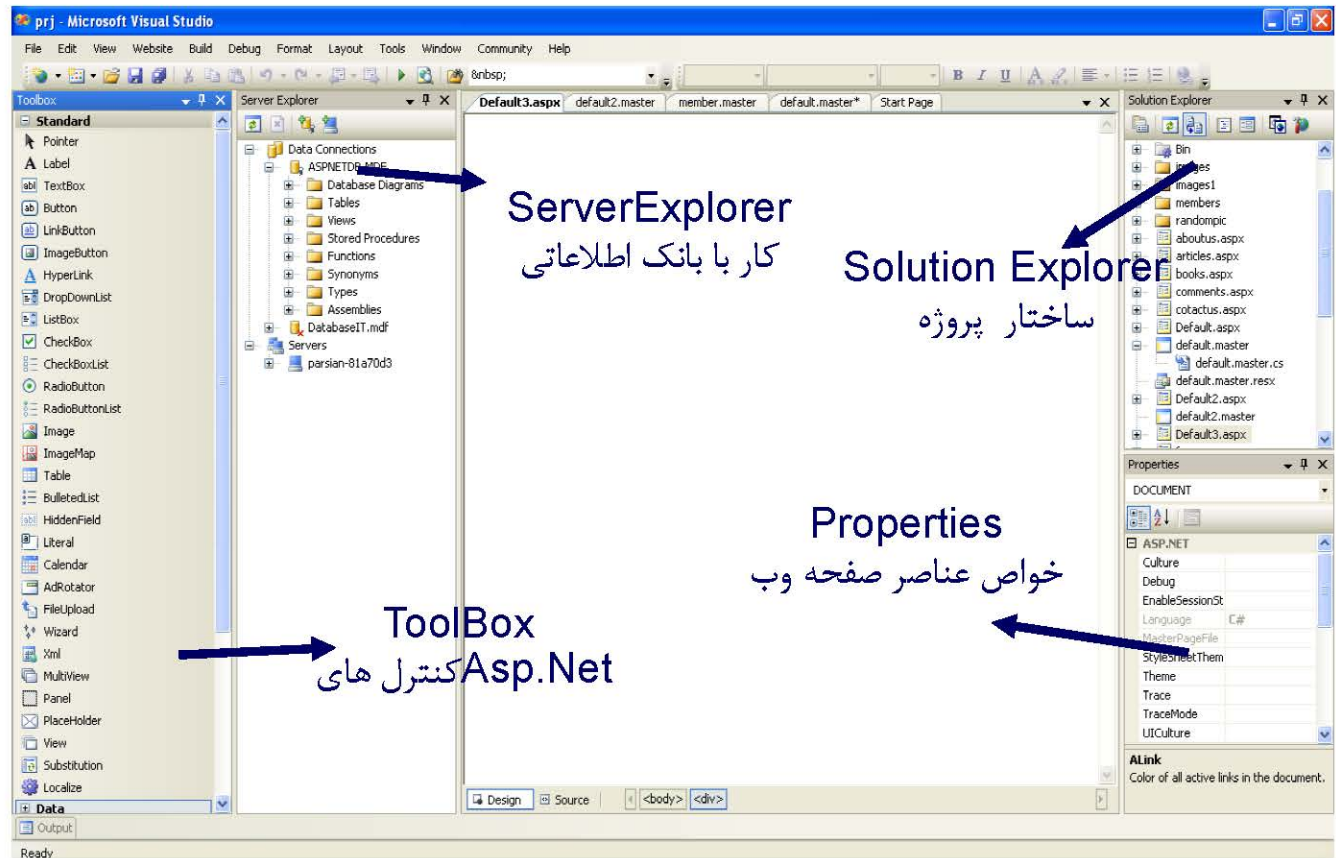
- برای نصب IIS مراحل زیر را به ترتیب انجام می دهیم :

7 : Control Panel → Programs and features → Turn Windows on or off
→ Internet Information Services

در یک مرحله از نصب سی دی ویندوز از شما درخواست می شود.

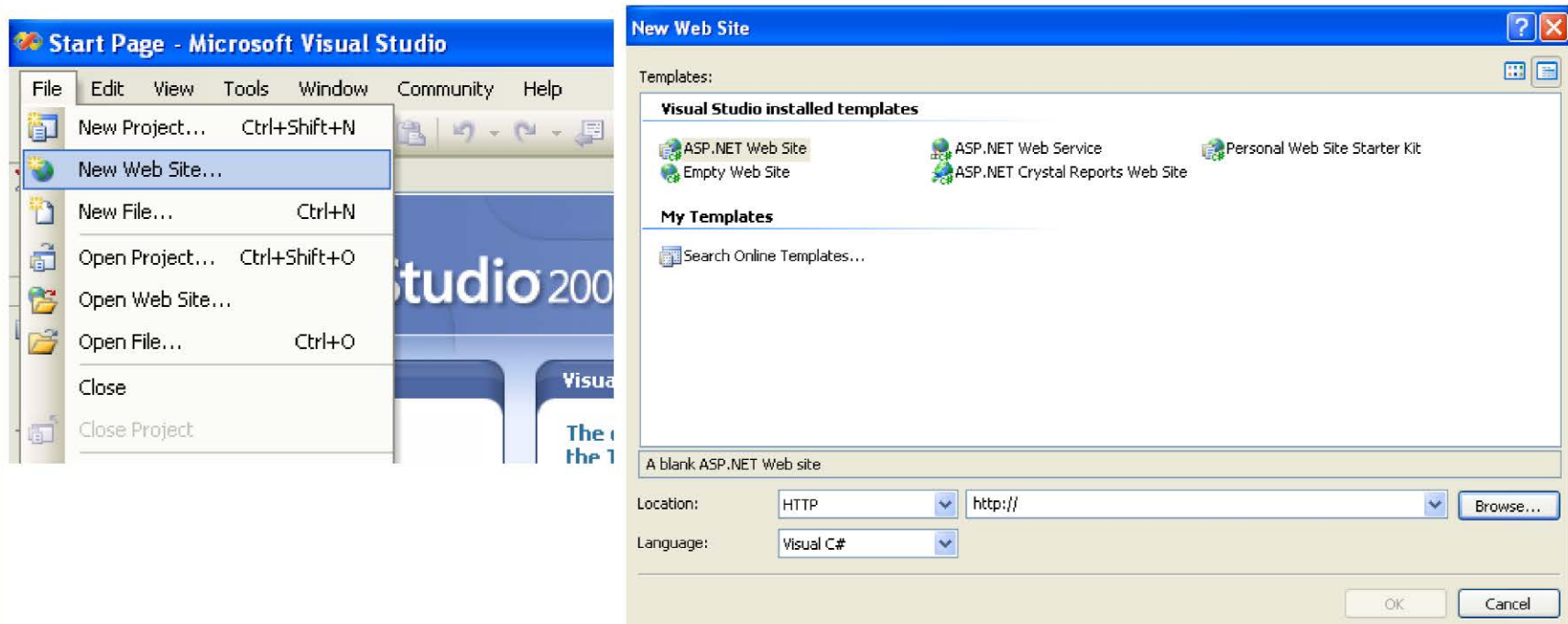


شروع کار با Visual Studio.Net



شروع کار با Visual Studio.Net

File → New Web Site → ASP.NET Web Site



شروع کار با Visual Studio.Net

اولین برنامه Asp.Net

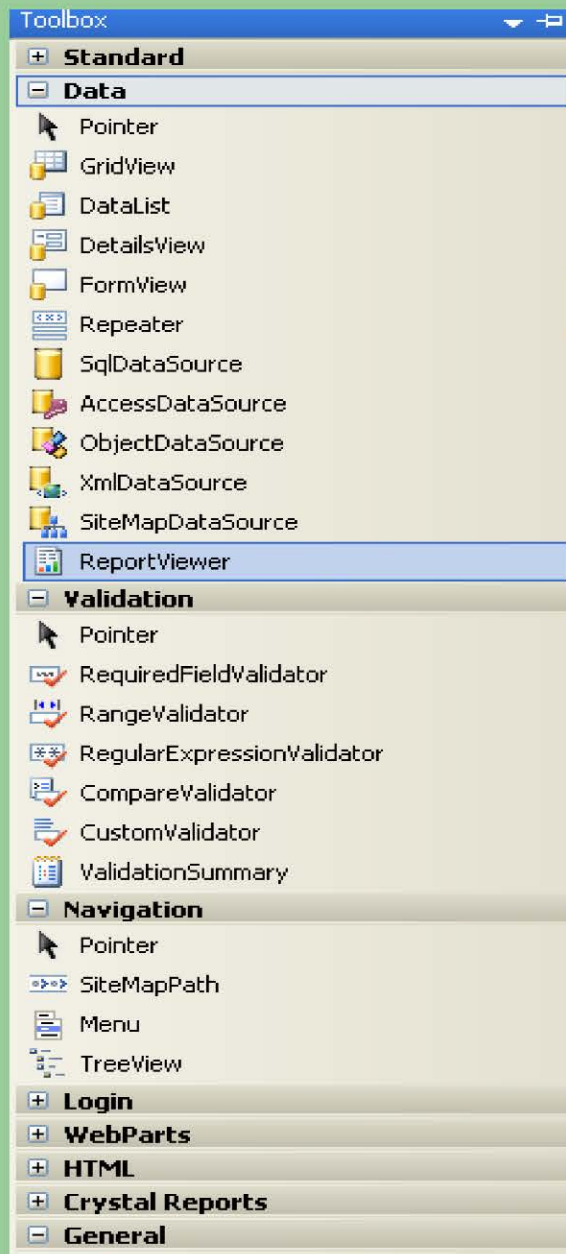
تمرین: در این برنامه نام کاربر از ورودی دریافت می شود و یک پیام خوش آمد گویی به همراه ساعت و تاریخ سیستم برای او به نمایش در می آید.

این برنامه شامل یک کنترل TextBox و یک کنترل Button و دو کنترل Label است

برای اجرای برنامه

۱- کافی است از منوی Debug گزینه Start Debugging یا کلید F5 را فشار دهیم

۲- در Solution Explorer روی فایل Asp.Net مورد نظر کلیک راست کرده گزینه View In Browser را انتخاب کنیم

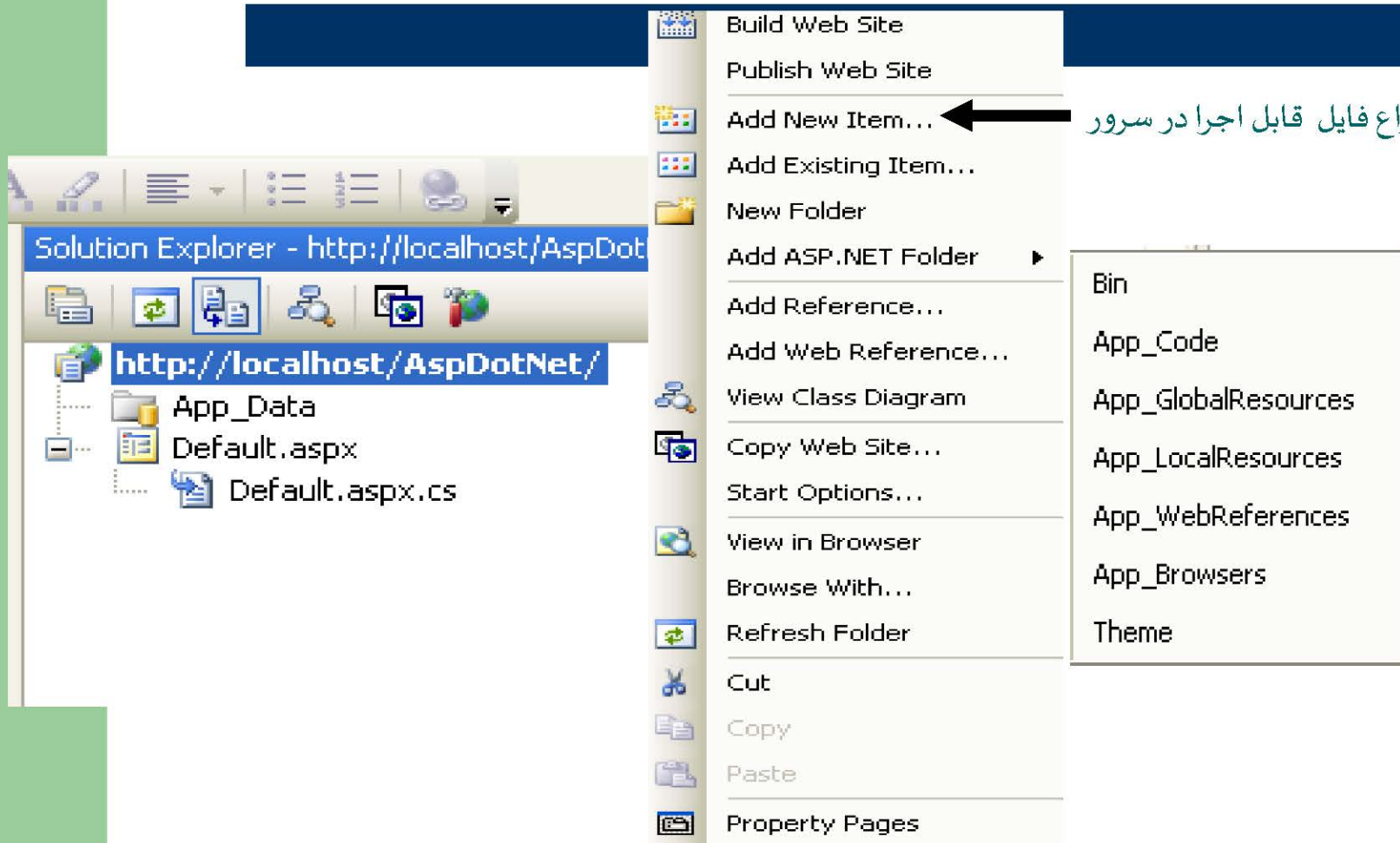


معرفی کنترل های ASP.Net

Standard	↗
Data	↗
Validation	↗
Navigation	↗
Login	↗
Webparts	↗
HTML	↗
Crystal Reports	↗

کار با Solution Explorer

برای ایجاد انواع فایل قابل اجرا در سرور

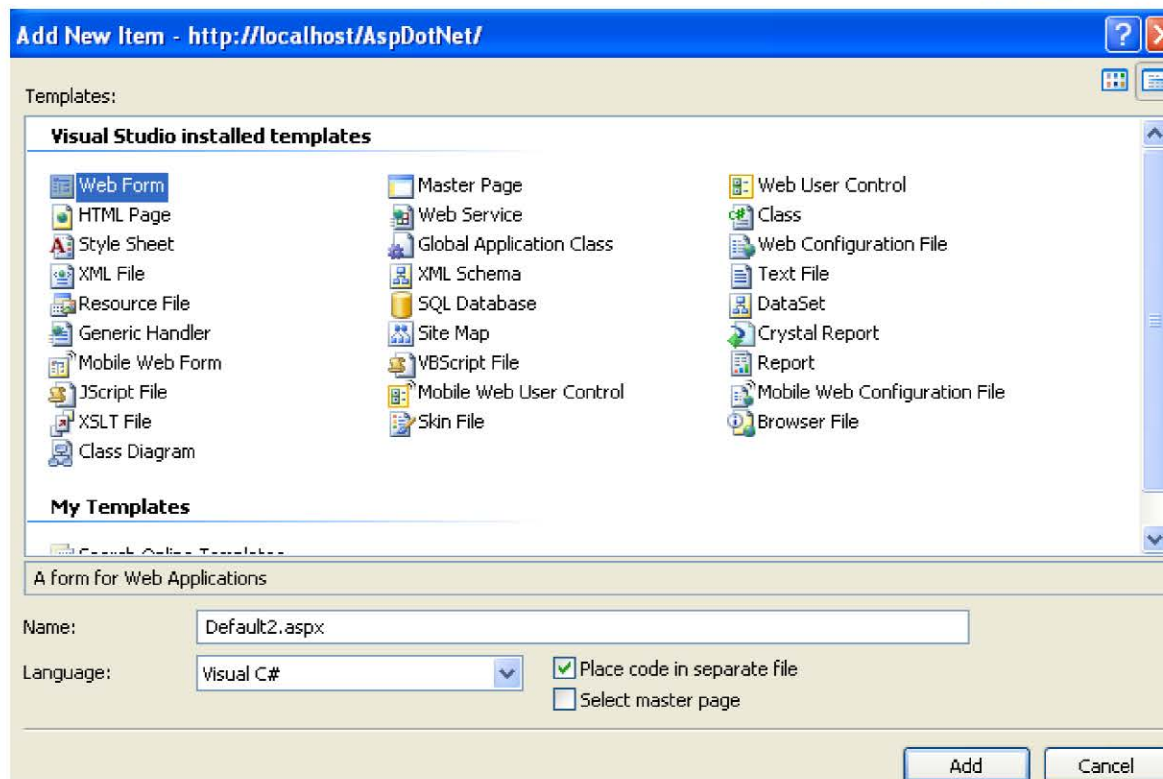


کار با Solution Explorer

پوشه App_Data : فایل بانک اطلاعاتی در این پوشه قرار می گیرد
پوشه Bin : حاوی فایل‌های کامپایل شده اسمبلی (.dll) و کامپوننت ها
و فایل‌های کد مرجع در آن نگهداری می شود
پوشه App_Code : کلاس ها (.vb _ .cs) و فایل‌های DataSet
بانک اطلاعاتی در آن نگهداری می شود
پوشه App_Themes : شامل فایل‌های (.css و .skin) است که برای
طراحی ظاهر و نمای کلی سایت بکار می رود

کار با Solution Explorer

• Add New Item



کار با Solution Explorer

Master Page چیست ؟ ظاهر و قالب یکسان برای کل صفحات وب سایت

- با استفاده از ویژگی فوق می توان ساختار و عناصر اینترفیس مورد نیاز یک سایت را تعریف نمود. تعریف هدر و یا فوتر برای صفحات و یا میله مسیریابی و حرکت ، نمونه هائی در این زمینه است که با تعریف آنان در یک مکان واحد موسوم به **Master page** ، امکان استفاده و به اشتراک گذاشتن آنان بین صفحات متفاوت موجود بر روی یک سایت فراهم می گردد . با استفاده از ویژگی فوق وضعیت مدیریت و نگهداری یک سایت بهبود یافته و از تکرار کدهای غیرضروری به منظور اشتراک ساختار و یا رفتار سایت ، پیشگیری بعمل می آید .
- در **Master page** کنترل ها و بخش هایی را که می خواهیم در تمام صفحات ظاهر شود قرار می دهیم

کار با Solution Explorer

ایجاد Master Page

Solution Explorer → کلیک راست روی نام پروژه → Add New Item → Master Page



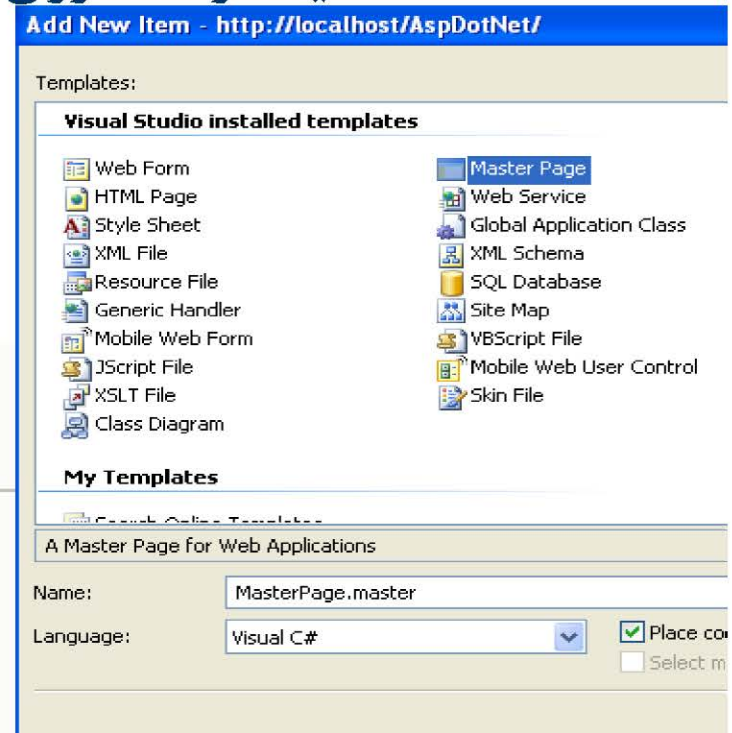
Build Web Site

Publish Web Site

Add New Item...

Add Existing Item...

New Folder



Style Sheet و Skins و Themes

با استفاده از ویژگی های فوق می توان شکل ظاهری یک سایت را متناسب با خواسته کاربر کاملاً سفارشی نمود . پس از تعریف اطلاعات مربوطه به هر Style ، می بایست آنان را در مکان هائی موسوم به "theme" ذخیره تا زمینه استفاده از آنان در سایر صفحات و یا کنترل های موجود بر روی یک صفحه فراهم گردد . ویژگی فوق همانند master page ، باعث بهبود وضعیت مدیریت و نگهداری سایت و پیشگیری از نوشتن کدهای تکراری برای دستیابی به styles مشترک می گردد .

Skin فایلی است که در آن تنظیمات ظاهری کنترل های **Asp.net** ایجاد می شود برای ایجاد آن بر روی پوشه Theme در Solution Explorer کلیک راست کرده و گزینه Add New Item را انتخاب کرده سپس از کادر باز شده گزینه Skin File را انتخاب می کنیم و در نهایت می توانیم در داخل فایل مورد نظر برای کنترل های **Asp.Net** مورد نظر تنظیمات دلخواه را انجام دهیم.
مثال :

1. `<asp:Image runat="server" ImageUrl="~/images/image1.jpg" />`
2. `<asp:label Runat="server" ForeColor="#004000" Font-Names="tahoma" Font-size="X-Small" />`
3. `<asp:TextBox Runat="server" ForeColor="#004000" Font-ames="tahoma" Font-size="X-Small" Borderstyle="solid" />`

Style Sheet و Skins و Themes

نحوه اعمال تم

اعمال تم به یک صفحه :

```
<%# page language="c#" Theme="summer" %>
```

اعمال تم به کل وب سایت :

در فایل Web.Config دستورات زیر را اضافه می کنیم.

```
<configuration>  
<system.web>  
<pages theme="summser" />  
</system.web>  
</configuration>
```

Style Sheet و Skins و Themes

ایجاد شیوه نامه (Style Sheet)

برای ایجاد آن بر روی پوشه Theme در Solution Explorer کلیک راست کرده و گزینه Add New Item را انتخاب کرده سپس از کادر باز شده گزینه Style Sheet را انتخاب می کنیم و در نهایت می توانیم در داخل فایل مورد نظر برای کنترل های مورد نظر اعم از Html و Asp.Net تنظیمات دلخواه را انجام دهیم برای اینکار داخل فایل Style Sheet کلیک راست کرده Add Style Role را انتخاب می کنیم و در آخر در کادر باز شده برای عناصر مورد نظر تنظیمات دلخواه را انجام می دهیم.

۱- اعمال به یک صفحه

در بخش Source صفحه در قسمت <head> دستور زیر را وارد می کنیم:

```
<head runat="server">  
  <link href="Style.css" rel="stylesheet" type="text/css" />  
</head>
```

۲- اعمال به کل وب سایت

در فایل Web.Config دستورات زیر را اضافه می کنیم:

```
<configuration>  
  <system.web>  
    <pages StyleSheetTheme="ThemeName" />  
  </system.web>  
</configuration>
```

Style Sheet و Skins و Themes

توجه : برای اعمال تنظیمات به عناصر صفحه کافی است در بخش Design صفحه، روی عناصر کلیک کرده بعد در قسمت Properties مربوط به کنترل مورد نظر در قسمت Id ، Cssclass تنظیمات مورد نظر را وارد کنیم.

پیمایش سایت

سیستم پیمایش سایت که دارای قابلیت تعریف کل سایت در یک فایل XML است نقشه سایت یا Site Map نامیده می شود

- از نقشه سایت برای تعریف طرح کلی تمام صفحات در برنامه و نحوه ارتباط آنها با یکدیگر استفاده می شود.
- با استفاده از کنترل SiteMapDataSource می توان اطلاعاتی را که در فایل نقشه سایت است به کنترل های انقیاد داده مقید کنید.

نحوه ساخت فایل نقشه سایت (Site Map)

برای ایجاد آن بر روی نام پروژه در Solution Explorer کلیک راست کرده و گزینه Add New Item را انتخاب کرده سپس از کادر باز شده گزینه Site Map را انتخاب می کنیم سپس داخل فایل رفته و ساختار پروژه خود را بصورت فایل XML در آن ایجاد می کنیم.

پیمایش سایت

● کنترل SiteMapPath

مسیر خطی از محلی است که کاربر در آن صفحه است

● کنترل TreeView

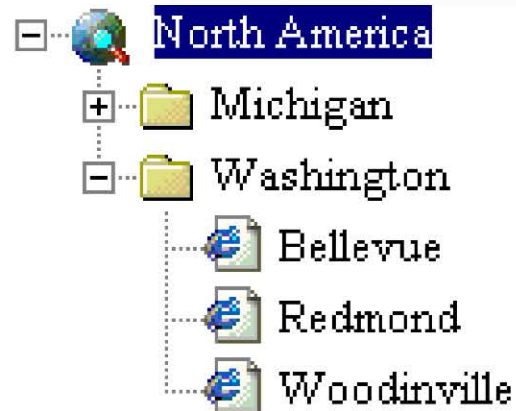
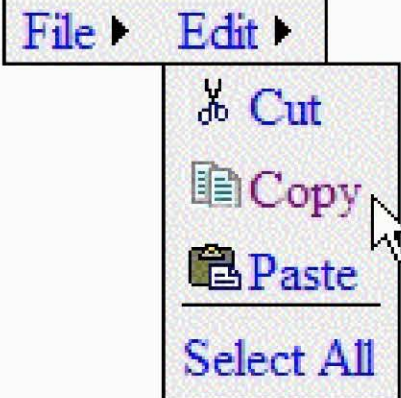
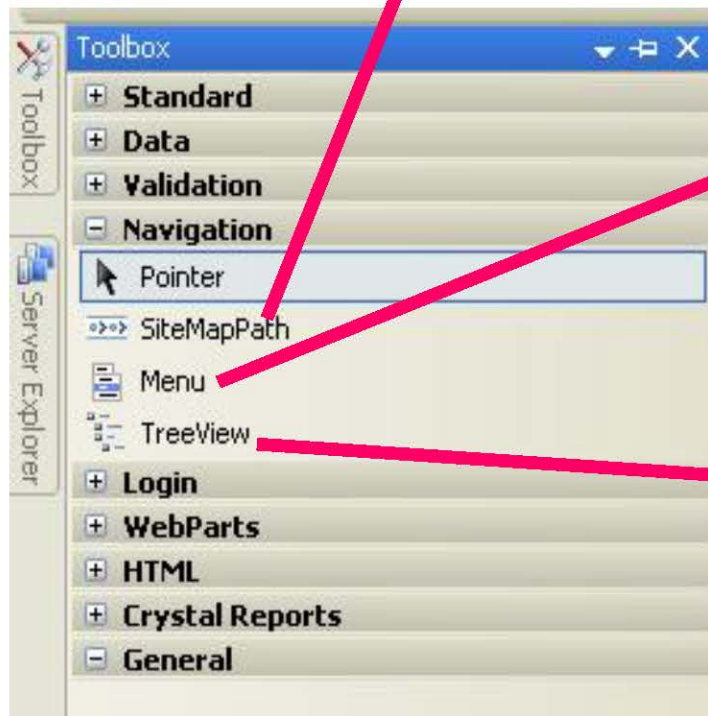
ساختار درختی و سلسله مراتبی وب سایت را نشان می دهد

● کنترل Menu

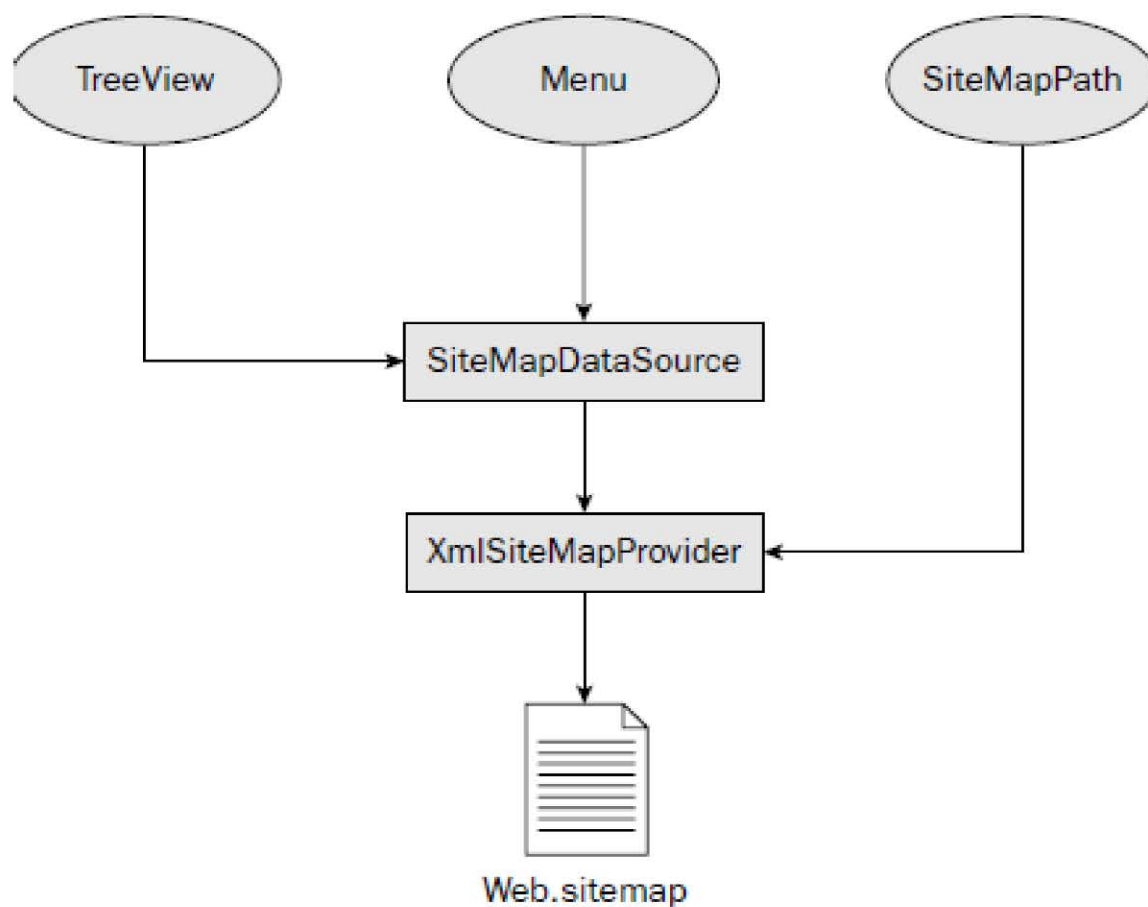
برای ساخت انواع منو از روی فایل SiteMap بکار می رود

پیمایش سایت

صفحه آغازین > عضویت در سایت



پیمایش سایت



مثالی از چگونگی ایجاد فایل Site Map

```
<?xml version="1.0" encoding="utf-8"?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" enableLocalization="true">
  <siteMapNode title="آغازین صفحه" url="default.aspx">
    <siteMapNode title="ما درباره" url="aboutus.aspx" />
    <siteMapNode title="ما با تماس" url="contactus.aspx" />
    <siteMapNode title="مقالات" url="articles.aspx" />
    <siteMapNode title="پیام ارسال سیستم" url="sendmessage.aspx" />
    <siteMapNode title="کتاب معرفی" />
    <siteMapNode title="دانشگاه سایت وب" url="university_website.aspx" />
    <siteMapNode title="کتاب معرفی" url="books.aspx" />
    <siteMapNode title="عبور رمز بازیابی" url="passwordrecovery.aspx" />
    <siteMapNode title="سایت در عضویت" url="registermember.aspx" />
    <siteMapNode title="اعضا" url="members/member.aspx" />
    <siteMapNode title="رمز تغییر" url="members/passwordchange.aspx" />
  </siteMapNode>
</siteMapNode>
</siteMap>
```

} Root

}

شرح کلی کنترل های اعتبار سنجی

کنترل	کاربرد
RequiredFieldValidator	بررسی می کند که آیا کنترل حاوی داده است یا خیر (آیا کاربر چیزی را وارد کرده است؟)
CompareValidator	بررسی می کند که آیا داده وارد شده با داده ی موجود در کنترل دیگر تطابق دارد یا خیر.
RangeValidator	بررسی می کند که آیا آیتم وارد شده بین دو مقدار تعریف شده قرار دارد یا خیر.
RegularExpressionValidator	بررسی می کند که آیا داده وارد شده با فرمت مشخص شده مطابقت دارد یا خیر.
CustomValidator	اعتبار داده ورودی را توسط اسکریپتی کلاینت سایت یا سمت سرور و یا هر دو انجام می دهد.
ValidationSummary	تمام موارد بررسی شده را در یک مکان نمایش می دهد یا به صورت کلی فقط یک پیغام را نمایش می دهد.

شرح کلی کنترل های اعتبار سنجی

```
<div>
  <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
  <asp:RequiredFieldValidator ID="RequiredFieldValidator1" ControlToValidate="TextBox1"
    runat="server" ErrorMessage="باید پر باشد" EnableClientScript="false"></asp:RequiredFieldVa
  <asp:CustomValidator ID="CustomValidator1" runat="server" ControlToValidate="TextBox1"
    OnServerValidate="CustomValidator1_ServerValidate" EnableClientScript="false"
    ErrorMessage="بیشتر از 8 کاراکتر بیشتر باشد"></asp:CustomValidator>
  <asp:Button ID="Btn_submit" Text="Salam" runat="server" />
  <asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server"
    ValidationExpression="\d{8}" ControlToValidate="TextBox1" EnableClientScript="false"
    ErrorMessage="عدد 8 رقمی باید وارد شود"></asp:RegularExpressionValidator>
</div>
*//////////////////////////////////*
Protected Sub CustomValidator1_ServerValidate(source As Object, args As ServerValidateEventArgs)
  If args.Value.Length >= 8 Then
    args.IsValid = True
  Else
    args.IsValid = False
  End If
```

شرح کلی کنترل های اعتبار سنجی

باید پر باشد

باید از 8 کاراکتر بیشتر باشد عدد 8 رقمی باید وارد شود

عدد 8 رقمی باید وارد شود

Web.Config

کاربرد بسیار مهم: ذخیره رشته اتصال به بانک اطلاعاتی در این فایل

1. `<connectionStrings>`
`<add name="NorthwindConnectionString" connectionString="Data Source=serverName;Initial Catalog=Northwind;User ID=username; Password=password" providerName="System.Data.SqlClient" />`
`</connectionStrings>`
2. `<connectionStrings>`
`<add name="ConnectionString" connectionString="Data Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\ASPNETDB.MDF;Integrated Security=True;" providerName="System.Data.SqlClient"/>`
`</connectionStrings>`

فایل Global.asax

- همانند فایل **Web.Config** در ریشه سایت قرار می گیرد
- کدهای موجود در فایل Global.asax در یکی از سه مورد زیر اجرا می شوند :
 1. زمانی که برنامه کاربردی کلاً آغاز یا متوقف می شود
 2. زمانی که هر کاربری وارد سایت می شود یا از آن خارج می شود
 3. زمانی که رویدادی در صفحات رخ می دهد مثل (ورود یک کاربر یا بروز خطا)

فایل Global.asax

```
<%@ Application Language="C#" %>
<script runat="server">
    void Application_Start(object sender, EventArgs e)
    {
        // Code that runs on application startup
    }
    void Application_End(object sender, EventArgs e)
    {
        // Code that runs on application shutdown
    }
    void Application_Error(object sender, EventArgs e)
    {
        // Code that runs when an unhandled error occurs
    }
    void Session_Start(object sender, EventArgs e)
    {
        // Code that runs when a new session is started
    }
    void Session_End(object sender, EventArgs e)
    {
        // Code that runs when a session ends.
    }
</script>
```

فایل Global.asax

کاربران آنلاین و کل بازدید ها

```
<script language="C#" runat="server">
    public void Application_OnStart()
    {
        Application["UsersOnline"] = 0;
    }
    public void Session_OnStart()
    {
        Application.Lock();
        Application["UsersOnline"] = (int)Application["UsersOnline"] + 1;
        updateCounter();
        Application.Unlock();
    }
    public void Session_OnEnd()
    {
        Application.Lock();
        Application["UsersOnline"] = (int)Application["UsersOnline"] - 1;
        Application.Unlock();
    }
    public void UpdateCounter()
    {
        اتصال به بانک اطلاعاتی و افزودن یک مقدار به تعداد کل کاربران
    }
</script>
```

ADO.NET

- اشیای اتصال «Connection» ↙
- اشیای فرمان «Command» ↙
- اشیای «DataAdapter» ↙
- اشیای «DataSet» ↙
- اشیای «DataReader» ↙

ADO.NET

↙ اشیای اتصال «Connection»

برای کار کردن با داده ها باید با یک منبع داده (Data Source) ارتباط برقرار کنید
منبع داده می تواند یک بانک اطلاعاتی مثل «Access , SQL Server , Oracle» باشد
برای برقراری دستیابی به منبع داده به یک تامین کننده (Provider) نیاز دارید
انواع تامین کننده داده ها عبارتند از :

SQL Server
OLE DB
Oracle

ADO.NET

❖ اشیاى اتصال «Connection»

SqlConnection □ : برای اتصال به بانک اطلاعاتی SQL Server

```
connectionString="Data Source=localhost; Integrated  
Security=True;User Instance=True;Initial Catalog=Northwind;"  
providerName="System.Data.SqlClient" ;
```

OleDbConnection □ : برای اتصال به بانک اطلاعاتی Access

OracleConnection □ : برای اتصال به بانک اطلاعاتی Oracle

ADO.NET

اشیای Command ↙

OleDbCommand

SqlCommand

OracleCommand

**توجه : شی Command دارای خواص و متدهای برای کار با بانک اطلاعاتی است
مثل :**

CommandText (تعریف دستور Sql)

Parameters (برای تعریف پارامتر)

ExecuteNonQuery (این متد برای عملیات درج ، حذف ، و بهنگام سازی استفاده می

شود)

ExecuteReader (برای بازیابی اطلاعات و ارسال نتایج به شی DataReader)

ExecuteScalar (بازیابی اطلاعات مربوط به اولین ستون مربوط به اولین سطر)

ADO.NET

← اشیای Command

- این اشیای نگه‌داری دستورات SQL بکار می‌روند
- برای انجام هر نوع تقاضا در بانک اطلاعاتی، نیاز به دستورات SQL داریم.
- برای استفاده از دستورات SQL ابتدا یک شی اتصال ایجاد می‌کنیم سپس آن اتصال را در خاصیت Connection مربوط به شی Command قرار می‌دهیم
- سپس دستور SQL را در شی Command قرار می‌دهیم
- سرانجام شی Command را در شی DataAdapter مورد استفاده قرار می‌دهیم

```
string strcon =
```

```
ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
```

```
SqlConnection constring = new SqlConnection(strcon);
```

```
string selectcom="select * from News";
```

```
SqlDataAdapter da = new SqlDataAdapter();
```

```
da.SelectCommand = new SqlCommand(selectcom, constring);
```

ADO.NET

↙ اشیای DataAdapter

وقتی شی اتصال به یک منبع داده ایجاد شد ، باید یک DataAdapter برای کار کردن با داده ها ایجاد شود

اشیای DataAdapter برای دریافت نتایج حاصل از اجرای دستورات SQL (Command) و قرار دادن آن در DataSet و اعمال تغییرات انجام شده در DataSet را به بانک اطلاعاتی برعهده دارد.

SqlDataAdapter ✓
OleDbDataAdapter ✓
OracleDataAdapter ✓

ADO.NET

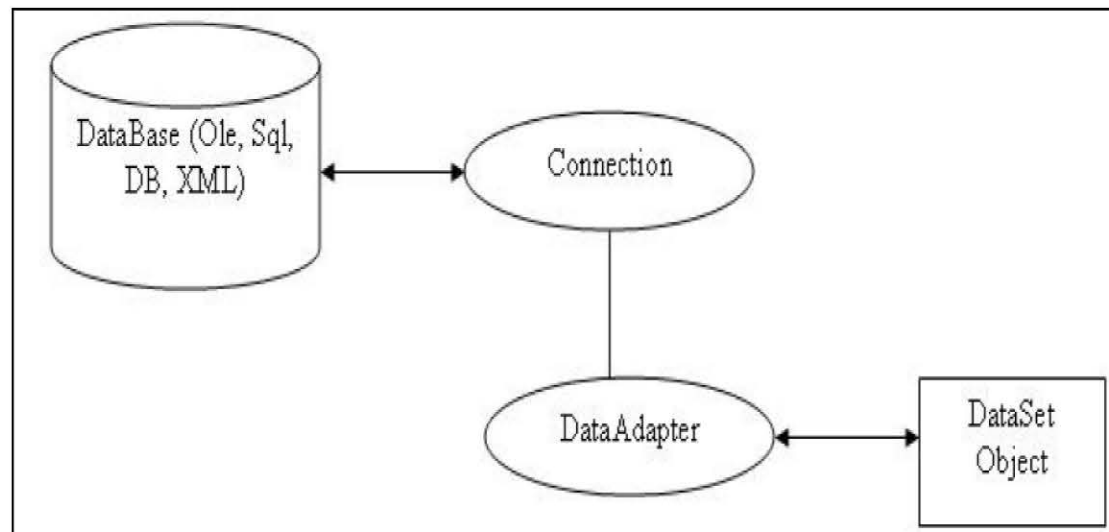
← اشیای DataAdapter

شرح	خاصیت
دستور sql را برای حذف رکوردها از بانک اطلاعاتی تعیین می کند	DeleteCommand
دستور sql را برای درج رکوردها در بانک اطلاعاتی تعیین می کند	InsertCommand
دستور sql را برای بازیابی رکوردها از بانک اطلاعاتی تعیین می کند	SelectCommand
دستور sql را برای بهنگام سازی رکوردها در بانک اطلاعاتی تعیین می کند	UpdateCommand
شرح	متد
این متد رکوردها را از جدول بانک اطلاعاتی به DataSet انتقال می دهد	Fill
دستورات Insert , Delete , Update را روی DataSet اعمال می کند	Update

ADO.NET

شیء DataSet

- DataSet برای ذخیره داده های بازیابی شده از منبع داده بکار می رود
- هر DataSet می تواند حاوی یک یا چند جدول باشد
- DataSet بدون اتصال به منبع داده باقی می ماند
- نقش DataAdapter این است که داده ها را از منبع داده به DataSet انتقال می دهد و همچنین داده های موجود در DataSet را به منبع داده می نویسد



ADO.NET

← اشیاای DataReader

این اشیا داده ها را فیلد به فیلد از منبع داده می خوانند.
هر وقت نیاز به سرعت دسترسی به اطلاعات داشتید از این اشیا استفاده کنید.

1. OleDbDataReader

2. SqlDataReader

3. OracleDataReader

برای کار با بانک اطلاعاتی از کجا شروع کنیم ؟

1. آماده شدن برای دسترسی و کار با داده ها از طریق ADO.NET

برای اینکار به دستورات زیر نیاز دارید :

```
Using System.Data;  
Using System.Data.SqlClient;  
Using System.Data.OleDb;
```

و یا در VB

```
Imports System.Data  
Imports System.Data.SqlClient  
Imports System.Data.OleDb
```

برای کار با بانک اطلاعاتی از کجا شروع کنیم ؟

۲- اتصال به بانک اطلاعاتی :

تعریف شی اتصال و تعیین یک نام برای آن

تعریف رشته اتصال

اتصال با بانک اطلاعاتی

اجرای متد Open شی اتصال

مثال :

```
SqlConnection sqlconn = new SqlConnection("Data  
Source=.\SQLEXPRESS; Initial Catalog=ASPNETDB; Integrated  
Security=True")  
Sqlconn.open();
```

و یا

```
Dim sqlconnection As New  
SqlConnection(ConfigurationManager.ConnectionStrings("connectionStrin  
g").ConnectionString)  
Sqlconn.open();
```

برای کار با بانک اطلاعاتی از کجا شروع کنیم ؟

۳- مطرح کردن تقاضا (تعریف دستورات SQL)

```
String strSql = "Select * from News ";  
Dim strSql AS String= "Select * from News "
```

یا روش دوم :
SqlCommand sqlcom = new SqlCommand ("Select * from News " , نام رشته
(اتصال);
Dim strSql AS New SqlCommand ("Select * from News " , نام رشته
(اتصال)

برای کار با بانک اطلاعاتی از کجا شروع کنیم ؟

۴- تعریف اشیای DataSet و DataAdapter

```
SqlDataAdapter Da = new SqlDataAdapter ();  
Da.SelectCommand= sqlcom;  
DataSet Ds = new DataSet();  
Da.Fill(Ds,"News");
```

۵- بستن رشته اتصال :

```
Sqlconn.Close();
```

به جای استفاده از DataSet و DataAdapter می توانستیم از شی SqlDataReader برای بازیابی اطلاعات استفاده کنیم !
مثال :

```
SqlDataReader reader = sqlcom. ExecuteReader();  
while (reader.Read())  
{  
    label1.text=reader["id"].ToString();  
    label2.text=reader["title"]. ToString();  
    label3.text=reader["Desc"]. ToString();  
}
```

برای کار با بانک اطلاعاتی از کجا شروع کنیم ؟

به جای استفاده از DataSet و DataAdapter می توانستیم از شی DataReader برای بازیابی اطلاعات استفاده کنیم !
مثال :

```
Dim reader As SqlDataReader
SQLCmd = New SqlClient.SqlCommand("Select * From Books Where ID='100'",
sqlconnection)
sqlconnection.Open() 'Open the connection
reader = SQLCmd.ExecuteReader 'Gets Data
If reader.Read() = True Then
    label1.text = reader("fname").ToString()
    label2.text = reader("lname").ToString()
End If
```

نکته:

- استفاده از روش DataSet و DataAdapter را روش Disconnected می گویند
- استفاده از روش DataReader را روش connected می گویند

برای کار با بانک اطلاعاتی از کجا شروع کنیم ؟

```
Using System.Data;  
Using System.Data.SqlClient;
```

```
SqlConnection sqlconn = new SqlConnection("Data Source=.\SQLEXPRESS; Initial  
Catalog=ASPNETDB;Integrated Security=True")
```

```
SqlCommand sqlcom= new SqlCommand (" Select * from News " ,  
sqlconn);  
Sqlconn.open();  
SqlDataAdapter Da = new SqlDataAdapter ();  
Da.SelectCommand= sqlcom;  
DataSet Ds = new DataSet();  
Da.Fill(Ds,"News");  
Sqlconn.Close();
```

برای کار با بانک اطلاعاتی از کجا شروع کنیم ؟

```
Imports System
```

```
Imports System.Data
```

```
Imports System.Data.SqlClient
```

```
-----  
Dim sqlconnection As New
```

```
SqlConnection(ConfigurationManager.ConnectionStrings("connectionString").ConnectionString)
```

```
Dim SQLCmd As New SqlClient.SqlCommand
```

```
Dim SQLdr As SqlClient.SqlDataReader
```

```
SQLCmd = New SqlClient.SqlCommand("Select * From Books Where ID='100",  
sqlconnection) 'Sets the Connection to use with the SQL Command
```

```
sqlconnection.Open() 'Open the connection
```

```
SQLdr = SQLCmd.ExecuteReader 'Gets Data
```

```
If SQLdr.Read() = True Then
```

```
    textbox1.text = SQLdr("fname").ToString()
```

```
    textbox2.text = SQLdr("lname").ToString()
```

```
End If
```

```
SQLdr.Close()
```

```
sqlconnection.Close()
```