

الرحمن  
الرحيم  
بسم الله

## IIS:Internet information services

برای نصب iis مراحل زیر را انجام می دهیم :

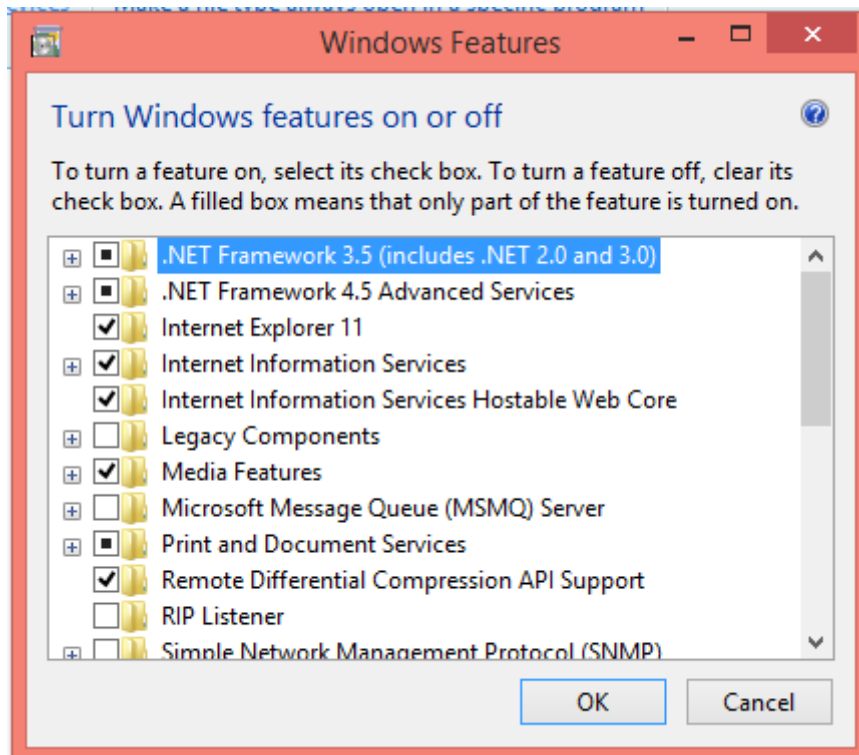
1- با کلیک بر روی گزینه control panel و سپس کلیک روی آیکن program

تا پنجره زیر بازگردد

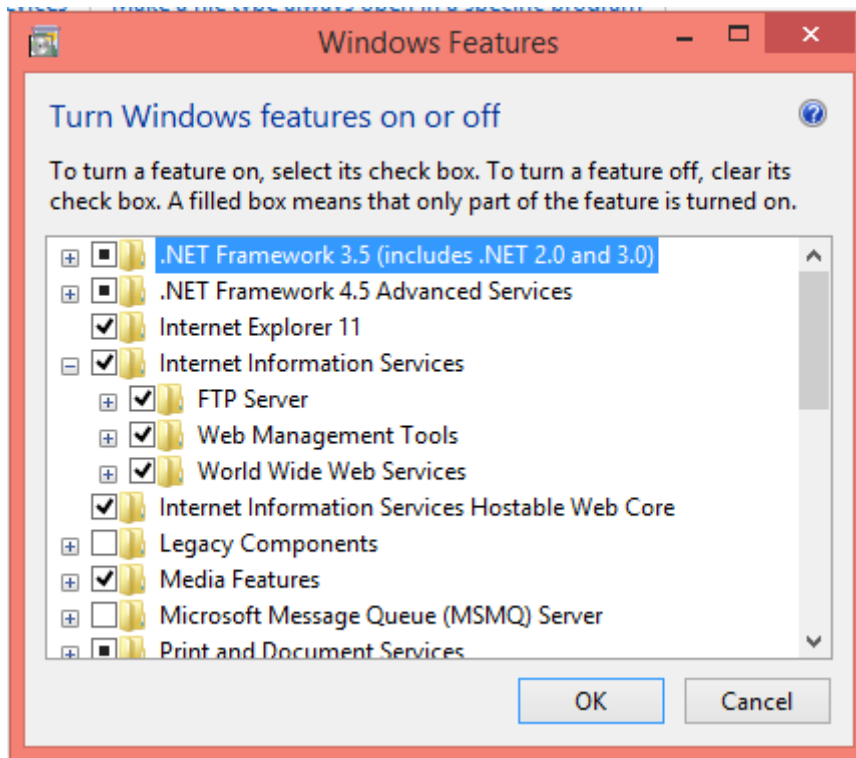


### Programs and Features

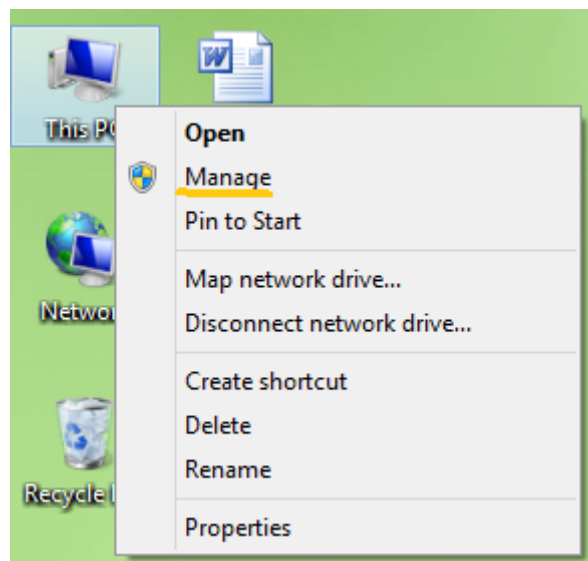
[Uninstall a program](#) | [Turn Windows features on or off](#) | [View installed updates](#) | [Run programs made for previous versions of Windows](#) | [How to install a program](#)



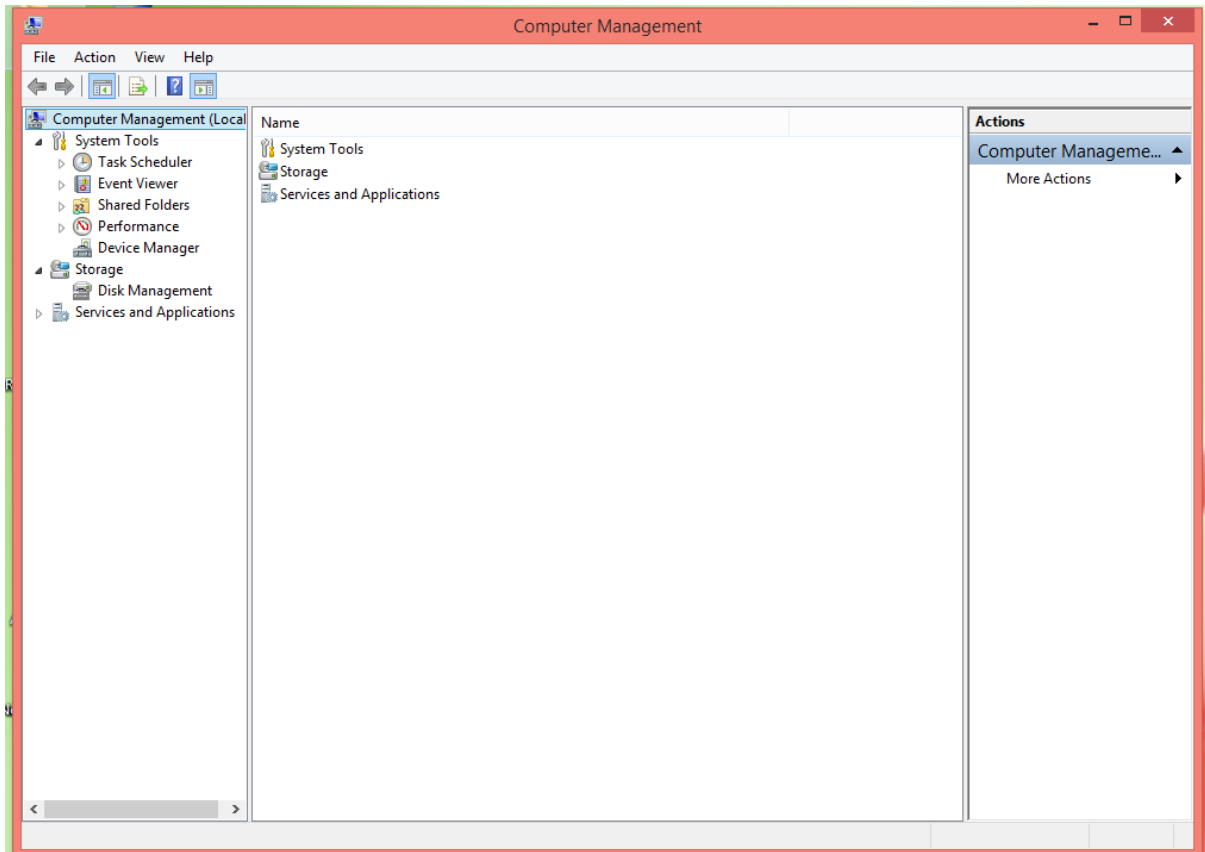
سپس با تیک زدن گزینه های مربوطه



برای مشاهده اینکه آیا سرویس iis فعال هست با خیر موارد زیر عمل می کنیم :  
 1- با راست کلیک بر روی my computer گزینه manage را کلیک می کنیم

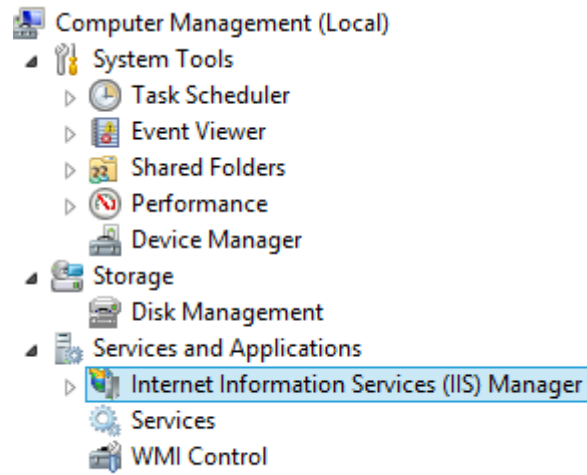


سپس از پنجره زیر

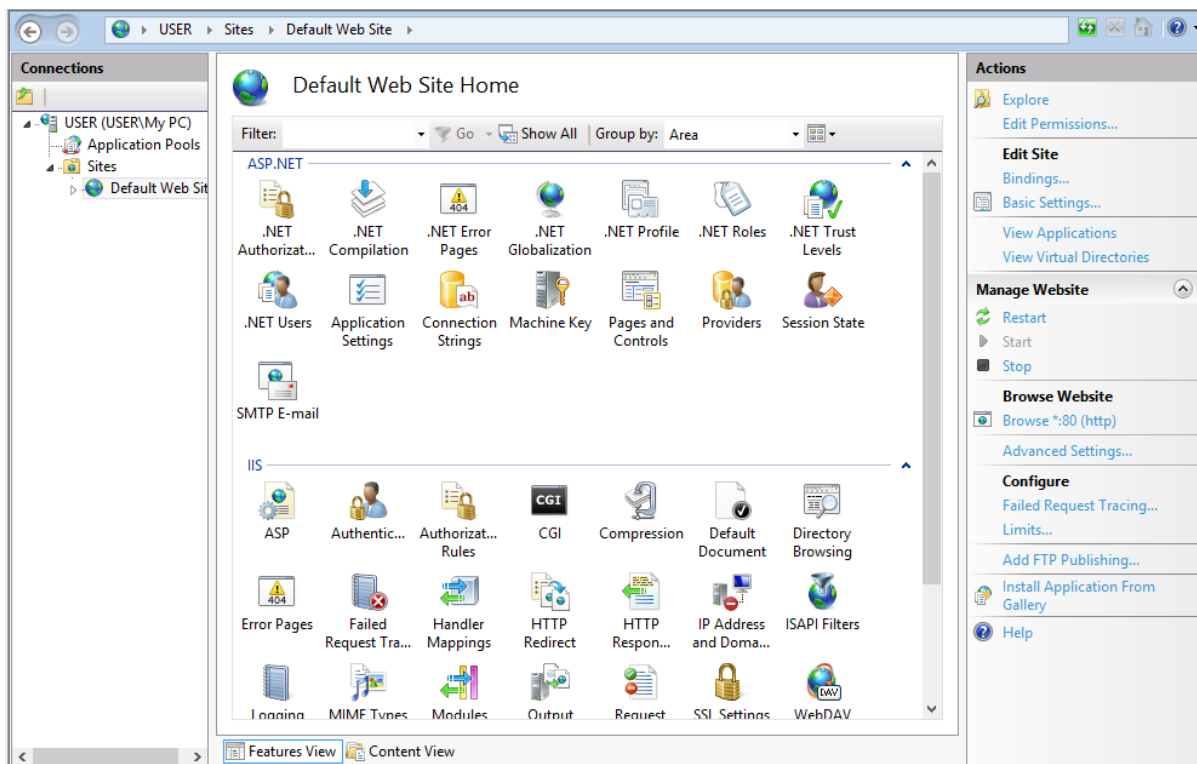


روی گزینه services and Applications کلیک می کنیم

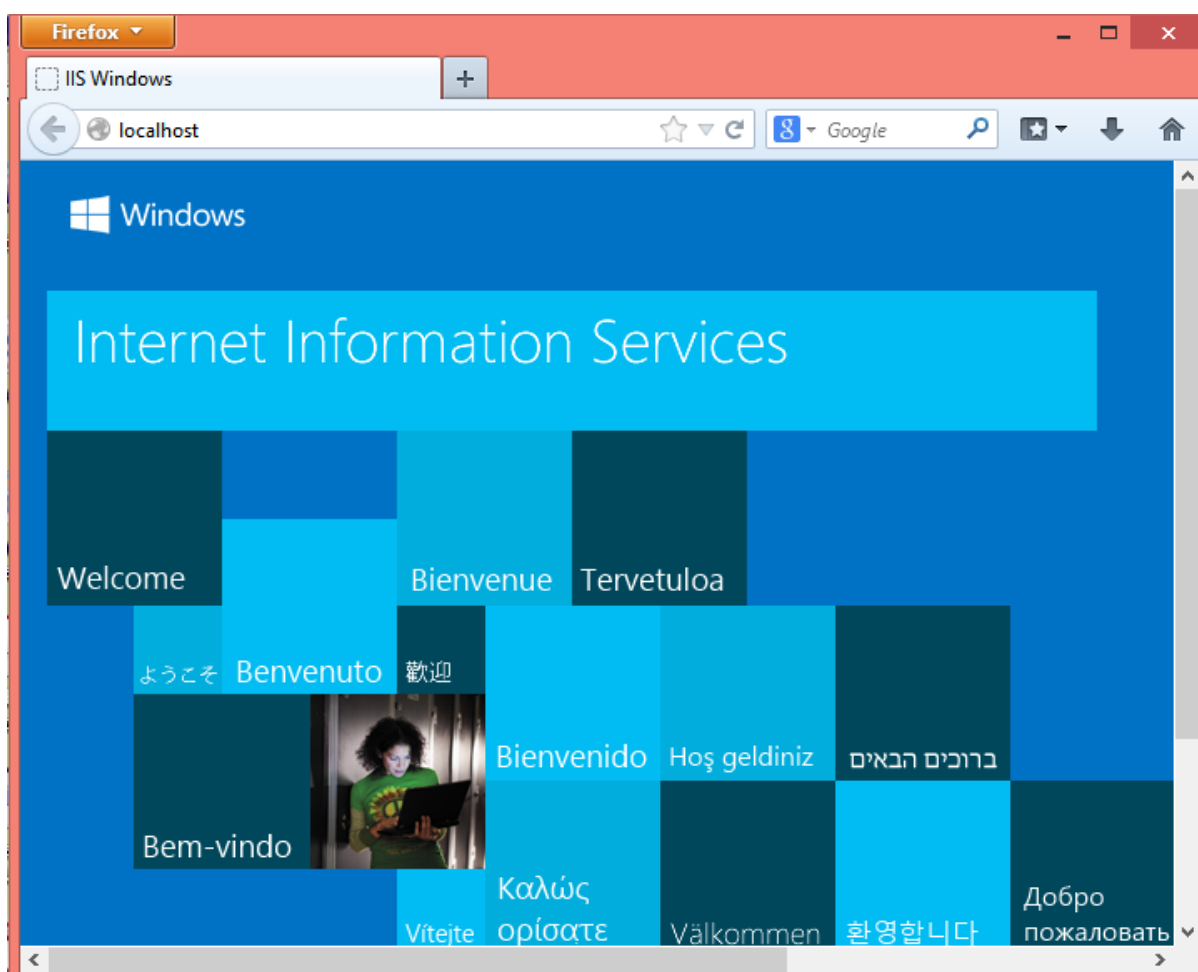
- روی گزینه



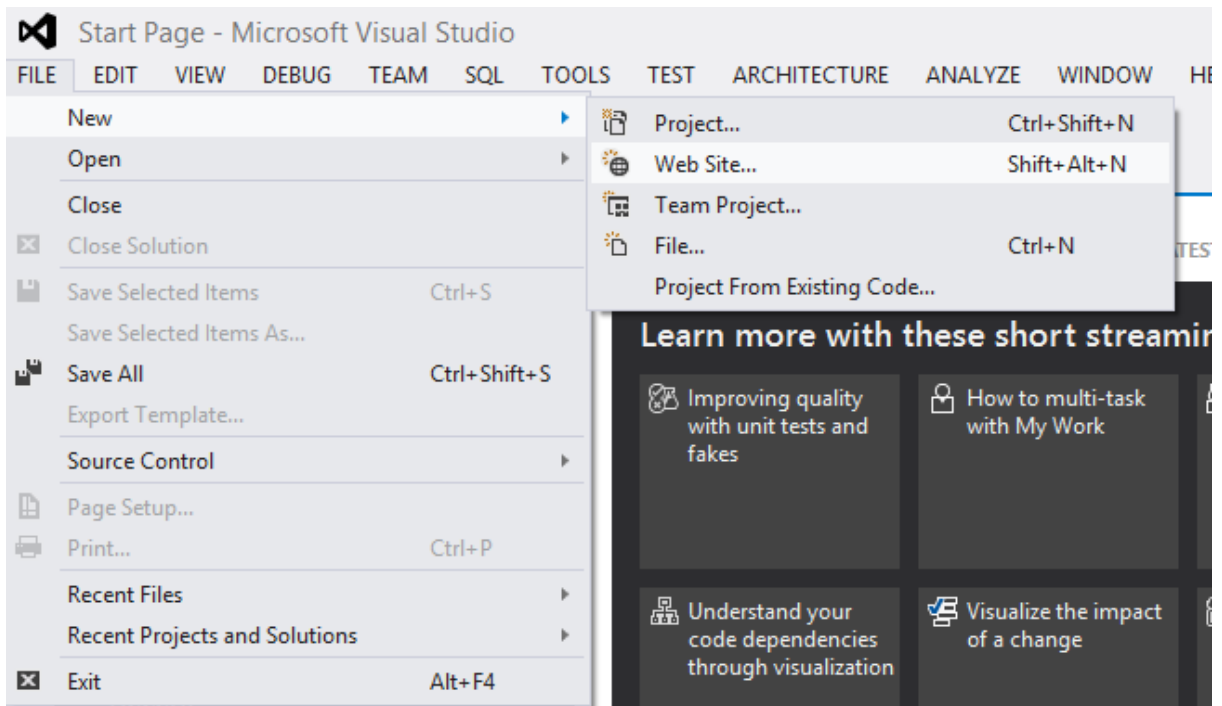
- منوی زیر ظاهر می گردد:



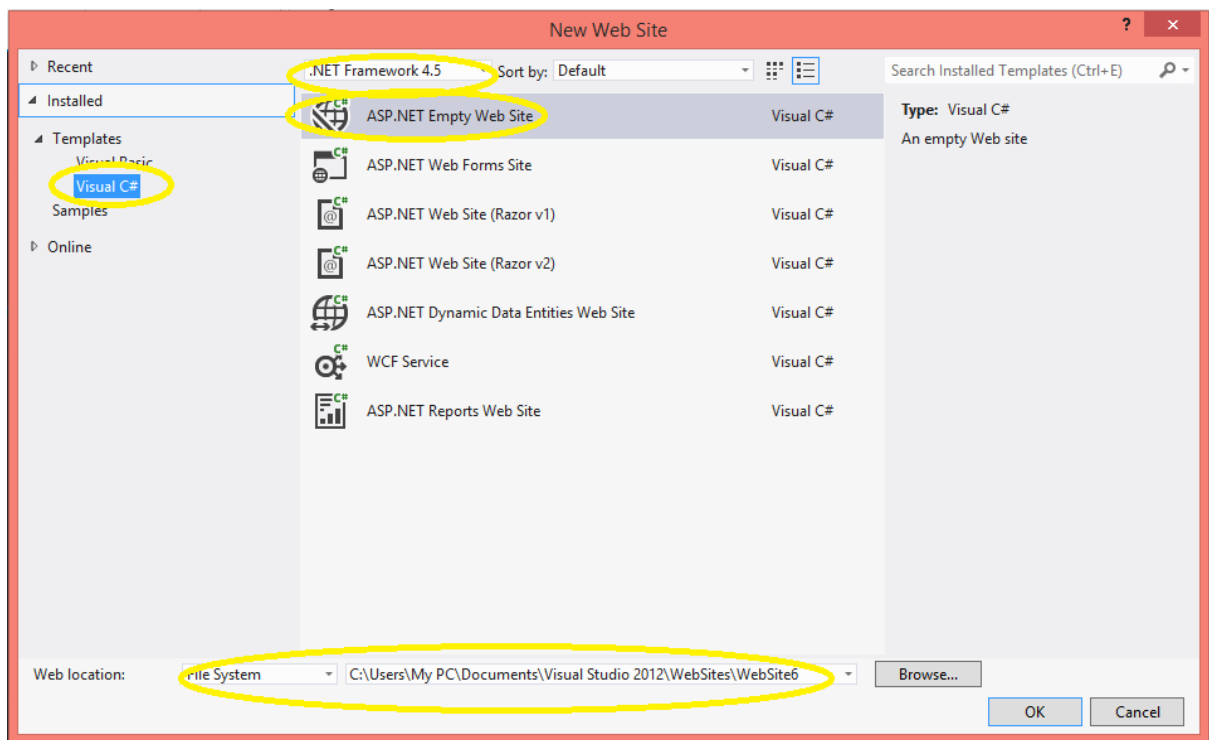
چنانچه گزینه start از بخش manage website در حالت اجرا باشد می توانیم وب سایت را browse کنیم



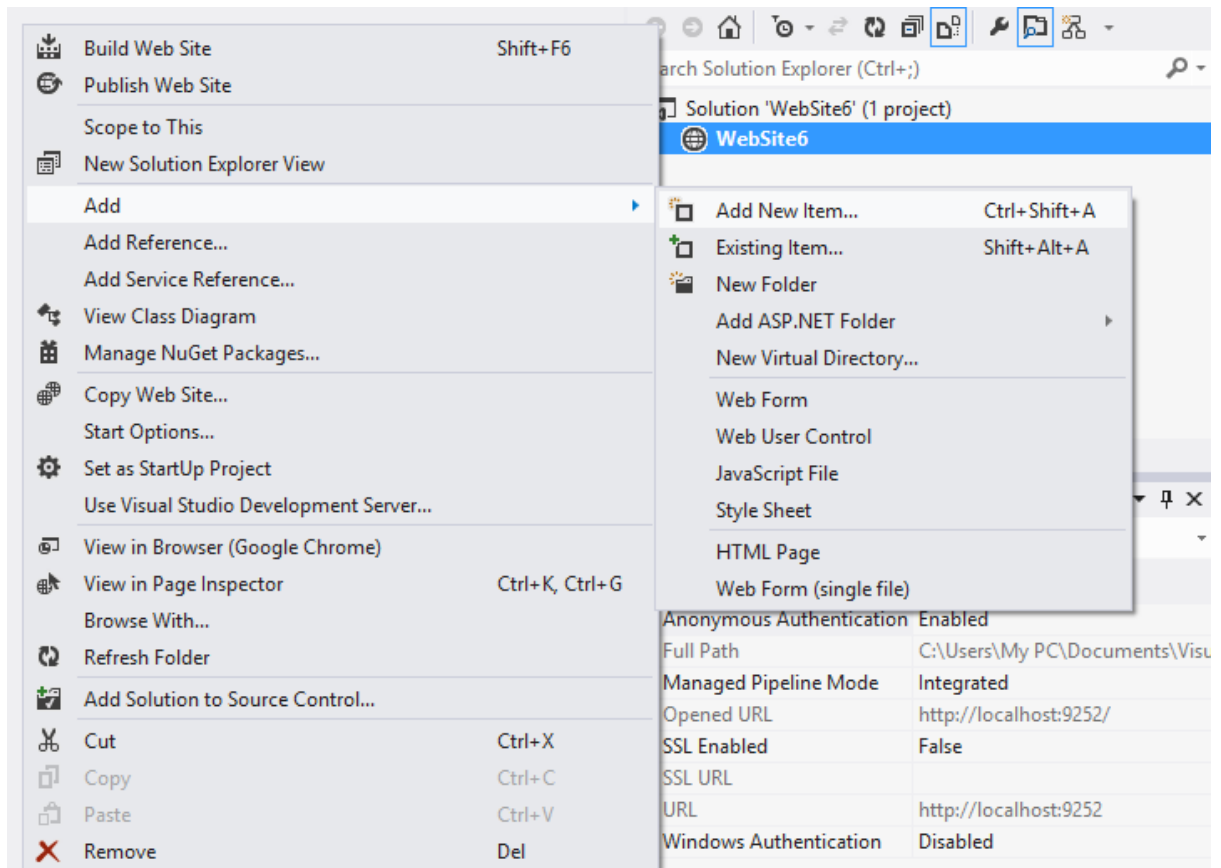
حال می توانیم از asp.net برای طراحی صفحات وب استفاده نماییم برای شروع مراحل زیر را انجام می دهیم  
 ۱- ابتدا نرم افزار visual studio را اجرا می نماییم و سپس از منوهای مشخص شده استفاده می نماییم :



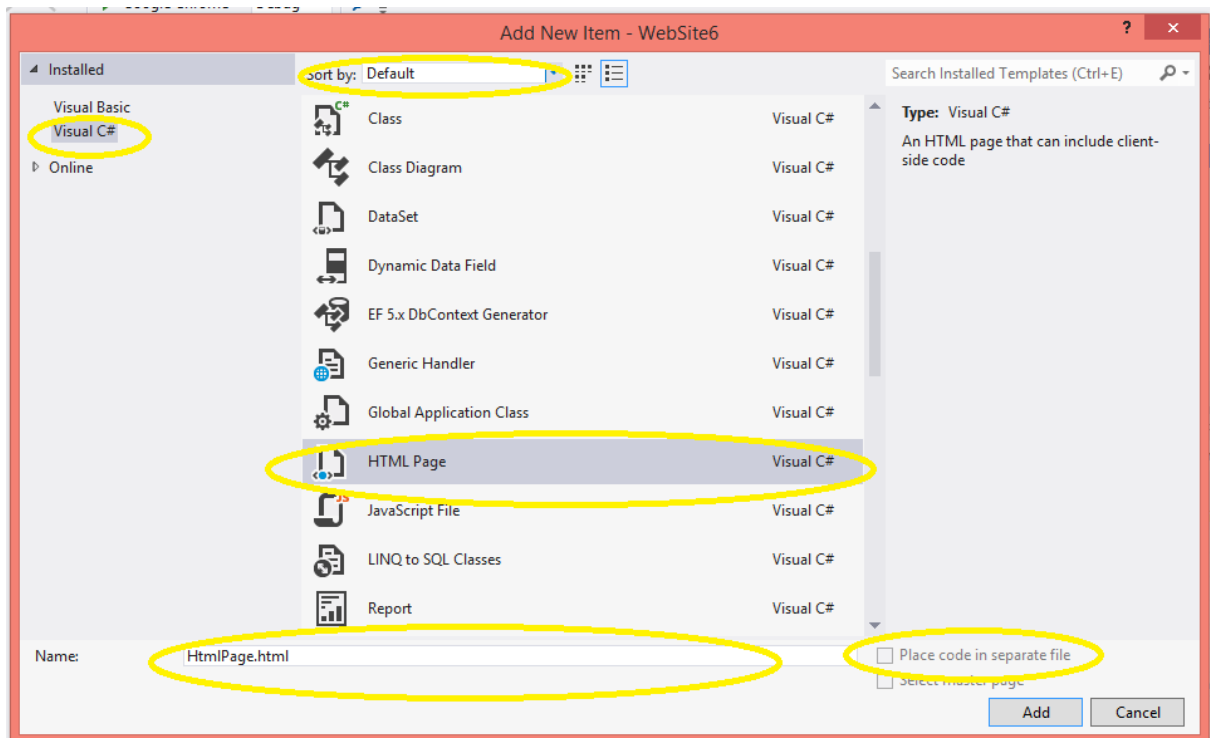
سپس به ترتیب گزینه های زیر را انتخاب می کنیم



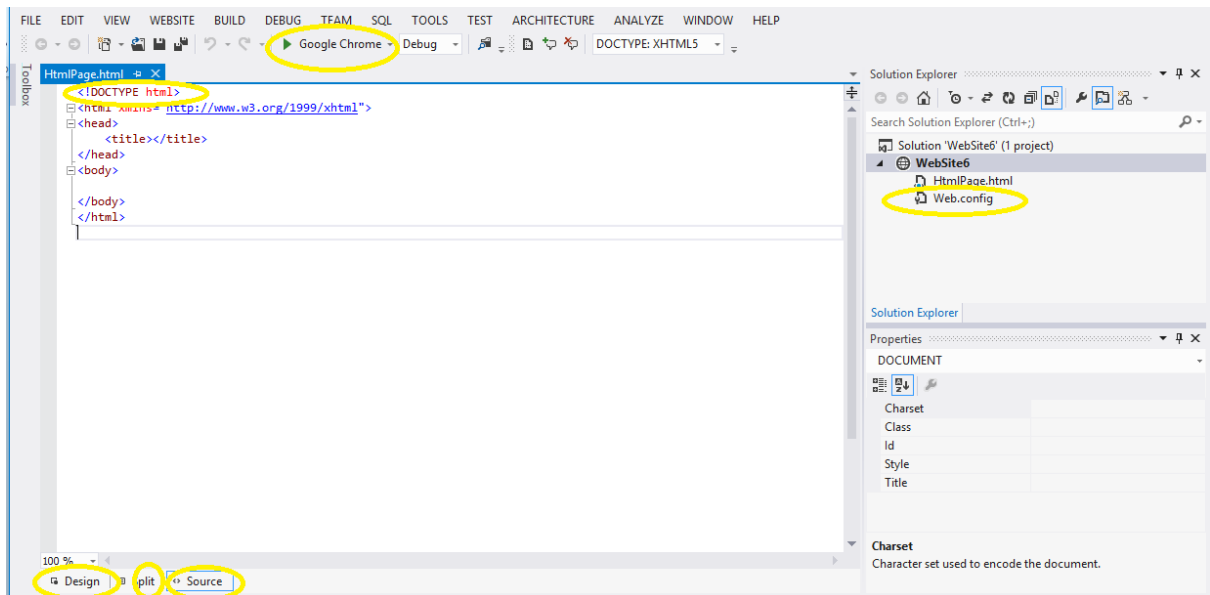
برروی فلدر وب سایت راست کلیک می نمایم :



منوی زیر ظاهر خواهد شد



به گزینه های زیر توجه کنید



یادآوری دستوراتی از html :

- HTML مخفف **Hyper Text Markup Language** است.
- HTML زبان نشانه گذاری است.
- یک زبان نشانه گذاری مجموعه ای از تگ های نشانه گذاری است.
- تگ ها محتوای سند را توصیف می کنند.
- اسناد HTML شامل تگ های HTML و متن های ساده می باشد.
- یک سند HTML **صفحه وب** نیز نامیده می شود.

## تگ های HTML

تگ های نشانه گذاری HTML معمولاً تگ های HTML نامیده می شود .

- تگ های HTML ، کلمات کلیدی (نام تگ) هستند که توسط براکت های زاویه ، مانند `<html>` احاطه شده اند.
- تگ های HTML معمولاً به صورت جفت می آیند مانند `<b>` و `</b>`
- تگ اول شروع و تگ دوم تگ پایان نامیده می شوند.
- تگ پایانی مانند تگ شروع، نوشته شده می شود ، اما با یک اسلش قبل از نام تگ.
- تگ های شروع و پایان برچسب های افتتاح و اختتام نیز نامیده می شود.

نمونه :

`<tagname> محتوا </tagname>`



## عناصر HTML :

تگ های HTML و عناصر HTML اغلب برای توصیف چیزی مشابه استفاده می شوند. اما به بیان دقیق، یک عنصر HTML همه چیزی است که بین تگ شروع و تگ پایان قرار دارد، از جمله تگ ها :

<p>این یک پاراگراف است</p>

## مرورگرهای وب

هدف مرورگر وب (مانند گوگل کروم، اینترنت اکسپلورر، فایرفاکس، سافاری (خواندن اسناد HTML و نمایش آنها به عنوان صفحات وب است. مرورگر ها تگ های HTML را نمایش نمی دهد، اما با استفاده از تگ ها محتوای صفحه را تفسیر می کند:

## نسخه های HTML

از روزهای اولیه وب، نسخه های بسیاری از HTML وجود داشته است :

سال	نسخه
1991	HTML
1993	HTML +
1995	HTML 2.0
1997	HTML 3.2
1999	HTML 4.01
2000	XHTML 1.0
2012	HTML5
2013	XHTML5

## اعلامیه <!DOCTYPE>

اعلامیه <!DOCTYPE> به مرورگر کمک می کند تا یک صفحه وب را به درستی نمایش دهد. بسیاری از اسناد و مدارک مختلف بر روی وب وجود دارد، و یک مرورگر تنها زمانی می تواند صفحه HTML را 100٪ به درستی نمایش دهد که نوع و نسخه مورد استفاده آن HTML را بداند .

## اعلامیه های مشترک

## HTML 5

```
<!DOCTYPE html>
```

## HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

## XHTML 1.0

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

در سال 1992 در دانشگاه مینه سوتا سیستمی به نام web به وجود آمد که دارای دو ویژگی خاص بود.

### Graphic, Multi Media

Hyper Text (فقط کلیک کردن و در سایت حرکت کردن) صفحات این سیستم توسط برنامه ای به نام HTML ساخته شد. همچنین برای رد و بدل کردن اطلاعات از پروتکل http استفاده میشود:

HTTP(Hyper Text Transfer Protocol HTML Web page)

HTML یک text عادی و در حقیقت زبانی برای مارک کردن فایل‌های text به یکدیگر می باشد که آن را با TAG مشخص کرده و به صورت <tag name> می نویسند.

فرمت کلی :

یک فایل HTML از دو بخش Body و Head، تشکیل می شود. شکل ساده یک فایل HTML به صورت زیر است.

```
<HTML>  
<HEAD>  
  <TITLE This is the title> </TITLE>  
</HEAD>  
<BODY>  
  This is the the body  
</BODY>  
</HTML>
```

### تگ head:

در برجسب HEAD از برجسب به نامهای TITLE و BASE و META استفاده می شود.

```
<HEAD>  
<TITLE> </TITLE>  
< META > </META>
```

< BASE> </BASE>

</HEAD>

## :TITLE

برای تعیین لقب صفحه (چیزی که در قسمت Status Bar دیده می شود).

## :META

(1) برای تعیین نام و منبعی که برنامه توسط آن نوشته شده.

(2) بهنگام کردن صفحات web توسط این برچسب انجام می شود.

(3) انتقال به یک صفحه دیگر web در زمان معین.

مثال برای حالت اول: در این حالت برای وارد کردن آدرس web خودمان به موتورهای جستجو در web (مثل yahoo و google و ...) از META استفاده می کنیم:

```
<META name="keyword" content="Hedayat,students,zahiri,yaghoubi,schoolnet"/>
```

```
<META name="description" content="This is Hedayat high school"/>
```

مثال برای حالت دوم و سوم:

```
<META name="vali" http_equiv="refresh" content="زمان بر حسب ثانیه"/>
```

با این برچسب صفحه web بعد از 1 دقیقه بهنگام (refresh) خواهد شد. در مثال بالا اگر در قسمت content به صورت زیر عمل کنیم صفحه web بعد از 60 ثانیه به [www.schoolnet.ir](http://www.schoolnet.ir) خواهد رفت:

```
Content="60; URL= http://www.schoolnet.ir"
```

## :BASE

برای مشخص کردن مبدا آدرس دهی از صفحات web می باشد.

```
<BASE href="آدرس"/>
```

```
<BASE href="http://www.schoolnet.ir/~zahiri/index.htm">
```

نکته مهم: در برنامه نویسی HTML برچسب ها به دو صورت با پایان و بی پایان نوشته می شوند.

جلسه دوم : یادآوری دستوراتی از **html** :

---

<TAG>.....</TAG> با پایان

<TAG> بی پایان

قسمت دوم یک فایل HTML را **Body** تشکیل می دهد که دارای **Attribute** های زیر می باشد.

```
<BODY bgcolor="رنگ پس زمینه صفحه">
```

```
background="آدرس عکسی که به عنوان پس زمینه در صفحه وب قرار می گیرد."
```

topmargin="یک فضای خالی بالای صفحه بر حسب پیکسل ایجاد ما کند"  
leftmargin="یک فضای خالی سمت چپ صفحه بر حسب پیکسل ایجاد ما کند"  
text="color" رنگ متن را مشخص می کند  
link="color"  
alink="color"  
vlink="color"/>

نکته مهم: در برنامه HTML و در نوشتن تگها بزرگ و یا کوچک نوشتن حروف هیچ تاثیری ندارد.

با این برچسب می توانیم مشخصات متن را به دلخواه خود درآوریم و فرمت کلی آن به صورت زیر است.

<FONT> .....</FONT>

این تگ دارای Attribute های زیر می باشد:

color: رنگ متن

size: اندازه متن

face: نوع متن

مثال: می خواهیم کلمه Schoolnet را با فونت نازنین و با اندازه normal و رنگ آبی بنویسیم.

```
<HTML>
<HEAD>
</HEAD>
<BODY>
  <FONT size = "3" color ="blue" face="NAZANIN" Schoolnet> </FONT>
</BODY>
</HTML>
```

نکته: اگر بخواهیم اندازه را نسبی مشخص کنیم یعنی نسبت به آنچه که قبلا بوده به صورت زیر عمل می کنیم:

Size = +2

**:BOLD**

اول و آخر متن مورد نظر قرار گرفته و آنرا Bold می کند.

<B> text ... </B>

**:ITALIC**

اول و آخر متن مورد نظر قرار گرفته و آنرا Italic می کند.

<I> ... text </I>

**:UNDERLINE**

اول و آخر متن مورد نظر قرار گرفته و آنرا Underline می کند.

<U> ... text </U>

در HTML بوسیله تگ `<a> ... </a>` می توانیم یک متن یا عکس را به صفحه ای دیگر پیوند دهیم ( Hyper link). مهمترین Attribute در این تگ، href می باشد. فرمت کلی این تگ به صورت زیر است.

```
<a href = "URL"> text/image </a>
```

مثال: در جمله `click here to go to page zahiri home` کلمه zahiri را به آدرس `html.index` لینک کنیم:

```
<BODY>
```

```
<p>
```

```
click here to go to
```

```
<a href = "http://www.schoolnet.ir"> schoolnet </a>
```

```
home page
```

```
</p>
```

```
</BODY>
```

**<P> ... </P>**: در این تگ همه موضوعات آن در یک خط نوشته شده یا یک پاراگراف ایجاد می شود که در صورت بوجود آمدن پاراگراف در زیر خط اول، خط دوم را با فاصله زیاد می نویسد. برای حل این مشکل از تگ **<BR>** استفاده می کنیم.

تگ **<P>** دارای یک Attribute است:

```
<p align="left/center/right">
```

**<BR>**: این تگ از تگهایی است که پایان ندارد و آن را هر کجا که قرار دهیم کلمه بعدی را در یک خط پائین تر ولی با کمتر می نویسد.

**<NOBR> ... </NOBR>**: اگر بخواهیم در آخر خط شکستگی نداشته باشیم بین دو بخشی که شکسته می

شود از این تگ استفاده می کنیم .

مثال: **<NOBR>a2</NOBR>**

a, 2 را هرگز از هم جدا نمی کند.

در HTML دارای شش نوع HEADING هستیم.

```
<H1> ... </H1>
```

```
<H2> ... </H2>
```

```
.  
.
```

```
<H6> ... </H6>
```

خود این تگها خاصیت راست چین و چپ چین و یا وسط چین شدن را هم دارند که برای فعال کردن آن از روش زیر استفاده می کنیم.

مثال: `<H2 align="center"> vali </H2>`

تگ `<HR>` برای ما تک خط افقی سه بعدی ایجاد می کند و دارای Attribute های زیر می باشد.

`<HR align="left/center/right"`

`width="طول خط بر حسب پیکسل یا درصد"`

`size="ضخامت خط بر حسب پیکسل"`

`noshade:` . با نوشتن این کلمه خط سه بعدی نمی شود.

`Color="رنگ خط">`

در HTML هر چیزی را که بین تگ `<PRE> ... </PRE>` به هر صورتی که بنویسیم با همان شکل در صفحه وب نشان می دهد.

مثال:

```
<PRE> Vali
```

```
Ali ...
```

```
Reza a b cd
```

```
</PRE>
```

برای ما یک بلوک در متن ایجاد می کند.

مثال:

`<DIV style="color: red"> ... </DIV>`

استفاده دیگر برای تعیین **Direction** می باشد که سمت نوشتن را از راست به چپ یا چپ به راست می کند.

`<DIV dir="rtl/ltr"> ... </DIV>`

**rtl= right to left**

**ltr= left to right**

`<big> ... </big>`: متن که بین آن باشد یک فونت درشتتر می نویسد.

`<small> ... </small>`: متن که بین آن باشد یک فونت کوچکتر می نویسد.

برای درست کردن توان و اندیس دو تگ داریم.

مثال: برای نوشتن و از دو تگ زیر استفاده می کنیم.

`a<SUB> 1 </SUB>`

`a<SUP> 20 </SUP>`

<IMG>: برای قرار دادن یک تصویر در web از این تگ استفاده می کنیم.

به چند دلیل نباید از تگ image زیاد استفاده کرد.

1- به علت زیاد شدن تعداد عکسها صفحه دیر load می شود.

2- ایجاد مشکل در Search Engine.

3- عدم قابل استفاده بودن برای همه(عدم سرعت کافی، هزینه دار بودن برای user و ...)

طرز قرار دادن عکس:

طرز قرار دادن عکس:

```
<IMG src="آدرس فایل تصویر را در این قسمت می نویسیم"  
align="left/middle/right/top/bottom"  
height="ارتفاع عکس بر حسب پیکسل"  
width="طول عکس بر حسب پیکسل"  
alt=" "  
title=""
```

هر چیزی که در این قسمت بنویسیم قبل از load شدن تصویر و یا بعد از کامل شدن صفحه اگر موس را روی عکس ببریم این متن دیده خواهد شد.(در یک مستطیل زرد رنگ)

```
Border="ضخامت جدول دور عکس را بر حسب پیکسل نشان می دهد"  
hspace="فاصله عمودی دور تصویر بر حسب پیکسل"  
vspace="فاصله افقی دور تصویر بر حسب پیکسل">
```

مثال:

```

```

سوال: برای لینک کردن یک عکس چه کاری را باید انجام دهیم؟

جواب:

```
<A href="URL" >  
  <IMG src="URL"/>  
</a>
```

در بسیاری از مواقع در یک صفحه وب مجبوریم قسمتی از یک عکس را به صفحه ای لینک کنیم. برای این کار در html از Image Map استفاده می کنیم.  
مثال: می خواهیم در t1.gif در منطقه یکی دایره و دیگری چهارگوش را به صفحه ای دیگر لینک کنیم:

```
<MAP name="Zahiri" >  
<AREA shape="circle" coord="50,50,30" href="1.htm">  
<AREA shape="rect" cords="100,70,130,100" href="2.htm">  
</MAP>  
...  
<IMG src="t1.gif" usemap="#Zahiri">
```

## عنوان HTML

عنوان ها یا سر برگ های HTML با تگ های <h1> تا <h6> تعریف می شود.

```
<h1>This is a heading</h1>
```

```
<h2>This is a heading</h2>
```

```
<h3>This is a heading</h3>
```

## پاراگراف HTML

پاراگراف های HTML با تگ <p> تعریف می شوند.

```
<p>This is a paragraph</p>
```

## لینک های HTML

لینک HTML با تگ <a> تعریف می شود.

```
<a href="http://www.w3.webmehraz.ir">This is a Link</a>
```

## عکس های HTML

عکس های HTML با تگ img تعریف می شود.



```

```

## جدول HTML

جدول با تگ `<table>` تعریف شده است .

جدول به ردیف ها (با تگ `<tr>` تقسیم می شود، و هر سطر به سلول های داده (با برچسب `<td>` تقسیم می شود `td` مخفف "table data" و دارای محتوای یک سلول داده می باشد. تگ `<td>` می تواند شامل متن، لینک ها، عکس ها، لیست ها، فرم ها، جداول و غیره باشد.

```
<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

## جدول HTML و عنصر حاشیه

اگر شما عنصر حاشیه را مشخص نکنید، جدول بدون حاشیه نمایش داده خواهد شد. گاهی اوقات این می تواند مفید باشد، اما بیشتر اوقات ، ما می خواهیم حاشیه ها را نشان دهیم . برای نمایش جدول با حاشیه ، صفت `border` را مشخص کنید:

```
<table border="1">
<tr>
<td>Row 1, cell 1</td>
<td>Row 1, cell 2</td>
</tr>
</table>
```

## عناوین جدول HTML

اطلاعات عنوان در یک جدول با برچسب `<th>` تعریف می شود . تمام مرورگرهای بزرگ ، متن را در عنصر `<th>` به صورت متمرکز و ضخیم نشان می دهند.

```

<table border="1">
<tr>
<th>Header 1</th>
<th>Header 2</th>
</tr>
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>

```

مثال جدول:

```

<table>
  <tr>
    <td width="770px"height="230px"colspan="2"></td>
  </tr>
</table>

```

طرح بندی - HTML استفاده از عنصر <div>

عنصر div یک سطح بلوک برای گروه بندی عناصر HTML است. در مثال زیر از پنج عنصر div به منظور ایجاد یک طرح بندی چند ستونی استفاده شده است، که همان نتیجه در مثال قبل را ایجاد می کند:

```

<!DOCTYPE html>
<html>
<body>
<div id="container" style="width:400px ">
<div id="header" style="background-color:#FFA500 ">;
</ body>
< /html>

```

شبه کلاس lang:

شبه کلاس lang: به شما اجازه می دهد تا قوانین ویژه ای را برای زبان های مختلف تعریف کنید. توجه IE8: از شبه کلاس lang: پشتیبانی می کند. به شرطی که <!DOCTYPE> مشخص شده باشد. در نمونه زیر lang: علامت کوتیشن را برای عناصر q با lang="no" تعریف می کند:

```
<html lang="fa-IR"></html>
```

ویژگی ها (attributes):

```
<p dir="rtl"></p>
```

```
<p dir="ltr"></p>
```

```
<font></font>
```

```
<font size="4"face="tahoma"color="green"></font>
```

- . ساختن منو

```
<OL>
```

```
<LH> آموزش و طراح </LH>
```

```
<LI> V </LI>
```

```
<LI> A </LI>
```

```
<LI> L </LI>
```

```
<LI> I </LI>
```

```
</OL>
```

روش دوم

```
<UL>
```

```
<LH> ... </LH>
```

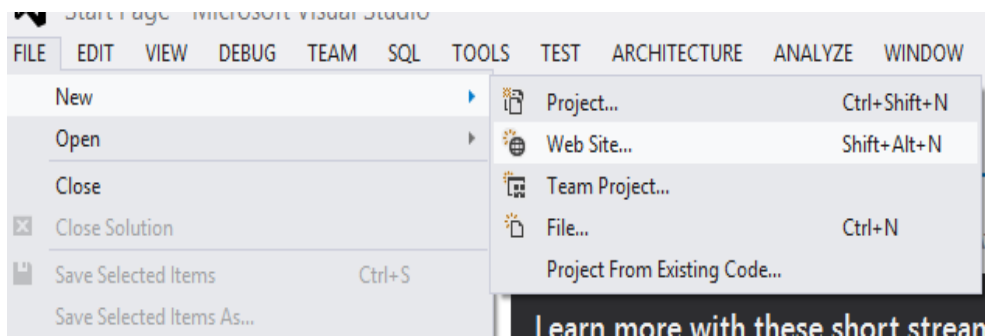
```
<LH> ... </LH>
```

```
<LH> ... </LH>
```

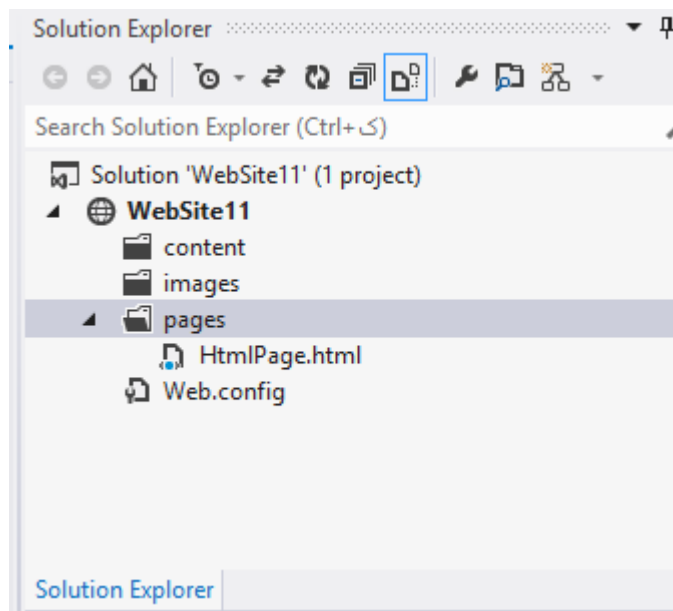
```
</UL>
```

جلسه سوم: طراحی لایه های صفحه با تکنولوژی div

1- ابتدا فایل جدیدی را باز می کنیم



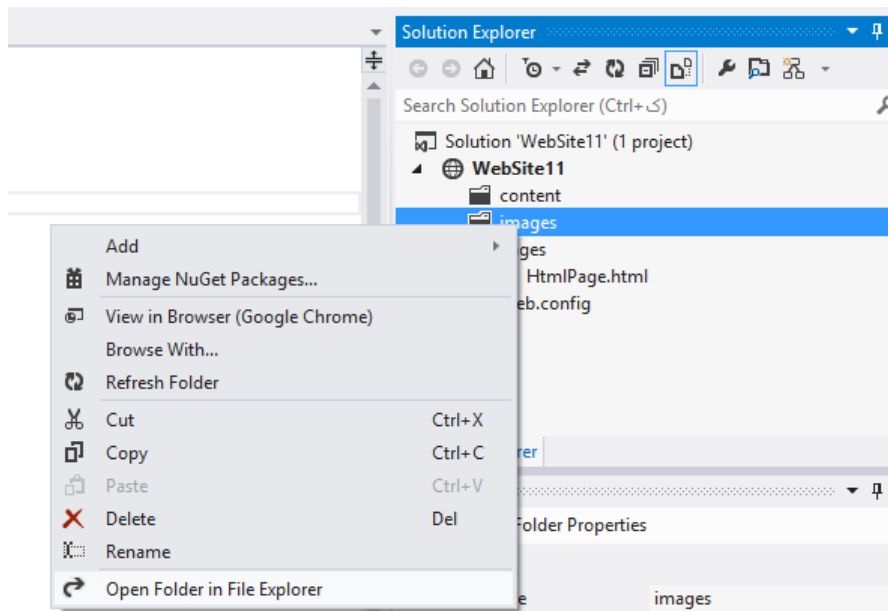
2- بروی پوشه کاری سه فلدر و یک صفحه html ایجاد می نمایم



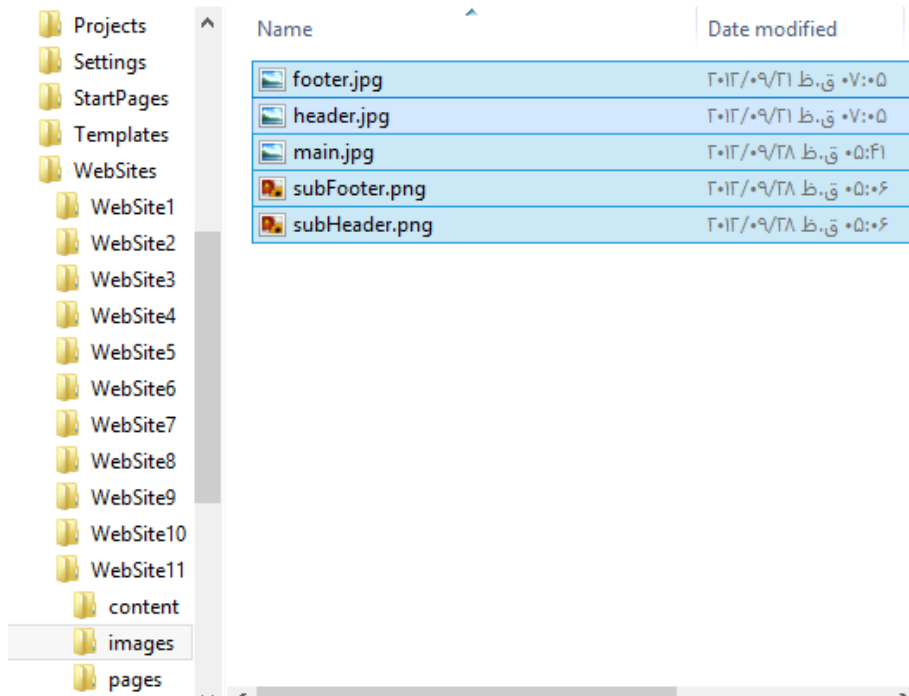
3- در صفحه html کد های زیر در بخش body اضافه می نمایم

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title></title>
  </head>
  <body>
    <div></div>
    <div></div>
    <div></div>
    <div></div>
    <div></div>
  </body>
</html>
```

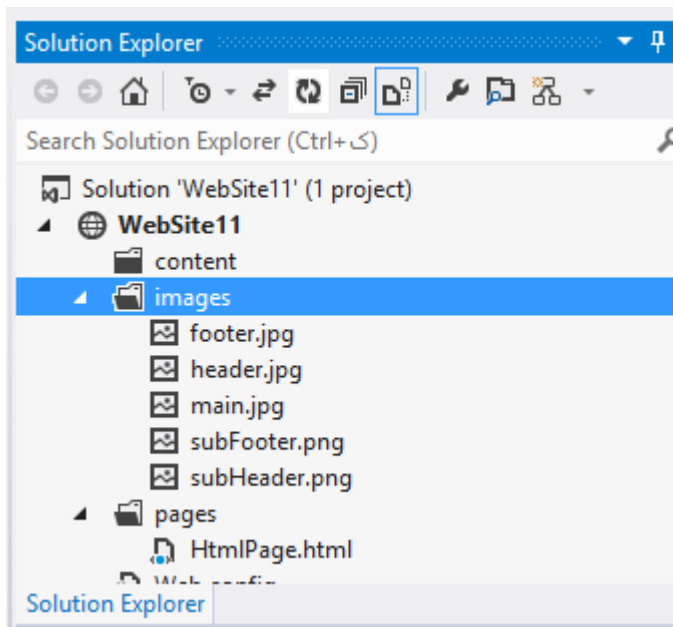
4- برای لود کردن عکس های مورد نیاز در فلدر images به روش زیر عمل می نمایم :



فایل های مورد نیاز در پوشه مربوطه کپی می نماییم :



5- بعد از کپی کردن فایلها در پوشه مربوطه پنجره را بسته و فلدر images را رفرش می کنیم .



6- در صفحه html کد های زیر را اضافه می کنیم

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
</head>
<body>
  <div style="width: 770px; height: 230px; background-image:
url(../Images/header.jpg);"></div>
  <div style="width: 770px; height: 52px; background-image:
url(../Images/subHeader.png)"></div>
  <div style="width: 770px; height: 230px; background-image:
url(../Images/main.jpg)"></div>
  <div style="width: 600px; height: 230px; float: left"></div>
  <div style="width: 770px; height: 52px; background-image:
url(../Images/subFooter.png)"></div>
  <div style="width: 770px; height: 102px; background-image:
url(../Images/footer.jpg)"></div>
</body>
</html>
```

توجه : در این برنامه ویژگی div را با عبارت style مشخص کرده ایم که از CSS داخلی یا internal استفاده شده است در محیط CSS همانند محیط C# برای دادن مقدار به یک پارامتر از : و برای جدا سازی مقادیر از ; استفاده می نمایم

در DIV های تو در تو برای مشخص شدن ارنج و ترتیب و جای آنها از ویژگی float با دو مقدار left و right استفاده شده است .

CSS - مخفف کلمه **Cascading Style Sheets** میباشد.

-اولین ورژن استایلها در سال ۱۹۹۶ ساخته شد. که اولین ورژن آن CSS1 بود.

-دیگر وزن آن CSS2 می باشد که در سال ۱۹۹۸ ساخته شد که بیشتر برای ویرایش صفحات، نحوه نمایشها تگها و ... کاربرد دارد.

-فایل استایلها با فرمت CSS ذخیره میشوند.

-جدا کردن ظاهر صفحات اعم از طرح بندی و رنگ بندی و ... از محتوای صفحات مانند کد های HTML و ...

### انواع استایلها:

همانطوری که قبلا نوشتیم استایلها بر سه نوع میباشد:

#### ۱) (internal). Inline Style:

این نوع استایل در داخل تگهای HTML به کار گرفته میشوند، و این استایل فقط بروی همان تگ به خصوص تاثیر خواهد گذاشت و به صورت زیر نوشته خواهد شد.

```
<p style="font-family: Tahoma;">This is a paragraph </p>
```

در مثال بالا با استفاده از حالتیهای مختلف در استایلها نوع فونت آنرا تغییر دادیم و همانطوری که در تعریف این نوع استایل نوشتیم این استایل فقط و فقط تغییرات را بروی همین پاراگراف اعمال خواهد کرد.

#### ۲) (external). Embedded or Global Style:

در این نوع، استایل نوشته شده در تمامی صفحه تاثیر خواهد گذاشت، اینرا بین دو تگ (<head></head>) به صورت زیر نوشته خواهد شد.

```
<style type="text/css">
```

```
pp { color: red; text-align: left; font-size: 8pt }
```

```
</style>
```

این نوع حالت از تغییرات تعریف شده بروی تمامی تگها پاراگراف اعمال خواهد شد که رنگ متن: قرمز، جهت متن: چپ، و اندازه فونت ۸ پوینت خواهد بود. توجه داشته باشید در این مثال سه حالت مختلف یک استایل را برای تگ پاراگراف تعریف کردیم که با استفاده از (;) حالتیهای مختلف آن از هم جدا شدند.

اگر بخواهید این حالت نوشته شده را برای چند تگ دیگر اعمال کنید می توانید تگها را با استفاده از کاما (,) از هم جدا کنید برای مثال:

```
h1, h2, h3, h4 { color: gold; font-family: sans-serif }
```

### ۳. Linked or External style sheet

نوع آخر هم معروف به استایل‌های خارجی هستند که به صورت لینک فایل استایل را به صفحات خود لینک می‌دهید. فایل‌های استایل با فرمت CSS ذخیره میشوند و به صورت زیر به صفحات لینک داده میشود.

```
<head><link rel="stylesheet" type="text/css" href="mystyle.css"> </head>
```

در این نوع از استایلها بعد از اینکه فایل استایل خود را آماده کردید آنرا با فرمت CSS ذخیره کرده و آنرا همانند بالا فایل استایل را به قالب لینک می‌کنیم. برای راحت بود در کار می‌توانید ابتدا استایلها را به صورت Internal استفاده کرده و بعد از اینکه کارتان تمام شد کدهای نوشته شده بین دو تگ (<style></style>) را در نت پد کپی کرده و آنرا با فرمت CSS ذخیره کرده و سپس با استفاده از فرمول بالا آن فایل را به قالب لینک دهید. یکی از مزیت‌های این نوع استایل این میباشد که می‌توانید با داشتن یک فایل استایل برای چندین صفحه استفاده کنید و از دیگر مزیت‌های آن با این کار کدهای قالب را به چند فایل تقسیم کرده و قالب را منظم تر میشود. در مقالات بعدی کاربرد های بیشتری از سی اس اس را براتون شرح می‌دهم.

کار عملی :

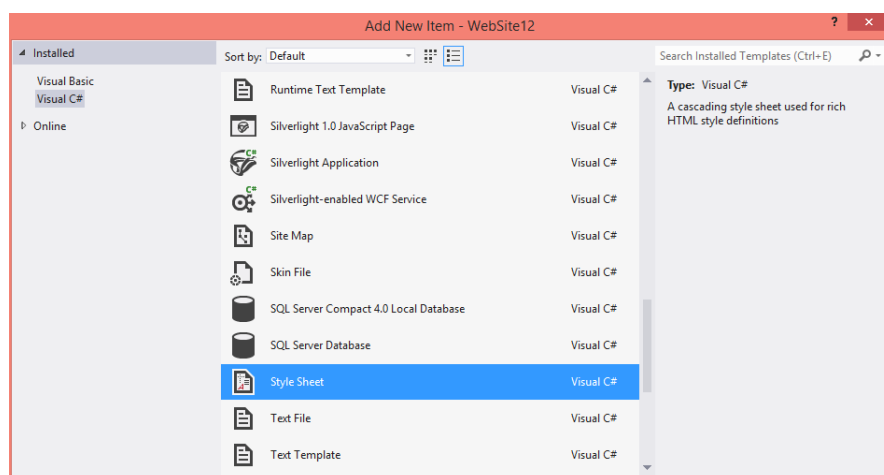
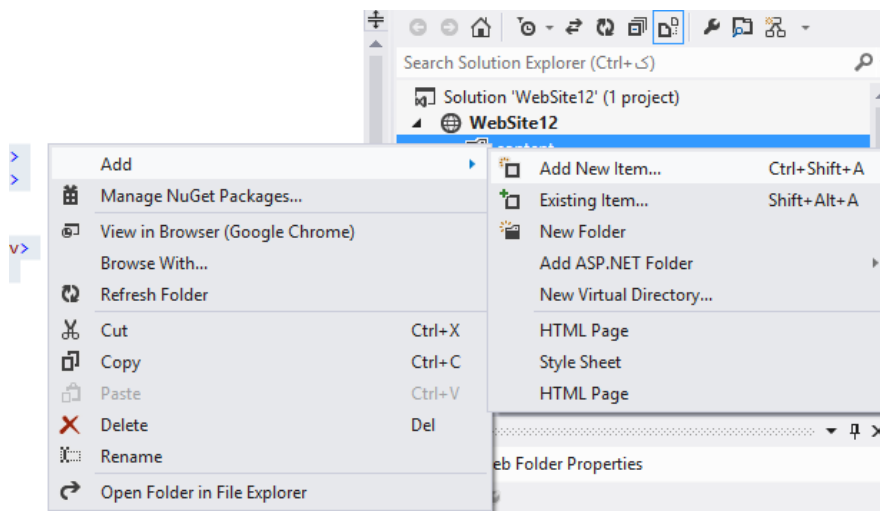
1- پروژه قبلی را مجدداً باز می‌نمایم

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
</head>
<body>
  <div style="width: 770px; height: 230px; background-image:
url(..\Images/header.jpg);"></div>
  <div style="width: 770px; height: 52px; background-image:
url(..\Images/subHeader.png)"></div>
  <div style="width: 770px; height: 230px; background-image:
url(..\Images/main.jpg)"></div>
  <div style="width: 600px; height: 230px; float: left"></div>
  <div style="width: 770px; height: 52px; background-image:
url(..\Images/subFooter.png)"></div>
  <div style="width: 770px; height: 102px; background-image:
url(..\Images/footer.jpg)"></div>
</body>
```



</html>

همانطور که ملاحظه می کنید از CSS داخلی استفاده شده است . برای راحتی کار و تغییرات در صفحات طولانی و متعدد از CSS خارجی استفاده می کنیم که برای اینکار مراحل زیر انجام می دهیم . در داخل فلدر content دو فایل CSS با نام های site1 و site2 باز می کنیم .



2- در فایل site1 از روش (دات) و فایل site2 از روش # (شارپ) استفاده می نمایم و style های داخل فایل اصلی به داخل فایل های مربوطه می آورم .

محتوای site1 :

```
.header
{
    width: 770px; height: 230px; background-image:
url(..\Images/header.jpg);
}
```

```
.subHeader
{
    width: 770px; height: 52px; background-image:
url(..\Images/subHeader.png);
}
```

```

}

.main
{
    width: 770px; height: 230px; background-image:
url(..../Images/main.jpg);
}

.content
{
    width: 600px; height: 230px; float: left;
}

.content_left
{
    width: 40px; height: 40px; float: left;
}

.content_right
{
    width: 121px; height: 70px; float: right; direction: rtl;
}

.menu
{
    width: 170px; height: 230px; float: right;
}

.subFooter
{
    width:770px; height:52px; background-
image:url(..../Images/subFooter.png);
}

.footer
{
    width:770px; height:102px;background-
image:url(..../Images/footer.jpg);
}

#header
}

```

محتوای فایل **site2** :

```

        width: 770px; height: 230px; background-image:
url(..../Images/header.jpg;(
{

#subHeader
}
        width: 770px; height: 52px; background-image:
url(..../Images/subHeader.png;(
{

#main
}
        width: 770px; height: 230px; background-image:
url(..../Images/main.jpg;(
{

#content
}
        width: 600px; height: 230px; float: left;
{

#content_left
}
        width: 40px; height: 40px; float: left;
{

#content_right
}
        width: 121px; height: 70px; float: right; direction: rtl;
{

#menu
}
        width: 170px; height: 230px; float: right;

```

```

{

#subFooter
}

width:770px; height:52px; background-
image:url(../Images/subFooter.png);(
{

#footer
}

width:770px; height:102px;background-
image:url(../Images/footer.jpg);(
}

```

توجه:

برای استفاده از فایل CSS در برنامه دو مرحله زیر انجام می دهیم :

- 1- ابتدا با درگ کردن فایل CSS به انتهای تگ **head** بعد از **title** قرار می دهیم
- 2- در داخل هر دیو مشخص می کنیم که کدام بخش از CSS بخواند . برای اینکار اگر از نقطه استفاده کرده باشیم در داخل فایل **html** از عبارت **class** استفاده می کنیم و نام استایل شیت را بنویسیم

```
<div class="main"></div>
```

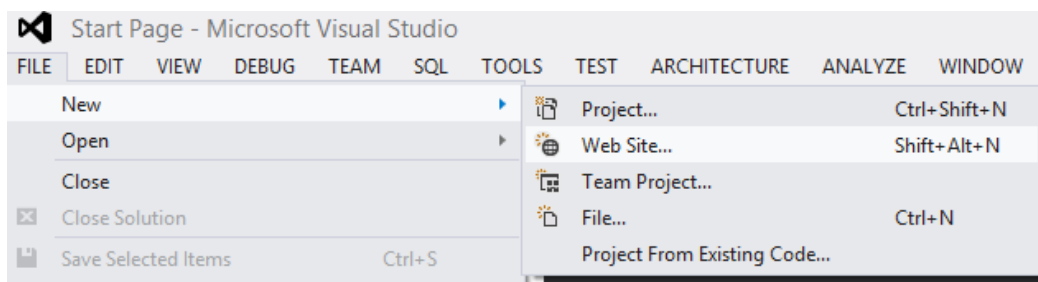
نام را بصورت **intelligence** شده می آورد و این از ویژگی **class** است .

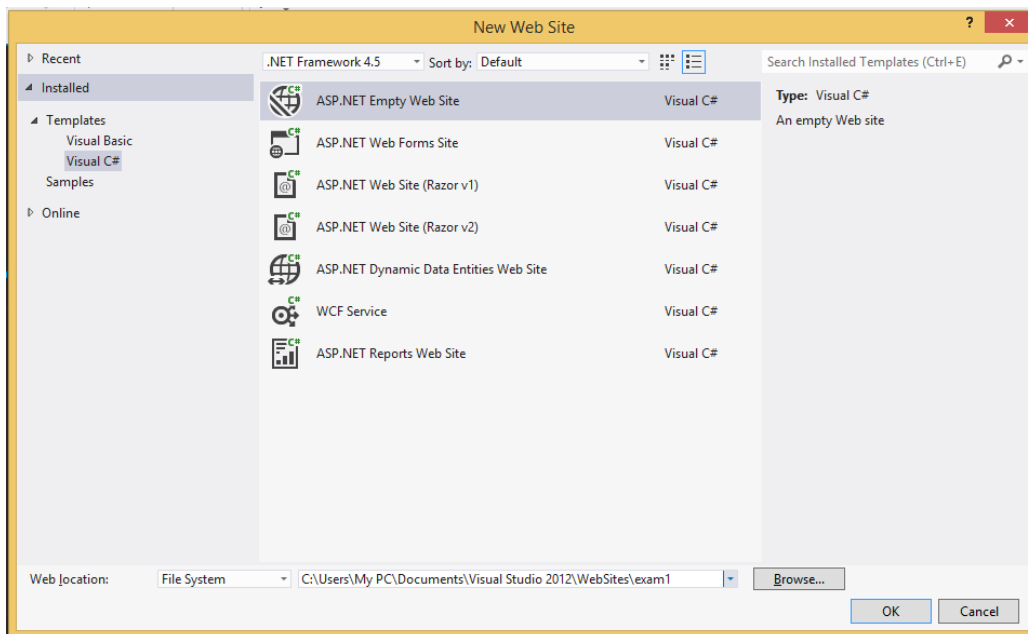
اگر در فایل CSS از **#** استفاده کنیم باید در فایل **html** از عبارت **id** استفاده نماییم

```
<div id="subFooter"></div>
```

## جلسه پنجم: روش form application

برای اجرای برنامه به روش form application مراحل زیر عمل می کنیم :

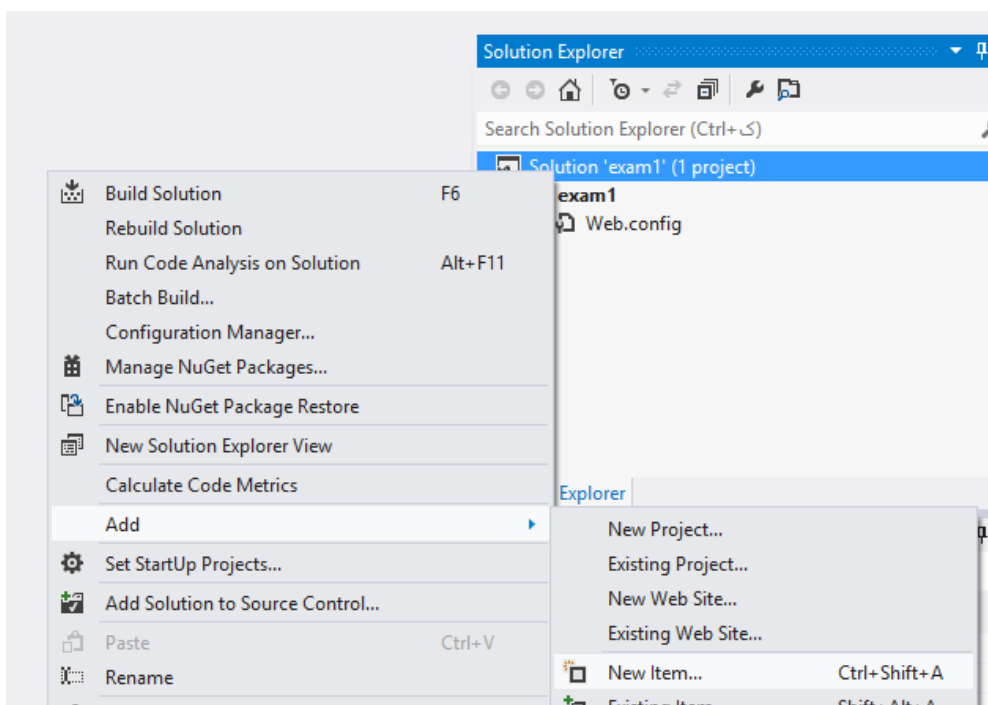




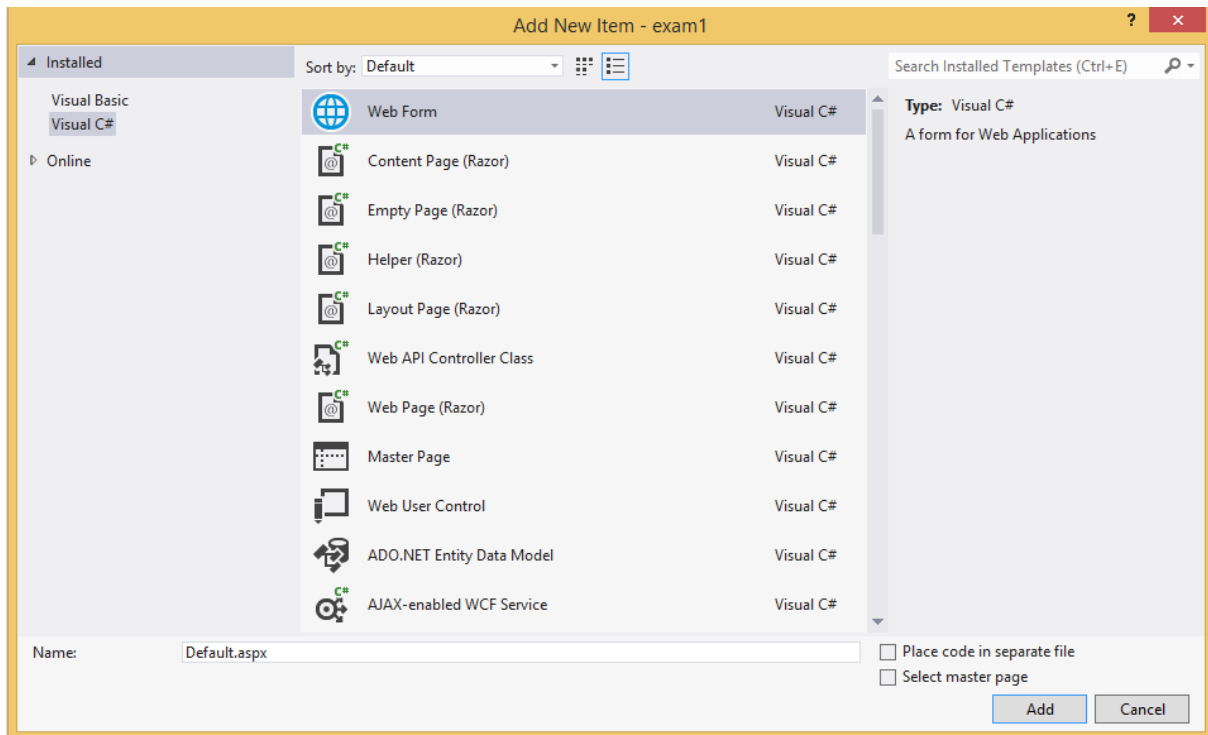
و نام exam1 ایجاد می کنیم .

برای کار کردن با asp.net نیاز به صفحاتی داریم که فرمت asp.net دارند. برای ایجاد یک فایل به روش زیر عمل می کنیم .

بر روی پوشه exam1 راست کلیک می کنیم ، سپس گزینه add و add new item را کلیک می کنیم



سپس به روش زیر عمل می کنیم



در صفحه مربوطه کدهای زیر ظاهر می گردند

```
<%@ Page Language="C#" %>

<!DOCTYPE html>

<script runat="server">

</script>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

        </div>
    </form>
</body>
</html>
```

در پروژه ویندوز اپلیکیشنی وقتی زبانی انتخاب می کردیم در کل پروژه همان زبان استفاده می شد ولی در پروژه های ولی در وب سایت ها به صورت parpage کار می شود یعنی هر صفحه می تواند از زبان net, را پشتیبانی کند. در این فایل کدهای c# را در داخل تگ script می نویسیم

تگ `runat="server"` مشخص می کند که برنامه ما سرور ساید است یعنی کنترلرهای ما سمت سرور اجرا خواهند شد و نتیجه به سمت `client` برمی گردانند. اگر این تگ در برنامه نداشته باشیم یعنی `client side` هستند. فرق بین این پروژه با پروژه `html` در این است که کدهای `C#` باید در تگ `script` نوشته شود و عبارت `runat="server"` و کدهای برنامه باید در تگ فرم نوشته شود.

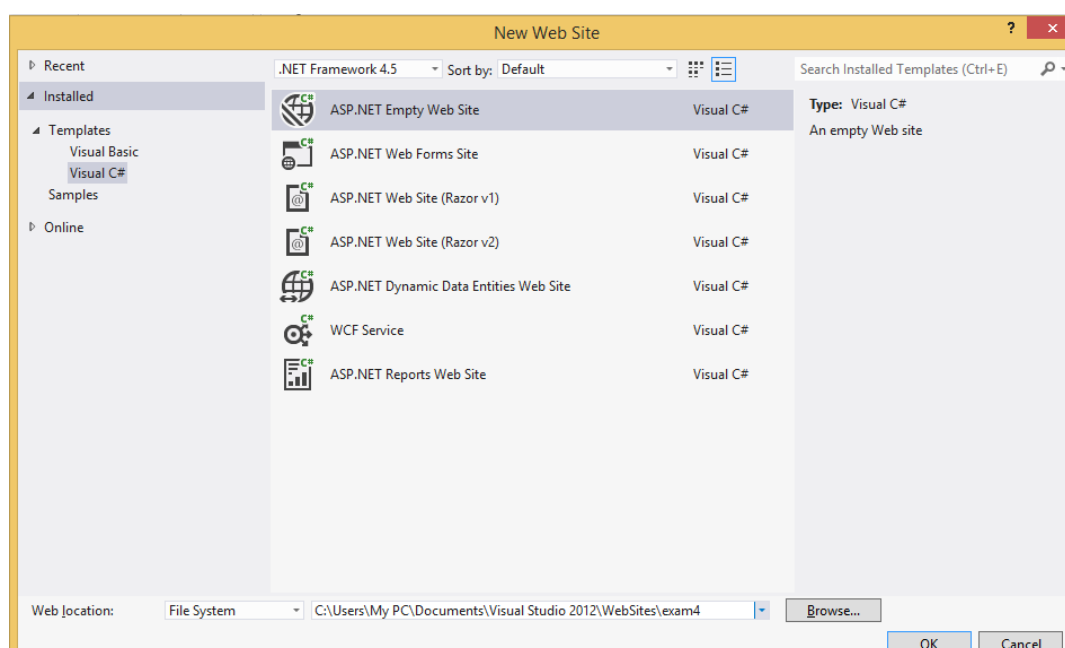
در نوشتن لیبل ها و آیدی ها و `CSS` ها از استاندارد کامل و بقیه موارد از استاندارد پاسکال استفاده می کنیم . استاندارد کامل `frm Main` استاندارد پاسکال مثل `Hello World` ( اولین کلمه بزرگ بقیه کوچک).

## جلسه ششم: مستریج چیست ؟

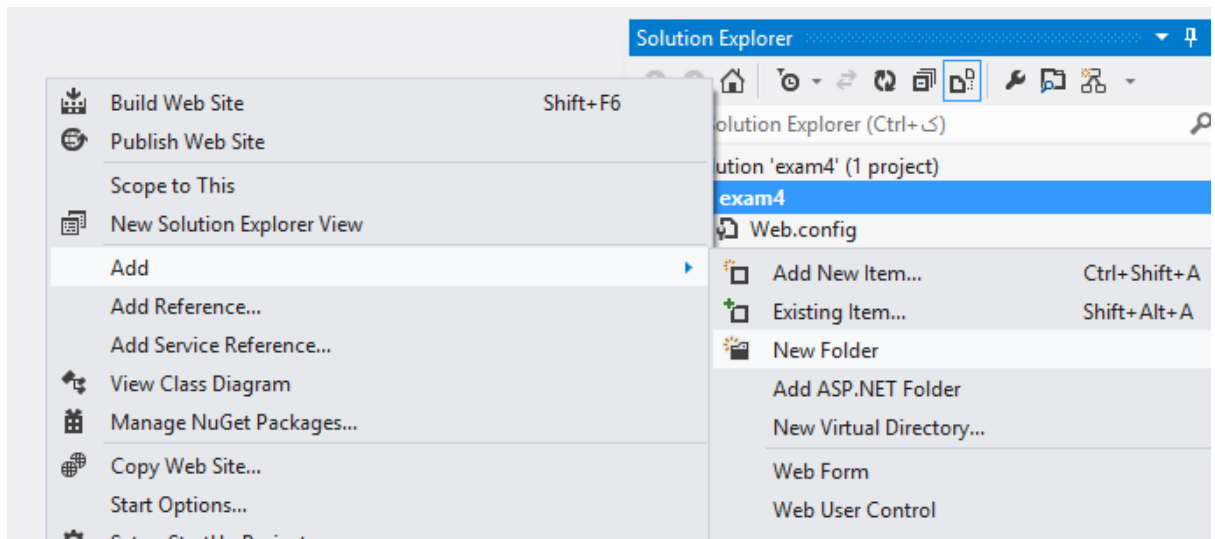
مستریج به طراح این امکان را می دهد تا یک صفحه یا قالب ( الگو ) پیش ساخته را برای ساخت صفحات دیگر ، طراحی نماید . در واقع مستریج ظاهر ، قالب و نحوه عملکرد کلی گروهی از صفحات را در پروژه یا وب سایت شما تعیین می کند . سپس می توانید صفحاتی را به عنوان صفحات محتوا ، به صورت تکی ایجاد نموده که این صفحات شامل محتوا متغیر سایت شما بوده و آن ها را به مستریج متصل می کنید .

هنگامی که کاربر صفحه محتوا ( `Content Page` ) را درخواست می کند ، سرور `ASP.NET` صفحه محتوا را با صفحه مستریج ترکیب کرده و تم های موجود را به آنها اعمال نموده و خروجی را در قالب یک فایل می سازد . برای درک بهتر مستریج ها ، یک مثال می زنیم .

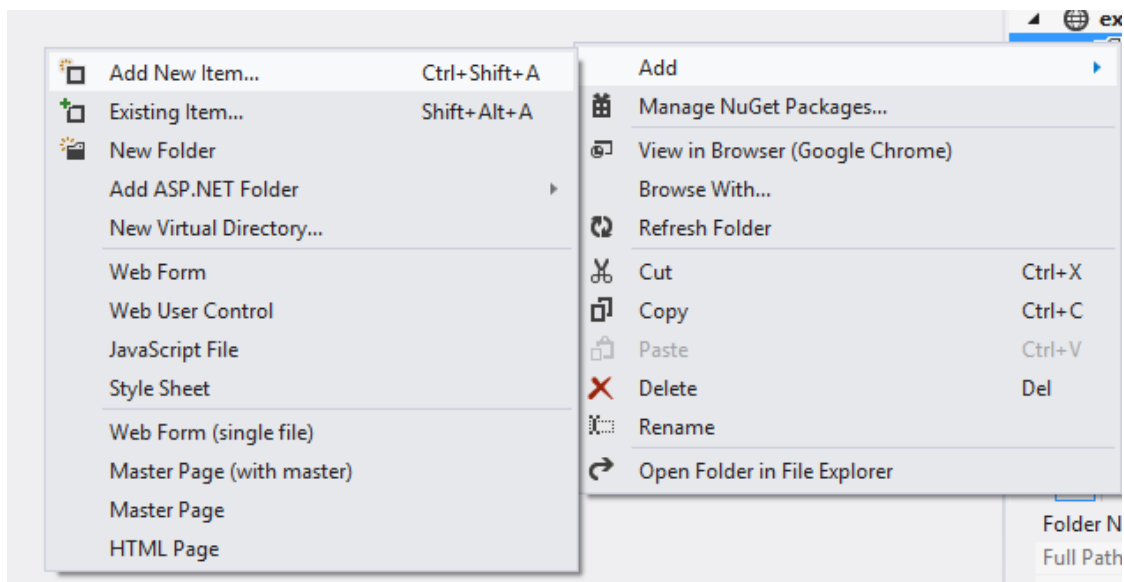
ابتدا مثل پروژه های قبلی عمل می کنیم :



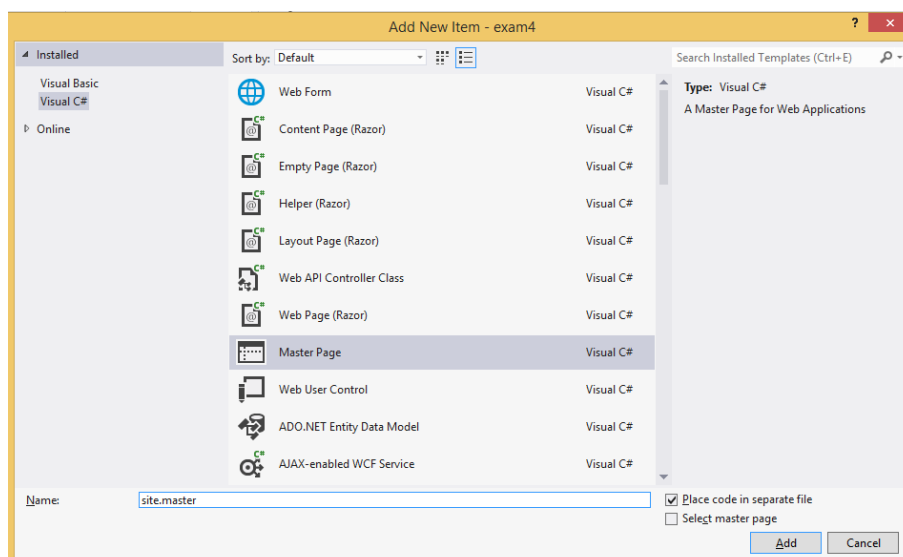
سپس پوشه ای به نام `master pages` ایجاد می کنیم .



روی فلدر master pages راست کلیک نموده و سپس add و add new item را کلیک می کنیم



سپس مراحل زیر را انجام می دهیم و نام site برای آن انتخاب می کنیم



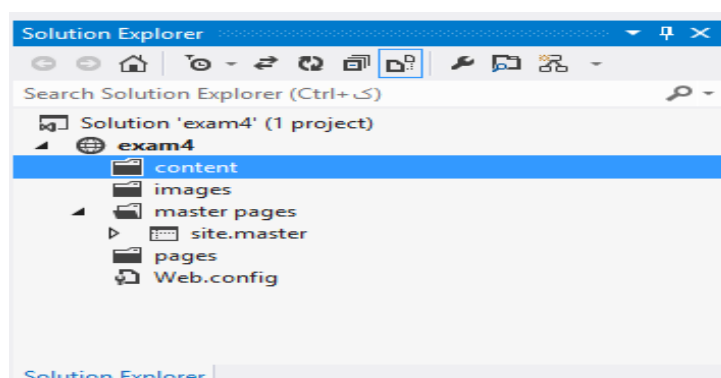


```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="site.master.cs" Inherits="master_pages_site" %>

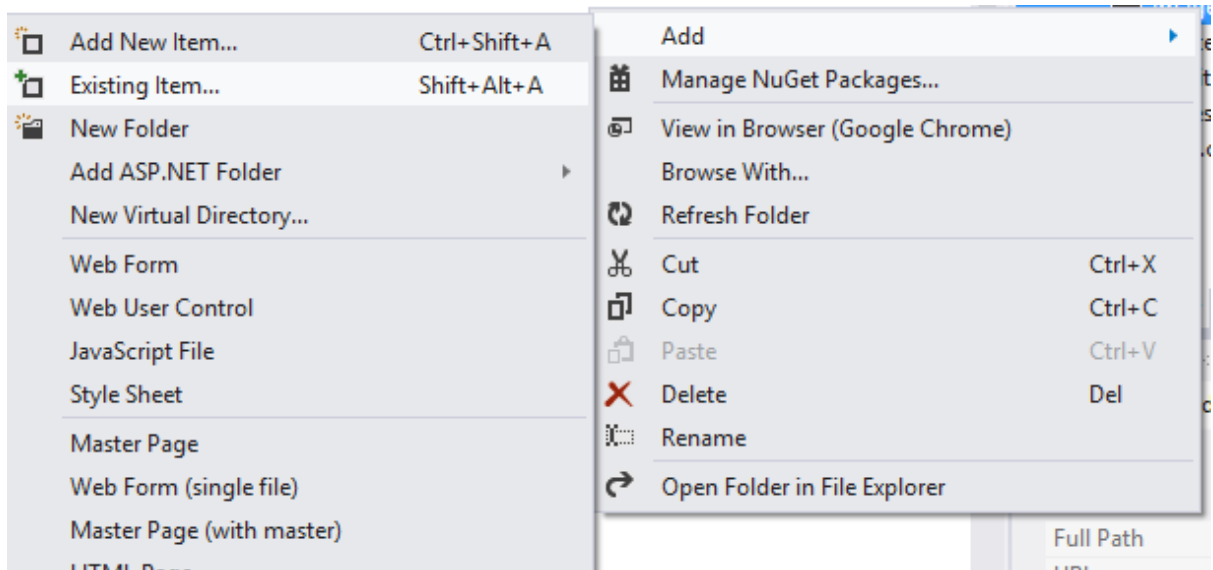
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <asp:ContentPlaceholder ID="head" runat="server">
    </asp:ContentPlaceholder>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <div></div>
            <div></div>
            <div>
                <asp:ContentPlaceholder
ID="ContentPlaceholder1" runat="server">
                </asp:ContentPlaceholder>
            </div>
            <div></div>
            <div></div>
        </div>
    </form>
</body>
</html>
```

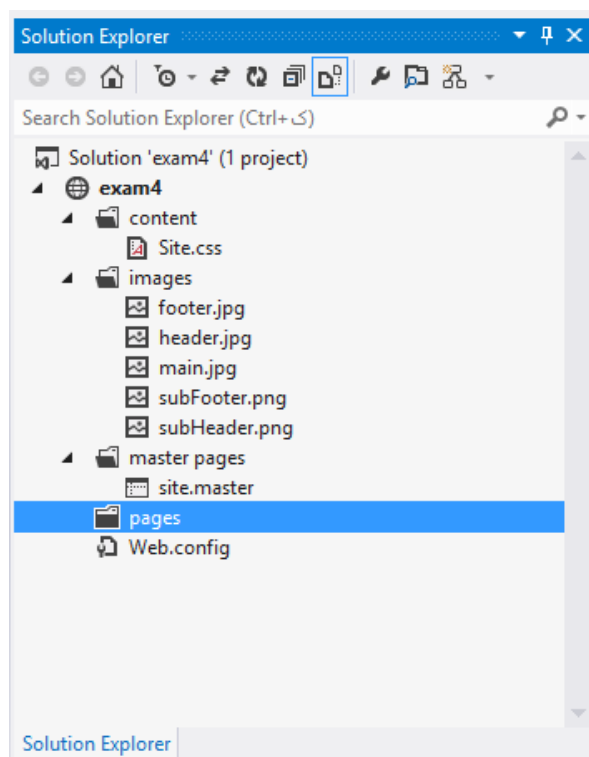
سپس فولدرهای pages و images و content به پروژه اضافه می کنیم .



در داخل فلدرهای محتوای پروژه های قبلی لود می کنیم



و سپس رفرش می کنیم



مثل تنظیمات CSS در فایل site.master تغییرات زیر لحاظ می کنیم

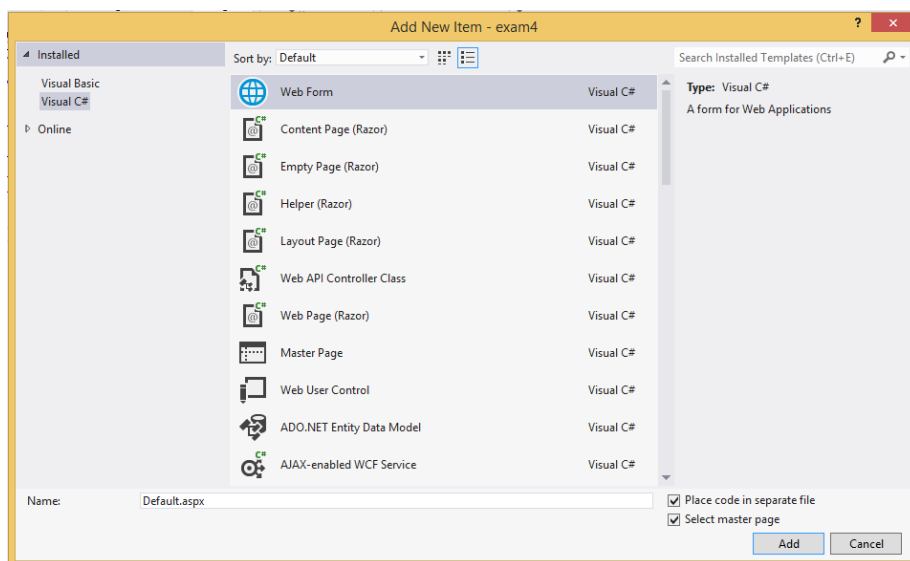
```
<%@ Master Language="C#" AutoEventWireup="true"  
CodeFile="site.master.cs" Inherits="master_pages_site" %>  
<!DOCTYPE html>  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">
```

```

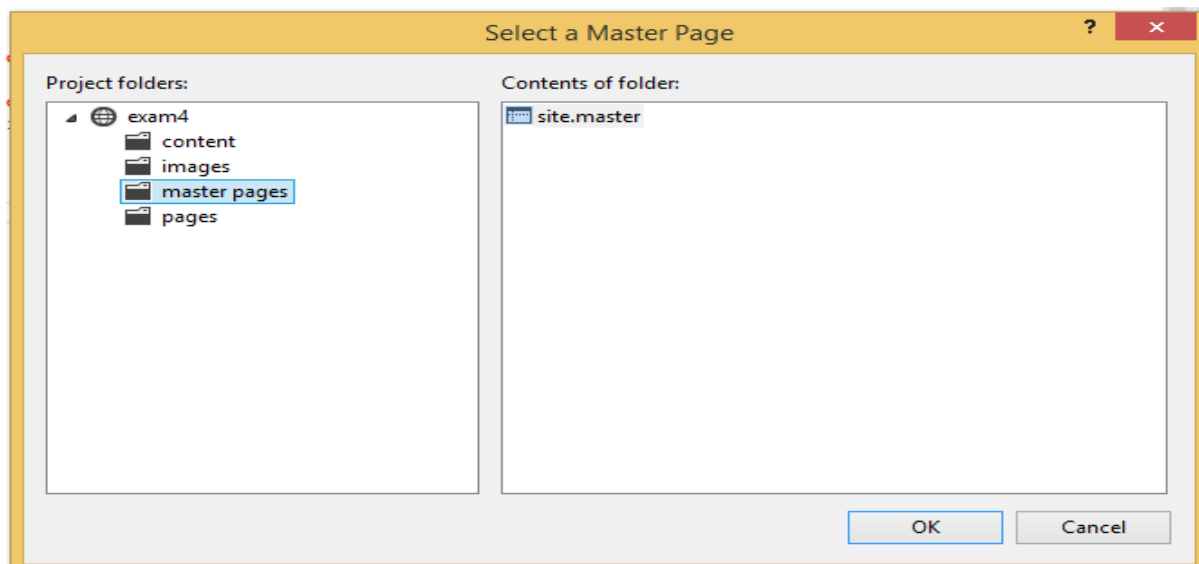
<title></title>
<link href="../../../content/Site.css" rel="stylesheet" />
<asp:ContentPlaceholder ID="head" runat="server">
</asp:ContentPlaceholder>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <div class="header"></div>
      <div class="subHeader"></div>
      <div class="main">
        <asp:ContentPlaceholder
ID="ContentPlaceholder1" runat="server">
        </asp:ContentPlaceholder>
      </div>
      <div class="footer"></div>
      <div class="subFooter"></div>
    </div>
  </form>
</body>
</html>

```

حال در design برنامه باید فرم ما کامل باشد. سپس روی فلدر pages راست کلیک می کنیم. گزینه add و add new item و صفحه زیر ظاهر می گردد. گزینه select master page را تیک می زنیم



سپس در صفحه زیر موارد مشخص شده انتخاب می کنیم و OK کلیک می کنیم.



تا کد default.aspx ظاهر گردد:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/master
pages/site.master" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="pages_Default" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="head"
Runat="Server">
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
</asp:Content>
```

استفاده از کنترل‌های html (کلاینت ساید):

```
<%@ Page Title="" Language="C#" MasterPageFile="~/master
pages/site.master" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="pages_Default" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="Server">
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="Server">
    First Name:
    <input type="text" />
    <br />
    Last Name:
```

```

<input type="text" />
<br />
Full Name:
<input type="text" />
<br />
<input type="submit" value="submit" />
<input type="button" value="cansel" />
</asp:Content>

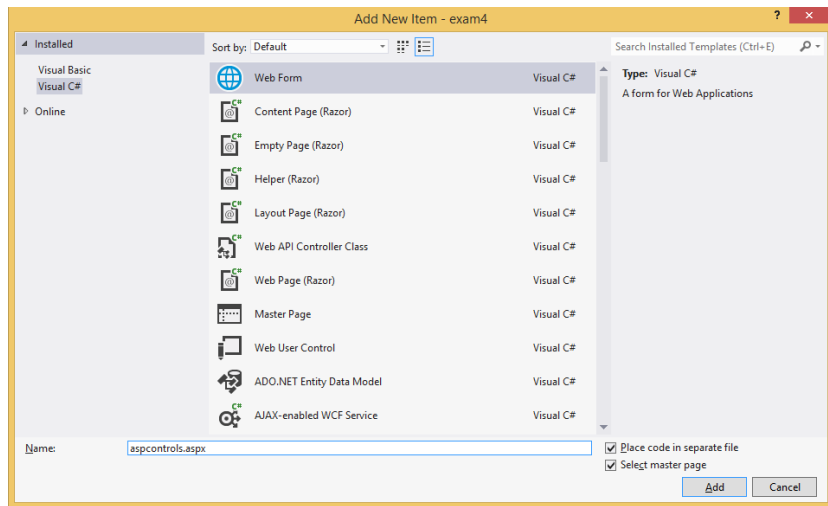
```



جلسه هفتم : code behine

این کنترل های مربوط به client side هست و معمولا به زبان html نوشته می شود که می توانیم از toolbar و گزینه های html با درگ کردن استفاده کنیم . اما چنانچه از کنترل های asp.net استفاده کنیم و به روش زیر عمل می کنیم :

ابتدا یک web form جدید باز می کنیم :



کد زیر ظاهر می گردد:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/master
pages/site.master" AutoEventWireup="true"
CodeFile="aspcontrols.aspx.cs" Inherits="pages_aspcontrols" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head"
Runat="Server">
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
</asp:Content>
```

برای اینکه از کنترلهای asp.net استفاده کنیم به شکل زیر عمل می کنیم :

```
<%@ Page Title="" Language="C#" MasterPageFile="~/master
pages/site.master" AutoEventWireup="true"
CodeFile="aspcontrols.aspx.cs" Inherits="pages_aspcontrols" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head"
Runat="Server">
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
    <div>
        <asp:Label ID = "lblfirstname" runat = "server" Text
="firstname:"/>
```

```

    <asp:TextBox ID="txtfirstname" runat="server" MaxLength
="20" />
    <br />
    <asp:Label ID="lbllastname" runat="server" Text
="lastname:" />
    <asp:TextBox ID="txtlastname" runat="server" MaxLength
="20" />
    <br />
    <asp:Label ID="lblfullname" runat="server"
text="fullname:" />
    <asp:TextBox ID="txtfullname" runat="server" MaxLength
="40" />
    <br />
    <input type="submit" value="Submit" runat="server" />
    <asp:Button ID="btncancel" runat="server" Text
="cancel" />
</div>

</asp:Content>

```

برای استفاده از تکنیک Legend :

```

asp:Content ID="Content1" ContentPlaceHolderID="cphHead" Runat="Server">
/asp:Content>
asp:Content ID="Content2" ContentPlaceHolderID="cphMain" Runat="Server">
  <fieldset>
    <legend>Legend</legend>
    <div class="field">
      <div class="lable">
        <asp:label id="lblFirtName" runat="server" text="FirtName" />
      </div>
      <div class="controls">
        <asp:textbox id="txtFirtName" runat="server" maxlength="20" />
      </div>
    </div>
  </fieldset>
/asp:Content>

```

```

    .controls
    {
        font-family:Georgia;
    }
    .field
    {
        padding:1px;
        margin:1px;
    }
    .button
    {
        font-family:Georgia;
    }
}

```

:

```

protected void btnSubmit_Click(object sender, EventArgs e)
{
    txtFirstName.Text =
        txtFirstName.Text.Trim();
    txtLastName.Text =
        txtLastName.Text.Trim();

    txtFullName.Text =
        txtFirstName.Text + txtLastName.Text;
}

```

```

protected void btnSubmit_Click(object sender, EventArgs e)
{
    txtFirstName.Text =
        txtFirstName.Text.Trim();
    txtLastName.Text =
        txtLastName.Text.Trim();

    txtFullName.Text =
        string.Format("{0} {1}",txtFirstName.Text,txtLastName.Text);
}
protected void btnClare_Click(object sender, EventArgs e)
{
    this.txtLastName.Text = string.Empty;
    this.txtFirstName.Text = string.Empty;
}

```

جلسه هشتم: تگ گروه بندی Fieldset , Legend



تگ Fieldset برای گروه بندی در فرم ها بسیار کاربرد دارد. این تگ در Asp.net معادل سروری کنترل Panel می باشد. برای تعیین عنوان یک گروه از تگ Legend در ابتدای تگ Fieldset استفاده می گردد.

**نکته ی بسیار مهم:** در صورت استفاده نمودن از کدنویسی سروری برای کنترل های Asp.net توصیه اکید می شود که از تگ های معادل آنها استفاده شود. بطور مثال بجای استفاده از کنترل Asp:Panel از تگ Fieldset و Legend استفاده شود. زیرا در نهایت کنترل های Asp.net به تگ های Html خالص تبدیل شده و برای جلوگیری از این سربار اضافه از تگ های مستقیم آنها استفاده نمایید.

مزیت دیگر این روش کنترل بهتر آنها در Style آنها بوسیله CSS می باشد که کنترل های Asp.net بعضا در این مورد مشکل زا هستند.

نمونه ای از کاربرد تگ Fieldset و Legend را در مثال زیر می بینید:

```
<fieldset>
  <legend>عنوان گروه</legend>

  <p>. این متن تست است</p>

  <p>. این متن تست شماره ۲ است</p>
</fieldset>
```

### خصوصیت اصلی

خصوصیت اصلی تگ Legend خصیصه ی Align یا تراز می باشد که محل قرار گیری عنوان را در این تگ تعیین میکند.

مقدار هایی که این خصوصیت می پذیرد عبارتند از Top , Bottom , Left , Right :

مثالی در این زمینه:

```
<fieldset>

  <legend align="left">عنوان گروه</legend>
```

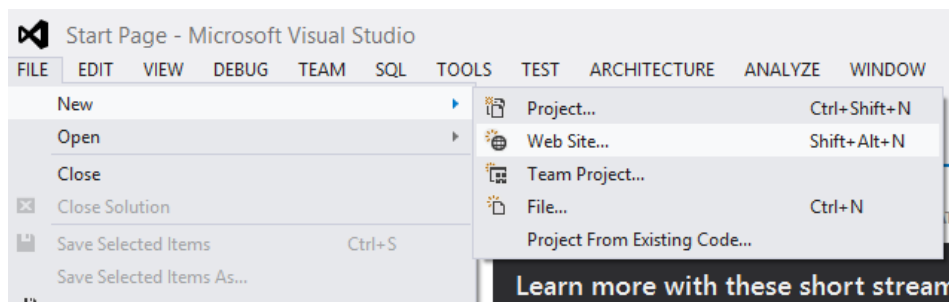
</p>.این متن تست است <p>

</p>.این متن تست شماره ۲ است <p>

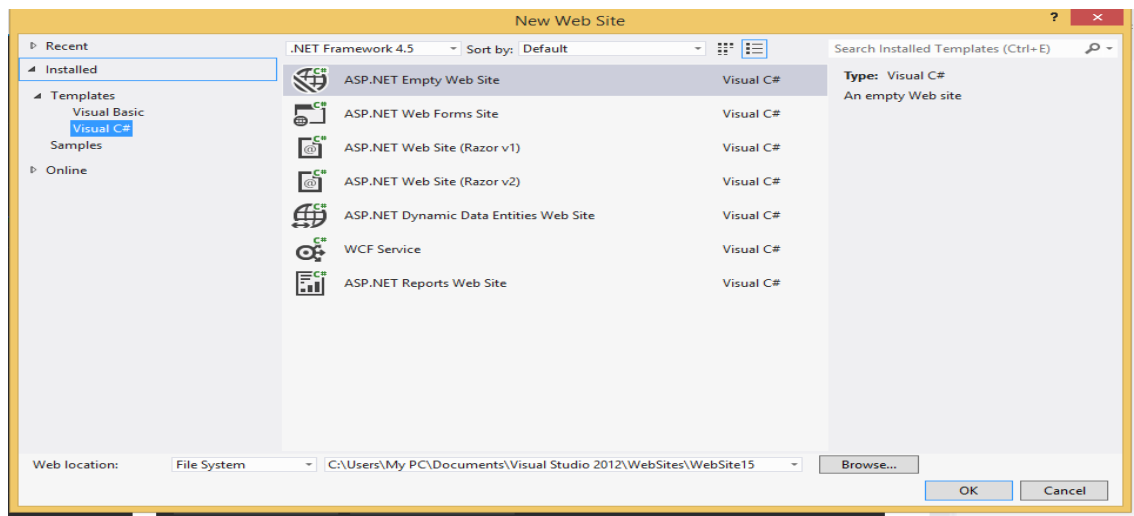
</fieldset>

آشنایی با مفهوم **user control**:

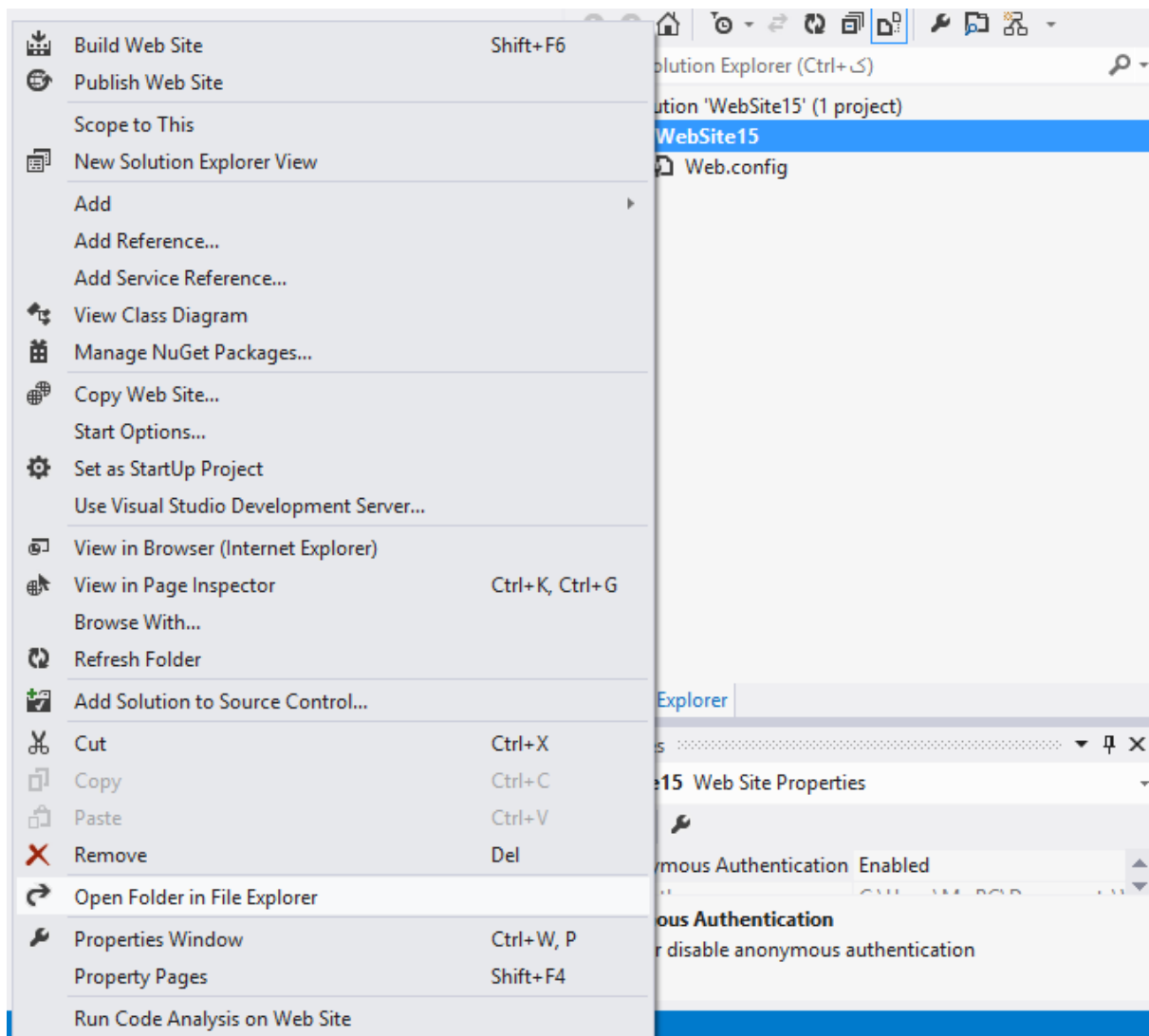
ابتدا پروژه جدیدی ایجاد می کنیم و فلدرهای قبلی را در آن قرار می دهیم:



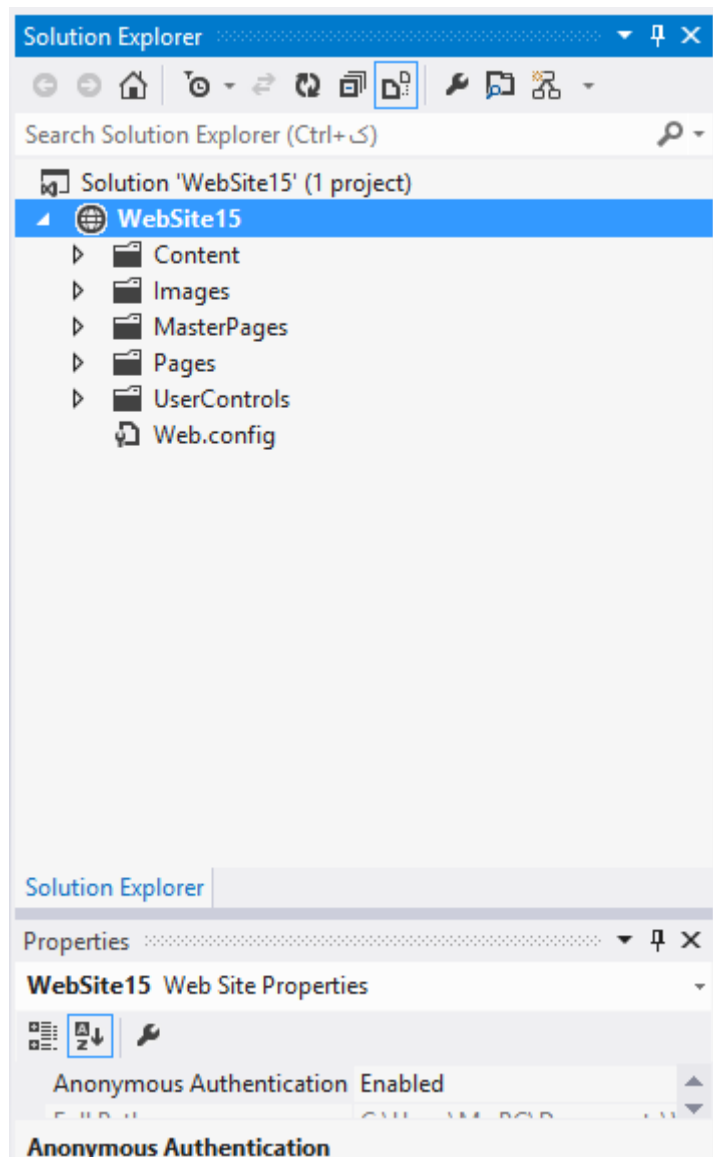
سپس:



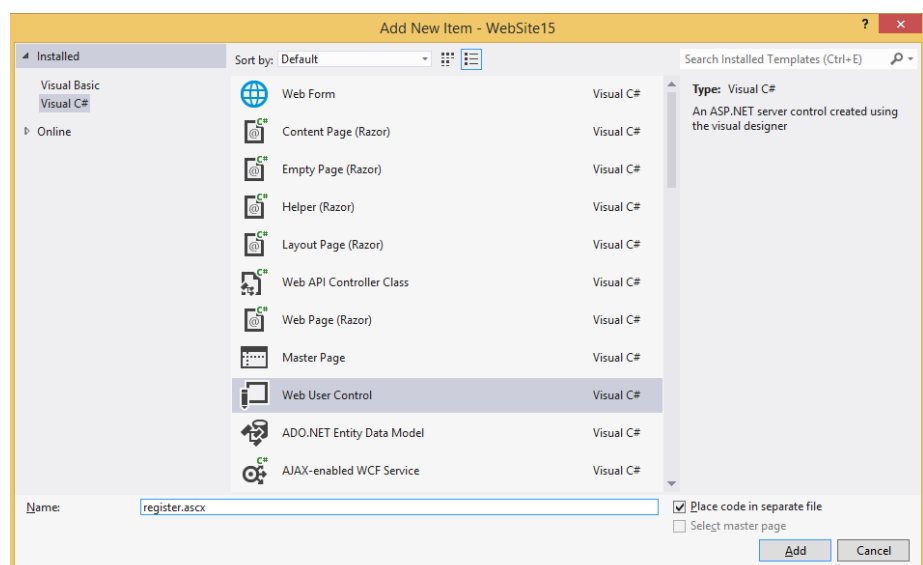
برای قرار دادن فلدر های قبلی روی نام پروژه راست کلیک می کنیم



در مکان باز شده فلدرها را کپی می نمایم و سپس رفرش می کنیم :



همان طور که می بینید یک پوشه ی جدیدی به نام user control ایجاد نموده و در داخل این پوشه یک آیتم جدید اضافه می کنیم :



در صفحه باز شده می توانیم کد کنترلی را ایجاد کنیم

```

<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="Register.ascx.cs" Inherits="UserControls_Register" %>
<link href="../Content/Site.css" rel="stylesheet" />
<center>
<fieldset>
<legend>Register</legend>
<div class="field">
<div class="lable">
<asp:Label ID="lblFirstName" runat="server" Text="FirstName : "
/>
</div>
<div class="controls">
<asp:TextBox ID="txtFirstName" runat="server" MaxLength="20" />
</div>
</div>
<div class="field">
<div class="lable">
<asp:Label ID="lblLastName" runat="server" Text="LastName : " />
</div>
<div class="controls">
<asp:TextBox ID="txtLastName" runat="server" MaxLength="20" />
</div>
</div>
<div class="field">
<div class="lable">
<asp:Label ID="lblEmail" runat="server" Text="Email : " />
</div>
<div class="controls">
<asp:TextBox ID="txtEmail" runat="server" MaxLength="20" />
</div>
</div>
<div class="field">
<div class="lable">
<asp:Label ID="lblPasswoed" runat="server" Text="Password : " />
</div>
<div class="controls">
<asp:TextBox ID="txtPassword" runat="server" MaxLength="20" />
</div>
</div>

```

```

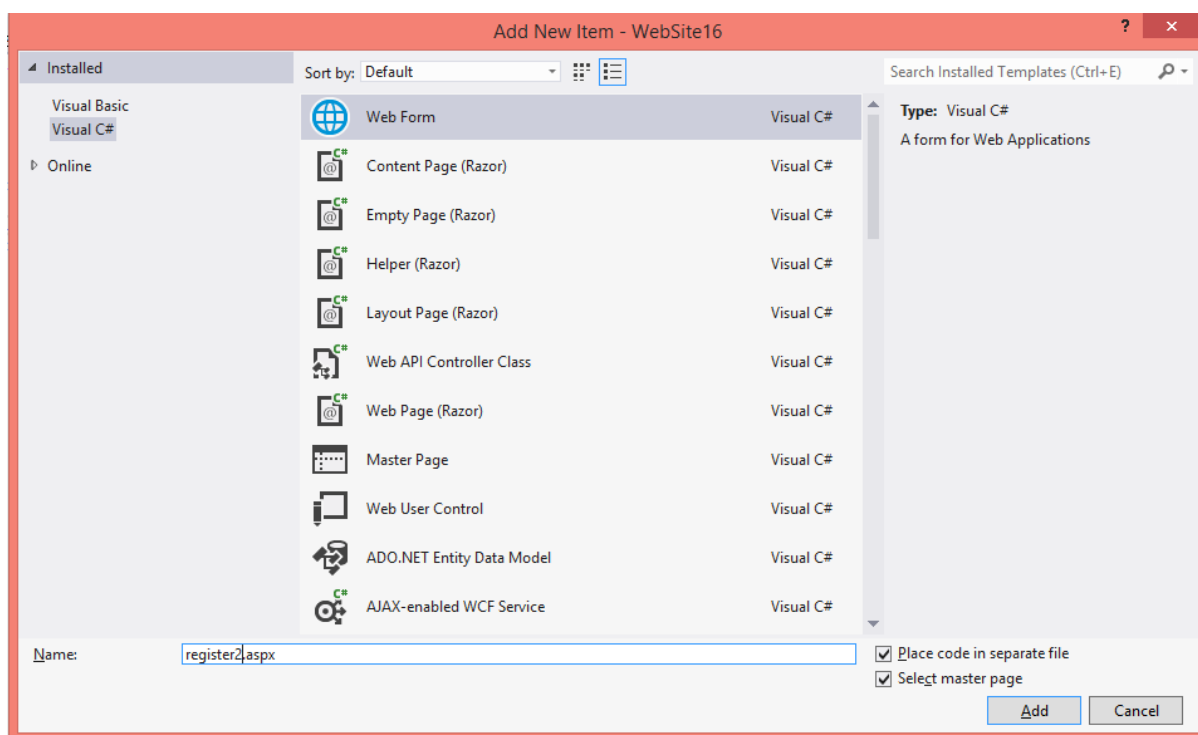
<div class="field">
<div class="button">
<asp:Button ID="btnRegister" runat="server" Text="Register" />
&nbsp;
<asp:Button ID="btnCancel" runat="server" Text="Cancel" />

</div>
</div>
</fieldset>
</center>

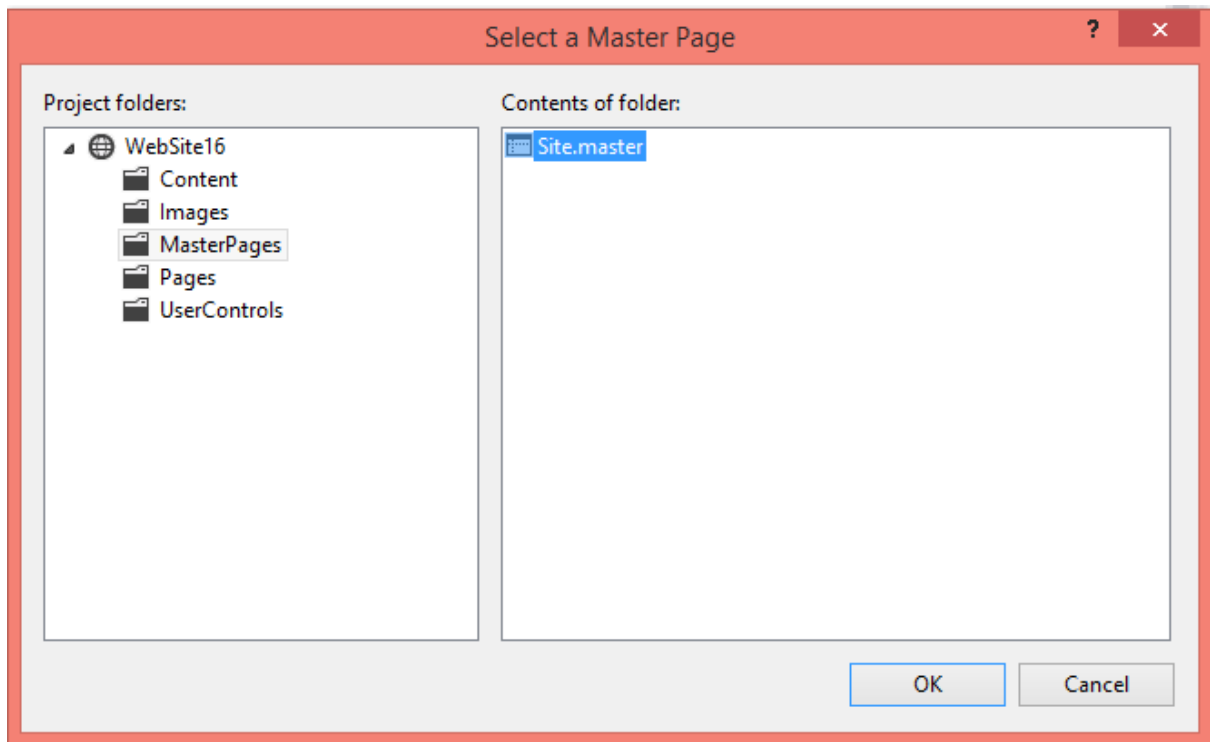
```

حال می‌توانیم این یوزر کنترل را در مستر پیج خود لحاظ نماییم تا هر زمان که از مستر پیج استفاده می‌نماییم اجزای یوزر کنترل نیز در صفحه اضافه گردد. برای اینکار به روش زیر اقدام می‌نماییم.

1- روی فولدر pages راست کلیک نموده و از **add new item** استفاده کرده و یک **web page** جدید ایجاد می‌کنیم



2- در حالی که تیک **select master page** خورده است گزینه **add** را کلیک می‌کنیم



3- در فولدر master pages مستر پیج site.master را انتخاب و دکمه ok را کلیک می کنیم .

4- در صفحه باز شده register2.aspx از فولدر usercontrols فایل register.ascx را مانند شکل زیر درگ می کنیم .



5- همزمان با درگ کردن دو خط زیر در فایل ایجاد می گردد

```
<%@ Register Src="~/UserControls/Register.ascx" TagPrefix="uc1"
TagName="Register" %>
```

```
<uc1:Register runat="server" ID="Register" />
```

6- حالا می توانیم به دفعات از این یوزر کنترل در برنامه استفاده کنیم

```

<%@ Page Title="" Language="C#"
MasterPageFile="~/MasterPages/Site.master"
AutoEventWireup="true" CodeFile="register2.aspx.cs"
Inherits="Pages_register2" %>

<%@ Register Src="~/UserControls/Register.ascx" TagPrefix="uc1"
TagName="Register" %>

<asp:Content ID="Content1" ContentPlaceHolderID="cphHead"
Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="cphMain"
Runat="Server">
    <uc1:Register runat="server" ID="Register" />
    <uc1:Register runat="server" ID="r2" />
</asp:Content>

```

### جلسه نهم: طراحی cms (سیستم مدیریت محتوا)

---

سیستم های مدیریت محتوا و یا به اصطلاح CMS مخفف Content Management Systems ابزار و برنامه هایی هستند که به شما کمک می کند تا مطالب خود را ایجاد و در یک منبع مشترک ذخیره سازی نمایید، و محتویات متنی و تصویری سایت خود را به صورت کاملاً پویا و آنلاین بروزرسانی کنید و از آن پس در مدیریت ارتباطات بین اجزا نیز به شما کمک خواهد نمود. در ادامه به معرفی برترین CMS ها و Forum هایی که قابلیت پشتیبانی کامل از زبان پارسی را نیز دارند می پردازیم:

**جوملا** یک سیستم مدیریت محتوای کد باز و قدرتمند است که برای هر نوع سایت ساده و یا پیچیده قابل بکارگیری می باشد. زبان برنامه نویسی آن PHP بوده و از بانک اطلاعاتی MySQL استفاده می کند. از ویژگیهای آن می توان به ذخیره گاه صفحه اشاره نمود که به افزایش قدرت اجرای آن کمک می کند.



**وردپرس** نیز یک سیستم مدیریت محتوای متن باز است که به خوبی با سایت های جستجوگر هماهنگ می شود ، پلاگین های بسیار زیادی متناسب با نیاز کار ارائه کرده است ، در زمینه های مختلف می توان از این سیستم استفاده کرد ، در ایران و سایر نقاط جهان طرفداران زیادی را داراست

**مامبو** را می توان مادر جوملا در نظر گرفت زیرا از بسیاری جهات ساختاری مشابه جوملا داشته و هر دو سیستم از یک سورس بهره جسته اند. مامبو با بهره گیری از زبان سطح بالای PHP و بکار گیری بانک های اطلاعاتی پهناور همانند جوملا امکان به جنبش در آوردن محتوای انتشار یافته در وبسایت را فراهم می کند.

**پی اچ پی نیوک** که از چند سال پیش طرفداران بسیاری در ایران پیدا کرده است ، یک سیستم مدیریت محتوای پیشرفته می باشد که بعنوان سیستم خودکار انتشار اخبار در اینترنت و اینترنت طراحی شده و از ویژگی های آن می توان به مدیریت آسان سایت و کاربران و همچنین تعامل صد در صد با بانک های اطلاعاتی اشاره نمود.

**مووبل تایپ** یک سیستم انتشار وبلاگ مبتنی بر زبان پرل می باشد که در سال ۲۰۰۱ اولین نسخه جدی آن منتشر شد. این سیستم تمامی خصیصه های مورد نیاز یک وبلاگ را پشتیبانی می کند ، مواردی از قبیل سیستم اعضا ، نظر گذاری ، انتخاب قالب ، بخش مدیریت وبلاگ و غیره

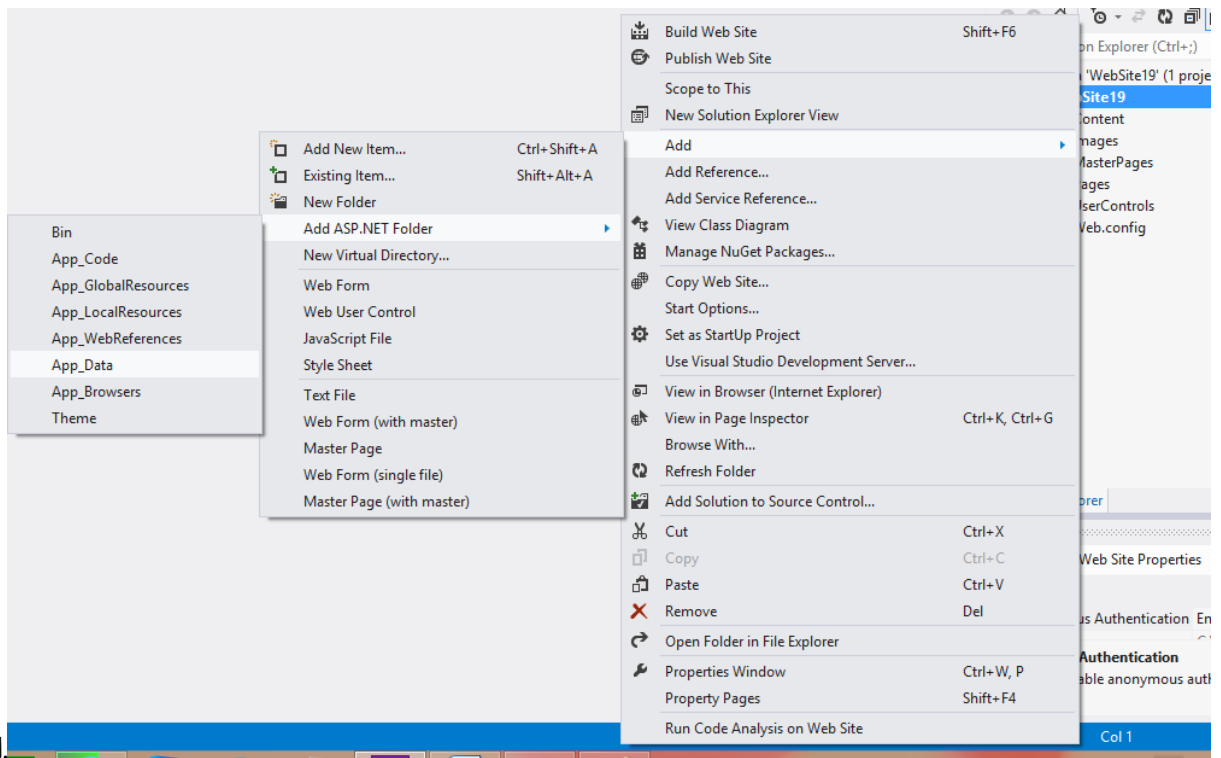
**107e** این سیستم مدیریت محتوای کد باز که از بانک اطلاعاتی معروف MySQL بهره می گیرد ، مدت زمان زیادی نیست که در دنیای وب مطرح شده ، با این حال از ویژگی های آن می توان به انعطاف پذیری و امکان انتشار سریع آن منطبق بر نیاز طراح و بکارگیری قالبهای مختلف اشاره نمود.

**PHPBB** مخفف PHP Bulletin Board یک سیستم انجمن اینترنتی کد باز است که با زبان سطح بالای PHP نوشته شده و قابلیت تنظیم قالب ، تنظیم زبان ، پشتیبانی از بانک های اطلاعاتی ، MySQL , Microsoft Access , Microsoft SQL Server , PostgreSQL و غیره از ویژگی های این انجمن می باشد.

**SMF** یک انجمن حرفه ایست که به شما در برقراری اجتماع آنلاین یاری می دهد. این انجمن از بسیاری جهات مشابه PHPBB بوده و دارای کامپوننت های جانبی مناسب بسیاریست که با نصب آنها می توانید تمامی نیازهای مدیریتی و کاربری خود را برآورده سازید.

**VBulletin** این انجمن بسایر قدرتمند و مبتنی بر زبان PHP ، در آخرین نسخه خود از سیستم آجاکس جهت پست دهی کاربران استفاده می کند و از امنیت بالایی برخوردار است. نکته قابل توجه در مورد VBulletin رایگان نبودن آن است!

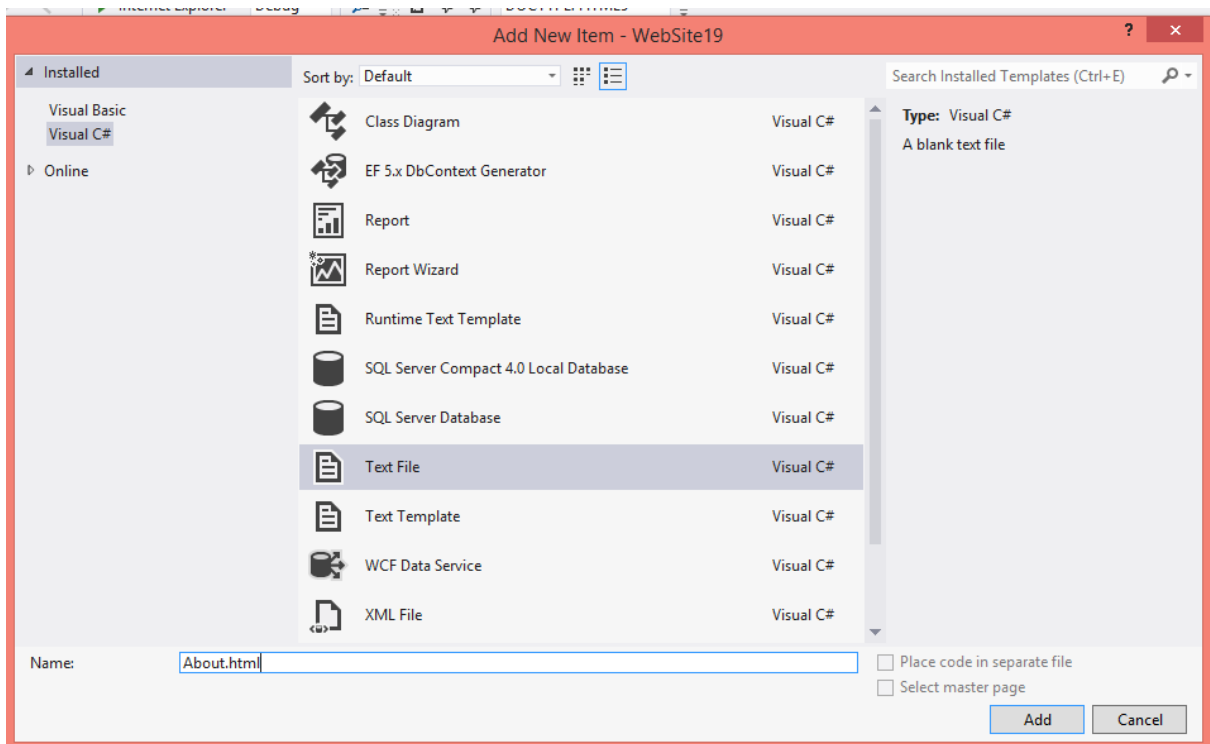
## 1- ابتدا پروژه قبلی را ایجاد و یک فولدر جدید cms ایجاد می کنیم.



این

فولدر امن asp.net است و امکان دسترسی آن برای دیگران ممکن نیست. Access denied پیام مبنی بر عدم دسترسی به فایل اعلام می نماید.

## 2- داخل آن فولدر pageContent ایجاد می کنیم و یک فایل html در آن ایجاد می کنیم



این صفحه می تواند مشخصات کاملی از admin یا شماره تماس یا اطلاعات دیگری که برای یک admin تعریف می گردد. مثلا داخل آن عبارتهای زیر اضافه می کنیم :



بنابراین admin قرار است از طریق صفحه خودش EdiPageControl با استفاده از تغییراتی که در About.html که ایجاد می کند ، کاربر در صفحه About.aspx بتواند ببیند.  
5- در صفحه About.aspx کد زیر اضافه می کنیم

```
<asp:Content ID="Content2" ContentPlaceHolderID="cphMain"
Runat="Server">
    <div>
        <asp:Literal id="litpagecontent" runat = "server" />
    </div>
</asp:Content>
```

6- در هدر دو لینک اضافه می کنیم

وارد مستر پیج می شویم و در قسمت ساب هدر کد های زیر اضافه می کنیم

```
<div class="subHeader">
    <asp:HyperLink ID="lnkAbout" runat="server"
Text="About" NavigateUrl="~/About.aspx" CssClass="link" />
    &nbsp;
    <asp:HyperLink ID="lnkEditPage" runat="server"
Text="Edit Page"
NavigateUrl="~/Administrator/EditPageContent.aspx"
CssClass="link" />
</div>
```

7-وارد صفحه کدهای سی شارپ about می شویم و کد های زیر اضافه می کنیم

```
protected void Page_Load(object sender, EventArgs e)
{
    if (Page.IsPostBack == false)
    {
        Initialize();
    }
}

private void Initialize()
{
    string strFileName =
        "About.html";
```

```

string strRootRelativePath =
    "~/App_Data/PageContent";
string strRootRelativePathName =
    string.Format("{0}/{1}", strRootRelativePath,
strFileName);

string strPathName =
    Server.MapPath(strRootRelativePathName);

if (System.IO.File.Exists(strPathName))
{
    System.IO.StreamReader oStreamReader = null;

    try
    {
        oStreamReader = new System.IO.StreamReader
            (strPathName, System.Text.Encoding.UTF8);
        litPagecontent.Text =
oStreamReader.ReadToEnd();

    }
    catch (Exception ex)
    {

        litPageContent.Text = ex.Message;
    }

    finally
    {
        if (oStreamReader != null)
        {
            oStreamReader.Dispose();
            oStreamReader = null;
        }
    }
}

```

متد : **try** یعنی تلاش

یعنی تلاش کن تا قطعه کدی که داخل **try** هست رو اجرا کنی  
و اگر با خطا مواجه شدی **catch** را اجرا کن

```
try
{
}
catch
{
}
```

حتما شما هم متوجه شدید که وقتی رخداد یک استثناء را با استفاده از **try** و **catch** کنترل می‌کنیم، هر چیزی که بعد از بسته شدن تگ **catch** بنویسیم، در هر صورت اجرا می‌شود.

```
try {
    int i=0;
    string s = "hello";
    i = Convert.ToInt32(s);
} catch (Exception ex)
{
    Console.WriteLine("Error");
}
Console.WriteLine("I am here!");
```

پس فلسفه استفاده از بخش **finally** چیست؟

در قسمت **finally** منابع تخصیص داده شده در **try** را آزاد می‌کنیم. کد موجود در این قسمت به هر روی اجرا می‌شود چه استثناء رخ دهد چه ندهد. البته اگر استثناء رخ داده شده در لیست استثناء هایی که برای آنها **catch** انجام دادیم نباشد، قسمت **finally** هم عمل نخواهد کرد مگر اینکه از **catch** به صورت سراسری استفاده کنیم. اما مهمترین مزیتی که **finally** ایجاد می‌کند در این است که حتی اگر در قسمت **try** با استفاده از دستوراتی مثل **return** یا **break** یا **continue** از ادامه کد منصرف شویم و مثلا مقداری برگردانیم، چه خطا رخ دهد یا ندهد کد موجود در **finally** اجرا می‌شود در حالی که کد نوشته شده بعد از **try catch finally** فقط در صورتی اجرا

می‌شود که به طور منطقی اجرای برنامه به آن نقطه برسد. اجازه بدهید با یک مثال توضیح دهم. اگر کد زیر را اجرا

کنیم:

```
public static int GetMyInt ()
{
    try {
        for (int i=10;i>=0;i--)
            Console.WriteLine(10/i);
        return 1;
    } catch
    {
        Console.WriteLine("Error!");
    }
    finally {
        Console.WriteLine("ok");
    }
    Console.WriteLine("can you reach here?");
    return -1;
}
```

برنامه خطای تقسیم بر صفر می‌دهد اما با توجه به کدی که نوشتیم، عدد -1 به خروجی خواهد رفت. در عین حال عبارت ok و can you reach here در خروجی چاپ شده است. اما حال اگر مشکل تقسیم بر صفر را حل کنیم، آیا باز هم عبارت can you reach here در خروجی چاپ خواهد شد؟

```
public static int GetMyInt ()
{
    try {
        for (int i=10;i>=1;i--)
            Console.WriteLine(10/i);
        return 1;
    } catch
    {
        Console.WriteLine("Error!");
    }
    finally {
        Console.WriteLine("ok");
    }
    Console.WriteLine("can you reach here?");
    return -1;
}
```

```

public int GetUserId(string nickname)
{
    SqlConnection connection = new SqlConnection(...);
    SqlCommand command = connection.CreateCommand();
    command.CommandText = "select id from users where
nickname like @nickname";
    command.Parameters.Add(new SqlParameter("@nickname",
nickname));
    try {
        connection.Open();
        return Convert.ToInt32(command.ExecuteScalar());
    }
    catch(SqlException exception)
    {
        // some exception handling
        return -1;
    } finally {
        if (connection.State == ConnectionState.Open)
            connection.Close();
    }
    // if all things works, you can not reach here
}

```

مشاهده می‌کنید که مقدار 1 برگردانده می‌شود و عبارت **can you reach here** در خروجی چاپ نمی‌شود ولی همچنان عبارت **ok** که در **finally** ذکر شده در خروجی چاپ می‌شود. یک مثال خوب استفاده از چنین وضعیتی، زمانی است که شما یک ارتباط با بانک اطلاعاتی باز می‌کنید، و نتیجه یک عملیات را با دستور **return** به کاربر بر می‌گردانید. مسئله این است که در این وضعیت چگونه ارتباط با دیتابیس بسته شده و منابع آزاد می‌گردند؟ اگر در حین عملیات بانک اطلاعاتی، خطایی رخ دهد یا ندهد، و شما دستور آزاد سازی منابع و بستن ارتباط را در داخل قسمت **finally** نوشته باشید، وقتی دستور **return** فراخوانی می‌شود، ابتدا منابع آزاد و سپس مقدار به خروجی بر می‌گردد.

**جلسه یازدهم: ادامه سیستم مدیریت محتوا cms :**

همانطور که قبلا گفته شد در فایل **About.aspx.cs** موارد زیر اضافه شد :

```

protected void Page_Load(object sender, EventArgs e)
{
    if (Page.IsPostBack==false)
    {

```



```

        Initialize();
    }

}

```

```

private void Initialize()
{
    string strFileName = "About.html";
    string strRootRelativePath = "~/App_Data/PageContent";
    string strRootRelativePathName = string.Format("{0}/{1}",strRootRelativePath,strFileName);

    string strPathName = Server.MapPath(strRootRelativePathName);

    if (System.IO.File.Exists(strPathName))
    {
        System.IO.StreamReader oStreamReader = null;

        try
        {
            oStreamReader = new System.IO.StreamReader
            (strPathName, System.Text.Encoding.UTF8);
            litPageContent.Text = oStreamReader.ReadToEnd();
        }
        catch (Exception ex)
        {
            litPageContent.Text = ex.Message;
        }

        finally
        {
            if (oStreamReader!=null)

```

در این تابع موارد زیر اجرا می گردد:

1- مسیر فایل را در رشته قرار می دهیم

2- با استفاده از دستور `Server.MapPath` آن را به آدرس فیزیکی تبدیل می کنیم

در این تابع موارد زیر اجرا می گردد:

1- اگر فایل مورد نظر ایجاد شده است

2- یک ابجکت به نام `oStreamReader` بساز و مقدار `null` در آن قرار ده

در این تابع موارد زیر اجرا می گردد:

1- `oStreamReader` را قالب دهی کن

2- از تابع `oStreamReader.ReadToEnd()` استفاده کن و محتوای فایل اختصاصی در تکس مربوطه قرار بده

```

    {
        oStreamReader.Dispose();
        oStreamReader = null;
    }
}

```

حال در داخل فایل EditPageContent.aspx مقادیر زیر اضافه می کنیم :

```

<div id="divPageMessages" runat="server" class="pageMessages"
visible="false" >
    <asp:Literal ID="litPageMessages" runat="server" />
</div>
<fieldset>
    <legend>Edit Pages</legend>
    <div class="field" >
        <div class="lable">
            <asp:Label ID="lblFileName" runat="server" Text="File Name
:" />
        </div>
        <div class="controls">
            <asp:TextBox ID="txtFileName" runat="server"
MaxLength="40" />
        </div>
    </div>

    <div class="field" >
        <div class="lable">
            <asp:Label ID="lblEdit" runat="server" Text="Edit : " />
        </div>
        <div class="controls">
            <asp:TextBox ID="txtEditPage" runat="server"
MaxLength="1000" TextMode="MultiLine" Rows="10" Columns="70" />
        </div>
    </div>
    <div class="field">
        <div class="button">
            <asp:Button ID="btnOpen" runat="server" Text="Open"
AccessKey="O" OnClick="btnOpen_Click" />
            &nbsp;
            <asp:Button ID="btnSave" runat="server" Text="Save"
AccessKey="S" OnClick="btnSave_Click" />
        </div>
    </div>
</fieldset>

```

در بروی دکمه open دابل کلیک نموده و کد زیر در آن اضافه می کنیم :

```
protected void btnOpen_Click(object sender, EventArgs e)
```

```

{
    txtFileName.Text =
        txtFileName.Text.Trim();

    if (txtFileName.Text==string.Empty)
    {
        string strErrorMessage =
            "You Did Not Specify File Name For Opening !";
        DisplayErrorMessage(strErrorMessage);
        return;
    }

    string strFileName = txtFileName.Text;
    string strRootRelativePath = "~/App_Data/PageContent";
    string strRootRelativePathName =
        string.Format("{0}/{1}",strRootRelativePath,strFileName);

    string strPathName = Server.MapPath(strRootRelativePathName);

    if (System.IO.File.Exists(strPathName)==false)
    {
        string strErrorMessage =
            string.Format("The File [{0}] Dose Not Exists For Opening
!",txtFileName.Text);
        DisplayErrorMessage(strErrorMessage);
        return;
    }

    System.IO.StreamReader oStreamReader = null;

    try
    {
        oStreamReader = new System.IO.StreamReader(strPathName,
System.Text.Encoding.UTF8);
        txtEditPage.Text = oStreamReader.ReadToEnd();

        string strInformationMessage =
            string.Format("The File [{0}] Opened Successfully . .
.",txtFileName.Text);
        DisplayInformationMessage(strInformationMessage);
    }
    catch (Exception ex)
    {

        DisplayErrorMessage(ex.Message);
    }
}

```

```

    }

    finally
    {
        if (oStreamReader!=null)
        {
            oStreamReader.Dispose();
            oStreamReader=null;
        }
    }
}
}
سپس بر روی دکمه save دابل کلیک می کنیم :
protected void btnSave_Click(object sender, EventArgs e)
{
    txtFileName.Text =
        txtFileName.Text.Trim();

    if (txtFileName.Text==string.Empty)
    {
        string strErrorMessage =
            "You Did Not Specify File For Opening!";
        DisplayErrorMessage(strErrorMessage);
    }

    string strFileName = txtFileName.Text;
    string strRootRelativePath = "~/App_Data/PageContent";
    string strRootRelativePathName =
        string.Format("{0}/{1}",strRootRelativePath,strFileName);

    string strPathName=Server.MapPath(strRootRelativePathName);

    System.IO.StreamWriter oStreamWriter = null;

    try
    {
        oStreamWriter = new System.IO.StreamWriter
            (strPathName, false, System.Text.Encoding.UTF8);
        oStreamWriter.Write(txtEditPage.Text);

        string strInformationMessage =
            string.Format("The File [{0}] Saved Successfully . .
            .",txtFileName.Text);
        DisplayInformationMessage(strInformationMessage);
    }
    catch (Exception ex)

```



برنامه های وب نیز می باشد . پس با انگیزه و دقت مضاعف کار را دنبال می نمائیم .

## ضرورت و ماهیت کنترل های Validation

زمانی که از یک TextBox برای درج داده بر روی یک فرم استفاده می گردد ، همواره این احتمال وجود خواهد داشت که کاربران اطلاعات را متناسب با انتظار یک برنامه وارد نمایند . به عنوان مثال ، فرض کنید که در یک فرم وب از دو TextBox برای درج داده عددی توسط کاربران استفاده شده است و در ادامه می بایست اطلاعات ورودی در اختیار یک روتین جهت انجام پردازش های مورد نیاز ( به عنوان نمونه ، انجام عملیات محاسباتی بر روی داده های ورودی ) قرار داده شود . در صورتی که کاربران داده را در محدوده مجاز وارد ننمایند و یا ورودی از نوع عددی نباشد ، روتین مورد نظر در انجام پردازش های خود با مشکل مواجه خواهد شد . بنابراین ، می بایست همواره قبل از این که داده ورودی در اختیار روتین های مورد نظر جهت انجام پردازش های مشخص شده قرار داده شود ، آنها را بررسی و در صورت رعایت مجموعه سیاست های تعریف شده ، امکان استفاده از آنها را برای اسکریپت ها فراهم نمود .

به منظور ارزیابی داده ورودی توسط کاربران با توجه به داده مورد نیاز اسکریپت ها ، از روش های متعددی می توان استفاده نمود . متداولترین نیاز ، وجود یک مقدار ورودی است . در چنین مواردی ، انتظار داریم قبل از این که یک اسکریپت بتواند پردازش های مورد نیاز را بر روی داده ورودی انجام دهد، در TextBox مقداری درج شده باشد . ماهیت داده ورودی در این مقطع مهم نمی باشد و مهم این است که داده ئی توسط کاربر در مکان مورد نظر وارد شده باشد . در برخی موارد لازم است که داده ورودی از یک نوع خاص باشد ( به عنوان نمونه یک مقدار عددی ) تا بتوان آن را در اختیار روتین های مورد نظر به منظور انجام پردازش های محاسباتی قرار داد . در برخی موارد دیگر ، لازم است که داده ورودی در یک محدوده خاص باشد و داده درج شده خارج از محدوده مورد نظر ، غیرمعتبر گردد . برای بررسی و ارزیابی صحت داده در ASP.NET مجموعه ای از کنترل های validation ارائه شده است . از کنترل های فوق می توان به منظور بررسی و تست عدم درج داده ، مقایسه داده ورودی با یک مقدار خاص ، بررسی وجود مقادیر در یک محدوده خاص و سایر فرم های داده استفاده نمود تا این اطمینان ایجاد گردد که داده مناسب و معتبر در اختیار اسکریپت ها و به منظور انجام پردازش ها قرار داده می شود .

کنترل های validation در ارتباط مستقیم با کنترل TextBox بوده و تست و ارزیابی خود را بطور اتوماتیک و پس از کلیک بر روی کنترل هائی نظیر Button ، LinkButton و ImageButton انجام خواهند داد . در صورتی که ماحصل بررسی یک کنترل validation توام با موفقیت نباشد ، validator یک پیام خطا را نمایش و به کاربران اعلام می نماید که داده ورودی فاقد شرایط لازم برای استفاده در برنامه می باشد . در شکل زیر کنترل های Validation ارائه شده در ASP.NET 2.0 ، نشان داده شده است .



کنترل های Validation

در ادامه به بررسی کنترل های فوق خواهیم پرداخت .

### کنترل <asp:RequiredFieldValidator>

متداولترین نوع ارزیابی داده ورودی ، بررسی آن به منظور اطمینان از درج داده توسط کاربر است . کنترل <asp:RequiredFieldValidator> ، مسئولیت بررسی داده ورودی در یک Textbox را برعهده گرفته و در صورت عدم درج داده توسط کاربر ، یک پیام خطا را نمایش خواهد داد . در چنین مواردی ، روتین هائی که می بایست پس از کلیک بر روی دکمه موجود از داده درج شده در TextBox استفاده نمایند ، فرصت انجام چنین کاری را به دلیل عدم درج داده توسط کاربر پیدا نخواهند کرد .

### شکل

### عمومی

شکل عمومی کنترل فوق به صورت زیر است :

```
<asp:RequiredFieldValidator id="id"
Runat="Server"
ControlToValidate="controlID"
Display="Dynamic|None|Static"
ErrorMessage="string"
SetFocusOnError="False|True"
ValidationGroup="name"
/>
```

توضیحات:

- از یک id صرفاً زمانی استفاده می شود که قرار است به کنترل از طریق اسکریپت مراجعه گردد .
- به خصلت `ControlToValidate` ، مقدار id کنترل `textBox` که مسئولیت بررسی آن به کنترل `RequiredFieldValidator` واگذار شده است ،نسبت داده می شود.
- خصلت `Text`، مشخصه `Text` به عنوان متنی استفاده می شود که کنترل تایید اعتبار روی صفحه نمایش می دهد .این می تواند یک آستریکس (\*) برای نشان دادن یک خطا باشد .یا متنی مانند `Enter Your .Name` .
- `CssClass`، این مشخصه به ما اجازه می دهد تا خاصیت `class` در `CSS` را تعیین کنیم که روی متن پیام خطا اعمال می شود.
- خصلت `ErrorMessage`، این مشخصه پیام خطای استفاده شده در کنترل `validationSummary` را نگه می دارد.وقتی مشخصه `Text` خالی باشد، مقدار `ErrorMessage` نیز به عنوان متنی استفاده می شود که روی صفحه ظاهر می شود.
- خصلت `SetFocusOnErrorMessage` ، یک `blinking cursor` را در کنترل `TextBox` مربوطه به منظور تسهیل در امر درج داده جدید قرار می دهد .
- این مشخصه تعیین می کند که آیا اسکریپت روی کلاینت ،روی اولین کنترلی که تولید خطا کرده است فوکوس را قرار دهد.
- خصلت `ValidationGroup` یک گروه از کنترل های `TextBox` را که مجموعه ای از تست های بررسی روی آنها اعمال خواهد شد ، مشخص می نماید (در مواردی که دکمه های متفاوت می توانند باعث فعال شدن تست های مختلفی گردند مثلاً برای کنترلهای صفحه `login` از یک گروه و برای `search` از گروه دیگر) .
- خصلت `Display` : کنترل `RequiredFieldValidator` ، یک فضای افقی را متناسب با طول رشته مربوط به پیام خطا اشغال خواهد کرد . در اغلب موارد ، پیام خطا در کنار `TextBox` مربوطه نمایش داده می شود . مکان فوق بر روی صفحه همواره نشان داده خواهد شد ( ولو این که خطائی اتفاق نیافتد ) . در صورتی که مقدار خصلت `Display` معادل `Dynamic` در نظر گرفته شود ، مکانی برای نمایش پیام خطا رزو نخواهد شد و بطور پویا و همزمان با بروز خطا ایجاد می گردد .با مقدار `None` ،کنترل اصلاً قابل دیدن نخواهد بود .این برای زمانی مفید است که از یک `ValidationSummary` استفاده کنید.
- خصلت `IsValid` ، معمولاً این مشخصه را در زمان طراحی اعمال نمی کنیم بلکه این کار را در زمان اجرا انجام می دهیم زیرا اطلاعاتی در باره این که آیا آزمایش تایید اعتبار با موفقیت انجام شده است فراهم می آورد . کلاس `page` نیز دارای یک مشخصه `IsValid` است که نتیجه ترکیب شده همه کنترلهای درون صفحه یا گروه تایید اعتبار را برمی گرداند.



در validationSummary از عبارت ErrorMessage استفاده می شود.

### مثال

در این مثال زمانی که کاربر بر روی button کلیک نمود، کنترل RequiredFieldValidator بطور اتوماتیک عملیات بررسی و ارزیابی داده ورودی در TextBox را آغاز می نماید (قبل از این که روتین GetData فراخوانده شود). در صورتی که TextBox خالی باشد، یک پیام خطاء نمایش داده شده و cursor مجدداً در TextBox قرار خواهد گرفت. در صورتی که در TextBox داده ئی توسط کاربر درج گردد، تست ارزیابی با موفقیت انجام و روتین مربوط فراخوانده شده و مقدار ورودی را در خروجی نمایش خواهد داد. در این مثال، مقدار خصلت Display کنترل RequiredFieldValidator، معادل Dynamic در نظر گرفته شده است، بنابراین در صورت عدم تولید یک پیام خطاء، مکانی بر روی صفحه برای نمایش آن در نظر گرفته نخواهد شد. بدین ترتیب به کنترل label اجازه داده می شود که در کنار TextBox نمایش داده شده و از فضای مشابه پیام خطاء استفاده نماید (استفاده مشترک از یک مکان بر روی فرم وب به منظور نمایش پیام خطاء و یا داده ورودی توسط کاربر).

```
<SCRIPT Runat="Server">

    Sub Get_Data (Src As Object, Args As EventArgs)
        Output.Text = "You entered '" & MyTextBox.Text & "'"
    End Sub

</SCRIPT>
<form Runat="Server">
<asp:TextBox id="MyTextBox" Runat="Server"/>
<asp:Button Text="Submit" OnClick="Get_Data" Runat="Server"/>
<asp:RequiredFieldValidator Runat="Server"
    ControlToValidate="MyTextBox"
    ErrorMessage="Please enter a data value"
    Display="Dynamic"
    SetFocusOnError="True"/>
<asp:Label id="Output" Runat="Server"/>

</form>

<asp:RequiredFieldValidator ID="rfvUserName" runat="server"
ControlToValidate="txtUserName" CssClass="validator"
Display="Dynamic" EnableClientScript="true"

SetFocusOnError="true" Text="UserName Is Required . . ."
/>
```

## کنترل <asp:RangeValidator>

کنترل فوق مسئولیت بررسی و ارزیابی داده ورودی در یک TextBox را برعهده گرفته و مأموریت آن حصول اطمینان از این موضوع است که داده درج شده در محدوده مورد نظر است. این محدوده توسط دو خصلت MinimumValue و MaximumValue مشخص می گردد. type در فرآیند فوق می تواند در ارتباط با نوع های متفاوت داده نظیر Date، اعداد صحیح، اعداد اعشاری و یا رشته (مقدار پیش فرض) اعمال گردد.

### شکل عمومی

شکل عمومی کنترل فوق به صورت زیر است:

```
<asp:RangeValidator id="id" Runat="Server"
  ControlToValidate="controlID"
  Display="Dynamic|None|Static"
  ErrorMessage="string"
  MaximumValue="value"
  MinimumValue="value"
  SetFocusOnError="False|True"
  Type="Currency|Date|Double|Integer|String"
  ValidationGroup="name"
/>
```

### توضیحات

- از یک id صرفاً زمانی استفاده می شود که قرار است به کنترل از طریق اسکریپت مراجعه گردد.
- خصلت ErrorMessage، پیام مورد نظر در صورت بروز خطا را مشخص می نماید.
- خصلت SetFocusOnErrorMessage، یک blinking cursor را در کنترل TextBox مربوطه به منظور تسهیل در امر درج داده جدید قرار می دهد.
- خصلت ValidationGroup یک گروه از کنترل های TextBox را که مجموعه ای از تست های بررسی روی آنها اعمال خواهد شد، مشخص می نماید (در مواردی که دکمه های متفاوت می توانند باعث فعال شدن تست های مختلفی گردند).
- خصلت Display: کنترل RangeValidator، یک فضای افقی را متناسب با طول رشته مربوط به پیام خطا اشغال خواهد کرد. در اغلب موارد، پیام خطا در کنار TextBox مربوطه نمایش داده می شود. مکان فوق بر روی صفحه همواره نشان داده خواهد شد (ولو این که خطائی اتفاق نیافتد). در صورتی که مقدار خصلت Display معادل Dynamic در نظر گرفته شود، مکانی برای نمایش پیام خطا رزو نخواهد شد و بطور پویا و همزمان با بروز خطا ایجاد می گردد.

- **Type** : در صورتی که نوع داده ورودی مشخص نگردد ، نوع آن به صورت پیش فرض **string** در نظر گرفته خواهد شد . یک **TextBox** خالی ، به عنوان یک نوع داده معتبر ارزیابی خواهد شد . بنابراین ، لازم است که به همراه کنترل **RangeValidator** از یک کنترل **RequiredFieldValidator** نیز استفاده گردد تا این اطمینان حاصل شود که با عدم درج داده در **TextBox** مربوطه با آن به عنوان یک داده معتبر برخورد نخواهد شد .
- خصلت های **MinimumValue** و **MaximumValue** ، حداقل و حداکثر محدوده مجاز برای داده ورودی را مشخص می نمایند .

### مثال

در این مثال به منظور ارزیابی و تست داده ورودی در یک **TextBox** از دو کنترل **RangeValidator** و **RequiredFieldValidator** استفاده شده است . بنابراین می بایست حتما" در **TextBox** مقداری درج گردد . با توجه به این که مقدار خصلت **Type** معادل **integer** و مقادیر خصلت های **MinimumValue** و **MaximumValue** به ترتیب صفر و نه در نظر گرفته شده اند ، داده ورودی می بایست عددی بین صفر تا نه باشد .

```
<SCRIPT Runat="Server">

Sub Get_Data (Src As Object, Args As EventArgs)
    Output.Text = "You entered '" & MyTextBox.Text & "'"
End Sub

</SCRIPT>

<form Runat="Server">

Enter a value between 0 and 9:<br/>
<asp:TextBox id="MyTextBox" Runat="Server"/>
<asp:Button Text="Submit" OnClick="Get_Data" Runat="Server"/>
<asp:RangeValidator Runat="Server"
    ControlToValidate="MyTextBox"
    Type="Integer"
    MinimumValue="0"
    MaximumValue="9"
    ErrorMessage="Please enter an integer in the range 0 to 9"
    Display="Dynamic">
```

```

SetFocusOnError="True"/>
<asp:RequiredFieldValidator Runat="Server"
  ControlToValidate="MyTextBox"
  ErrorMessage="Please enter a data value"
  Display="Dynamic"
  SetFocusOnError="True"/>
<asp:Label id="Output" Runat="Server"/>

</form>

```

### کنترل <asp:CompareValidator>

کنترل فوق مسئولیت بررسی مقدار درج شده در یک TextBox را برعهده دارد. بدین منظور داده ورودی با یک مقدار خاص و یا مقدار یک کنترل موجود بر روی فرم مقایسه می گردد. نوع داده درج شده در TextBox می تواند از نوع Currency، Date، اعشاری، صحیح و یا رشته (مقدار پیش فرض) باشد.

#### شکل عمومی

شکل عمومی کنترل فوق به صورت زیر است:

```

<asp:CompareValidator id="id" Runat="Server"
ControlToCompare="controlID"
ControlToValidate="controlID"
Display="Dynamic|None|Static"
ErrorMessage="string"
Operator="Equal|NotEqual|GreaterThan|GreaterThanEqual|LessThan
|LessThanEqual|DataTypeCheck"
SetFocusOnError="False|True"
Type="Currency|Date|Double|Integer|String"
ValidationGroup="name"
ValueToCompare="value"
/>

```

#### توضیحات

- از یک id صرفاً زمانی استفاده می شود که قرار است به کنترل از طریق اسکریپت مراجعه گردد.
- ControlToCompare، این مشخصه حاوی ID کنترلی است که مقایسه کننده با آن مقایسه می شود. وقتی این مشخصه تعیین شود، ValueToCompare بی اثر می شود.

- Operator، این مشخصه تعیین کننده نوع عملیات مقایسه است .
- به خصلت `ControlToValidate` ، مقدار `id` کنترل `textBox` که مسئولیت بررسی آن به کنترل `<asp:CompareValidator>` واگذار شده است ،نسبت داده می شود.
- خصلت `ErrorMessage` ، پیام مورد نظر در صورت بروز خطاء را مشخص می نماید .
- خصلت `SetFocusOnErrorMessage` ، یک `blinking cursor` را در کنترل `TextBox` مربوطه به منظور تسهیل در امر درج داده جدید قرار می دهد .
- خصلت `ValidationGroup` یک گروه از کنترل های `TextBox` را که مجموعه ای از تست های بررسی روی آنها اعمال خواهد شد ، مشخص می نماید (در مواردی که دکمه های متفاوت می توانند باعث فعال شدن تست های مختلفی گردند) .
- خصلت `Display` : کنترل `<asp:CompareValidator>` ، یک فضای افقی را متناسب با طول رشته مربوط به پیام خطاء اشغال خواهد کرد . در اغلب موارد ، پیام خطاء در کنار `TextBox` مربوطه نمایش داده می شود . مکان فوق بر روی صفحه همواره نشان داده خواهد شد ( ولو این که خطائی اتفاق نیافتد ) . در صورتی که مقدار خصلت `Dynamic` معادل `Display` در نظر گرفته شود ، مکانی برای نمایش پیام خطاء رزو نخواهد شد و بطور پویا و همزمان با بروز خطاء ایجاد می گردد .
- مقدار ورودی می تواند با یک مقدار مشخص شده توسط خصلت `ValueToCompare` و یا مقدار یک کنترل دیگر موجود در صفحه ( مشخص شده توسط خصلت `ControlToCompare` ) ، مقایسه گردد . به صورت پیش فرض ، عملیات مقایسه برای "برابری" انجام خواهد شد . در صورت نیاز می توان از سایر عملگرهای مقایسه ای که توسط خصلت `Operator` مشخص می گردند ، استفاده نمود . عملیات مقایسه بر اساس نوع داده درج شده در `Textbox` که توسط خصلت `Type` مشخص می گردد ، انجام خواهد شد .
- خصلت `Type` : در صورتی که نوع داده ورودی مشخص نگردد ، نوع آن به صورت پیش فرض `string` در نظر گرفته خواهد شد . یک `TextBox` خالی ، به عنوان یک نوع داده معتبر ارزیابی خواهد شد . بنابراین ، لازم است که به همراه کنترل `<asp:CompareValidator>` از یک کنترل `RequiredFieldValidator` نیز استفاده گردد تا این اطمینان حاصل شود که با عدم درج داده در `TextBox` مربوطه با آن به عنوان یک داده معتبر برخورد نخواهد شد .

### مثال

در این مثال ، کاربر می بایست یک عدد مثبت را وارد نماید و در صورتی که مقدار ورودی منفی باشد یک پیام خطاء نمایش داده می شود . به همراه کنترل `CompareValidator` از یک کنترل `RequiredFieldValidator` نیز استفاده شده است تا عدم درج داده ( خالی بودن ) به عنوان یک داده معتبر ارزیابی نگردد . مقدار ورودی می بایست از نوع اعشاری (`Type=Double`) و بزرگتر از (`Operator=GreaterThan`) صفر (`ValueToCompare=0`) باشد تا به عنوان یک داده معتبر ارزیابی گردد .

```

<SCRIPT Runat="Server">

Sub Get_Data (Src As Object, Args As EventArgs)
    Output.Text = "You entered '" & MyTextBox.Text & "'"
End Sub

</SCRIPT>

<form Runat="Server">

Enter a positive number:<br/>
<asp:TextBox id="MyTextBox" Runat="Server"/>
<asp:Button Text="Submit" OnClick="Get_Data" Runat="Server"/>
<asp:CompareValidator Runat="Server"
    ControlToValidate="MyTextBox"
    ValueToCompare="0"
    Type="Double"
    Operator="GreaterThan"
    ErrorMessage="Please enter a number greater than 0"
    Display="Dynamic"
    SetFocusOnError="True"/>
<asp:RequiredFieldValidator Runat="Server"
    ControlToValidate="MyTextBox"
    ErrorMessage="Please enter a data value"
    Display="Dynamic"
    SetFocusOnError="True"/>
<asp:Label id="Output" Runat="Server"/>

</form>

```

### کنترل <asp:CustomValidator>

در زمان بررسی و ارزیابی داده ورودی ممکن است به مواردی برخورد نمائیم که با ترکیب یک RequiredFieldValidator ، RangeValidator و یا CompareValidator خواسته ما تامین نگردد. در چنین مواردی می توان از کنترل <asp:CustomValidator> به منظور انجام تست های اضافه استفاده نمود .

```
<asp:CustomValidator id="id" Runat="Server"
ControlToValidate="controlID"
Display="Dynamic|None|Static"
ErrorMessage="string"
SetFocusOnError="False|True"
ValidationGroup="name"
OnServerValidate="subprogram"
/>
```

### توضیحات

- خصلت های کنترل <asp:CustomValidator> مشابه سایر کنترل های validation می باشند با این تفاوت که از خصلت OnServerValidate به منظور فراخوانی یک برنامه فرعی برای بررسی و ارزیابی داده ورودی نیز استفاده می گردد .
- یک TextBox خالی به عنوان یک داده معتبر در نظر گرفته خواهد شد . بنابراین می بایست به همراه کنترل CustomValidator از یک کنترل RequiredFieldValidator نیز استفاده گردد تا عدم درج داده به عنوان یک داده معتبر ارزیابی نگردد .
- برنامه فرعی صدا زده شده دارای آرگومان های خاص ServerValidateArgs می باشد . در صورتی که بررسی و ارزیابی صحت داده ورودی توام با موفقیت نباشد ، مقدار خصلت IsValid معادل false خواهد شد . از آرگومان Value ، به عنوان مرجعی به منظور مراجعه به مقدار TextBox ( کنترل مشخص شده توسط خصلت ControlToValidate ) استفاده می گردد .
- خصلت ErrorMessage ، پیام مورد نظر در صورت بروز خطاء را مشخص می نماید .
- خصلت SetFocusOnErrorMessage ، یک blinking cursor را در کنترل TextBox مربوطه به منظور تسهیل در امر درج داده جدید قرار می دهد .
- خصلت ValidationGroup یک گروه از کنترل های TextBox را که مجموعه ای از تست های بررسی روی آنها اعمال خواهد شد ، مشخص می نماید (در مواردی که دکمه های متفاوت می توانند باعث فعال شدن تست های مختلفی گردند ) .
- خصلت Display : کنترل <asp:CustomValidator> ، یک فضای افقی را متناسب با طول رشته مربوط به پیام خطاء اشغال خواهد کرد . در اغلب موارد ، پیام خطاء در کنار TextBox مربوطه نمایش داده می شود . مکان فوق بر روی صفحه همواره نشان داده خواهد شد ( ولو این که خطائی اتفاق نیافتد ) . در

صورتی که مقدار خصلت **Display** معادل **Dynamic** در نظر گرفته شود ، مکانی برای نمایش پیام خطا رزو نخواهد شد و بطور پویا و همزمان با بروز خطا ایجاد می گردد .

## مثال

در این مثال به منظور ارزیابی و تست داده ورودی در یک **TextBox** از دو کنترل **CustomValidator** و **RequiredFieldValidator** استفاده شده است . داده ورودی می بایست یک عدد صحیح بین صفر تا نود و نه باشد .

پس از کلیک بر روی دکمه **Submit** ، روتین **Get\_Data** فعال و قبل از هر چیز عملیات بررسی و ارزیابی صحت داده ورودی انجام خواهد شد . کنترل **RequiredFieldValidator** در ابتدا بررسی لازم در خصوص درج داده در **TextBox** را انجام و در ادامه روتین **Validate\_TextBox** توسط خصلت **OnServerValidate** کنترل **CustomValidator** فراخوانده می شود . روتین فوق ، تست های لازم را بر روی داده ورودی انجام خواهد داد ( برای مراجعه به داده ورودی از **Args.Value** استفاده شده است ) . در صورتی که هر یک از تست های انجام شده توأم با موفقیت نباشد ، خصلت **Args.IsValid** مقدار **false** را خواهد گرفت ( داده ورودی می بایست یک عدد مثبت بین صفر تا نود و نه باشد ) .

پس از انجام فرآیند بررسی داده ورودی ، روتین **Get\_Data** ، اجرا خواهد شد . روتین فوق در ابتدا و پس از کلیک بر روی دکمه **button** فراخوانده می گردد و اجرای آن تا زمانی که عملیات بررسی و ارزیابی داده ورودی به اتمام نرسیده باشد ، به تاخیر خواهد افتاد . پردازش های انجام شده در روتین **Get\_Data** مشروط به انجام موفقیت آمیز تست ارزیابی و صحت داده است . بنابراین ، همه چیز وابسته به شرط **Page.IsValid** شده است و در صورتی که **Args.IsValid** معادل **false** شده باشد ، مقدار **Page.IsValid** نیز **false** خواهد شد.

```
<SCRIPT Runat="Server">
```

```
Sub Validate_TextBox (Src As Object, Args As ServerValidateEventArgs)
```

```
    If Not IsNumeric(Args.Value) Then
```

```
        MyValidator.ErrorMessage = "Please enter a number"
```

```
        Args.IsValid = False
```

```
    Else
```

```
        If Not Args.Value Mod 1 = 0 Then
```

```
            MyValidator.ErrorMessage = "Please enter an integer"
```

```
            Args.IsValid = False
```

```
        End If
```

```
        If Args.Value < 0 Then
```



```

    MyValidator.ErrorMessage = "Please enter a positive integer"
    Args.IsValid = False
End If
If Args.Value > 99 Then
    MyValidator.ErrorMessage = "Please enter a positive integer between 0
and 99"
    Args.IsValid = False
End If
End If

End Sub
Sub Get_Data (Src As Object, Args As EventArgs)

If Page.IsValid Then
    Output.Text = "You entered '" & MyTextBox.Text & "'"
End If

End Sub
</SCRIPT>

<form Runat="Server">

Enter a positive integer:<br/>
<asp:TextBox id="MyTextBox" Runat="Server"/>
<asp:Button Text="Submit" OnClick="Get_Data" Runat="Server"/>
<asp:CustomValidator id="MyValidator" Runat="Server"
    ControlToValidate="MyTextBox"
    Display="Dynamic"
    SetFocusOnError="True"
    OnServerValidate="Validate_TextBox"/>
<asp:RequiredFieldValidator Runat="Server"
    ControlToValidate="MyTextBox"
    ErrorMessage="Please enter a data value"
    Display="Dynamic"
    SetFocusOnError="True"/>
<asp:Label id="Output" Runat="Server"/>

```

```
</form>
```

مثال :

```
<div class="controls">
    <asp:TextBox ID="txtPassword" runat="server"
MaxLength="40"
        CssClass="password" />
    <asp:RequiredFieldValidator ID="rfvPassword"
runat="server"
        ControlToValidate="txtPassword" CssClass="validator"
        Display="Dynamic" EnableClientScript="true"
        SetFocusOnError="true" Text="Password Is Required . .
. "
        />
    <asp:RegularExpressionValidator ID="revPassword"
runat="server"
        ControlToValidate="txtPassword" CssClass="validator"
        Display="Dynamic" EnableClientScript="true"
SetFocusOnError="true"
        Text="Password Is Not Valid . . ."
        ValidationExpression="[a-zA-Z0-9]{6,20}" />
</div>
</div>
```

جلسه سیزدهم: طراحی منو:

```
<asp:Menu ID="mnuMain" runat="server" RenderingMode="Table"
MaximumDynamicDisplayLevels="3"
Orientation="Vertical" DisappearAfter="400"
DynamicHorizontalOffset="2"
BackColor="#FFFD6" ForeColor="#990000"
StaticSubMenuIndent="20px"
StaticPopOutImageUrl="~/Images/RightToLeftArrow.gif"
DynamicPopOutImageUrl="~/Images/RightToLeftArrow.gif">

<Items>
<asp:MenuItem Text="Home . . ." NavigateUrl="~/Default.aspx"
ToolTip="Home" />

<asp:MenuItem Text="Learn . . ." NavigateUrl="~/Default.aspx"
ToolTip="Learn">
```

```

<asp:MenuItem Text="87-91" NavigateUrl="~/Default.aspx"
ToolTip="87-91" />
<asp:MenuItem Text="92" NavigateUrl="~/Default.aspx"
ToolTip="92" />
</asp:MenuItem>

<asp:MenuItem Text="Help . . ." NavigateUrl="~/Default.aspx"
ToolTip="Help"/>
<asp:MenuItem Text="ContactUs . . ."
NavigateUrl="~/Default.aspx" ToolTip="ContactUs" />

<asp:MenuItem Text="AboutUs . . ." NavigateUrl="~/Default.aspx"
ToolTip="AboutUs">
<asp:MenuItem Text="System" NavigateUrl="~/Default.aspx"
ToolTip="System" />
<asp:MenuItem Text="Team" NavigateUrl="~/Default.aspx"
ToolTip="Team">

<asp:MenuItem Text="Support" NavigateUrl="~/Default.aspx"
ToolTip="Support" />
<asp:MenuItem Text="Sales" NavigateUrl="~/Default.aspx"
ToolTip="Sales" />

</asp:MenuItem>
</asp:MenuItem>

<asp:MenuItem Text="News . . ." NavigateUrl="~/Default.aspx" />

</Items>
<StaticSelectedStyle BackColor="#FFCC66" />
<StaticHoverStyle BackColor="#990000" ForeColor="White" />
<StaticMenuItemStyle HorizontalPadding="5px"
VerticalPadding="2px" CssClass="fixMenuItem" />
<DynamicSelectedStyle BackColor="#FFCC66" />
<DynamicMenuStyle BackColor="#FFFBD6" CssClass="fixMenu" />
<DynamicHoverStyle BackColor="#990000" ForeColor="White" />
<DynamicMenuItemStyle HorizontalPadding="5px"
VerticalPadding="2px" CssClass="fixMenuItem" />
</asp:Menu>

```

طراحی منوهای ساده و کارآمد که دسترسی به تمامی صفحات و موضوعات سایت شما را فراهم نماید ، یک موضوع حیاتی است . کاربران بایستی در هر صفحه بتوانند به صفحات اصلی دیگر دسترسی داشته و بدانند در کجای سایت قرار دارند .

**ASP.Net** دارای کنترل های درون ساخته ای است که به وسیله آنها می توانید ، انواع منو ها را ایجاد نمایید . همچنین **ASP.Net** دارای یک امکان به نام نقشه وب سایت می باشد ، که به وسیله آن ، کلیه مسیرهای مورد نظر خود را درون یک فایل با پسوند **sitemap** . تعیین می کنید . فایل **sitemap** ، حکم یک فایل داده ای از نوع **XML** را داراست که می توان سایر کنترل های منوی **ASP.Net** را به آن متصل نمود . در این حالت کنترل های منو اطلاعات فایل و مسیرها را از فایل **sitemap** دریافت می کنند . این فایل ، نقشه مکان صفحات و فایل های وب سایت شماست .

**ASP.Net** دارای 3 نوع کنترل برای ایجاد منوها و سیستم های مسیریاب به شرح زیر می باشد :

- منوهای دینامیک ( کنترل . Menu )
- منوهای درختی ( کنترل . TreeView )
- منوهای مسیریاب ( کنترل . SiteMapPath )

در ادامه به توضیح و آموزش کار با این کنترل ها خواهیم پرداخت .

### مرحله اول - ایجاد فایل نقشه وب سایت : ( sitemap )

گفتیم که در **ASP.Net** ، می توانید مسیرها و فایل های اصلی سایت را در یک فایل نقشه سایت با پسوند **sitemap** . تعیین نمایید . این فایل به زبان **XML** نوشته شده و می تواند به عنوان منبع داده ای برای تمامی کنترل های منوی **ASP.Net** به کار رود .

کد زیر ، کد یک فایل ساده نقشه سایت را نشان می دهد . به آن دقت نموده ، سپس توضیحات لازم را ارائه خواهیم کرد :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<siteMap>
  <siteMapNode title="Home" url="~/Default.aspx">
    <siteMapNode title="Web Design" url="~/Web/Index.aspx">
```

کد

```

<siteMapNode title="HTML" url="~/HTML/Index.aspx"/>
<siteMapNode title="CSS" url="~/CSS/Index.aspx"/>
</siteMapNode>
</siteMapNode>
</siteMap>

```

قوانین ایجاد و تعیین مسیرها در یک فایل **sitemap** همانند کد فوق ، به شرح زیر است :

- تمامی مسیرها و تگ ها بایستی درون تگ اصلی **<SiteMap>** تعریف شوند.
- این فایل ها به **زبان XML** نوشته می شوند.
- تگ **<SiteMap>**، فقط می تواند یک عنصر **<SiteMapNode>** را به عنوان فرزند داشته باشد که معمولا به صفحه اصلی یا **Home** اشاره می کند.
- هر تگ **<SiteMapNode>** می تواند به تعداد دلخواه عنصر فرزند ( **Child Nodes** ) به صورت تگ درونی **<SiteMapNode>** داشته باشد.
- درون هر تگ **<SiteMapNode>**، عنوان لینک توسط خاصیت **title** و آدرس آن نیز توسط خاصیت **url** تعیین می شود.

**نکته :** فایل **sitemap** با نقشه فایل ، بایستی حتما در پوشه اصلی سایت ( **root** ) قرار بگیرد . همچنین هر سایت یک فایل نقشه می تواند داشته باشد .

### آموزش طراحی منوهای دینامیک به وسیله کنترل : **Menu**

از کنترل **<asp:Menu>** ، می توان برای ایجاد منوهای استاندارد در **ASP.Net** استفاده نمود . این منوها معمولا به صورت **Pop-Up** بوده و آیتم های اصلی آن ثابت بوده و سپس با قرار گرفتن موس بر روی آیتم های اصلی ، زیر منوها باز می شوند .

```

<asp:SiteMapDataSource id="nav1" runat="server" />
<form id="Form1" runat="server">
  <asp:Menu ID="Menu1" runat="server" DataSourceId="nav1" />
</form>

```

کد

خروجی

[Home](#) •

شکل کلی ایجاد این منو با استفاده از کنترل **Menu** و یک فایل **SiteMap** به صورت زیر است :

### توضیح کد :

در ابتدا یک کنترل `<asp:SiteMapDataSource>` را ایجاد کرده و `id` آن را تعیین می کنیم . این کنترل یک کنترل سرور داده است که می تواند به فایل نقشه سایت ( **SiteMap** ) متصل شده ، اطلاعات آن را خوانده و سپس به کنترل منوی مورد نظر انتقال دهد .

سپس کنترل **Menu** را تعریف کرده و با قرار دادن مقدار خاصیت `DataSourceId` آن برابر با `nav1` که `id` کنترل `asp:SiteMapDataSource` است ، آنها را به هم متصل کرده ایم . در این صورت این کنترل منو ، اطلاعات دریافتی از کنترل سرور داده خود را نمایش خواهد داد .


**نکته مهم :** کنترل `asp:SiteMapDataSource` به صورت اتوماتیک به فایل نقشه سایت ( `web.sitemap` ) که در پوشه اصلی سایت ( `root` ) قرار دارد ، متصل می شود .

### ایجاد منو با استفاده از کنترل : **TreeView**

کنترل `<asp:TreeView>` ، برای ایجاد یک منو با ساختار درختی استفاده می شود ، که آیتم های خود را با رعایت سلسله مراتب نمایش می دهد .

لینک هایی که در سطح پایین تر باشند ، به صورت پیش فرض نمایان بوده و با زدن آیکن های + یا - در کنترل ، می توان آنها را باز یا بسته نمود .

شکل کلی تعریف و ایجاد یک منو با کنترل **TreeView** به صورت زیر است :

کد	<pre>&lt;asp:SiteMapDataSource id="nav1" runat="server" /&gt; &lt;form id="Form1" runat="server"&gt;   &lt;asp:TreeView ID="Menu2" runat="server" DataSourceId="nav1" /&gt; &lt;/form&gt;</pre>
خروجی	

### توضیح کد مثال :

همانند مثال قبل ، کنترل `asp:SiteMapDataSource` را به عنوان فایل داده ای استفاده کرده تا اطلاعات فایل نقشه سایت ( `sitemap` ) را خوانده و به کنترل `asp:TreeView` انتقال دهد .

همچنین توسط خاصیت DataSourceId ، کنترل asp:TreeView را به کنترل asp:SiteMapDataSource متصل کرده ایم .

## آموزش ایجاد منو با استفاده از کنترل SiteMapPath

کنترل SiteMapPath منوی خود را به صورت یک سری آیتم پشت سر هم که مسیر خاصی را نمایش می دهند ، نشان می دهد . به وسیله این نوع منوها ، می توانید در هر لحظه بدانید در کجای سایت قرار داشته و همچنین با کلیک بر روی آیتم های قبل از آدرس صفحه جاری ، به مسیر پیموده شده قبلی خود باز گردید . برخلاف کنترل های Menu و TreeView ، کنترل SiteMapPath از کنترل asp:SiteMapDataSource به عنوان منبع داده ای استفاده نکرده و خود به صورت پیش فرض و اتوماتیک به فایل نقشه سایت متصل می شود .  
کد تعریف این کنترل به صورت زیر است :

کد	<pre>&lt;form id="Form1" runat="server"&gt;     &lt;asp:SiteMapPath ID="Menu3"     runat="server" /&gt; &lt;/form&gt;</pre>
----	---

## جلسه چهاردهم : آموزش آپلود کردن وب سایت روی هاست در دات نت پنل (WebSite Panel)

همیشه یکی از مشکلاتی که افراد پس از خرید هاست دارند این است که می پرسند:

- چطور دامنه را به هاست متصل کنیم؟(تنظیمات DNS)
- چطور سایتمان را در سرور هاست آپلود کنیم؟
- چگونه سایتمان را پیکربندی کنیم؟(بحث در فایل web.config)
- چطور یک دیتا بیس در هاست ایجاد کنیم؟
- چگونه محتویات دیتا بیس را از سیستم خود به هاست منتقل کنیم؟
- چطور دیتا بیس را به سایت متصل کنیم؟
- چطور ftp را فعال کنیم و چطور از آن استفاده کنیم؟

نحوه تنظیم DNS (روی هاست) تنظیم (DOMAIN NAME SERVER)

برای تنظیم دامنه خود روی سرور هاست ابتدا به سایتی که دامنه را خریداری کرده اید بروید، وارد حساب کاربری خود شده و دامنه خود را انتخاب کنید. سپس گزینه تغییر DNS را انتخاب کنید. 2 تا DNS برای دامنه خود set کنید.

اولی [cns5.my-hosting-panel.com](http://cns5.my-hosting-panel.com)

دومی [cns6.my-hosting-panel.com](http://cns6.my-hosting-panel.com)

اگر از قبل DNS دامنه شما روی سایت دیگری قرار داده شده است قبلی ها را پاک کنید و این 2 را وارد کنید. حال باید تا حداکثر 48 ساعت منتظر بمانید تا تغییرات اعمال شود. معمولا کمتر از 12 ساعت انجام میشود.

نحوه آپلود کردن سایت روی سرور هاست

از طریق **username** و پسوردی که به وسیله این شرکت به ایمیلتان ارسال شده به کنترل پنل خود وارد شوید روی **Domains** کلیک کنید، سپس **Add Domain** ، سپس روی **Domain** کلیک کنید. نام دقیق دامنه یتان را در کادر وارد کنید، فقط پشت **Create Web Site** تیک بزنید و دکمه **Add Domain** را فشار دهید. وبسایت شما ایجاد شده است. در این مرحله اگر تغییرات DNS به خوبی اعمال شده باشد می توانید سایت خود را در اینترنت ببینید. منتها یک صفحه قراردادی را می بینید با عنوان **UNDER CONSTRUCTION** که خالی می باشد. در هر مرحله از کار اگر مشکلی وجود داشت می توانید با ایمیل [support@iranhost24.com](mailto:support@iranhost24.com) با پشتیبانی سایت در ارتباط باشید. حالا وقت آن است که فایل های خود را آپلود کنید. از کنترل پنل روی **File Manager** کلیک کنید. سپس روی نام وب سایتان کلیک کنید و بعد به پوشه ی **wwwroot** کلیک کنید. اینجا همان جایی است که می توانید فایل هایتان را **upload** کنید. برای راحتی کار توصیه می کنیم فایل هایتان را به صورت **zip** آپلود کنید و سپس **unzip** کنید. سایتتان را امتحان کنید. اگر درست کار نمی کند، نگران نباشید احتمالا مشکل از تنظیمات **web.config** می باشد.

تنظیمات فایل **WEB.CONFIG** روی هاست

این فایل از مهمترین فایل های سایت شما می باشد که اگر بخوبی تنظیم نشده باشد سایت شما با **error** مواجه می شود. اگر سایت شما به صورت **local** یا محلی در سیستم کامپیوترتان کار می کند ولی در سرور مشکل دارد، ابتدا فایل **config** را از داخل سایت **edit** کنید و این سطر را داخل **<system.web>** اضافه کنید تا **error** سایت با توضیحات آن مشخص شود تا سر از کارتان در بیاورید:

```
<customErrors mode="Off"/>
```

حالا وقتی سایت را اجرا می کنید دقیقا ارور ها داخل مرورگر به نمایش در می آید. در این مرحله شاید عیب کارتان را پیدا کنید و آن را بر طرف کنید. همچنین باید اطمینان حاصل کنید که تگ های زیر را به **web.config** اضافه کرده اید:



```

<system.serviceModel>
<serviceHostingEnvironment aspNetCompatibilityEnabled="true" />
</system.serviceModel>
<system.webServer>
<directoryBrowse enabled="false"/>
<defaultDocument>
<files>
<clear />
<add value="default.aspx"/>
</files>
</defaultDocument>
</system.webServer>

```

اگر اضافه نکرده اید حتما همین الان اضافه کنید. دقت کنید در کد بالا باید به جای `default.aspx` نام صفحه ی اصلی سایتتان را اضافه کنید که می تواند اسامی مختلفی باشد مانند: `index.html, index.php, index.aspx, default.htm, default.asp & ...` این صفحه با ورود به سایتتان نمایش داده می شود. تا اینجای کار کلی به یک سایت ایده آل نزدیک شده اید. حالا وب سایتتان را تست کنید، اگر بازهم `error` گرفتید اصلا نگران نباشید. این طبیعیه، مگر یادتان رفته هنوز پایگاه داده یا دیتابیس را ایجاد و متصل نکرده اید؟

ایجاد دیتا بیس و مدیریت آن

ایجاد و مدیریت دیتا بیس در پنل `WebSite Panel` بسیار آسان می باشد. شما ابتدا از داخل پنل وارد قسمت دیتا بیس می شوید، یک دیتا بیس و یک کاربر دیتا بیس (`Create User`) ایجاد می کنید. وارد دیتا بیسی که ساخته اید می شوید، برای مدیریت آن روی `Browse Database` کلیک کنید. از اینجا میتوانید دیتا بیس را به ایجاد کنید یا از دیتا بیسی که قبلا ساخته اید `Script` بگیرید و به صورت `Query` وارد دیتا بیس جدید کنید. `ATTACH` کردن یا `SCRIPT` گرفتن از پایگاه داده (دیتا بیس)

شاید بارها در داخل ویندوز یک فایل دیتا بیس `Attach` کرده اید و از آن در برنامه هایتان استفاده کرده اید. اما در هاست این روش روش مناسبی نیست و اکثر مواقع با مشکلاتی روبه رو می شود. راه بهتر و آسانتر این است که از دیتابیس `Script` بگیرید. آن را در دیتا بیس سرور اجرا کنید. مراحل ایجاد و اجرای `Script` به اینصورت می باشد:

ابتدا از داخل پنل هاست به دیتا بیس مورد نظر وارد می شوید و `Browse Database` را کلیک می کنید. حالا درون قسمت مدیریت دیتا بیس هستید. گزینه ی `tools` و سپس `Query` را انتخاب می کنید، `Script` مورد نظر را در اینجا کپی میکنید و `Submit` را کلیک می کنید. اگر نمی دانید `Script` را باید از کجا بیابید که در اینجا کپی کنید به این طریق عمل کنید:

sqlServer را از داخل ویندوز باز کنید، روی دیتا بیس مورد نظر راست کلیک کنید، Tasks و سپس Generate scripts را انتخاب کنید، بقیه مراحل روتین است طی کنید تا Script ساخته شود و در یک پنجره به نمایش در آید. کار هنوز تمام شده نیست، در خط اول script جلوی use نام دیتا بیسی که در سرور ساخته اید را جایگزین کنید [USE نام دیتا بیس]

سپس از طریق find and replace تمام نوشته های [dbo] را با نام کاربری دیتا بیس در سرور جایگزین کنید [Database\_username] ---> [dbo] .

این script آماده ی کپی و اجرا شدن در سرور می باشد.  
نحوه متصل کردن دیتا بیس به سایت

با فرض این که دیتا بیس و سایت شما به صورت کامل در سرور مستقر می باشند باید یک ارتباط بین این دو برقرار شود که این کار بسیار آسان و با افزودن چند خط به web.config میسر می شود:

```
<connectionStrings>  
<add name="connectionStringName" connectionString="Data  
Source=204.93.178.157;  
Initial Catalog=DatabaseName; User ID=DatabaseUserName;  
Password=DatabaseUserPassword"  
providerName="System.Data.SqlClient" />  
</connectionStrings>
```

این تگ ها باید داخل تگ <configuration> باشند.  
طریقه ایجاد اکانت FTP و آپلود از طریق این پروتکل

ابتدا از داخل پنل مدیریت هاست یک اکانت ftp می سازیم. سپس با یکی از نرم افزار های ftp مانند FileZilla به راحتی فایل ها را upload می کنیم IP host. برای این اتصال 204.93.156.140 و port=21 می باشد.

```
<?xml version="1.0"?>  
<configuration>  
<system.web>  
<compilation debug="true" targetFramework="4.0"/>  
<httpRuntime targetFramework="4.0"/>  
<customErrors mode="Off"/>  
</system.web>  
</configuration>
```

در این راهکار قصد داریم تا نحوه قرار دادن یک کنترل **Menu** و تنظیم آن برای نمایش فهرست محتویات سایت **Asp.Net** را آموزش دهیم . یکی از امکانات مناسبی که در هر سایت باید قرار بگیرد ، منو یا فهرست موضوعات سایت است . به وسیله کنترل **Menu** در **Asp.Net** و بدون نوشتن کد خاصی به راحتی می توانید منوی مورد نظر خود را بسازید . کنترل **Menu** انواع حالت های نمایش منو را چه بصورت استاتیک و چه بصورت داینامیک در اختیار شما قرار می دهد .

در حالت منوی استاتیک ، تمام شاخه های اصلی منو و زیر شاخه های آن باز بوده و کاربر قادر به مشاهده و کلیک بر روی آنهاست .

اما در حالت داینامیک معمولا سرعنوان ها فقط نمایش داده شده و با حرکت موس بر روی هر سرعنوان ، زیر منوی آن بصورت **Pop-Up** نمایش داده می شود .

کنترل **Menu** این امکان را می دهد که ترکیبی از هر دو حالت استاتیک و داینامیک را نیز بکار ببرید .

اما نحوه تعیین کردن موضوعات برای نمایش در فهرست **Menu**؟! . شما می توانید منو ها و زیرمنوهای کنترل را بصورت از پیش تعیین شده در هنگام طراحی تعریف نمایید و یا اینکه کنترل را به یک منبع داده ای که دارای ساختار درختی و سلسله مراتبی است ، مثل **XmlDataSource** پیوند دهید . در این راهکار موارد زیر آموزش داده خواهند شد :

- ساخت یک منوی ساده و استاتیک و سپس اتصال هر عنوان به صفحه مربوطه .
- ساخت یک منوی پیشرفته تر که برای دریافت فهرست موضوعات خود به یک فایل **XML** به نام **Web.sitemap** متصل می شود .
- تنظیم جهت چیدمان و نمایش منو .
- تلفیق منو های استاتیک با داینامیک و نمایش در هنگام اجرا .

### پیش نیازها :

برای اجرای این راهکار نیاز دارید تا برنامه های زیر بر روی سایت شما نصب شده باشد :

1. **Microsoft Visual Studio** .
2. **Microsoft .NET Framework version 2.0** یا بالاتر ( همراه با ویژوال استودیو نصب می شود )

مرحله اول - طراحی یک سایت **ASP.Net** :

در مرحله اول از راهکار بایستی یک سایت ASP.Net را برای اجرای پروژه خود ایجاد نمایید. فرض بر این است که سایت را قبلاً ایجاد کرده و یا با نحوه اجرای آن آشنایی دارید، در غیر اینصورت به راهکار شماره 1-1: نحوه ایجاد یک وب سایت ASP.Net در ویژوال استودیو بروید.

## مرحله دوم - ساخت یک منوی ساده و ایستاتیک:

برای ساخت یک منوی ساده در صفحه مورد نظرمان مراحل زیر را انجام دهید:

1. صفحه مورد نظر را باز کرده و به حالت Design بروید.
2. از منوی Toolbox و بخش Navigation یک کنترل Menu را کشیده و بر روی صفحه قرار دهید.
3. در این مثال، قصد داریم تا یک منوی افقی طراحی نماییم. بنابراین کنترل Menu را انتخاب کرده و از قسمت Properties، مقدار خاصیت Orientation را بر روی Horizontal قرار دهید.

### تنظیم کردن منوی قرار داده شده بر روی صفحه:

در این بخش، بوسیله ویرایشگر Menu Item Editor، آیتم های کنترل منو را تنظیم می کنیم:

1. بر روی کنترل Menu راست کلیک کرده و سپس گزینه Edit Menu Items را انتخاب نمایید. پنجره Menu Item Editor، باز می شود.
  2. در زیر قسمت Items، آیکن Add a root item را انتخاب کنید.
  3. در زیر قسمت Properties آیتم جدید، مقدار خاصیت Text را بر روی Home و Navigate URL را به Default.aspx، تغییر دهید.
  4. مجدداً با کلیک بر روی گزینه Add a root item، یک آیتم جدید دیگر را به منو اضافه کنید.
  5. مقدار خاصیت Text آیتم جدید را بر روی Books و خاصیت Navigate آن را بر روی Books.aspx تعیین کنید.
  6. کار فوق را برای گزینه دیگر به نام Apps انجام داده و آدرس صفحه مقصد را نیز Apps.aspx تعیین نمایید.
- حال اگر به بخش Design صفحه دقت نمایید، آیتم هایی که برای کنترل Menu را تعیین کرده اید، مشاهده خواهید کرد.

7. صفحات Home.aspx، Books.aspx و Apps.aspx را به پروژه خود اضافه نمایید. حال زمان تست کردن منوی ساخته شده است، مراحل زیر را انجام دهید:

1. به صفحه اصلی یا Default.aspx رفته و کنترل های Ctrl + F5 را برای اجرای پروژه فشار دهید.

2. صفحه اجرا شده و کنترل Menu را با آیتم های تعیین شده بر روی صفحه مشاهده خواهید کرد.
3. موس را بر روی هر آیتم ببرید ، آدرس مقصد آیتم در منوی Status Bar مرورگر نمایش داده می شود . با کلیک بر روی هر گزینه به صفحه مقصد آن منتقل خواهید شد.

### مرحله سوم - ساخت یک کنترل Menu متصل به یک فایل نقشه سایت ( Site Map ) :

در بخش قبل ، یک کنترل Menu ساده را ایجاد و آیتم های آن را به صورت دستی در هنگام Design صفحه تعیین کردیم . در این بخش قصد داریم تا به جای تعیین آیتم های کنترل Menu از قبل ، آن را به یک فایل نقشه سایت به عنوان یک XmlDataSource متصل کنیم . این کار به کنترل این امکان را می دهد که ساختار و آیتم های خود را از یک فایل مجزا XML در هنگام اجرای صفحه دریافت نماید . شما می توانید هر زمان که بخواهید فایل XML مورد استفاده را بدون نیاز به کار با کنترل Menu ، آپدیت و به روز رسانی نمایید . سپس این تغییرات به صورت اتوماتیک در کنترل Menu اعمال می شود .

برای انجام این کار مراحل زیر را به ترتیب انجام دهید . در این مثال از یک کنترل Menu دوم استفاده می کنیم :

1. مجدد از منوی Toolbox و قسمت Navigation ، یک کنترل دیگر Menu را انتخاب کرده و بر روی صفحه قرار دهید.

2. اکنون ما به یک فایل نقشه سایت ( site map ) نیاز داریم . برای ایجاد این فایل مراحل زیر را انجام دهید:

- در منوی Solution Explorer ، بر روی نام وب سایت کلیک راست کرده و گزینه Add New Item را انتخاب کنید.

- از پنجره باز شده ، یک فایل Site Map را انتخاب کرده و بر روی گزینه Add کلیک نمایید.

3. (3) کد XML زیر را در فایل Site Map ایجاد شده قرار دهید . مطابق تصویر زیر:

```

<siteMap>
  <siteMapNode title="Home" description="Home" url="default.aspx" >
    <siteMapNode title="Products" description="Our products"
url="Products.aspx">
      <siteMapNode title="Hardware" description="Hardware choices"
url="Hardware.aspx" />
      <siteMapNode title="Software" description="Software choices"
url="Software.aspx" />
    </siteMapNode>
    <siteMapNode title="Services" description="Services we offer"
url="Services.aspx">
      <siteMapNode title="Training" description="Training classes"
url="Training.aspx" />
      <siteMapNode title="Consulting" description="Consulting services"
url="Consulting.aspx" />
      <siteMapNode title="Support" description="Support plans"
url="Support.aspx" />
    </siteMapNode>
  </siteMapNode>
</siteMap>

```

4. کد XML فوق ساختار و آیتم های کنترل Menu را تعیین می کند . تگ های SiteMap ode درون تگ <siteMapNode> اول ، بصورت عنصر فرزند یا زیر منو ، منوی اصلی نمایش داده می شوند.
5. (4) فایل Site Map را ذخیره نمایید.

### اتصال کنترل Menu به فایل : Site Map

اکنون نوبت آن است که یک کنترل داده وب مسیریاب مثل SiteMapPath را بر روی صفحه قرار داده و آن را به فایل Site Map متصل کنید . سپس کنترل Menu خود را نیز برای دریافت اطلاعات به کنترل سرور داده مسیریاب متصل نمایید .

برای این منظور مراحل زیر را انجام دهید :

1. صفحه اصلی مورد نظر خود را باز کرده و به حالت Design بروید.
2. بر روی گزینه Smart tag کنترل Menu کلیک نمایید تا پنجره Menu Tasks باز شود.
3. از منوی باز شده ، کنترل کرکره ای Choose Data Source را انتخاب کرده و گزینه New Data Source را انتخاب نمایید.

4. پنجره Data Source Configuration Wizard ، باز می شود . گزینه Site Map را انتخاب کرده و یک ID در قسمت Specify an ID for the data source ، برای آن تعیین کرده و Ok نمایش دهید .

پس از انجام مراحل فوق ، کنترل Menu به فایل (site map) متصل شده است . برای تست صفحه با زدن کلیدهای Ctrl + F5 آن را اجرا نمایید . صفحه باز شده و منو بر روی آن قابل مشاهده است . با حرکت موس بر روی منو آیتم های آن را مشاهده کرده و کلیک نمایید .

### نظیم درجات نمایش منو های استاتیک و داینامیک :

همانطور که گفتیم ، کنترل Menu دارای دو حالت برای نمایش منوها و زیرمنوهای خود است ( استاتیک و داینامیک )

در نمایش استاتیک ، تمام منوها و زیرمنوهای کنترل بصورت باز هستند و کل کنترل قابل مشاهده است . کاربر می تواند بر روی هر کدام از منوها نیز کلیک نماید . اما در حالت داینامیک ، فقط منوها یا زیرمنوهایی که طراح تعیین کرده در ابتدا نمایش داده می شوند و بقیه مخفی هستند . زیرمنوهای مخفی در هنگامی که کاربر اشاره گر موس را بر روی منوی Parent آن ببرد ، نمایش داده می شود .

کنترل Menu ای که در مثال قبل ساختیم ، بجز سر منوهای اصلی آن ، بقیه منوهای داینامیک و مخفی هستند . بوسیله کنترل Menu شما این توانایی را دارید که تعیین نمایید چه درجه ای از منوها و زیرمنوها ، استاتیک بوده و یا داینامیک باشند . بوسیله مراحل زیر این قابلیت را برای کنترل منوی خود تنظیم می کنیم :

1. بر روی کنترل Menu در صفحه کلیک کرده و به منوی Properties آن بروید .
2. مقدار خاصیت **StaticDisplayLevels** ، آن را به 2 تغییر دهید . در این حالت 2 درجه از منوهای کنترل ( سرمنوها و یک مرحله زیرمنوهای آنها ) بصورت استاتیک و بقیه درجات منوها ، مخفی بوده و به صورت داینامیک باز می شوند .
3. با اجرای صفحه ، تفاوت این حالت را در عمل مشاهده کنید .

## آموزش ASP.NET < کنترل های پیمایش < کنترل SITEMAPPATH

### کنترل : SiteMapPath

کنترل SiteMapPath یک مسیر نمایش از صفحه ابتدا تا صفحه جاری را بصورت یک لینک چند بخشی به کاربر نمایش می دهد .

```
<siteMap>
  <siteMapNode title="Home" description="Home" url="~/default.aspx" >
    <siteMapNode title="Services" description="Services we offer"
url="~/Services.aspx" />
    <siteMapNode title="Training" description="Training classes"
url="~/Training.aspx" />
```

شما می توانید با کلیک بر روی هر یک از بخش های این کنترل به یک مرحله قبلتر از صفحه جاری بروید .  
این کنترل امکانات زیادی برای تغییر شکل و تنظیم لینک ها در اختیار طراح قرار می دهد .

کنترل **SiteMapPath** اطلاعات خود را از یک فایل **SiteMapPath** دریافت میکند . فایل **SiteMapPath** فایلی است که شامل اطلاعات کلیه صفحات و مسیرها در سایت شما همراه با توضیحی مثل متن لینک , عنوان و آدرس مقصد ( URL ) می باشد .

ایجاد یک فایل **SiteMapPath** برای سایت خودتان بسیار مفید است زیرا می توانید از این فایل برای دادن اطلاعات به سایر کنترل های پیمایش **Asp .Net** استفاده کرد و همچنین آن را به عنوان راهنما برای موتورهای جستجو قرار دهید .

همچنین چنانچه تغییر در مسیر ها و فایل های سایتتان بوجود آید کافی است اطلاعات فایل **SiteMapPath** را به روز کنید .

در ورژن های قدیم تر **Asp** و یا زبان های دیگر چنانچه لینک یا آدرسی در سایت تغییر می کرد ، مجبور بودیم تمام لینک ها و صفحات که آن لینک در آن وجود دارد را اصلاح نماییم . اما با امکاناتی که کنترل های پیمایش **Asp.Net** دارند با تغییر لینک در فایل داده ای آدرس های فایل **SiteMapPath** این تغییر بصورت اتوماتیک در تمام سایت اعمال می شود .

کد زیر یک فایل نمونه **SiteMapPath** را نمایش می دهد :



```

<siteMapNode title="Consulting" description="Consulting services"
url="~/Consulting.aspx" />
  </siteMapNode>
</siteMapNode>
</siteMap>

```

شکل کلی قرار دادن یک کنترل Site Map بر روی صفحه بصورت زیر است :

```

< asp:SiteMapPath ID="SiteMapPath1" Runat="server"> </asp:SiteMapPath >

```

شما به راحتی و بدون هیچ کد نویسی خاصی می توانید به وسیله کنترل SiteMapPath مسیر پیمایش برای سایت خود بسازید .

توجه داشته باشید که به وسیله کنترل SiteMapPath کاربر می تواند مسیر عقب ( صفحاتی که قبلا مرور کرده و یا در هرم لینک های سایت در رده بالاتری هستند ) برود . اما به کاربر امکان حرکت به سمت جلو را نمی دهد . همواره آخرین لینک در کنترل SiteMapPath صفحه جاری است .

### کنترل : Menu

از کنترل Menu ، می توانید برای ایجاد منوهای ایستاتیک و یا دینامیک در صفحات ASP.Net استفاده نمایید . منوهای همانند لیست موضوعات در همین سایت و یا منویی از محصولات و ... . شما می توانید آیتم های موجود در کنترل منو را به صورت دستی برای آن تعیین کرده و یا با اتصالش به یک پایگاه یا فایل داده ای ، آیتم های آن منبع داده ای را توسط کنترل نمایش دهید . بدون نوشتن کد یا دستوری ، می توانید عملکرد این کنترل را به راحتی در محیط ویژوال استودیو تنظیم نمایید . شکل کلی تعریف و نمایش یک کنترل Menu در صفحه های ASP.Net به صورت زیر است :

```

<asp:Menu ID="Menu1" runat="server">
  <Items>
    <asp:MenuItem Text="طراحی وب">
      <asp:MenuItem Text="HTML" NavigateUrl="~/HTML/Index.aspx" />
      <asp:MenuItem Text="CSS" NavigateUrl="~/CSS/Index.aspx" />
    </asp:MenuItem>
    <asp:MenuItem Text="برنامه نویسی وب">
      <asp:MenuItem Text="ASP.Net"
NavigateUrl="~/ASP_NET/Standard/Index.aspx" />
      <asp:MenuItem Text="PHP" NavigateUrl="~/PHP/Index.aspx" />
    </asp:MenuItem>
  </Items>
</asp:Menu>

```

```
</asp:MenuItem>
</Items>
</asp:Menu>
```

طراحی وب

برنامه نویسی وب

## نکات مهم درباره کنترل **Menu**:

1. آیتم های کنترل **Menu** ، در مجموعه **< Items >** کنترل ، تعیین می شوند.
2. برای ساخت هر آیتم ، درون مجموعه **< Items >** کنترل ، یک تگ **< asp:MenuItem >** تعیین می کنیم.
3. عنوان هر آیتم ، توسط خاصیت **Text** ، درون تگ آن تعیین می شود.
4. آدرسی که در صورت کلیک بر روی هر آیتم ، کاربر به آن ارسال می شود را توسط خاصیت **NavigateUrl** تعیین می کنیم.

## نحوه قرار دادن یک کنترل **Menu** بر روی صفحه **ASP.Net**:

1. ابتدا یک صفحه **ASP.Net** در محیط **Visual Studio** ایجاد کرده و یا صفحه از قبل طراحی شده خود را باز کنید.
2. از منوی **Toolbox** و از قسمت کنترل های **Navigation** یک کنترل **Menu** را کشیده و روی صفحه قرار دهید.
3. همچنین می توانید به صورت مستقیم در قسمت کدنویسی **Source** صفحه کد مربوط به کنترل را تایپ نمایید.

## کنترل **TreeView**:

کنترل **TreeView** برای نمایش اطلاعات سلسله مراتبی مثل محتویات یک کتاب یا فهرست مطالب یک سایت به شکل درخت وار استفاده می شود .

## امکانات و قابلیت های کنترل **TreeView**:

کنترل **TreeView** دارای ویژگی ها و قابلیت های زیر می باشد :

- اتصال اتوماتیک به یک منبع داده ای که باعث می شود تا گره ها (Nodes) کنترل به یک منبع سلسله مراتبی مثل فایل XML متصل شوند.
- امکان ساخت یک منو و یا فهرست برای مطالب سایت با کمک از کنترل **SiteMapDataSource**
- متن گره ها می توانند نوشته عادی یا لینک به صفحات دیگر باشند.
- می توانید ظاهر کنترل را با استفاده از **Theme** و استایل های مختلف به شکل دلخواه درآورید.
- دسترسی حین اجرا و قابل برنامه ریزی به کنترل **TreeView** که این امکان را به شما می دهد تا گره ها و لینک های کنترل را بصورت دینامیک تعیین نمایید.
- امکان **Callback** کردن صفحه به سرور در هنگام لینک بر روی گره ها.
- قابلیت نمایش یک کادر انتخابی ( **CheckBox** ) در کنار هر گره.

شکل کلی استفاده از یک کنترل **TreeView** به صورت زیر است :

```
<asp:TreeView ID="TreeView1" runat="server">
  <Nodes>
    <asp:TreeNode Text="طراحی وب">
      <asp:TreeNode Text="HTML" NavigateUrl="~/HTML/Index.aspx" />
      <asp:TreeNode Text="CSS" NavigateUrl="~/CSS/Index.aspx" />
    </asp:TreeNode>
    <asp:TreeNode Text="برنامه نویسی وب">
      <asp:TreeNode Text="ASP.Net" NavigateUrl="~/ASP_NET/Standard/Index.aspx" />
      <asp:TreeNode Text="PHP" NavigateUrl="~/PHP/Index.aspx" />
    </asp:TreeNode>
  </Nodes>
</asp:TreeView>
```



نحوه قرار دادن یک کنترل **TreeView** بر روی صفحه **ASP.Net** :

1. ابتدا یک صفحه ASP.Net در محیط Visual Studio ایجاد کرده و یا صفحه از قبل طراحی شده خود را باز کنید.
2. از منوی Toolbox و از قسمت کنترل های Navogation یک کنترل TreeView را کشیده و روی صفحه قرار دهید.
3. همچنین می توانید به صورت مستقیم در قسمت کدنویسی Source صفحه کد مربوط به کنترل را تایپ نمایید.

## نحوه کار با کنترل :TreeView

کنترل TreeView نمایش چندین نوع اطلاعات را داراست :

1. اطلاعات استاتیک یا ثابت که در هنگام طراحی توسط طراح صفحه تعیین می شود.
2. اتصال به یک منبع داده ای و دریافت اطلاعات به آن.
3. اضافه شدن اطلاعات به صورت برنامه ریزی شده در حین اجرا بنا بر عملکرد کاربر.

### 1) نمایش اطلاعات ثابت یا استاتیک :

شما می توانید گره ها و لینک های کنترل TreeView از قبل و در هنگام طراحی تعیین نمایید . برای این منظور بایستی گره های مورد نظر خود را در مجموعه < nodes > کنترل تعیین کنید . هر گره توسط یک تگ < asp:TreeNode > ایجاد شده و یک فرزند کنترل TreeView محسوب می شود .

```

<asp:TreeView ID="TreeView1" runat="server">
  <Nodes>
    <asp:TreeNode Text="طراحی وب">
      <asp:TreeNode Text="HTML"
NavigateUrl="~/HTML/Index.aspx" />
      <asp:TreeNode Text="CSS" NavigateUrl="~/CSS/Index.aspx"
/>
    </asp:TreeNode>
  </Nodes>
</asp:TreeView>

```

طراحی وب

[HTML](#)

[CSS](#)

## 2) اتصال کنترل TreeView به یک منبع داده ای :

شما می توانید کنترل TreeView را به یک منبع داده ای که از خاصیت IHierarchicalDataSource پشتیبانی می کند مثل XmlDataSource متصل نمایید . در هنگام اتصال کنترل به یک منبع داده ای می توانید تعیین نمایید تا گره های کنترل به کدام فیلد بانک اطلاعاتی متصل شود . برای دریافت اطلاعات بیشتر به بخش اتصال کنترل TreeView به یک منبع داده ای بروید .

## 3) نمایش اطلاعات به صورت برنامه ریزی شده با استفاده از TreeNodesCollection :

شما می توانید اطلاعات کنترل TreeView را بصورت برنامه ریزی شده و در حین اجرا به کنترل ارسال نمایید . برای این منظور از خاصیت Nodes کنترل استفاده می شود . این خاصیت با استفاده از کلاس TreeNodesCollection کار می کند . برای دریافت اطلاعات بیشتر به بخش توضیح Nodes بروید .

## جلسه پانزدهم : آموزش دسترسی به پایگاه داده – ASP.NET

ASP.NET امکان دسترسی و استفاده از منابع داده ی زیر را به برنامه نویس می دهد:

MySQL), Oracle, SQL Server, Access.Databases (e.g.

XML documents

Business Objects

Flat files

ASP.NET فرایندهای پیچیده و سنگین دسترسی به داده را پنهان می کند و همچنین سطح بسیار بالاتری از کلاس ها و اشیاء که دسترسی به داده از طریق آن ها بسیار سهل است را ارائه می دهد. این کلاس ها تمامی کد نویسی های پیچیده که برای اتصال به داده، بازیابی داده، پرسمان از داده (data querying) و دستکاری داده ها انجام می شود را پنهان می کند.

ADO.NET یک تکنولوژی است که به مثابه ی یک پل ارتباطی بین اشیاء کنترلی ( control ASP.NET (object و منبع داده (data source) backend عمل می کند. در این فصل به نحوه ی دسترسی به داده و کار با آن به طور مختصر خواهیم پرداخت.

### بازیابی و نمایش داده

به منظور بازیابی و نمایش داده در ASP.NET به دو نوع data control احتیاج داریم:

**data source control** (کنترل منبع داده) – اتصال به داده، انتخاب و گزینش داده و دیگر کارها و

عملیاتی همچون صفحه بندی (paging) و ذخیره ی داده در حافظه ی پنهان (caching) را مدیریت می کند.

**data view control** (کنترل مقید سازی و نمایش داده) – داده را متصل کرده سپس نمایش می دهد،

همچنین امکان دستکاری و مدیریت داده ها را برای برنامه نویس فراهم می آورد.

به مفاهیم مقید سازی و اتصال داده (data binding) و کنترل های منبع داده ای ( data source control

در بخش های بعدی با جزئیات بیشتر خواهیم پرداخت. در این قسمت، برای دسترسی به داده از یک

کنترل SqlDataReader و از کنترل GridView جهت نمایش و مدیریت داده بهره می گیریم.

همچنین از پایگاه داده ی Access که دربردارنده ی اطلاعاتی درباره ی کتاب های NET. موجود در بازار است

استفاده خواهیم کرد. اسم پایگاه داده ASPDotNetStepByStep.mdb بوده و جدول داده ای ( data

table) که بکار می بریم را DotNetReferences نام گذاری کرده ایم.

جدول مورد نظر دارای ستون های زیر خواهد بود:

ID, Title, AuthorLastName, AuthorFirstName, Topic, Publishe و .

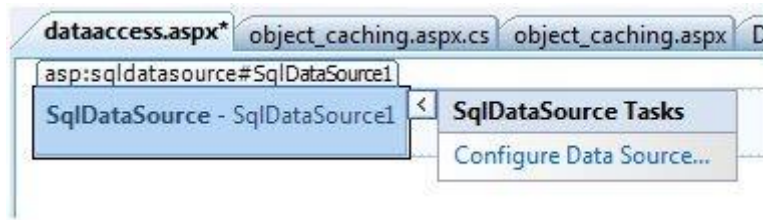
ذیل تصویری از جدول داده ای مذکور مشاهده می کنید:



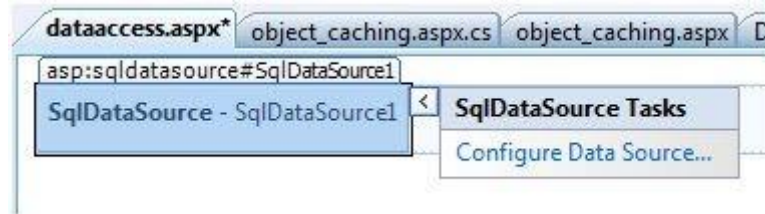
ID	Title	AuthorLastName	AuthorFirstName	Topic
1	Essentials .NET	Box	Don	Gestalt of .NET
2	Programming Microsoft Visual C...	Shepherd	George	C++ in the .NET World
3	ASP.NET Step by Step	Shepherd	George	ASP.NET from square one
4	Programming Microsoft ASP.NET	Esposito	Dino	ASP.NET comprehensive reference
5	Windows Forms Programming in	Sells	Chris	Windows UIs usin .NET
6	Applied Microsoft .NET Framework	Richter	Jeffrey	Comprehensive .NET refrence
7	.NET Compact Framework Progra	Yao	Paul	How to do .NET on small devices
8	.NET Framework Essentials	Thal	Thuan	How to do .NET development
9	Microsoft Visual Basic .NET Progra	MacDonald	Matthew	Digestible Visual Basic examples
10	Designing Microsoft ASP.NET App	Reilly	Douglas	ASP.NET Design topics
11	The C	Hejlsberg	Anders	Definitive C# Reference
12	Programming Windows with C	Petzold	Charles	The original Windows programming autho
13	The CLR Infrastructure Annotated	Miller	Jim	Info from someone really close to the CLR

اکنون مراحل زیر را طی می کنیم:

یک وب فرم ایجاد کرده، سپس کنترل `SqlDataSourceControl` را به آن اضافه کنید.



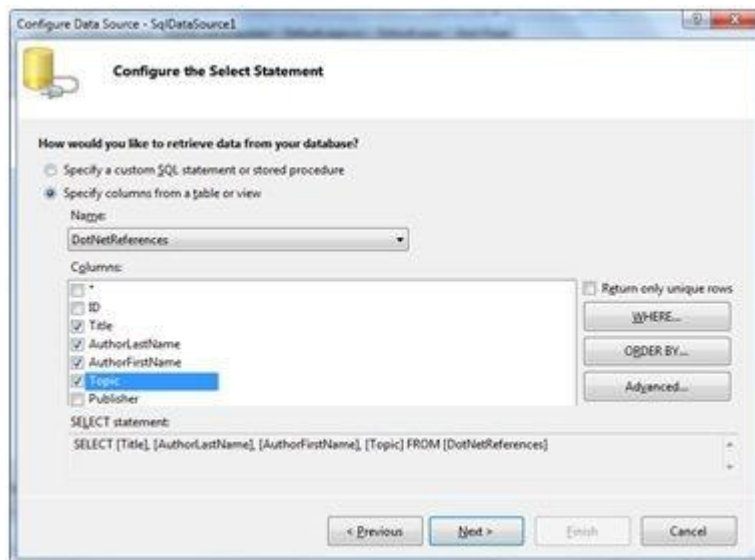
روی گزینه ی **Configure Data Source** کلیک کنید.



حال روی دکمه ی **New Connection** کلیک کرده تا اتصال با یک پایگاه داده برقرار گردد.

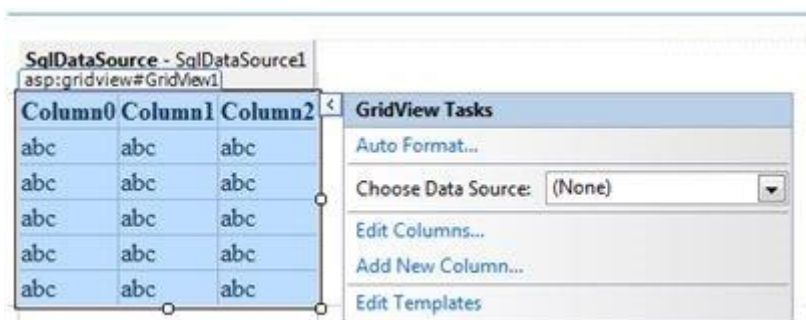


پس از اینکه اتصال با پایگاه داده برقرار شد، آن را برای استفاده در آینده ذخیره کنید. در مرحله ی بعدی از شما خواسته می شود دستور **Select** را پیکربندی (**config**) کنید:



اکنون ستون ها را انتخاب کرده و دکمه ی **next** را فشار دهید تا مراحل به پایان برسد. به دکمه های **WHERE**، **ORDER BY**، **Advanced** دقت کنید. دکمه های نام برده به شما امکان ارائه ی عبارت های **Where (clause)**، **Orderby** و همچنین تعریف دستورات **insert(command)**، **update** و **delete** اس کیو ال را به ترتیب فراهم می کند. از این طریق شما قادر خواهید بود داده ها را مدیریت یا دستکاری کنید.

کنترل **GridView** را به فرم اضافه کنید. منبع داده ای مورد نظر را انتخاب کرده و با استفاده از گزینه ی **AutoFormat** کنترل را قالب بندی (فرمت) کنید.

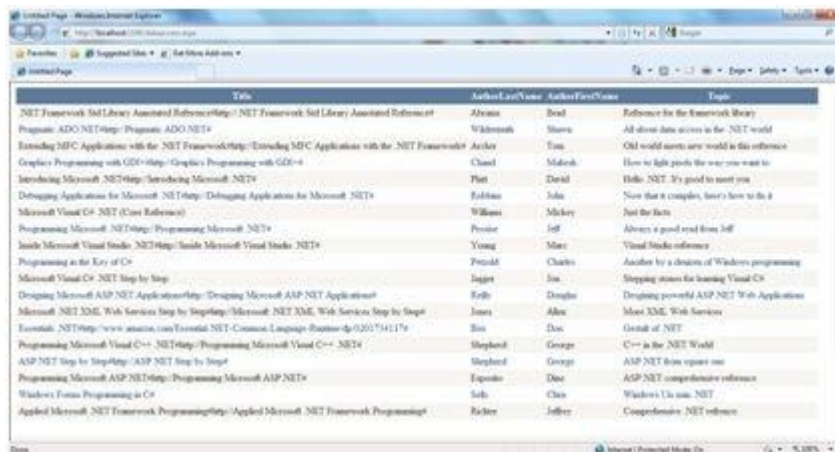


پس از انجام این کار، کنترل **GridView** قالب بندی شده عنوان های ستون ها را نمایش می دهد. اکنون برنامه آماده ی اجرا است.



SqlDataSource - SqlDataSource1			
Title	AuthorLastName	AuthorFirstName	Topic
abc	abc	abc	abc
abc	abc	abc	abc
abc	abc	abc	abc
abc	abc	abc	abc
abc	abc	abc	abc

برنامه را اجرا کنید.



Content file کد

```
<% @ Page Language="C#" AutoEventWireup="true" CodeBehind="dataaccess.as
px.cs" Inherits="DataAccess.dataaccess" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<html xmlns="http://www.w3.org/1999/xhtml" >
```

```
<head runat="server">
<title>
Untitled Page
</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString
="<%"$ ConnectionStrings:ConnectionString5 %>" ProviderName="<%"$
```

```

ConnectionStrings:ConnectionString5.ProviderName %>" SelectCommand=
"SELECT * FROM [EEE]"></asp:SqlDataSource>
<asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False" CellPadding="3"
DataSourceID="SqlDataSource1" BackColor="#DEBA84" BorderCol
or="#DEBA84" BorderStyle="None" BorderWidth="1px" CellSpacing
="2">
<RowStyle BackColor="#FFF7E7" ForeColor="#8C4510" />
<Columns>
<asp:BoundField DataField="Title" HeaderText="Title"
SortExpression="Title" />
<asp:BoundField DataField="AuthorLastName"
HeaderText="AuthorLastName" SortExpression="AuthorLastN
ame" />
<asp:BoundField DataField="AuthorFirstName"
HeaderText="AuthorFirstName" SortExpression="AuthorFirstN
ame" />
<asp:BoundField DataField="Topic"
HeaderText="Topic" SortExpression="Topic" />
</Columns>
<FooterStyle BackColor="#F7DFB5" ForeColor="#8C4510" />
<PagerStyle
ForeColor="#8C4510" HorizontalAlign="Center" />
<SelectedRowStyle BackColor="#738A9C"
Font-Bold="True" ForeColor="White" />
<HeaderStyle BackColor="#A55129" Font-Bold="True"
ForeColor="White" />
<SortedAscendingCellStyle BackColor="#FFF1D4" />
<SortedAscendingHeaderStyle BackColor="#B95C30" />
<SortedDescendingCellStyle BackColor="#F1E5CE" />
<SortedDescendingHeaderStyle BackColor="#93451F" />
</asp:GridView>

</div>
</form>
</body>
</html>

```



در این مقاله قصد آموزش چگونگی ایجاد یک **table adapter** و **dataset** در سی شارپ را دارم. در تجربه هایی که داشته ام ، **dataset** همیشه آزار دهنده بوده است ، زیرا برای تغییر یک خاصیت در دیتابیس باید جدول ها و **table adapter** ها را در فایل **xds** آپدیت کنید و نسخه ی جدید نرم افزار را انتشار دهید.

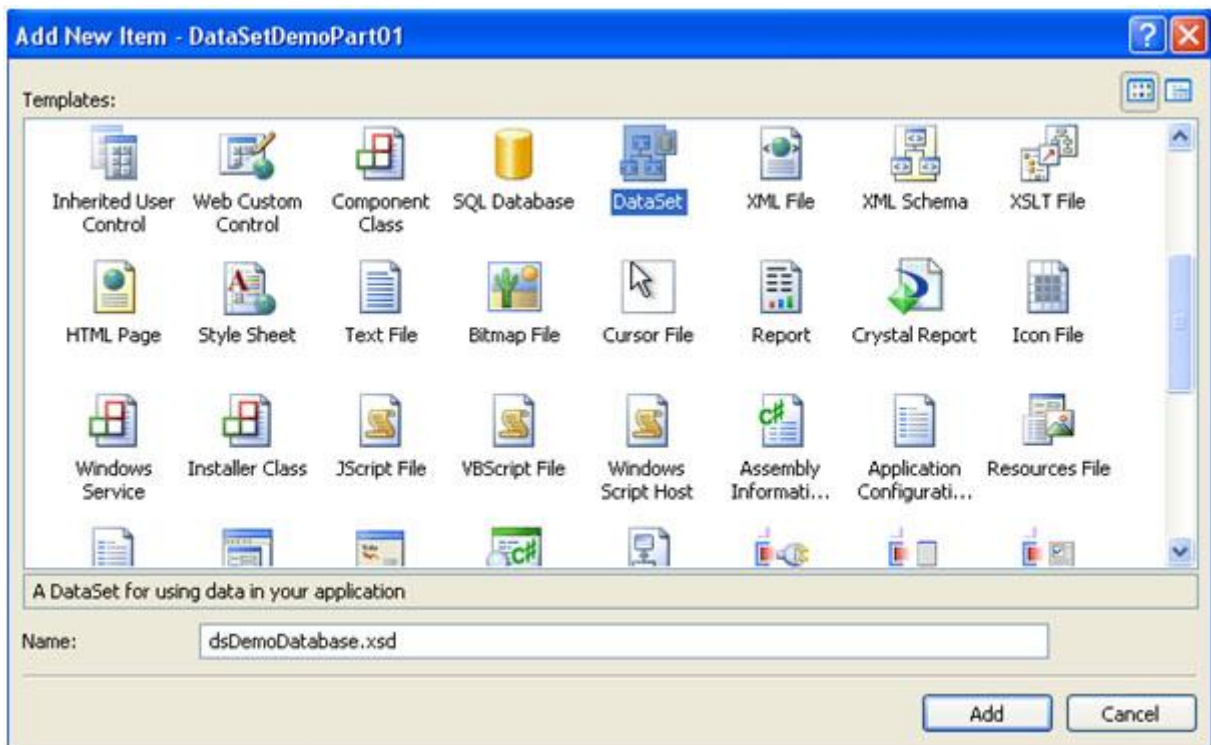
اما ایجاد **connection** و **query** ها با **dataset** آسان تر و سریع تر از ایجاد یک کلاس **Connection Manager** است ، (اما همیشه این بهترین کار نیست). اگر قصد ساخت یک برنامه ای دارید که شامل **Data Grid View** میشود و نیاز به افزودن ، آپدیت و حذف در آن است ؛ **dataset** بهترین گزینه است.

### ایجاد dataset و table adapter

۱ چارچوب خود را ایجاد کنید.

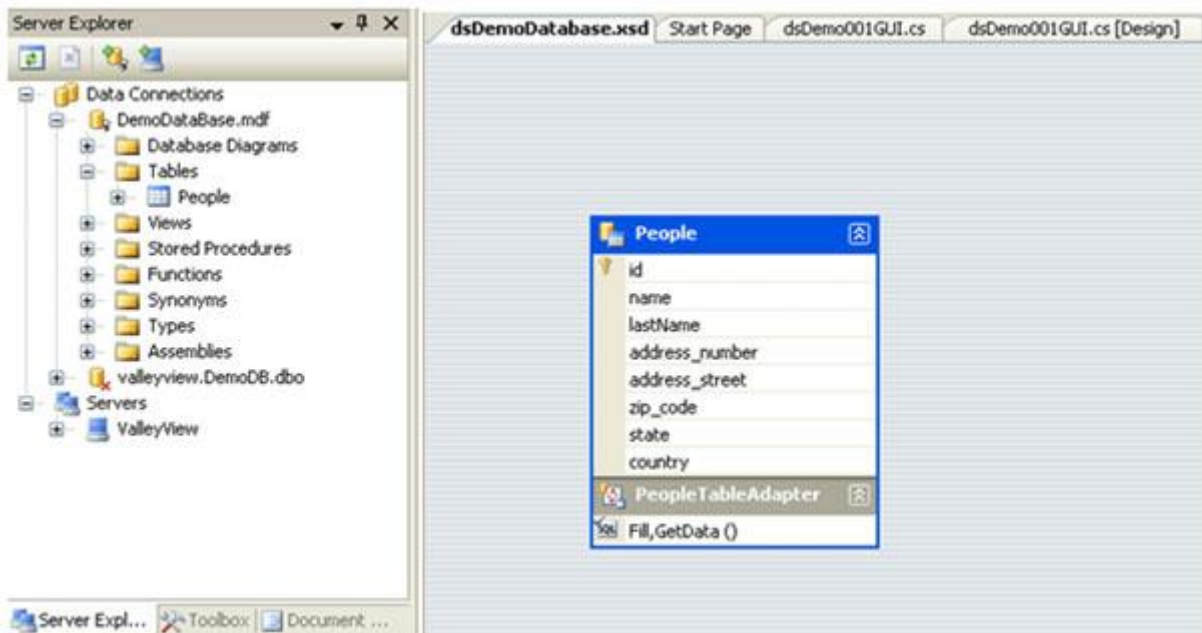
۲ یک کنترل **Data Grid View** به برنامه اضافه کنید.

۳ یک فایل **Dataset (xds)** ایجاد کنید.



۴ فایل xds خود را باز کنید.

۵ جدولی که قصد دارید با آن کار کنید را از قسمت Server explorer بکشید. جدول و table adapter شما ایجاد خواهد شد.



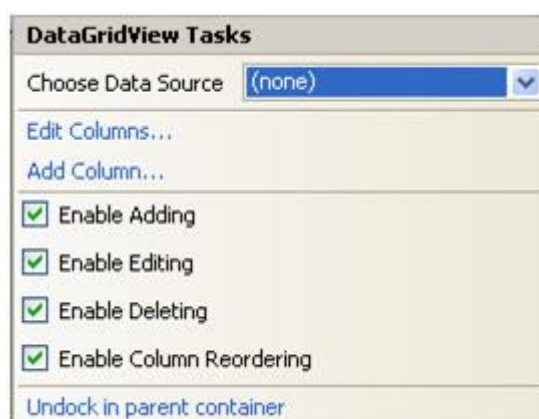
اگر بر روی table adapter کلیک کنید و مشخصات آن را چک کنید ، می‌توانید ببینید که دستورات Select ، insert ، update و delete به صورت اتوماتیک تولید شده اند.

به این بستگی دارد که جدول تان چگونه طراحی شده است. زمانی که من با یک جدول بدون کلید اصلی کار میکردم ، دستور update تولید نشده بود. میتوانید دستور را اضافه کنید ، اما زمانی که موقع update یک مورد در data grid view پیش بیاید ، ممکن است مشکلاتی پیش بیاید و update صورت نگیرد.

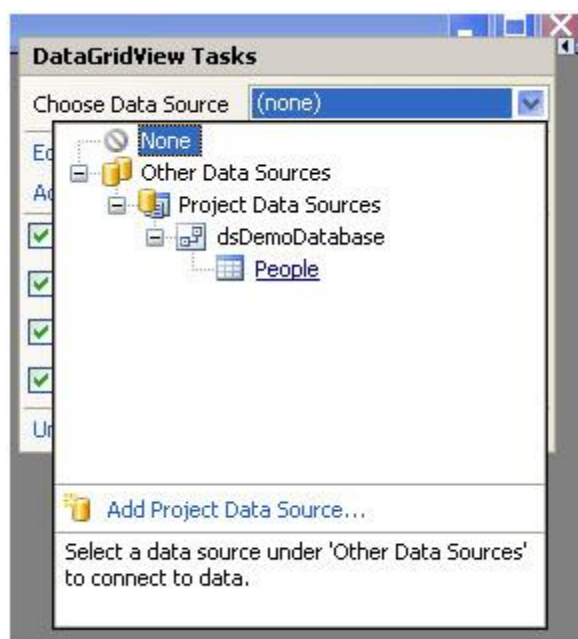
اگر میخواهید با این مشکلات رو به رو نشوید ، یک کلید اصلی در هنگام طراحی به جدولتان در دیتابیس اختصاص دهید.

حال قطعه کدی که به صورت اتوماتیک در رویداد load form ایجاد شده است را چک کنید. این قطعه کد data grid view را با داده های موجود در جدولتان پر میکند.

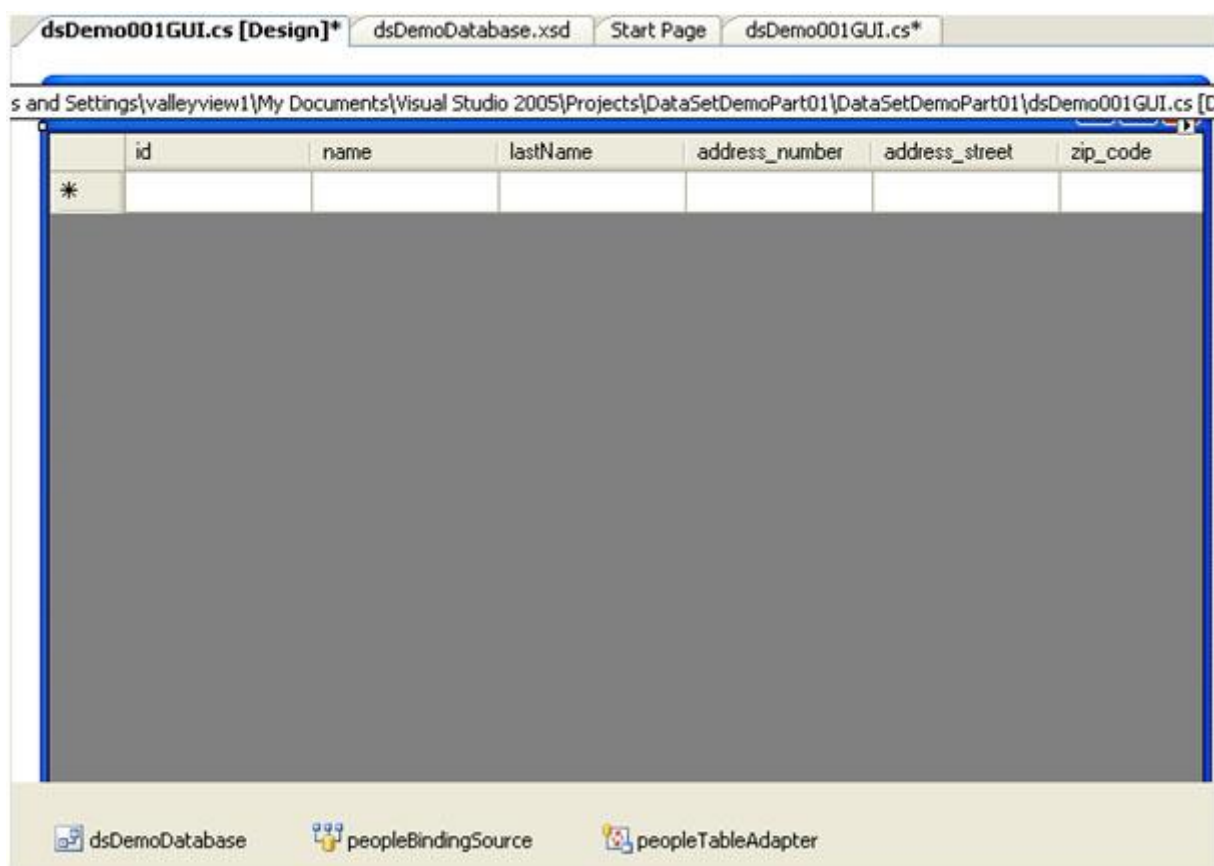
۶ برای دسترسی به خصوصیات update.insert و delete در کنترل data grid view خود بر روی دکمه ی کنترل option کلیک کنید.



۷ در همان قسمت ، Data Source و Data Table خود را انتخاب کنید.



حال ستون های data grid view به صورت اتوماتیک ساخته میشود. همچنین اشیای table adapter ، اشیای Binding Source و dataset .



همچنین در رویداد Load Form این کد برای پر شدن با اطلاعات جدول در هر بار لود شدن فرم window استفاده میشود.

```
private void dsDemo001GUI_Load
{
// TODO: This line of code loads
this.peopleTableAdapter
}
```

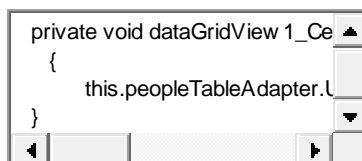
```
private void dsDemo001GUI_Load(object sender, EventArgs e)
{
// TODO: This line of code loads data into the 'dsDemoDatabase.People'
table. You can move, or remove it, as needed.
    this.peopleTableAdapter.Fill(this.dsDemoDatabase.People);
}
```

۸ حال باید یک button که برای حذف مورد انتخاب شده است ، اضافه کنید.

۹ حال وقت آن رسیده است که از رویداد برای آپدیت موارد از **data grid view** استفاده کنید. **Grid view** را انتخاب کنید و به بخش مشخصات رویداد بروید.

۱۰ رویداد **CellEndEdit** را ایجاد کنید.

۱۱ در قسمت کد به متد رویداد بروید و کد زیر را تایپ کنید.

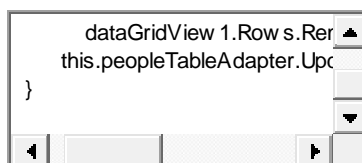


```
private void dataGridView1_Ce
{
    this.peopleTableAdapter.U
}
```

```
private void dataGridView1_CellEndEdit(object sender,
DataGridViewCellEventArgs e)
{
    this.peopleTableAdapter.Update(dsDemoDatabase);
}
```

حال برای تنظیم **grid view** خود احتیاج دارید که **Add Row** مربوط به **Allow User** را **True** کنید.

حال یک دکمه ی رویداد کلیک ایجاد کنید و کد زیر را در آن وارد کنید :



```
dataGridView1.Rows.Remove
this.peopleTableAdapter.Up
}
```

```
private void button1_Click(object sender, EventArgs e)
{
    foreach (DataGridViewRow row in dataGridView1.SelectedRows)
        dataGridView1.Rows.Remove(row);
    this.peopleTableAdapter.Update(dsDemoDatabase);
}
```

دکمه سطر انتخاب شده از **data grid view** را حذف میکند و سپس متد **Update** را از شیء **Table Adapter** فراخوانی میکند. همچنین خاصیت انتخاب **grid view** به انتخاب همه ی سطر ها تغییر میکند تا از بروز استثنا در زمان حذف سطر جلوگیری کند.



بسته به کاری که در حال انجامش هستید ، **Table Adapter** متد **Update** را فراخوانی میکند و جدول در **Dataset** و **base** را آپدیت میکند. اگر میخواهید سطر ها را حذف کنید ، **delete query** را فراخوانی کنید ؛ اگر میخواهید سطر اضافه کنید ، **insert query** را فراخوانی کنید و اگر میخواهید مقادیر سلول های موجود را تغییر دهید ، **query Update** را فراخوانی کنید.

### جمع بندی

برای استفاده از **Dataset** در یک جدول باید یک جدول با طراحی در دیتابیس خود داشته باشید (شامل یک کلید اصلی و افزودن خودکار ستون).

**Dataset** میتواند مقدار زیادی از زمان برای ایجاد **Connection string** و **queries** را برایتان ذخیره کند.

در این مطلب آموزشی شما میتوانید تمامی پروژه های برنامه نویسی به زبان **#C** را خودتان انجام بدین

در برنامه نویسی تنها مطلب مهم برقراری با پایگاه داده که **SQL** است و همچنین که بتوانید دستوراتی مانند **Insert**، **update** و **Delete** را توسط فرم برنامه خود به دیتابیس اعمال نمایید

تعریف : به مجموعه کامپوننت هایی که برای دسترسی به داده های بانک اطلاعاتی در **NET** استفاده می شود **Ado.Net** گفته می شود .

می توانیم بگوییم که دو روش برای اتصال به بانک وجود دارد اتصال متصل (**online**) و غیر متصل .

کار با بانک اطلاعاتی بصورت متصل :

هنگام استفاده از اشیا و متد های مربوط به این نوع اتصال ارتباط بین فرم و بانک در تمام مدت باید برقرار باشد برای همین موضوع این نوع کار با بانک متصل می نامند. معمولا این نوع کار با بانک از سرعت بسیار بالاتری نسبت به روش دوم که غیر متصل نام دارد برخوردار است:

کلاسهای ارتباط با بانکهای اطلاعاتی :

توجه - در اینجا در مثالها از بانک اطلاعاتی **SqlServer** استفاده میشود.

برای استفاده از این کلاس ها باید فضای نام زیر را با استفاده از **using** به برنامه اضافه کرد .



System.Data.SqlClient ;

1- کلاس SqlConnection :

این کلاس وظیفه برقراری ارتباط بین برنامه و بانک اطلاعاتی را بر عهده دارد .  
- هنگامی که می خواهید یک نمونه از آن کلاس را ایجاد کنید باید پارامتری را به نام **Connection String** به آن ارسال کنید .  
**Connection String** رشته ای است که شامل تمام داده های مورد نیاز برای برقراری اتصال به یک بانک اطلاعاتی می شود .  
ویژوال استودیو با استفاده از ویزارد **AddConnection** و اطلاعاتی که کاربر وارد می کند چنین رشته ای را ایجاد کرده و در اختیار **SqlConnection** قرار می دهد .  
اغلب بهتر است که متن لازم برای **ConnectionString** را خودمان بنویسیم که به صورت زیر است :

**“Data Source = local;Initial Catalog = university;Integrated Security = True”**

در کد بالا: **local** نام سروری است که بانک بر روی آن قرار دارد که در اینجا چون سرور خود کامپیوتر ماست مقدار آن را **local** قرار داده ایم که می توانیم به جای آن از “ (نقطه) هم استفاده کنیم .  
**University** نام بانکی است که قرار است ما با اطلاعات آن کار کنیم .  
-متن **ConnectionString** به صورت پارامتر به شیء جدید **SqlConnection** فرستاده می شود به صورت زیر:

```
SqlConnection Con = New SqlConnection( "Data Source = local;Initial Catalog  
= university;Integrated Security = True");
```

در کد بالا : Con یک نمونه جدید از نوع SqlConnection است که برای استفاده از آن آن را ساخته ایم .

متصل و قطع کردن اتصال به یک بانک اطلاعاتی :

با استفاده از متدهای Open و Close در کلاس SqlConnection به بانک متصل شده و یا اتصال خود را قطع کنیم .

```
Con.Close();وCon.Open();
```

2- کلاس SqlCommand :

این کلاس حاوی یک دستور Sql برای اجرا بر روی داده های دریافت شده از بانک اطلاعاتی است این دستور می تواند یک دستور SELECT برای انتخاب داده هایی خاص ، یک دستور INSERT برای درج داده های جدید در بانک اطلاعاتی ، یک دستور DELETE برای حذف داده ها از بانک اطلاعات و یا حتی فراخوانی یک پروسیجر ذخیره شده در بانک اطلاعاتی می باشد .  
ایجاد آن به صورت زیر می باشد :

```
SqlCommand Cmd = New SqlCommand();
```

نکته – در برنامه های بانک اطلاعاتی معمولاً از اشیای ایجاد شده از کلاس SqlCommand به تنهایی استفاده نمی کنند بلکه آنها را همراه با DataSet ها و DataAdapter ها به کار می برند .  
همچنین اشیای SqlCommand می توانند به همراه اشیای ایجاد شده از کلاس SqlDataReader مورد استفاده قرار گیرند .

خاصیت connection کلاس SqlCommand :

قبل از اینکه بتوانیم از یک شیء از کلاس SqlCommand استفاده کنیم باید بعضی از خاصیت های آن را تنظیم

کنیم اولین خاصیتی که باید تنظیم شود خاصیت Connection است .  
این خاصیت می تواند یک مقدار از نوع SqlConnection را دریافت کند :

```
Cmd .Connection = Con ;
```

توضیح کد بالا :

Cmd همان شیئی است که از کلاس SqlCommand قبلا ساخته ایم .

Con شیئی است که از نوع SqlConnection قبلا ساخته ایم .

خاصیت CommandText کلاس SqlCommand :

خاصیت بعدی که باید از کلاس SqlCommand تنظیم شود خاصیت CommandText است .

این خاصیت متنی را دریافت می کند که می تواند حاوی یک دستور Sql و یا فراخوانی یک پروسیجر ذخیره شده در بانک اطلاعاتی باشد که باید روی داده ها اجرا شود .

متد ExecuteNonQuery :

این متد دستورات را بر روی بانک اطلاعات اجرا می کند .

برای استفاده از این متد باید ابتدا اتصال خود را به بانک اطلاعاتی برقرار کنید سپس با فراخوانی این متد دستور موجود در شیء Command را اجرا کنید .

کلاس SqlDataReader :

از طریق این کلاس می توانیم اطلاعات را از بانک دریافت کنیم . که به صورت زیر تعریف می شود :

```
SqlDataReader dr;
```

متد ExecuteReader :

برای دریافت اطلاعات از بانک از متد ExecuteReader شیء تقاضا استفاده می کنیم ، یک شیء از نوع

SqlDataReader تعریف کرده و مقدار ExecuteReader را برابر آن قرار می دهیم

```
SqlDataReader dr = Cmd.ExecuteReader();
```

سپس از طریق متد Read شی DataReader اطلاعات را در یک حلقه واکشی می کنیم .

ارائه مراحل کلی یک ارتباط و مثال

برای تقاضا از یک جدول در بانک اطلاعاتی باید مراحل زیر طی شود :

1- اضافه نمودن فضا نام های مورد نیاز : برای ارتباط با بانک های اطلاعاتی به فضا نام System.Data و همچنین فضا نام System.Data.SqlClient برای کار با دیتابیس SQLServer و فضا نام System.Data.OleDb برای کار با بانک Access نیازمندیم .

2- تعیین رشته اتصال (ConnectionString) : رشته اتصال رشته شامل تنظیماتی جهت اتصال به بانک اطلاعاتی می باشد ، این رشته برای هر بانک متفاوت خواهد بود

3- تعیین شی اتصال (Connection Object) : کلاسی است برای برقراری ارتباط با بانک اطلاعاتی است ، این شی که از کلاس DbConnection ارث بری می کند اعمالی مانند باز و بسته کردن اتصال با بانک را از طریق رشته اتصال انجام می دهد.

4- تعیین رشته تقاضا (Text Command) : همان دستورات SQL است که جهت یک تقاضا ارائه می گردد ، این تقاضا جهت دریافت اطلاعات (Select) درج اطلاعات (Insert) ، ویرایش اطلاعات (Update) و یا حذف اطلاعات (Delete) یا... صورت می گیرد .

5- تعیین شی تقاضا (Command Object) : کلاسی است جهت ارسال و دریافت تقاضا از طریق شی اتصال به بانک اطلاعاتی

6- باز کردن اتصال

7- اجرای درخواست

8- دریافت اطلاعات (در صورتی که تقاضا Select باشد)

9- بستن اتصال

- «درج ، حذف و بروز رسانی» برای درج ، حذف و بروز رسانی به صورت متصل و مستقیم از متد

ExecuteNoneQuery مربوط به شی Command استفاده می شود . این شی درخواست مربوط به

insert,delete,update یا هر درخواست دیگری را بدون هیچ درخواستی انجام میدهد و خروجی آن تعداد

سطر های تحت تاثیر درخواست می باشد.

```

string connectionString = "Data Source=(local);Initial
Catalog=university;Integrated Security=true;";
SqlConnection Con = new SqlConnection(connectionString);
string commandText = "insert into student(name,family) values('ali','arefi')";
SqlCommand Cmd = new SqlCommand(CommandText, Con);
Con.Open();
Cmd.ExecuteNonQuery();
Con.Close();

```

معرفی خاصیت ها و دو کد نمونه Placeholder ها: متغیرهایی هستند که در یک دستور Sql قرار می گیرند و می توانند در زمان اجرای برنامه جای خود را با عبارتی خاص عوض کنند این متغیرها با علامت @ در یک دستور مشخص می شوند . و هنگامی که از آنها در یک دستور Sql استفاده کنیم قبل از اجرای دستور باید تمامی آنها را با مقادیر مناسب تعویض کنیم . که این کار به صورت اتوماتیک توسط برنامه در زمان اجرای دستور انجام می شود. اما باید پارامترهایی را ایجاد کرده و آن را در لیست parameters در شیء ایجاد شده از کلاس SqlCommand قرار دهیم تا برنامه بداند هنگام اجرای دستور هر placeholder را باید با مقدار چه متغیری در برنامه عوض کند . نکته – هیچ ضرورتی ندارد که نام یک placeholder همانم فیلدی باشد که قرار است مقدار placeholder در آن قرار بگیرد . خاصیت parameters کلاس SqlCommand: برای دسترسی به لیست پارامترهایی که در یک شیء از کلاس SqlCommand وجود دارد می توانیم از خاصیت parameters در این کلاس استفاده کنیم . این خاصیت حاوی لیستی از placeholder ها به همراه متغیرهای وابسته به آنها است بنابراین در کد قبل از اجرای دستور ، باید به وسیله ی این لیست مشخص کنیم که هر placeholder با مقدار چه متغیری باید تعویض شود . مثالی از درج رکورد در بانک اطلاعات : در این کد می خواهیم در جدول student از بانک university ، یک رکورد اضافه کنیم این جدول شامل سه فیلد می باشد که اطلاعات آن توسط کاربر در TextBox ها وارد می شود و برنامه با گرفتن این اطلاعات آنها را در جدول بانک ذخیره می کند .

```

Strcon= "Data Source = (local);Initial Catalog = university;Integrated Security
= True"; SqlConnection Con = New SqlConnection(Strcon);

```

```

SqlCommand Cmd = New SqlCommand();

Cmd.Connection = Con;

Cmd.CommandText= " insert into student
id=@id,lname=@lname,fname=@fname";
Cmd.Parameters.AddWithValue("@id",TextBox1.Text);

Cmd.Parameters.AddWithValue("@lname",TextBox2.Text);

Cmd.Parameters.AddWithValue("@fname",TextBox3.Text);

Con.Open();

Cmd.ExecuteNonQuery();

Con.Close();

```

توضیح کد بالا : در خط 1 یک متغیر از نوع **String** تعریف کردیم و راطلاعات اتصال به بانک را در آن قرار دادیم (connectinString) .خط 2 : شیئی از نوع **SqlConnection** ساختیم و رشته **StrCon** را به عنوان پارامتر به آن فرستادیم .خط 3 : شیئی از نوع **SqlCommand** ساختیم .خط 4 : خاصیت **Connection** کلاس **SqlCommand** را برابر شیئی ساخته شده از کلاس **SqlConnection** قرار دادیم .خط 5 : دستور اجرایی **Sql** را به شیئی **SqlCommand** نسبت دادیم (با استفاده از خاصیت **CommandText** ) .خط 6 ، 7 ، 8 : با استفاده از خاصیت **Parameters** کلاس **SqlCommand** به **placeholder** ها مقدار دادیم .خط 9 : اتصال به بانک را برقرار می کنیم .خط 10 : متد اجرایی **ExecuetNonQuery** را اجرا می کنیم .خط 11 : اتصال برنامه با بانک را قطع می کنیم .مثالی از اصلاح ( **update** ) اطلاعات یک رکورد :حالا می خواهیم اطلاعات یک رکورد از جدول **student** را اصلاح کنیم و تغییرات را ثبت نماییم . برای این منظور جدول مورد نظر دارای یک کلید است که می توان با استفاده از آن به تمامی اطلاعات رکورد موردنظر دسترسی پیدا کرد . در این جدول فیلد **id** (شماره دانشجویی) کلید جدول است .

```

Strcon= "Data Source = (local);Initial Catalog = university;Integrated Security
= True";

```

```

SqlConnection Con = New SqlConnection(Strcon);
SqlCommand Cmd = New SqlCommand();
Cmd.Connection = Con;

Cmd.CommandText= "update student Iname=@Iname,fname=@fname where
id=@id";

Cmd.Parameters.AddWithValue("@id",TextBox1.Text);

Cmd.Parameters.AddWithValue("@Iname",TextBox2.Text);

Cmd.Parameters.AddWithValue("@fname",TextBox3.Text);

Con.Open();

Cmd.ExecuteNonQuery();
Con.Close();

```

در توضیح کد بالا باید این رو بگم که تمام مراحل آن مانند کد insert می باشد به جز دستور Sql که در خط 5 آمده است و در اینجا update می باشد .

## کلاس SqlDataAdapter

کلاس SqlDataAdapter: این کلاس در برنامه های بانک اطلاعاتی ، همانند پلی بین جداول اطلاعاتی و نیز داده های موجود در حافظه که به وسیله ی DataSet نگهداری می شوند ، عمل می کنند . و برای استفاده از آن در برنامه باید یک شیئی از نوع آن ساخته شود .

```
SqlDataAdapter da = New SqlDataAdapter ();
```

این کلاس برای دسترسی به بانک اطلاعاتی از شیء ایجاد شده از کلاس `SqlCommand` ای که به آن نسبت داده می شود استفاده می کند. و برای دسترسی به بانک اطلاعات از کلاس `SqlCommand` و `SqlConnection` استفاده می کند.

```
da . SelectCommand = New SqlCommand();
```

خاصیت `SelectCommand` :

کلاس `SqlDataAdapter` دارای خاصیتی این خاصیت است. خاصیت `SelectCommand` حاوی شیئی از نوع `SqlCommand` است که از دستور موجود در آن شیء برای دریافت داده های موردنیاز در برنامه از بانک اطلاعاتی به کار می رود یعنی `SqlDataAdapter`، دستوری را که در خاصیت `SqlCommand` نگهداری می شود را روی بانک اطلاعاتی اجرا کرده و نتایج آن را در کلاس هایی مانند `DataSet` و یا `DataTable` قرار می دهد تا در برنامه مورد استفاده قرار گیرند. علاوه بر این، کلاس `SqlDataAdapter` دارای خاصیت هایی به نام `InsertCommand`، `DeleteCommand` و `UpdateCommand` است که به هر یک شیئی از نوع `SqlCommand` را قبول می کنند و `DataAdapter` از دستور ذخیره شده در هر یک از آنها به ترتیب بای حذف، درج و ویرایش داده ها در بانک اطلاعاتی استفاده می کند.

\* هنگامی که بخواهید با استفاده از کلاس `DataAdapter` اطلاعات مورد نیاز خود را از یک بانک اطلاعاتی دریافت کنید ابتدا باید خاصیت `SelectCommand` را در `DataAdapter` تنظیم کنید

\* خاصیت `SelectCommand` شیئی از نوع `SqlCommand` دریافت کرده که این شیء مشخص می کند داده ها چگونه باید از بانک اطلاعات انتخاب شده و نیز چه داده هایی باید انتخاب شوند.

\* اشیاء از نوع `SqlCommand` نیز دارای خاصیت هایی هستند که قبل از استفاده باید آنها را تنظیم کرد این خاصیت ها عبارتند از :



Connection - یک شیء از کلاس SqlConnection در این قسمت قرار گرفته و نحوه ی اتصال به بانک اطلاعاتی را مشخص می کند .

```
da.SelectCommand.Connection = Con;
```

CommandText : دستور Sql و یا پروسیجر ذخیره شده در بانک اطلاعاتی که باید توسط این شیء اجرا شود ، در این قسمت ذخیره می شود .

```
da.SelectCommand.CommandText = "select fields from table ";
```

توضیح کد بالا : در اینجا نوع دستور select می باشد و منظور از fields ، نام فیلدهایی است که می خواهیم اطلاعات آنها را استخراج کنیم اگر بخواهیم همه ی آنها را استخراج کنیم از \* استفاده می کنیم و منظور از table نیز نام جدولی است که می خواهیم اطلاعات را از آن استخراج کنیم . نمونه کد : برای مثال می خواهیم اطلاعات فیلدهای نام ، نام خانوادگی و شماره دانشجویی از جدول دانشجو را استخراج کنیم و برای کارمورد نظر استفاده کنیم (پس ما در اینجا فقط اطلاعات را استخراج می کنیم).

```
SqlConnection Con = New SqlConnection(connectionstring) ;
```

```
SqlDataAdapter da = New SqlDataAdapter ();
```

```
da.SelectCommand.Connection = Con ;
```

```
da.SelectCommand.CommandText = "select fname,lname,id from student ;"
```

دسترسی اطلاعات و اتصال داده ها

دسترسی به اطلاعات :

در ویژوال #C برای دسترسی به اطلاعات و نمایش آنها سه کامپوننت مهم و اصلی وجود دارند که عبارتند از :

Binding Source ، Binding Navigator ، Table Adapter ، DataSet .

\* کامپوننتهای Binding Navigator ، Binding Source و DataSet را می توانید در قسمت Data جعبه ابزار ببینید .

\* کامپوننت TableAdapter نیز بر اساس مسیری که برای دسترسی به اطلاعات درون بانک اطلاعاتی و نمایش آنها طی می کنیم به صورت اتوماتیک ایجاد می شود . در ادامه توضیح مختصری در مورد کامپوننتهای مطرح شده با هم مرور می کنیم .

## کامپوننت DataSet :

مانند یک موتور اطلاعاتی کوچک عمل می کند با استفاده از DataSet ابتدا به بانک وصل می شویم اطلاعات مورد نیاز را در حافظه DataSet قرار می دهیم سپس ارتباط با بانک را قطع می کنیم از این پس هر تغییری که خواستیم می توانیم بر روی اطلاعات درون DataSet اعمال کنیم سپس در آخر تمام تغییرات را بر روی بانک اطلاعاتی اعمال کنیم .

- با استفاده از این کامپوننت اطلاعات درون جدولی نگهداری می شوند و با استفاده از کامپوننت DataView به چندین روش پرس و جوهای را روی داده ها انجام داد .

## کامپوننت DataGridView :

این کنترل برای نمایش داده های موجود در یک بانک اطلاعاتی در فرم برنامه به کار می رود . برای کار با آن کافی است آن را به منبع داده های خود ، مثلاً یکی از جدولهای موجود در بانک اطلاعاتی متصل کرده و سپس این کنترل را تنظیم کنیم تا داده های جدول مورد نظر همانند یک جدول نمایش دهد (ستونهای این جدول نام فیلدها و ردیفهای آن اطلاعات مربوط به فیلدها که هر کدام در یک رکورد نگهداری می شوند) .

- علاوه بر این به وسیله این کنترل می توانید عنوان ستونهای داده ها و یا نوع نمایش آنها را نیز بدخواه تعیین کنیم

## کامپوننت BindingSource :

این کنترل همانند پلی برای ایجاد ارتباط بین داده های موجود در منبع داده ای شما (مانند DataSet) و کنترل هایی که برای نمایش داده ها مورد استفاده قرار می گیرند (مانند TextBox) به کار می رود . بنابراین هنگامی که بخواهید به وسیله ی کنترل هایی و یا به هر دلیل دیگری بخواهید به آنها د منبع اطلاعاتی دسترسی داشته باشید ، این ارتباط باید از طریق این کامپوننت صورت بگیرد .

## کامپوننت BindingNavigator :

این کنترل یک رابط گرافیکی استاندارد برای حرکت بین رکوردهای موجود در یک بانک اطلاعاتی ایجاد می کند . همچنین مانند کنترل DataGridView می تواند به کنترل BindingSource متصل شده و از طریق آن به داده های موجود در برنامه دسترسی داشته باشد .

## کامپوننت TableAdapter :

این کامپوننت در جعبه ابزار وجود ندارد بلکه با توجه به روشی که کامپوننت های داده ای دیگر را در برنامه قرار داده و آنها را تنظیم می کنید و به صورت اتوماتیک ایجاد می شود

- این کامپوننت حاوی پرس و جوهایی برای انتخاب داده های موجود در بانک اطلاعاتی و نیز اطلاعاتی در مورد نحوه اتصال برنامه به بانک است .

- همچنین حاوی متدهایی است که به وسیله آنها می توان داده ها را از جداول بانک اطلاعاتی بدست آورد و در

کامپوننت هایی مانند **DataSet** قرار داد و سپس در برنامه از آن داده ها استفاده کرد .  
- این کامپوننت این قابلیت را دارد که بر اساس دستور **Select** ای که برای انتخاب داده ها از بانک اطلاعاتی برای آن وارد می کنید دستورات **Update**، **Insert** و نیز **Delete** مناسب برای تغییر داده های انتخاب شده در بانک اطلاعاتی ایجاد کند .

\* اتصال داده ها :

اتصال داده یعنی اینکه داده های را که به وسیله ی کامپوننت **BindingSource** به آنها دسترسی دارید را به یک کنترل خاص نسبت دهید (مثلا به یک **DataGridView** یا **TextBox** ، ... ) .  
به عبارت دیگر یک کنترل را بتوانید به نحوی تنظیم کنید که داده های مورد نیاز خود را به وسیله کامپوننت های دسترسی داده ها در برنامه دریافت کند و سپس آنها را به صورت اتوماتیک به کاربر نمایش دهد .  
- در **#C** بعضی از کنترل ها وجود دارند که مخصوص این کار طراحی شده اند مانند کنترل **DataGridView** و **TextBox** .

jQuery چیست برای اینکه بدانیم به راحتی باید گفت که یک کتابخانه یا چارچوب کاری برای زبان برنامه نویسی سمت سرورس گیرنده جاوا اسکریپت (JavaScript framework) است که هدف اصلی آن فراهم آوردن زمینه ی استفاده ی آسان از زبان نام برده در برنامه نویسی تحت وب (صفحات وب) می باشد jQuery. یک کتابخانه از جاوا اسکریپت است. از آن جایی که کتابخانه ی JQuery بسیاری از امکانات و عملیات قابل اجرا توسط زبان جاوا اسکریپت را گرفته و در قالب یک سری تابع می گنجانند که می توانید به راحتی هر چه تمام تر فراخوانی و اجرا کنید (با کم ترین میزان کدنویسی بیشترین عملیات ممکن را اجرا کنید)، می توان آن را یک لایه ی سطح بالا (abstraction layer) نیز نامید. با این حال، لازم است به خاطر داشته باشید که کتابخانه ی ذکر شده به هیچ وجه نمی تواند کاملا جایگزین JavaScript گردد. اگرچه JQuery بسیاری از موارد میزان کدنویسی را کاهش داده و در عین حال همان عملیات قابل اجرا توسط جاوا اسکریپت را اجرا می کند، اما نباید فراموش کنید که توابع و دستورات jQuery هم در حقیقت همان کد زبان جاوا اسکریپت است. با توجه به آنچه گفته شد، در لزومی ندارد برای استفاده از کتابخانه ی مزبور حتما یک متخصص در زمینه ی برنامه نویسی با جاوا اسکریپت باشید. در واقع، JQuery سعی دارد بسیاری از کارهایی که می توان با استفاده از کدهای جاوا اسکریپت پیاده سازی کرد، همچون دستکاری DOM و فراخوانی توابع AJAX، را تسهیل نماید. از این رو لازم نیست درباره ی جاوا اسکریپت اطلاعات بسیار بالایی داشته باشید.

تعدادی زیادی کتابخانه ی جاوا اسکریپت وجود دارد که در حال حاضر JQuery پرکاربردترین و کارآمدترین آن ها محسوب می شود. از جمله دلایل محبوبیت این کتابخانه می توان به قابلیت توسعه پذیری بالا اشاره کرد. بنابراین شما می توانید برای هر کاری که تصورش را می کنید، افزونه (plugin) پیدا کنید.

قدرت و کارایی بالا، دامنه ی وسیع افزونه های پرکاربرد، ساختار نگارشی روان و کارآمد این چارچوب کاری را از دیگر کتابخانه های JQuery تمایز بخشیده است.

پس فهمیدیم که jquery چیست (همچنین برخی افراد معادل فارسی آن یعنی جی کوئری را استفاده می کنند)... در مطالب آموزشی بعدی به بررسی سایر خصوصیات و موارد فنی jquery می پردازیم.

## نحوه استفاده از JQuery در صفحات وب

جهت استفاده از JQuery، بایستی آن را به صفحاتی که می خواهید از کدهای آن استفاده کنید، اضافه نمایید. برای این منظور می بایست ابتدا این کتابخانه را از سایت [www.jquery.com](http://www.jquery.com) دانلود نمایید. با مراجعه به سایت مزبور با دو گزینه برای دانلود مواجه می شوید:

1. نسخه ی "Production" نسخه ی "Development"

نسخه ی "Production" برای وب سایت ها live و آماده می باشد. این نسخه ی فشرده و کم حجم (minify) شده تا کم ترین میزان فضا را اشغال کند و برای کاربرانی که مرورگرها آن ها باید فایل JQuery را همراه با باقی اطلاعات سایت بارگیری کند، بهترین گزینه می باشد. اما برای آزمایش و توسعه، نسخه ی "Development" گزینه ی ارجح تلقی می شود. نسخه ی یاد شده فشرده سازی (minify) نشده و به همین خاطر در صورت برخورد با خطا می توانید مشاهده کنید خطا در کجا رخ داده است.

پس از دانلود فایل JQuery JavaScript، لازم است آن را به وسیله ی تگ `<script>` تی ام ال به صفحه ی وب خود متصل کنید (خصیصه ی type تگ نام برده را با مقدار "text/javascript" src="jquery-1.5.1.js" تنظیم

نمایید یا به عبارتی دیگر در تگ script به آن ارجاع دهید). بهترین کار جایگذاری فایل دانلود شده ی jquery.js در پوشه ی متعلق به صفحه ای است که می خواهید از jQuery استفاده کند. سپس آن را در بخش < head > بگنجانید (در بخش head صفحه به آن ارجاع دهید):

```
1<script type="text/javascript" src="jquery-1.5.1.js"></script>
```

اکنون بخشی از صفحه ی شما به شکل زیر خواهد:

```
1<title>jQuery test</title>
2<script type="text/javascript" src="jquery-1.5.1.js"></script>
```

یک روش جدیدتر (جای دانلود jQuery و میزبانی آن) این است که jQuery را از یک CDN تحویل محتوا روی شبکه دریافت کنید Google و Microsoft. هر دو نسخه های مختلفی از کتابخانه ی مذکور و دیگر کتابخانه های جاوا اسکریپت را میزبانی می کنند. با این کار دیگر نیازی به دانلود و ذخیره ی فایل jQuery نیست. بعلاوه از آنجایی این فایل از یک URL مشترک با سایت دیگر دریافت می شود، به احتمال زیاد زمانی که کاربران به سایت شما مراجعه می کنند و مرورگر کتابخانه را درخواست می کنند، فایل نام برده از قبل در حافظه ی نهان (cache) آماده می باشد (به این دلیل که سایت دیگری دقیقاً همین نسخه و فایل را از قبل استفاده می کند). همچنین بیشتر CDN ها اطمینان حاصل می کنند، وقتی کاربری فایلی را از آن درخواست می کند، آن فایل از طریق نزدیکترین سرویس دهنده به کاربر (از نظر موقعیت جغرافیایی و مکانی) تحویل داده شود.

می توانید jQuery را از یک CDN، درست مانند زمانی که آن را دانلود می کنید، مورد استفاده قرار دهید. تنها چیزی که تغییر می کند، آدرس URL می باشد. برای نمونه، جهت اضافه کردن ویرایش jQuery 1.5.1 از سرورهای Google به صفحه ی وب مورد نظر، دستور زیر را وارد می کنید:

```
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.5.1/jquery.min.js"></script>
```

## مثال Hello World! در JQuery

در مبحث قبلی با نحوه ی اضافه کردن jQuery به صفحه ی وب برای بهره وری از امکانات فوق العاده زیاد آن آشنا شدید. پیش از شروع به استفاده از کتابخانه ی jQuery بایستی با مفاهیم پایه ای آن آشنایی پیدا کنید و نظر خود را به مثال ساده ی زیر جلب نمایید:

نمونه یک

```
<div id="divTest1">Hello, world!</div>
<script type="text/javascript">
  $("#divTest1").text("Hello, world!");
</script>
```

همان طور که مشاهده می کنید، یک تگ با شناسه ی "divTest1" داریم. از علامت \$ که به مثابه ی یک میانبر برای دسترسی به المان های HTML ایفای نقش می کند، استفاده می کنیم. بنابراین با استفاده از این علامت تمامی المان های HTML با شناسه ی "divTest1" را انتخاب کرده و مقدار "Hello, world!" را در آن قرار می دهیم (text) آن را با رشته ی "Hello, world!" تنظیم می کنیم.

حتی چاپ کردن همین متن ساده در جاوا اسکریپت به کدنویسی بسیار بیشتر نیاز داشت:

```
<div id="divTest2">Hello, world!</div>
<script type="text/javascript">
  document.getElementById("divTest2").innerHTML = "Hello, world!";
</script>
```

اگر المان HTML فقط یک class داشت و از ID نیز برخوردار نبود، این کد به مراتب طولانی تر می شد. به طور معمول، باید صبر کنید تا صفحه ی وب وارد مرحله یا وضعیت READY شود و سپس اقدام به دستکاری محتویات آن نمایید. مثال های ساده ی بالا (و همچنین تعدادی از عملیات پیچیده ی دیگر) در بیشتر مرورگرها، پیش از آماده شدن صفحه نیز قابل اجرا هستند. اما این امر در مورد تمامی عملیات با jQuery پیش از بارگذاری کامل صفحه و آماده شدن (آن) امکان پذیر نیست. خوشبختانه، jQuery اجرای این عملیات را نیز بسیار آسان ساخته است. در مبحث بعدی به تشریح انتخابگرها (selector) خواهیم پرداخت.

### رخداد Ready در Jquery

همان طور که در درس قبلی ذکر شد، توصیه می شود تا اتمام بارگذاری صفحه و آماده شدن آن صبر کنید و سپس اقدام به کار با آن نمایید. این امر همچنین برای شما این امکان را فراهم می کند تا کد جاوا اسکریپت خود را در بخش head صفحه (یا به طور مستقیم و یا از طریق لینک به یک فایل جاوا اسکریپت خارجی) و قبل از تگ body قرار دهید. در jQuery این کار با قرار دادن کد در رخداد document ready امکان پذیر می باشد. در زیر، مثال درس قبلی را بکار می بریم، اما این بار داخل رخداد ready قرار داده شده است.

### نمونه یک

```
<div id="divTest1"></div>
<script type="text/javascript">
  function DocumentReady() {
    $("#divTest1").text("Hello, world!");
  }
  $(document).ready(DocumentReady);
</script>
```

در این مثال یک تابع ایجاد کرده و آن را DocumentReady نام گذاری می کنیم که به مجرد آماده شدن صفحه برای کار با DOM و دستکاری المان های آن، فعال می شود. در خط آخر کد، با استفاده از متد ready()، تابع خود را به رخداد ready تخصیص داده و بدین وسیله به jQuery اطلاع می دهیم که با آماده شدن صفحه می خواهیم که تابع را فراخوانی کند.

می توان با بهره گیری از قابلیت تعریف تابع بی نام (anonymous function) در جاوا اسکریپت، کد فوق را کوتاه تر نیز کرد. منظور این است که بجای اعلان تابع و تخصیص اسم به آن، فقط یک تابع ایجاد می کنیم و بلافاصله ارجاع به تابع ready() را به آن پاس می دهیم. اگر چندان آشنایی با زبان جاوا اسکریپت ندارید، در آن صورت استفاده از این روش کمی پیچیده بنظر می رسد. اما با آشنایی بیشتر خواهید دانست که در این روش، کد نویسی کاهش یافته و فضای کمتری نیز اشغال می شود.

### نمونه دو

```
<div id="divTest2"></div>
<script type="text/javascript">
```

```

$(document).ready(function () {
    $("#divTest2").text("Hello, world!");
});
</script>

```

با این وجود، تیم طراحان و توسعه دهندگان این کتابخانه احساس کردند که این کد را کوتاه تر هم می توانند بکنند. از اینرو یک نسخه ی overload از سازنده (constructor) ایجاد کردند که تابع ready را به عنوان پارامتر می پذیرد و کد را با این روش به مراتب کوتاه تر کردند.

نمونه سه

```

<div id="divTest3"></div>
<script type="text/javascript">
    $(function () {
        $("#divTest3").text("Hello, world!");
    });
</script>

```

در مثال فوق، تابع بی نام مستقیماً به سازنده ارسال می شود که آن را به رخداد ready تخصیص می دهد. اگر این کد را تست کنید، متوجه می شوید که به مجرد بارگذاری صفحه، event مربوطه اجرا می گردد. گاهی این رخداد چنین سریع فعال می شود که اصلاً متوجه اجرای آن نمی شوید. همان طور که قبلاً گفته شد، توصیه می شود برای استفاده از jQuery در صفحه وب خود کدهای آن را داخل تابع رخداد ready قرار دهید. در بیشتر مثال های این آموزش همین رویه دنبال خواهد شد، مگر برای کوتاه تر شدن مثال که در تعدادی از نمونه ها این کد را لحاظ نمی کنیم.

## زنجیره سازی متدها یا فراخوانی متدها به صورت زنجیر وار (METHOD CHAINING)

یکی دیگر از خصوصیت های خارق العاده jQuery، این است که بیشتر متدهای آن یک شی jQuery برمی گردانند. شی بازگشتی خود در فراخوانی متد دیگر بکار شما می آید. این قابلیت به شما اجازه می دهد دستورات را مانند زنجیر به هم متصل کنید که در نهایت زمینه را مهیا می کند تا چندین متد را در بر روی مجموعه یکسان از المان ها اجرا نمایید. همچنین از آنجایی که مرورگر دیگر مجبور نیست همان المان ها را چندین بار پیدا کند، در زمان صرفه جویی می شود.

**مثال:**

نمونه یک

```

<div id="divTest1" style="color: blue;">Hello, world!</div>
<script type="text/javascript">
    $("#divTest1").text("Hello, world!").css("color", "blue");
</script>

```

در این مثال، یک شی جدید jQuery نمونه سازی می کنیم و المانی که شناسه ی آن divTest1 می باشد را با کاراکتر \$ انتخاب می نماییم. این علامت یک میانبر بوده و جایگزینی برای کلاس jQuery می باشد. در نتیجه یک شی jQuery دریافت می کنیم که به ما این امکان را می دهد تا المان انتخابی را مطابق نیاز ویرایش کنیم. ما از آن شی برای فراخوانی متد text() استفاده می کنیم. این متد متن المان انتخابی را تنظیم می کند. متد ذکر شده بار دیگر شی jQuery را باز می گرداند و به ما اجازه می دهد متد یا عملیات دیگری (متد css()) را بر روی مقدار برگشتی، اجرا کنیم. می توانیم بسته به نیاز خود توابع دیگری را به انتهای تابع دیگر متصل کرده و فراخوانی کنیم، اما در کل باید از طولانی

کردن بیش از حد کد خودداری نمود. خوشبختانه، جاوا اسکریپت در خصوص دستور نگارشی (syntax) خیلی سخت گیر نیست، بنابراین شما می توانید آن را مطابق میل فرمت دهی کنید. به عنوان مثال، می توان دستور ذکر شده در بالا را به صورت زیر نیز نوشت:

نمونه دو

```
<div id="divTest2" class="bold" style="color: blue;">Hello, world!</div>
<script type="text/javascript">
    $("#divTest2").text("Hello, world!")
                                .removeClass("blue")
                                .addClass("bold")
                                .css("color", "blue");
</script>
```

جاوا اسکریپت خود فضای های بی مورد در کد را در زمان تفسیر دور انداخته و آن را به عنوان یک خط طولانی کد با تعداد زیادی فراخوانی تابع (method call) اجرا می کند (کدی که تعداد زیادی تابع در آن صدا زده شده اجرا می کند). توجه داشته باشید که برخی از متدهای به عنوان نتیجه، شی برنمی گردانند. این درحالی است که برخی دیگر از متدها بسته به پارامترهای ارسالی به آن، یا شی برمی گرداند و یا هیچ شی ای برنمی گرداند. به عنوان نمونه، می توان به متد text() اشاره کرد. در صورت عدم ارسال آرگومان به آن، متن جاری المان انتخابی بجای شی بازگردانده می شود، اما اگر فقط یک پارامتر به آن پاس دهید، متن شی jQuery تنظیم شده و یک شی جدید jQuery بازگردانده می شود.

## معرفی JQuery Selectors

یکی از عملیات رایج که برای انجام آن از جاوا اسکریپت استفاده می شود، خواندن و ویرایش محتویات صفحه می باشد. برای این منظور، ابتدا بایستی المانی که قصد دستکاری آن را دارید پیدا کنید. در اینجا است که انتخابگر (selector) jQuery به کمک شما می آید. با کد خالص جاوا اسکریپت، پیدا کردن المان ها و تغییر آن ها بسیار دشوار می باشد. مگر اینکه قصد یافتن تنها یک المان را داشته باشید که دارای شناسه ی منحصر بفرد باشد jQuery. به شما این امکان را می دهد تا المان ها را بر اساس شناسه، کلاس ها، نوع (type)، مقدار attribute ها و غیره گزینش نموده و بازبایی کنید. این روش در واقع مبتنی بر selector های jQuery می باشد.

از آنجایی انتخاب المان ها با jQuery یک فعل بسیار رایج است، constructor در اشکال و فرم های مختلف به کار می آید، بدین صورت که سازنده با پذیرفتن یک selector query به عنوان آرگومان، با کم ترین میزان کد، المان مد نظر را یافته و بیشترین کارایی را ارائه می دهد. شما می توانید با نوشتن دستور jQuery() یا درج میانبر و جایگزین آن (\$)، یک شی jQuery نمونه سازی نمایید. بنابراین، انتخاب یک مجموعه المان به آسانی نمونه ی زیر می باشد:

```
$(<query here="">) </query>
```

با شی بازگشتی، می توانید اقدام به استفاده و ویرایش المان هایی که در کوئری انتخاب کرده بودید، نمایید.

انتخاب المان ها بر اساس شناسه و کلاس (استفاده از المان ها، شناسه و کلاس ها)



## #ID انتخابگر

یکی از پرکاربردترین انواع selector ، انتخابگر ( #id ) گزینش المان ها بر اساس شناسه ( می باشد، همان طور که در مثال "Hello, world" نظاره گر آن بودیم. در مثال نام برده، از خصیصه ی ID یک تگ HTML برای یافتن و انتخاب یک المان منحصر بفرد استفاده کردیم. جهت مکان یابی یک المان با شناسه ی مشخص، یک کاراکتر هش و به دنبال آن ID المان مورد نظر را درج نمایید:

```
$("#divTest")
```

**مثال:**

نمونه یک

```
<div id="divTest"></div>
<script type="text/javascript">
  $(function () {
    $("#divTest").text("Test");
  });
</script>
```

اگرچه تنها یک المان در صفحه وجود دارد که با کوئری ذکر شده در این مثال منطبق می باشد، باید توجه داشته باشید که نتیجه ی برگشتی یک لیست است. بدین معنی که ممکن است نتیجه حاوی چندین المان باشد، البته در صورتی که کوئری با بیش از یک المان منطبق باشد. یک مثال رایج در این زمینه، منطبق شدن کوئری با تمامی المان هایی است که از یک یا چند کلاس CSS استفاده می کنند.

## گزینش المان بر اساس کلاس آن ( CLASS ) انتخابگر

المان هایی که دارای کلاس مشخص هستند را می توان با نوشتن کاراکتر نقطه " . " و اسم کلاس با کوئری منطبق نموده و انتخاب کرد و در نهایت به صورت شی برگرداند.

نمونه دو

```
<ul>
  <li class="bold">Test 1</li>
  <li>Test 2</li>
  <li class="bold">Test 3</li>
</ul>
<script type="text/javascript">
```

```
$(function () {  
    $(".bold").css("font-weight", "bold");  
});  
</script>
```

## انتخاب المان بر اساس اسم آن (ELEMENT SELECTOR)

همچنین می توان المان ها را بر اساس اسم تگ آن ها انتخاب کرد. به عنوان مثال، می توانید تمامی لینک ها یک صفحه را بدین صورت (با ذکر اسم تگ آن ها) انتخاب کرد:

```
$("a")
```

و یا تمامی تگ های div را به شکل زیر انتخاب نمود:

```
$("div")
```

در صورت استفاده از یک انتخاب گر چند-المانه (مانند انتخابگر کلاس که در مثال قبلی بکار گرفته شد) و همچنین آگاهی از نوع دقیق المان های مورد انتخاب، توصیه می شود نوع المان را پیش از انتخابگر مشخص نمایید. نه تنها این روش دقیق تر است، بلکه پردازش آن برای jQuery آسان تر صورت پذیرفته که به واکنش هر چه سریعتر سایت کمک می کند. در زیر نسخه ی بازنویسی شده ی مثال قبلی را مشاهده می کنید:

```
$("#span.bold").css("font-weight", "bold");
```

این مثال تمامی المان های span که اسم کلاس آن ها bold می باشد را bold می کند. البته، این روش را می توان با دیگر selectorها نیز پیاده سازی کرد.

## پیدا کردن و انتخاب المان ها بر اساس ATTRIBUTE

در آموزش قبلی، دیدیم که چگونه می توان المان ها را با توجه به کلاس یا شناسه ی آن ها انتخاب کرد( به آموزش انتخاب عناصر وب بر اساس کلاس و شناسه با jquery مراجعه کنید). این دو خاصیت (property) به خاطر اینکه برای استایل دهی به المان ها با CSS مورد استفاده قرار می گیرند، با هم مرتبط می باشند. اما با کتابخانه ی jQuery، می توان المان ها را بر اساس هر نوع خصیصه (attribute) پیدا و انتخاب کرد. در کتابخانه ی jQuery چندین انتخابگر خصیصه (attribute selector) وجود دارد که در مقاله ی حاضر به شرح برخی از آن ها می پردازیم.

### انتخاب المان بر اساس خصیصه ی معین

می توان المان ها را بر اساس attribute یکسان انتخاب کرد. دقت داشته باشید که مثال بعدی ایجاب نمی کند که attribute مقدار معینی داشته باشد یا حتی آن attribute اصلا مقداری داشته باشد. دستور نگارش برای نوشتن و استفاده از این selector به صورت زیر می باشد:

نمونه یک

```

<span title="Title 1">Test 1</span><br>
<span>Test 2</span><br>
<span title="Title 3">Test 3</span><br>
<script type="text/javascript">
    $(function () {
        $("[title]").css("text-decoration", "underline");
    });
</script>

```

بنابراین، یک [] درج کرده و داخل آن اسم attribute دلخواه را ذکر می کنید. در نمونه ی بالا، با استفاده از یک انتخاب گر خصیصه (attribute selector)، کلیه ی المان هایی که attribute آن ها title می باشد را گزینش کرده و سپس با استفاده از تابع css(). به المان های مزبور underline اعمال می کنیم. همان طور که بیشتر گفته شد، این کوئری با تمامی المان هایی که خصیصه ی آن ها title می باشد، صرف نظر از مقدارشان، منطبق می شود. اما گاهی لازم است یک المان را که attribute آن دارای مقدار مشخص است، پیدا کنیم.

### انتخاب المان هایی که ATTRIBUTE آن ها دارای مقدار مشخص می باشد

در زیر مثالی را مشاهده می کنید که همه ی المان هایی که خصیصه ی آن ها دارای مقدار مشخص است را پیدا می کند:  
نمونه دو

```

<a href="http://www.google.com" target="_blank">Link 1</a><br>
<a href="http://www.google.com" target="_self">Link 2</a><br>
<a href="http://www.google.com" target="_blank">Link 3</a><br>
<script type="text/javascript">
    $(function () {
        $("a[target='_blank']").append("[new window]");
    });
</script>

```

Selector صرفاً به jQuery اعلان می کند تمامی لینک هایی (المان های a) که دارای خصیصه ی target هستند و مقدار آن ها برابر رشته ی "\_blank" می باشد را پیدا کرده و سپس متن "[new window]" را به آن ها پیوست (append) نماید. حال اگر بخواهیم المان هایی که مقدار attribute آن ها برابر با مقدار ذکر شده نباشند، انتخاب کنیم، چه اقدامی را بایستی اتخاذ کنیم؟

در پاسخ به این سوال باید گفت که کافی است انتخاب گر را با استفاده از عملگر "!" نقیض نمایید:

```

$("a[target!='_blank']").append("[same window]");

```

تنها فرق این نمونه با مثال قبلی در استفاده از عملگر "!" پیش از علامت = می باشد. نمونه های دیگر:

پیدا کردن المان های input که مقدار خصیصه ی name آن ها با رشته ی معین (مثلاً 'txt') شروع می شود (با استفاده از عملگر: ^=):

```

$("input[name^='txt']").css("color", "blue");

```

پیدا کردن المان های input که مقدار خصیصه ی name آن با رشته ی معین (مثلا letter) پایان می یابد (با استفاده از عملگر: \$=)

```
$("#input[name$='letter']").css("color", "red");
```

پیدا کردن المان های input که مقدار خصیصه ی name آن ها برابر با رشته ی 'txt' می باشد:

```
$("#input[name*='txt']").css("color", "blue");
```

### انتخاب المان ها بر اساس رابطه ی پدر و فرزندی (Parent/child relation selectors)

jQuery زیاده شما این امکان را می دهد تا المان ها را بر اساس عنصر پدر آن ها انتخاب کنید. دو روش وجود دارد: یکی اینکه تنها المان هایی که فرزند مستقیم عنصر پدر هستند با کوئری منطبق شده و انتخاب شوند، دیگری اینکه تمامی المان هایی که حتی به طور غیر مستقیم با عنصر پدر ارتباط دارند نیز به صورت سلسله مراتبی انتخاب شوند (برای مثال، فرزند فرزند فرزند عنصر پدر).

دستور نگارشی برای یافتن تمامی المان هایی که نوادگان مستقیم یک عنصر مشخص هستند، به صورت زیر می باشد:

```
$("#div > a")
```

این selector تمامی لینک هایی که فرزند مستقیم المان div هستند را انتخاب می کند. حال اگر عملگر <را با یک جای خالی (space) جایگزین نمایید، تمامی لینک های داخل المان div که حتی فرزند غیرمستقیم آن محسوب می شوند نیز انتخاب می شوند.

```
$("#div a")
```

در اینجا مثالی را می بینید که در آن تگ bold که فرزند مستقیم المان div با شناسه ی TestArea1 است، آبی رنگ می شود:

نمونه یک

```
<div id="divTestArea1">
  <b style="color: blue;">Bold text</b>
  <i>Italic text</i>
  <div id="divTestArea2">
    <b style="color: blue;">Bold text 2</b>
    <i>Italic text 2</i>
  </div>
  <b style="color: blue;">Bold text 3</b>
</div>
<script type="text/javascript">
  $("#divTestArea1 > b").css("color", "blue");
</script>
```

اگر این مثال را اجرا کنید، می بینید که تنها تگ bold اول آبی رنگ می شود. حال، چنانچه این کد را با روش دوم پیاده سازی کنید، هر دو تگ bold آبی رنگ می گردند .

همان طور که در نمونه ی زیر مشاهده می کنید، مثال فوق را با اعمال یک تغییر (جایگزین کردن عملگر <با space) بازنویسی کردیم که در آن هر دو تگ bold انتخاب و آبی رنگ می شوند (نوه یا فرزندان غیرمستقیم المان div نیز دستکاری می شوند):

```

<div id="divTestArea1">
  <b style="color: blue;">Bold text</b>
  <i>Italic text</i>
  <div id="divTestArea2">
    <b style="color: blue;">Bold text 2</b>
    <i>Italic text 2</i>
    <div>
      <b style="color: blue;">Bold text 3</b>
    </div>
  </div>
</div>
<script type="text/javascript">
  $("#divTestArea1 b").css("color", "blue");
</script>

```

jQuery با ارائه ی تابعی به نام parent() به شما این اجازه را می دهد تا در صورت لزوم به بالای سلسه مراتب (نمودار درختی وراثت) پیمایش کنید. با این تابع می توان پدر یک عنصر فرزند را یافت.

### افکت محو کردن المان ها به صورت تدریجی

اجرای افکت ساده ی animation با jQuery بسیار ساده می باشد. یکی از افکت های اضافی بر سازمان که jQuery از آن پشتیبانی می کند، محو/نمایش کردن المان ها به صورت تدریجی می باشد. در زیر مثالی ساده مشاهده می کنید که یک کادر پنهان را به تدریج نمایان می سازد:

نمونه یک

```

<div id="divTestArea1" style="padding: 50px; background-color: #89BC38; text-align: center; display: none;">
  <b>Hello, world!</b>
</div>
<a href="javascript:void(0);" onclick="ShowBox();">Show box</a>
<script type="text/javascript">
  function ShowBox() {
    $("#divTestArea1").fadeIn();
  }
</script>

```

می توانید المان های مختلف نظیر div، span، linkها را با استفاده از متد ذکر شده، به صورت تدریجی پنهان نمایید. این متد می تواند سه پارامتر بپذیرد. اولین پارامتر سرعت اجرای افکت نمایان سازی تدریجی را مشخص می کند (می توان مقادیر "fast"، "slow" و میلی ثانیه را به عنوان این پارامتر به تابع fadeIn() پاس دهید). مثال:

نمونه دو

```

<div id="divTestArea21" style="width: 50px; height: 50px; display: none; background-color: #89BC38;"></div>
<div id="divTestArea22" style="width: 50px; height: 50px; display: none; background-color: #C3D1DF;"></div>
<div id="divTestArea23" style="width: 50px; height: 50px; display: none; background-color: #9966FF;"></div>
<a href="javascript:void(0);" onclick="ShowBoxes();">Show boxes</a>
<script type="text/javascript">
  function ShowBoxes() {

```

```

    $("#divTestArea21").fadeIn("fast");
    $("#divTestArea22").fadeIn("slow");
    $("#divTestArea23").fadeIn(2000);
}
</script>

```

پارامتر دوم می تواند یک تابع callback باشد که در آن نام یک تابع را ذکر می کنید تا با به اتمام رسیدن افکت (نمایش تدریجی عنصر مورد نظر) اجرا گردد. در زیر مثالی را مشاهده می کنید که اثر هر دو متد fadeIn و fadeout که اثری مخالف تابع مذکور دارد) را به نمایش می گذارد:

```

<script src="http://www.tahlildadeh.com/Jquery/jquery-1.9.1.min.js"></script>
<div id="divTestArea3" style="width: 50px; height: 50px; background-color:
rgb(137, 188, 56); opacity: 0.0456002;"></div>
<script type="text/javascript">
    $(function () {
        $("#divTestArea3").fadeIn(2000, function () {
            $("#divTestArea3").fadeOut(3000);
        });
    });
</script>

```

نمونه سه

گاهی لازم است یک المان را بسته وضعیت فعلی آن به تدریج نمایان ساخته یا پنهان نمایید. ابتدا بررسی می کنیم آیا المان مورد نظر نمایان است یا خیر. سپس بسته به وضعیت آن، تابع fadeIn() یا fadeOut() را صدا می زنید. اما jQuery یک متد به نام fadeToggle() ارائه کرده است که بین دو حالت نمایان و پنهان تغییر حالت می دهد. این متد همان پارامترهای دو متد نام برده را می پذیرد و استفاده از آن بسیار ساده می باشد:

نمونه چهار

```

<script src="http://www.tahlildadeh.com/Jquery/jquery-1.9.1.min.js"></script>
<div id="divTestArea4" style="width: 50px; height: 50px; display: none;
background-color: #89BC38;"></div><br>
<a href="javascript:void(0);" onclick="ToggleBox();">Toggle box</a>
<script type="text/javascript">
    function ToggleBox() {
        $("#divTestArea4").fadeToggle("slow");
    }
</script>

```

## نمایش کشویی/اسلایدی المان ها در jquery

در درس قبلی با نحوه ی محو کردن تدریجی المان ها با استفاده از افکت fading آشنا شدید(مطلب مربوط به آموزش افکت fadein-fadeout در jquery). اما گاهی نمایش اسلایدی و کشویی المان ها بر ایفکت نمایش/پنهان سازی تدریجی ارجحیت دارد. برای این منظور jQuery چندین متد کارآمد ارائه می دهد.

در مثال زیر با بهره گیری از متد `slideDown()` یک المان `div` که شناسه ی آن `divTestArea1` می باشد را به صورت کشویی نمایش می دهیم:

نمونه یک

```
<div id="divTestArea1" style="padding: 50px; background-color: #89BC38; text-align: center; display: none;">
  <b>Hello, world!</b>
</div>
<a href="javascript:void(0);" onclick="ShowBox();">Show box</a>
<script type="text/javascript">
  function ShowBox() {
    $("#divTestArea1").slideDown();
  }
</script>
```

حال برای پنهان نمودن این کادر به صورت کشویی، از متد `slideUp()` استفاده می کنیم. این دو متد پارامترهای یکسان را می پذیرند که همگی اختیاری هستند. اولین پارامتر سرعت نمایش کشویی المان را تنظیم می کند. می توانید سرعت اجرای ایفکت را به سه مقدار "fast" ، "slow" یا میلی ثانیه تنظیم نمایید. مثال:

نمونه دو

```
<div id="divTestArea21" style="width: 50px; height: 50px; display: none; background-color: #89BC38;"></div>
<div id="divTestArea22" style="width: 50px; height: 50px; display: none; background-color: #C3D1DF;"></div>
<div id="divTestArea23" style="width: 50px; height: 50px; display: none; background-color: #9966FF;"></div>
<a href="javascript:void(0);" onclick="ShowBoxes();">Show boxes</a>
<script type="text/javascript">
  function ShowBoxes() {
    $("#divTestArea21").slideDown("fast");
    $("#divTestArea22").slideDown("slow");
    $("#divTestArea23").slideDown(2000);
  }
</script>
```

با اجرای مثال فوق مشاهده می کنید که کادر اول با سرعت بسیار بالایی به صورت کشویی نمایش داده می شود، کادر دوم با کمی تاخیر و کادر سوم با سرعت بسیار پایین اجرا می شود که اجرای آن تقریباً دو ثانیه به طول می انجامد. پارامتر دوم یک تابع `callback` می باشد که با اجرای کامل ایفکت نمایش اسلایدی اجرا می شود. استفاده از این پارامتر کاملاً اختیاری می باشد. در زیر مثال دیگری را مشاهده می کنید که اثر متد `slideUp()` را به نمایش می گذارد:

نمونه سه

[?](#)

```

<div id="divTestArea3" style="width: 50px; height: 50px; display: none;
background-color: #89BC38;"></div>
<script type="text/javascript">
  $(function () {
    $("#divTestArea3").slideDown(2000, function () {
      $("#divTestArea3").slideUp(3000);
    });
  });
</script>

```

در مثال فوق، تابع callback به محض اجرای کامل ایفکت نمایش اسلایدی عنصر به پایین (متد SlideDown()) صدا زده می شود و بلافاصله پس از آن متد SlideUp() فراخوانی می گردد (و المان مورد نظر باز به صورت کشویی جمع می شود). تیم طراحان کتابخانه ی jQuery متد دیگری به نام slideToggle() را ارائه کرده است که بین دو حالت slidedown و slideup تغییر وضعیت می دهد. این متد بسته به وضعیت جاری المان، آن را به صورت کشویی نمایش می دهد یا در صورت نمایان بودن) المان مورد نظر را جمع می کند.

نمونه چهار

[?](#)

```

<div id="divTestArea4" style="width: 50px; height: 50px; display: none; background-color: #89BC38;"></div><br>
<a href="javascript:void(0);" onclick="ToggleBox();">Toggle box</a>
<script type="text/javascript">
  function ToggleBox() {
    $("#divTestArea4").slideToggle("slow");
  }
</script>

```

### ایجاد animation های سفارشی با استفاده از متد animate() در jquery

در مباحث پیشین، افکت های نمایش اسلایدی و پنهان سازی تدریجی را که امکانات درون ساخته ی این کتابخانه ی جاوا اسکریپت هستند، تشریح کردیم. به کمک متد animate()، می توان animation های سفارشی ایجاد نمود. به طوری که تمامی خواص عددی CSS یک المان توسط این متد قابل مدیریت و دستکاری می باشد. با این متد می توان به طور مثال، یک کادر را به آرامی بر روی صفحه به حرکت درآورد یا به آن قابلیت پرش به بالا و پایین اعمال نمود.

نمونه یک

[?](#)

```

<div style="height: 60px;">
  <div id="divTestBox1" style="height: 50px; width: 50px; background-color: #89BC38; position: absolute;"></div>
</div>
<script type="text/javascript">
  $(function () {
    $("#divTestBox1").animate(
      {
        "left": "500px"
      }
    );
  });
</script>

```



اولین و تنها پارامتر الزامی متد `animate()` ، مجموعه خواص CSS است که مایلید تغییر دهید. در این مثال، یک المان `div` با موقعیت `absolute` داریم که با کمک متد `animate()` جی کوئری به آن دستور می دهیم زمانی که مقدار خاصیت `left` سی اس اس آن به 200 پیکسل رسید، آن را متوقف کند. دومین پارامتر به شما این امکان را می دهد تا سرعت اتمام انیمیشن و متحرک سازی را بر حسب میلی ثانیه یا با پاس دادن مقادیر "slow" (یا "fast" به ترتیب معادل 600 یا 200 میلی ثانیه) به متد یاد شده، تنظیم نمایید. با این دستور می توان سرعت حرکت کادر سبز رنگ بر روی صفحه را کاهش دهیم.

#### نمونه دو

```

1 <script type="text/javascript">
2     $(function () {
3         $("#divTestBox2").animate(
4             {
5                 "left": "200px"
6             },
7             5000
8         );
9     });
10 </script>

```

سومین پارامتر که به متد `animate()` ارسال می شود یک تابع `callback` است که به محض اتمام انیمیشن صدا زده می شود. این امکان در اجرای یک سری انیمیشن مختلف به صورت پشت سرهم بسیار مفید واقع می شود. مثال:

#### نمونه سه

```

1 <script type="text/javascript">
2     $(function () {
3         $("#divTestBox3").animate(
4             { "left": "100px" },
5             1000,
6             function () {
7                 $(this).animate(
8                     { "left": "20px" },
9                     500,
10                    function () {
11                        $(this).animate({ "left": "50px" },
12                            500);
13                    }
14                )
15            }
16        );
17    });
18 </script>

```

در این نمونه، متد `animate` را فراخوانی کرده و خاصیت `left` المان دارای شناسه ی `"divTestBox3"` را به عنوان پارامتر به آن پاس داده ایم. سپس از آن خواستیم تا زمانی که مقدار این خاصیت به 100 پیکسل نرسیده، انیمیشن را ادامه دهد. همچنین سرعت اتمام انیمیشن را 1 ثانیه (1000 میلی ثانیه) مشخص کرده ایم که با به پایان رسیدن آن بلافاصله انیمیشن دیگری آغاز می گردد. انیمیشن دوم، همان طور که از کد مشخص است، المان مزبور را به اندازه ی 20 پیکسل و در طی نیم

ثانیه به عقب برمی گرداند. بار دیگر با خاتمه یافتن انیمیشن دوم، انیمیشن دیگری شروع می شود که در آن المان به سمت راست رانده شده تا مقدار خاصیت left بر روی 50 تنظیم گردد .

با این حال، از آنجایی که jQuery دارای امکان صف بندی برای اجرای animation ها می باشد، می توانید مثال بالا را بسیار ساده تر پیاده سازی کنید. ناگفته نماند که این امکان بیشتر زمانی بکار می آید که بخواهید چندین انیمیشن به صورت پشت سرهم اجرا نمایید. اگر بخواهید زمانی که animation به پایان رسید، اتفاق دیگری رخ دهد، مثال ذکر شده در بالا باز روش مناسب می باشد. نسخه ی صف بندی شده ی انیمیشن بالا را در مثال زیر مشاهده می کنید (انیمیشن ها پشت سرهم اجرا می شوند):

نمونه چهار

```
1 <script type="text/javascript">
2     $(function () {
3         $("#divTestBox4").animate({ "left": "100px" }, 1000);
4         $("#divTestBox4").animate({ "left": "20px" }, 500);
5         $("#divTestBox4").animate({ "left": "50px" }, 500);
6     });
7 </script>
```

### متوقف سازی انیمیشن ها با استفاده از متد stop()

در فصل قبلی با نحوه ی ایجاد انیمیشن های سفارشی با استفاده از متد animate() و اجرای آن ها یکی پس از دیگری آشنا شدیم (آموزش متد animate در jQuery). این کار را با دو روش پیاده سازی کردیم: چندبار صدا زدن متد animate() (قابلیت queue قرار دادن انیمیشن ها در صف و اجرای آن ها به صورت پشت سرهم) جی کوئری. گاهی لازم است یک متد را پیش از اتمام آن، متوقف نمایید. برای این منظور jQuery متدی به نام stop() را فراهم می نماید. متد نام بر روی تمامی ایفکت های جی کوئری (توابعی که افکت اجرا می کند) نظیر نمایش کشویی/اسلایدی، نمایش/پنهان سازی تدریجی و نیز تمامی انیمیشن های سفارشی که با متد animate() ایجاد می شود، قابل اجرا می باشد.

مثال:

نمونه یک

```
<a href="javascript:void(0);" onclick="$('#divTestArea1').slideDown(5000);">Show bo
<a href="javascript:void(0);" onclick="$('#divTestArea1').stop();">Stop</a>
<div id="divTestArea1" style="padding: 100px; background-color: #89BC38; tex
center; display: none;">
    <b>Hello, world!</b>
</div>
```

به منظور کاهش اندازه ی کد، توابع به صورت درون خطی و inline در رخدادهای onclick دو لینک مورد نظر فراخوانی شده اند. با کلیک بر روی اولین لینک، متد slideDown() بر روی المان div اجرا شده و آن را با سرعت پایین به صورت کشویی نمایش می دهد. حال به محض کلیک بر روی لینک دوم، اجرای انیمیشن (نمایش کادر سبز رنگ حاوی رشته ی "Hello World" به صورت اسلایدی) درجا متوقف می گردد. متد stop() دو پارامتر می پذیرد. اولین پارامتر مشخص می کند آیا صف اجرای انیمیشن بایستی پاک شود یا خیر (از اجرای انیمیشن های صف بندی شده جلوگیری می کند). در حالت پیش فرض بر روی false تنظیم می باشد، بدین معنی که انیمیشن جاری متوقف شده ولی انیمیشن های دیگری که در صف قرار دارند اجازه ی اجرا را دارند.

نمونه دو

```

<a href="javascript:void(0);"
onclick="$('#divTestArea2').slideDown(5000).slideUp(5000);">Show box</a>
<a href="javascript:void(0);" onclick="$('#divTestArea2').stop();">Stop</a>
<a href="javascript:void(0);" onclick="$('#divTestArea2').stop(true);">Stop all</a>
<a href="javascript:void(0);" onclick="$('#divTestArea2').clearQueue().hide();">Res
<div id="divTestArea2" style="padding: 100px; background-color: #89BC38; tex
center; display: none;">
    <b>Hello, world!</b>
</div>

```

یک انیمیشن دیگر به لینک "Show box" اضافه کردیم. این انیمیشن سبب می شود کادر با سرعت پایین و به صورت کشویی نمایش داده شود و پس از اتمام آن دوباره به صورت کشویی جمع گردد. سیستم صف بندی در جی کوئری اطمینان حاصل می کند که این گام ها به ترتیب مشخص شده و پشت سرهم، اجرا شوند. بر روی لینک "Reset" کلیک کنید تا دوباره کادر سبز رنگ محو می گردد و سپس بار دیگر بر روی لینک "Show box" و در نهایت بر روی لینک "Stop" کلیک کنید. خواهید دید که انیمیشن اول متوقف می شود و راه برای اجرای انیمیشن دوم باز می گردد. اما اگر بر روی دکمه ی "Stop all" کلیک کنید، مقدار true پاس داده شده به عنوان آرگومان باعث می شود تمامی انیمیشن های قرار گرفته در صف خالی شده و به دنبال آن تمامی انیمیشن هایی که قرار است بر روی المان اجرا شود، متوقف گردند. دومین پارامتر مشخص می کند آیا انیمیشن جاری همان جا متوقف شود یا فوراً پایان یابد. ارسال این پارامتر به متد stop() تفاوت قابل توجهی ایجاد می کند. در مثال قبلی، اگر بر روی لینک stop کلیک کنید، انیمیشن درجا متوقف شده و به همان صورت باقی می ماند. با اجرای مثال زیر این تفاوت را به صورت مشهودتری مشاهده خواهید کرد:

نمونه سه

```

<a href="javascript:void(0);" onclick="$('#divTestArea3').slideDown(5000);">Show bo
<a href="javascript:void(0);" onclick="$('#divTestArea3').stop(true);">Stop</a>
<a href="javascript:void(0);" onclick="$('#divTestArea3').stop(true, true);">St
finish</a>
<a href="javascript:void(0);" onclick="$('#divTestArea3').clearQueue().hide();">Res
<div id="divTestArea3" style="padding: 100px; background-color: #89BC38; tex
center; display: none;">
    <b>Hello, world!</b>
</div>

```

با کلیک بر روی لینک "stop" انیمیشن درجا متوقف می شود ولی اگر بر روی لینک "stop but finish" کلیک کنید، انیمیشن فوراً پایان می یابد.

### مقدمه ای بر دستکاری مدل شی گرای سند (HTML مدل DOM)

یکی از امکانات ویژه ی جاوا اسکریپت و در نتیجه جی کوئری، قابلیت مدیریت و دستکاری DOM می باشد DOM. سرواژه ی کلمات Document Object Model بوده و مکانیزمی برای تعامل با سندهای HTML، XHTML یا XML و دسترسی به عناصر درون سند HTML را فراهم می آورد. این مدل در واقع به شما این امکان را می دهد به وسیله ی یک زبان برنامه نویسی (زبان جاوا اسکریپت) درون سندهای html پیمایش نموده و المان های درون آن را دستکاری نمایید. پیمایش در صفحه ی HTML و دستکاری المان های آن توسط جاوا اسکریپت کمی دشوار می باشد. خوشبختانه، jQuery تعدادی متد برای دستکاری و کار با DOM ارائه می کند که فرآیند نام برده را به مراتب آسان تر می سازد. در اولین مثال "Hello, world!" آموزش حاضر، پیدا کردن یک المان و تنظیم متن آن را با جاوا اسکریپت خالص و

همچنین با استفاده از جی کوئری مقایسه کردیم. در مباحث بعدی این برتری جی کوئری در مقایسه با جاوا اسکریپت خالص، به خصوص در سهولت در بکاربردن آن، با وضوح بیشتری به نمایش گذاشته می شود.

### خواندن (get) و تنظیم کردن (Set) محتویات سند [text(), .html(), .val()]

آسان ترین کار ممکن در دستکاری DOM، بازیابی و تنظیم متن (text)، مقادیر (value) و HTML یک صفحه می باشد. این سه آیتم اگرچه در نگاه اول مشابه به نظر می رسد، اما در حقیقت تفاوت هایی با هم دارند. Text نمایش متنی محتویات درونی المان های معمولی در نظر گرفته می شود، مقادیر مربوط به المان های فرم هستند که به سرور ارسال می گردند و HTML نیز در واقع همان text است، با این تفاوت که تمامی تگ ها (markup) را شامل می شود. خوشبختانه، برای هر یک از آیتم های ذکر شده، جی کوئری یک متد ویژه ارائه می کند که هم property ها را بازیابی می کند و هم مقادیر آن ها را تنظیم می نماید. این سه متد عبارتند از text()، .html() و .val(). در مثال زیر تفاوت میان این سه متد و نیز سهولت در استفاده از آن ها به نمایش گذاشته شده است:

نمونه یک

```
1 <script type="text/javascript">
2     $(function () {
3         alert("Text: " + $("#divTest").text());
4         alert("HTML: " + $("#divTest").html());
5         alert("Value: " + $("#divTest").val());
6         alert("Text: " + $("#txtTest").text());
7         alert("HTML: " + $("#txtTest").html());
8         alert("Value: " + $("#txtTest").val());
9     });
10 </script>
```

فراخوانی یکی از متدها بدون اینکه پارامتری به آن ها ارسال شود، صرفا property مد نظر را برمی گرداند. اگر بخواهیم مقدار property را تنظیم کنیم، بایستی به هر یک از این متدها یک پارامتر پاس دهیم، مانند نمونه ی زیر:

نمونه دو

```
1 <script type="text/javascript">
2     $(function () {
3         $("#divText").text("A dynamically set text");
4         $("#divHtml").html("<b><i>A dynamically set HTML
string</i></b>");
5         $("#txtTest").val("A dynamically set value");
6     });
7 </script>
```

می توان با ارسال یک پارامتر اضافی به هر یک از سه متد ذکر شده، آن ها را overload کرد. برای این منظور یک تابع callback به عنوان اولین و تنها پارامتر پاس می دهیم. توابع ذکر شده خود دارای دو پارامتر ورودی به شرح زیر می باشد. پارامتر اول بیان کننده ی اندیس اولین المان در لیست المان های انتخاب شده می باشد. و پارامتر دوم بیان کننده مقدار که درون المان انتخاب شده می باشد و شما می توانید به عنوان خروجی تابع یک رشته را برگردانید که در پایین از تلفیق دو رشته جدید و قدیم این خروجی را تولید کرده ایم:

نمونه سه

```
<script type="text/javascript">
    $(function () {
        $("p").text(function (index, oldText) {
            return "Existing text: " + oldText + ". New text: A dynamically set
text (#" + index + ")";
        });
    });
</script>
```

با سه المان p مثال را شروع می کنیم که تنها تفاوت آن ها در نوشته ای (text) است که نمایش می دهند. سپس هر سه المان مزبور را در کد jQuery انتخاب کرده و نسخه ی overload شده ی متد text() را برای جایگزین کردن نوشته ی کنونی با متن جدید (با استفاده از دو پارامتر ورودی که به تابع نام برده پاس می دهیم) بکار می بریم. این پارامترها عبارتند از: 1. اندیس یا شماره ی مکان قرار گیری المان جاری و 2. محتوای المان text (p جاری). این نوشته ی جدید به عنوان خروجی به jQuery بازگردانده می شود که متن جاری را با متن جدید جایگزین می کند. در اینجا آموزش انجام عملیات get و set در jquery به پایان رسید. همانگونه که دیدید، به سادگی هر چه تمام تر توانستیم که محتوای مورد نظرمان را در صفحه تغییر داده و به آن چیزی که می خواستیم تبدیل کنیم.

#### خواندن (get) و تنظیم (set) مقدار attribute ها [attr()]

در مبحث قبلی با نحوه ی بازیابی و تنظیم متن و محتویات HTML آشنا شدیم. به همین نحو، می توان attribute های یک المان را به آسانی ویرایش نمود. برای این منظور می توان متد attr() را بکار برد. این متد در ساده ترین حالت، یک پارامتر می پذیرد: اسم attribute ای که می خواهیم بازیابی کنیم.

نمونه یک

```
<a href="http://www.google.com" id="aGoogle1">Google Link</a>
<script type="text/javascript">
    $(function () {
        alert($("#aGoogle1").attr("href"));
    });
</script>
```

در این مثال، مقدار خصیصه ی "href" لینک صفحه ی خود را بازیابی کرده و سپس آن را برای کاربر به نمایش می گذاریم. برای ویرایش یک attribute، کافی است یک پارامتر دیگر به تابع مذکور ارسال کنیم:

نمونه دو

```
<a href="http://www.google.com" id="aGoogle2">Google Link</a>
<script type="text/javascript">
    $(function () {
        $("#aGoogle2").attr("href", "http://www.google.co.uk");
    });
</script>
```

با تغییری که اعمال شد، لینک مرورگر را به نسخه ی انگلیسی (مختص به کشور انگلستان Google) هدایت می کند. متد attr() همچنین می تواند یک سری جفت کلید/مقدار را به عنوان پارامتر پذیرفته و از این طریق چندین خصیصه (attribute) را به طور همزمان مقدار دهی کند.

نمونه سه

```
<a href="http://www.google.com" id="aGoogle3">Google Link</a>
<script type="text/javascript">
  $(function () {
    $("#aGoogle3").attr(
      {
        "href": "http://www.google.co.uk",
        "title": "Google.co.uk"
      });
  });
</script>
```

متد attr() همچنین قابل overload شدن می باشد که در آن یک تابع callback به عنوان پارامتر به متد مزبور ارسال می شود. سپس با استفاده از (پارامترهای) این تابع به اندیس المان انتخابی و همچنین مقدار جاری attribute دسترسی پیدا می کنیم و آن را تغییر می دهیم. مثال:

نمونه چهار

```
<a href="http://www.google.com/" class="google">Google.com</a><br>
<a href="http://www.google.co.uk/" class="google">Google UK</a><br>
<a href="http://www.google.de/" class="google">Google DE</a><br>
<script type="text/javascript">
  $(function () {
    $("a.google").attr("href", function (index, oldValue) {
      return oldValue + "img?tab=wi";
    });
  });
</script>
```

با کد فوق، تمامی لینک های (Google المان های a با کلاس google) را طوری مقداردهی می کنیم که بجای صفحه ی پیش فرض به صفحه ی Image search ارجاع دهند. همان طور که مشاهده می کنید، این کار را با افزودن پارامتر جدیدی به خصیصه href ترتیب دادیم.

## خواندن و تنظیم کلاس های CSS

همان طور که attribute ها و محتوای المان های سند HTML را ویرایش کردیم، به همان راحتی هم می توانیم کلاس های CSS المان ها را دستکاری کنیم. jQuery. این زمینه را برای شما فراهم می کند تا به راحتی خصیصه ی style المان ها را با استفاده از متد css() و با ارائه ی چندین متد دیگر، کلاس های یک المان را ویرایش نمایید. ویرایش را از خصیصه ی class المان های HTML آغاز می کنیم. خصیصه ی class تا چندین اسم کلاس می گیرد که ممکن است به کلاس CSS تعریف شده در صفحه ی سبک دهی (Stylesheet) جاری اشاره داشته یا نداشته باشد. غالباً همین طور است، اما گاهی فقط اسم کلاس به المان های خود تخصیص می دهیم تا با استفاده از جی کوئری بتوانیم (بر اساس اسم کلاس) به المان مورد نظر دسترسی داشته باشیم. به عبارتی دیگر با کد جی کوئری به راحتی می توان یک المان

را بر اساس اسم کلاس آن انتخاب نموده و به آن دسترسی پیدا کرد. به منظور تست، چندین انتخابگر ساده ی CSS در Stylesheet خود تعریف کرده ایم:

```
.bold {
    font-weight: bold;
}
.blue {
    color: blue;
}
```

- در اینجا سه متد بسیار پر کاربرد که با کلاس کار می کنند را معرفی می کنیم:
1. `hasClass()`: بررسی می کند آیا المان یا المان های مورد نظر کلاس مشخصی (با `classname` مشخص) را دارند یا خیر. اسم کلاس را به عنوان پارامتر به این متد پاس می دهیم.
  2. `addClass()`: یک کلاس معین را به المان مورد نظر تخصیص می دهد.
  3. `removeClass()`: کلاس مورد نظر را از المان انتخابی حذف می کند، در واقع اسم کلاس که به عنوان پارامتر به متد ارسال شده را از المان انتخاب شده حذف می نماید.

نمونه یک

```
<a href="javascript:void(0);" onclick="ToggleClass(this);">Toggle class</a>
<script type="text/javascript">
    function ToggleClass(sender) {
        if ($(sender).hasClass("bold"))
            $(sender).removeClass("bold");
        else
            $(sender).addClass("bold");
    }
</script>
```

در مثال فوق، به مجرد اینکه که کاربر بر روی لینک کلیک می کند، خود لینک (`this`) را به عنوان پارامتر به متد `toggleClass()` ارسال می کنیم. در آن، بررسی می کنیم آیا `sender` دارای کلاسی به نام `bold` هست یا خیر و در صورت وجود کلاسی با این نام آن را حذف می کنیم، در غیر این صورت آن را به المان مورد نظر اضافه می کنیم. این عملیات بطور مرتب انجام می شود. تیم جی کوئری برای اینکه کاربران مجبور به نوشتن چندین خط کد جهت اجرای همین عملیات ساده نباشند، متد بسیار کارآمد `toggleClass()` را پیاده سازی می کند.

```
<a href="javascript:void(0);"
onclick="$ (this).toggleClass ('bold');">Toggle class</a>
```

می توان چندین المان را انتخاب کرده و چندین کلاس را به طور همزمان به آن اضافه کرده یا از آن حذف نمود.

```
<script type="text/javascript">
  $(function () {
    $("#divTestArea1 span, #divTestArea1 b").addClass("blue");
    $("#divTestArea1 div").addClass("bold blue");
  });
</script>
```

در ابتدا، تگ های span و b را انتخاب می کنیم و سپس یک کلاس به آن اضافه می نماییم و آن کلاسی با نام blue می باشد. در گام بعدی، تگ div را انتخاب می کنیم و به آن دو کلاس که توسط space از هم جدا شده اند، تخصیص می دهیم: کلاس های bold و blue. در شرح متد removeClass نیز باید بگوییم که به همین صورت عمل می کند. نام کلاس ها را به عنوان پارامتر به متد نام برده پاس می دهیم، سپس آن ها را داخل پرانتز متد با خط فاصله از هم جدا می نماییم. حال کلاس های مد نظر از المان دلخواه حذف می گردند.

### متدهای APPEND() و PREPEND()

با امکانات کتابخانه ی jQuery می توان به راحتی المان های جدید به المان های جاری پیوست یا اضافه کرد. در این کتابخانه، متدهایی برای افزودن المان جدید به ابتدا/انتهای المان جاری، دریافت HTML به صورت رشته، المان های DOM و نیز اشیا جی کوئری به عنوان پارامتر وجود دارد. در مثال زیر خواهید دید افزودن المان های جدید به یک لیست با استفاده از توابع prepend و append چقدر آسان می باشد:

### نمونه یک

```
<a href="javascript:void(0);"
onclick="$ ('#olTestList1').append('<li>Appended item</li>');">Append</a>
<a href="javascript:void(0);"
onclick="$ ('#olTestList1').prepend('<li>Prepended item</li>');">Prepend</a>
<ol id="olTestList1">
  <li>Existing item</li>
  <li>Existing item</li>
</ol>
```

در نمونه ی ذکر شده در بالا، دو لینک را مشاهده می کنید. اولین لینک یک آیتم جدید به انتهای المان li اضافه می کند. لینک دوم، برخلاف لینک اول، یک آیتم جدید به ابتدای المان li الحاق می نماید. در این مثال، صرفاً یک قطعه HTML جدید درج می گردد. اما می توانستیم آیتم های جدید را توسط کد jQuery تولید کنیم یا آن را از طریق کد خالص جاوا



اسکرپت و المان های DOM ایجاد نماییم. در واقع هر دو متد append() و prepend() ، قادر به پذیرش تعداد نامحدودی المان به عنوان پارامتر می باشند. در مثال زیر، قابلیت پذیرش تعداد نامحدودی پارامتر و توانایی افزودن المان های جدید به اشکال مختلف را نمایش می دهیم:

نمونه دو

```

1 <a href="javascript:void(0);" onclick="AppendItemsToList();" >Append
2 items</a>
3 <ol id="olTestList2"></ol>
4 <script type="text/javascript">
5     function AppendItemsToList() {
6         var item1 = $("<li></li>").text("Item 1");
7         var item2 = "<li>Item 2</li>";
8         var item3 = document.createElement("li");
9         item3.innerHTML = "Item 3";
10        $("#olTestList2").append(item1, item2, item3);
11    }
</script>

```

همان طور که مشاهده می کنید، item1 المان است که توسط jQuery ایجاد شده. item2 یک رشته ی ساده ی HTML و item3 یک المان DOM تولید شده توسط javascript می باشد. این سه آیتم توسط متد append به انتهای المان (li لیست) پیوست شده است .

دو متد نام برده دارای مشتقاتی می باشند که عبارتند از:

1. appendTo()
2. prependTo()

این دو متد همان کار دو append و prepend را انجام می دهند اما به شیوه ای کاملا متفاوت.

**مثال:**

نمونه سه

```

<a href="javascript:void(0);" onclick="PrependItemsToList();" >Prepend
items</a>
<ol id="olTestList3"></ol>
<script type="text/javascript">
    function PrependItemsToList() {
        $("#olTestList3").prepend($("<li></li>").text("prepend(
item"));
        $("<li></li>").text("prependTo (
item").prependTo("#olTestList3");
    }
</script>

```

در این مثال، آیتم هایی را به انتهای لیست اضافه می کنیم. همان طور که خود مشاهده می کنید، نتیجه یکسان است و تنها تفاوت این دو متد مشتق با متدهای اصلی، در ترتیب انجام کار خلاصه می شود.

توابع before() و after() در jQuery

در مبحث پیشین با نحوه ی افزودن آیتم های جدید در داخل المان، با استفاده از دو متد append() و prepend() آشنا شدیم. اما در برخی موارد لازم است یک یا چند آیتم جدید را به بعد یا قبل المان های جاری پیوست کنیم. کتابخانه ی jQuery دو متد before() و after() را برای این منظور در نظر گرفته است. کاربرد دو متد مزبور را در مثال زیر مشاهده می کنید:

نمونه یک

```
<a href="javascript:void(0);"
onclick="$('input.test1').before('<i>Before</i>');">Before</a>
<a href="javascript:void(0);"
onclick="$('input.test1').after('<b>After</b>');">After</a>
<br><br>
<input type="text" class="test1" value="Input 1" name="txtInput1"><br>
<input type="text" class="test1" value="Input 2" name="txtInput2"><br>
```

بسته به اینکه بر روی کدام لینک کلیک کنید، یک تگ italic یا bold، با استفاده از کلاس "test1"، به قبل یا بعد المان input اضافه می شود. درست مانند توابع append() و prepend()، دو متد before() و after() نیز به شما اجازه می دهند از رشته های HTML، المان های DOM و اشیا jQuery به عنوان پارامتر ارسالی استفاده کنید (همچنین تعداد آرگومان ورودی این دو تابع می تواند نامحدود باشد). نمونه:

نمونه دو

```
<a href="javascript:void(0);" onclick="InsertElements();" >Insert
elements</a>
<br><br>
<span id="spnTest2">Hello world? </span>

<script type="text/javascript">
function InsertElements() {
var element1 = $("<b></b>").text("Hello ");
var element2 = "<i>there </i>";
var element3 = document.createElement("u");
element3.innerHTML = "jQuery!";

$("#spnTest2").after(element1, element2, element3);
}
</script>
```

در مثال فوق یک شی jQuery، رشته ی HTML، یک المان JavaScript DOM ایجاد کرده و سپس با استفاده از متد after() تمامی آن ها به بعد المان span اضافه می کنیم. توابع before() و after() دارای مشتقاتی هستند که عبارتند از insertBefore() و insertAfter(). نتیجه ای که از فراخوانی دو متد مشتق حاصل می شود با توابع اصلی آن ها یکسان است. تنها تفاوت در ترتیب انجام کار می باشد.

**آموزش کاربرد کنترل AJAX در ASP.NET**

AJAX سرواژه ی Asynchronous JavaScript and XML می باشد. ایجکس در واقع یک فناوری یا تکنولوژی ( corss-platform به آن دسته از نرم افزارها گفته می شود که در چندین بستر اجرای نرم افزار رایانه ای یا همان platform قابل اجرا هستند ) که زمان پاسخ دهی ( response time ) را به گونه ی قابل توجهی کاهش می دهد. به عبارت دیگر ایجکس مجموعه ای از استانداردها و فناوری های ویژه ی وب است که با بهره گیری از آنها می توان برنامه هایی مبتنی بر وب نوشت که به آسانی با کاربران تعامل ( interaction ) داشته باشند. با استفاده از این فناوریها و با کمک انتقال تکه های کوچک داده و اطلاعات از Server ، صفحات وب از حالت منفعل خارج می شوند و واکنشهایی مناسب با رویدادها انجام می دهند. با استفاده از این معماری، صفحات وب تعامل بسیار خوبی با کاربران خواهند داشت. علاوه بر آن، مهم ترین مزیت این معماری این است که دیگر برای انجام هر کاری، احتیاج نیست صفحه وب دوباره بارگذاری شود. کنترل های سمت سرور ایجکس ( AJAX server controls ) به صفحه ی وب اسکریپت اضافه می شوند که توسط مرورگر پردازش شده، سپس اجرا می گردد.

با این حال درست مانند دیگر کنترل های سمت سرور ASP.NET ، کنترل های سمت سرور ایجکس AJAX ( server controls ) نیز می توانند دارای متدها و event handler های ( مدیریت کننده ی رخداد ) مرتبط با آن متدها باشند که در سمت سرور ( server side ) پردازش می شوند. control toolbox در محیط برنامه نویسی یکپارچه ی visual studio شامل مجموعه ای از کنترل ها می باشد که 'AJAX Extensions' یا به عبارتی دیگر افزونه های ایجکس خوانده می شوند.

## کنترل ScriptManager

کنترل ScriptManager مهم ترین control تلقی می گردد، به گونه ای که کنترل های دیگر موجود در صفحه برای عملکرد صحیح به وجود ScriptManager احتیاج دارند. کنترل ScriptManager ، وظیفه مدیریت پردازشها ( script ) بر روی صفحات ASP.Net ای که امکان و قابلیت Ajax آنها فعال سازی شده است را بر عهده دارد. به صورت پیش فرض، کنترل ScriptManager اسکریپت های به وقوع پیوسته متعلق به عملیات ایجکس در صفحه را ، با مجموعه سایر پردازش های ( script ) صفحه مرتبط می کند. این کار امکان استفاده از قابلیت های مرورگر در سمت سرویس گیرنده ( client-side ) و بروز رسانی بخش های مختلف صفحه بدون بارگذاری مجدد یا تجدید کامل آن را می دهد. نحوه ی کلی نگارش ( syntax ) کنترل مذکور به ترتیب زیر می باشد:

```
<asp:ScriptManager ID="ScriptManager1" runat="server"> </asp:ScriptManager>
```

چنانچه وب سایتی ایجاد کنید که تکنولوژی ایجکس در آن بکار گرفته شده ( 'Ajax Enabled site' ) یا یک فرم وب ایجکس ( 'AJAX Web Form' ) از طریق پنجره ی محاوره ( dialog box ) به آن اضافه گردیده، در آن صورت مشاهده خواهید که فرم وب شما به صورت خودکار کنترل script manager را در خود خواهد داشت .

## کنترل ScriptManager

ترتیب کلیه ی اسکریپت های سمت سرویس گیرنده ( client-side script ) را به ازای تمامی کنترل های سمت سرویس دهنده ( server side control ) می دهد.

## کنترل UpdatePanel

UpdatePanel control در حقیقت یک کنترل دربرگیرنده ( container control ) است که از کلاس Control مشتق شده است . کنترل ذکر شده به مثابه ی یک ظرف ( نگهدارنده ) برای کنترل های فرزند child control عمل می کند . این کنترل همچنین دارای رابط ( interface ) مختص خود نیست . هنگامی که یکی از کنترل های موجود در آن، در صدد انجام و راه اندازی postback برمی آید، UpdatePanel مداخله کرده و باعث می شود ارسال ( post ) به صورت ناهمزمان 1 صورت گرفته ( آغاز شده ) و تنها قسمت مورد نظر صفحه بروز رسانی شود.

کنترل UpdatePanel این امکان را به برنامه نویس می دهد، تا برنامه های تحت وب کارآمد و با رابط کاربری همانند برنامه های رایانه های رومیزی بسازید . به وسیله این کنترل می توانید قسمت یا بخش هایی از صفحه را که می خواهید اطلاعات آن تغییر کرده را بروز رسانی کرده و بدین وسیله از تجدید یا بارگذاری دوباره و Postback کل صفحه به server جلوگیری کنید.

UpdatePanel control، با جدا سازی بخش مورد نظر از صفحه، حین به روز رسانی خود ، فقط آن قسمت های مرتبط را آپدیت کرده و مانع از بارگذاری مجدد یا بازسازی و Postback کل صفحه ی مورد نظر می شود. برای مثال اگر یک دکمه ی موجود در کنترل نام برده کلیک شود، تنها کنترل های داخل update panel تحت تاثیر قرار گرفته و کنترل های موجود در بخش های دیگر صفحه هیچ تغییری نمی کنند . به این فرایند postback جزئی / بخشی یا ناهمگام ( asynchronous / partial postback ) گفته می شود.

مثال:

ابتدا یک فرم وب ایجکس ( AJAX web form ) به ( application برنامه ی کاربردی ) خود اضافه کنید . این فرم به صورت پیش فرض کنترل Script Manger را در خود دارد . سپس یک update panel داخل آن قرار دهید ( درج کنید ) .

حال یک کنترل button به همراه یک کنترل label داخل update panel control جای گذاری کنید .

بار دیگر یک کنترل button به همراه کنترل lable اضافه می کنیم اما این بار خارج از panel مربوطه.

1 asynchronous

Source file :

```
<form id="form2" runat="server"> <div> <asp:ScriptManager ID="ScriptManager2"
runat="server" /> </div> <asp:UpdatePanel ID="UpdatePanel2" runat="server">
<ContentTemplate> <asp:Button ID="Button1" runat="server"
onclick="btnpartial_Click" Text="PartialPostBack"/> <br /> <br /> <asp:Label
ID="Label1" runat="server"></asp:Label> </ContentTemplate> </asp:UpdatePanel>
<p> </p> <p>Outside the Update Panel</p> <p> <asp:Button ID="Button2"
runat="server" onclick="btntotal_Click" Text="TotalPostBack" /> </p> <asp:Label
ID="Label2" runat="server"></asp:Label> </form>
```

هر دو کنترل های button دارای کد یکسان برای event handler می باشند:

```
string time = DateTime.Now.ToLongTimeString(); lblpartial.Text = "Showing time
from panel" + time; lbltotal.Text = "Showing time from outside" + time;
```

همان طور که مشاهده می کنید زمانی که صفحه اجرا می شود، اگر روی دکمه ی total post back کلیک شود،  
زمان بر

روی هر دو label بروز رسانی می شود. اما اگر دکمه ی partial post back از panel موردنظر کلیک شود، فقط  
label موجود در کنترل update panel بروز رسانی می شود.

یک صفحه می تواند دربردارنده ی چندین panel باشد که هر یک خود شامل کنترل های دیگر مانند grid بوده و  
بخشهای مختلف داده را نمایش دهد.

هنگامی که یک total post back رخ می دهد، محتوا update panel ( content ) به صورت پیش فرض بروز  
رسانی میشود. این حالت ( mode ) پیش فرض را می توان با اصلاح خاصیت UpdateMode ( property )  
تغییر داد.

## خاصیت های کنترل UpdatePanel

جدول زیر برخی از خاصیت های کنترل مذکور را برای شما لیست کرده است:

:ChildrenAsTriggers

مقداری بازیابی کرده یا تنظیم می کند که نشانگر این است که آیا postback های آمده از کنترل های فرزند ( child control ) محتوای panel را بروز رسانی می کنند یا خیر. چنانچه مایلید postback های آمده از child controlها باعث بروز رسانی محتویات panel شوند، این خاصیت را روی مقدار true تنظیم کنید.

:ContentTemplate

این خاصیت قالبی ( template ) که محتویات کنترل UpdatePanel را تعریف می کند، بازیابی یا مقداردهی  
می کند.

.ContentTemplateContainer

این property یک شیء control بازیابی می کند که می توان به آن به صورت برنامه نویسی / با برنامه نویسی (programmatically) کنترل فرزند اضافه کرد.

### **:IsInPartialRendering**

مقداری را برمی گرداند که تعیین می کند آیا کنترل UpdatePanel در نتیجه ی روی دادن postback ناهمگام (asynchronous postback) بروز رسانی می شود یا خیر. مشخص می کند آیا panel در نتیجه (به عنوان بخشی از postback) جزئی (partial postback) آپدیت می شود یا خیر.

### **:RenderMode**

مقداری بازیابی یا تنظیم می کند که نشان می دهد محتوای کنترل UpdatePanel در یک تگ <div> یا HTML <span> محصور شده است یا خیر. چنانچه به خاصیت RenderMode مقدار Inline تخصیص داده شود، محتوای کنترل UpdatePanel داخل تگ <span> render می شود. اگر خاصیت RenderMode روی مقدار Block تنظیم شده باشد، داخل <div> render می شود.

UpdateMode مقداری بازیابی یا تنظیم می کند که نشان می دهد چه زمانی محتوای کنترل UpdatePanel بروز رسانی می شود.

این خاصیت (mode rendering) حالت ارائه ی ماشینی تصویر (را با مشخص کردن برخی شرایط بازیابی و مقداردهی می کند.

Triggers آرایه ای از اشیاء trigger تعریف کرده که هر یک مربوطه به یک رخداد است. این رخدادهای باعث می شوند panel به صورت خودکار بروز رسانی شود. مقداری برمی گرداند که نشانگر آرایه یا مجموعه ای از اشیاء از نوع AsyncPostBackTrigger و PostBackTrigger می باشد.

خاصیت Triggers آرایه ای از کل trigger ها که برای کنترل UpdatePanel تعریف شده است برمی گرداند.

توابع کنترل UpdatePanel

متدهای کنترل نام برده به شرح زیر می باشند:

متد شرح

CreateContentTemplateContainer این تابع یک شیء Control ایجاد می کند که به عنوان یک ظرف

برای کنترل

های فرزند عمل می کند. کنترل های فرزند محتوایات کنترل

UpdatePanel را تعریف می کنند.

CreateControlCollection آرایه ای از کلیه ی کنترل های موجود در UpdatePanel control را بازیابی می کند.

Initialize در صورتی که (partial-page rendering ارائه ی ماشینی تصویر به صورت جزئی) فعال شده باشد، آرایه ای از trigger های کنترل UpdatePanel را مقداردهی اولیه می کند.

Update این متد باعث می شود محتوای کنترل UpdatePanel بروز رسانی شود. عملکرد کنترل UpdatePanel به مقادیر خاصیت های UpdateMode و ChildrenAsTriggers بستگی دارد.

UpdateMode ChildrenAsTriggers نتیجه

Always False پارامترهای غیر مجاز

Always True چنانچه کل صفحه بازسازی شود یا یک کنترل فرزند به هنگام postback باعث بازسازی صفحه شود،

UpdatePanel نیز refresh می شود.

Conditional False چنانچه کل صفحه بازسازی / بروز رسانی (refresh )

شود یا یک (triggering control) کنترلی که به عنوان

تعریف شده باشد (بیرون از آن باعث بازسازی trigger

صفحه شود، کنترل UpdatePanel نیز refresh می

شود.

Conditional True اگر کل صفحه بروز رسانی شود یا کنترل فرزند باعث

شود یک (postback بازگردانی اطلاعات) رخ دهد و یا

(triggering control) کنترلی که به عنوان trigger

تعریف شده باشد (بیرون از صفحه منجر به بازسازی

صفحه شود، UpdatePanel نیز بروز رسانی می شود.

کنترل UpdateProgress

به منظور نمایش یک نمایه تصویری یا یک پیام، که میزان پیشرفت فرایند عملیات بارگذاری و به روز رسانی صفحه را نشان می دهد، مورد استفاده قرار می گیرد. برای مثال می توان به زمانی اشاره کرد که کاربر در حال ثبت ورود log in می باشد یا زمانی که سرور در حال اجرای برخی عملیات مربوط به پایگاه داده بود و کاربر منتظر دریافت پاسخ از سرویس دهنده (server) می باشد، اشاره کرد.

نحوه ی نگارش کنترل UpdateProgress به صورت زیر می باشد:

```
<asp:UpdateProgress ID="UpdateProgress2" runat="server" DynamicLayout="true"
AssociatedUpdatePanelID="UpdatePanel1" > <ProgressTemplate> Loading... </ProgressTemplate>
</asp:UpdateProgress>
```

تکه کد فوق یک پیغام ساده را داخل تگ ProgressTemplate نمایش می دهد. البته بجای پیغام می توان یک تصویر یا هر کنترل مربوطه ی دیگری قرار داد. کنترل UpdateProgress به ازای هر postback ناهمگام ( asynchronous رخ می دهد یک بار نمایش داده می شود، مگر اینکه با استفاده از خاصیت AssociatedUpdatePanelID به یک updatepanel متصل شده باشد) شما می توانید به کمک property UpdatePanel یا خاصیت AssociatedUpdatePanelID کنترل UpdateProgress را به یک کنترل UpdatePanel متصل کنید. هنگامی که عمل Postback یا بازسازی ( refresh ) توسط کنترل UpdatePanel انجام می شود، کنترل UpdateProgress مربوط به آن نیز نمایش داده می شود. اگر کنترل UpdateProgress را به هیچ کنترل UpdatePanel خاصی متصل نکنید، با هر بار رخداد Postback یا به روز رسانی، آن کنترل در صفحه نمایش داده خواهد شد.

### خاصیت های کنترل UpdateProgress

AssociatedUpdatePanelID (شناسه ID) (مرتبط با کنترل update panel را بازگردانی کرده یا مقداردهی می کند).

Attributes (خاصیت های CSS) (attribute) (کنترل UpdateProgress را بازیابی کرده یا مقداردهی می کند).

DisplayAfter (مدت زمانی که باید سپری شود تا ( progress template قالب پیشروی پروسه)

نمایش داده شود را بر حسب میلی ثانیه خوانده و مقداردهی می کند. پیش فرض 500 می باشد.

DynamicLayout (مقداری خوانده و تنظیم می کند که نشان می دهد آیا progress template به

صورت پویا render ( dynamic ) می شود یا خیر.

این خاصیت تعیین می کند آیا فضای لازم برای محتوای progress template تنها زمانی به صفحه تخصیص داده شود که لازم است یا خیر.

ProgressTemplate

قالبی ( template ) که محتوای کنترل UpdateProgress را تعریف می کند، مقداردهی کرده یا برمی گرداند.

نشانگر قالبی است که حین عمل postback ناهمگام نمایش داده می شود. این فرایند زمان بیشتری نسبت به

DisplayAfter طول می کشد.

### توابع کنترل UpdateProgress :

GetScriptDescriptors

فهرستی از مولفه ها ( component )، عملکردها ( behavior ) و کنترل های سمت سرورس گیرنده

( client control ) بازیابی می کند که برای راه اندازی قابلیت های کنترل UpdateProgress سمت سرورس



گیرنده لازم و ضروری می باشد . این متد را برای تعریف کامپوننت ها، عملکردها یا کنترل های client از کتابخانه ی ایجکس مایکروسافت ( Microsoft AJAX Library ) که server control ها به آن نیاز دارند، پیاده سازی کنید.

### GetScriptReferences

آرایه ای از اشیاء ScriptReference بازمی گرداند که تعریف کننده ی منابع اسکریپت ( script resources ) بوده و کنترل مورد نظر به آن نیاز دارد.

### کنترل Timer

از این کنترل به منظور راه اندازی عمل postback به صورت خودکار استفاده می شود .کنترل Timer به صورت متناوب ، در فواصل زمانی تعیین شده ، صفحه را Postback می کند . همچنین اگر این کنترل را به همراه یک کنترل UpdatePanel به کار ببرید ، می توانید تنها آن بخشی از صفحه که کنترل UpdatePanel شامل آن می شود را به صورت متناوب ، بروز رسانی کنید .این کار از دو طریق زیر امکان پذیر می باشد:

#### 1. متصل کردن و مقداردهی خاصیت Triggers کنترل : UpdatePanel

```
<Triggers> <asp:AsyncPostBackTrigger ControlID="btnpanel2" EventName="Click" /> </Triggers>
```

2. با قرار دادن یک کنترل timer به صورت مستقیم داخل UpdatePanel که به عنوان trigger کنترل فرزند عمل می کند .توجه داشته باشید که تنها یک کنترل timer می تواند به عنوان trigger برای چندین UpdatePanel عمل کند.

```
<asp:UpdatePanel ID="UpdatePanel2" runat="server" UpdateMode="Always">  
<ContentTemplate> <asp:Timer ID="Timer2" runat="server" Interval="1000">  
</asp:Timer> <asp:Label ID="Label1" runat="server" Height="101px"  
style="width:304px" > </asp:Label> </ContentTemplate> </asp:UpdatePanel>
```

مثال:

```
<a href="javascript:void(0);" onclick="InsertElementsBefore();">Insert elements</a>  
<br><br>  
<span id="spnTest3">Hello world? </span>
```

```
<script type="text/javascript">  
function InsertElementsBefore() {  
    $("#spnTest3").before("<i></i>").text("before() ");  
    "<b></b>").text("insertBefore() ").insertBefore("#spnTest3");  
}  
</script>
```

## ویرایشگری به نام TinyMCE

در این مقاله قصد داریم نحوه ایجاد ویرایشگر rich textbox را در برنامه MVC توضیح دهیم. در واقع در اینجا به یک ویرایشگر سریعتر و قابل اعتماد تر از rich textbox نیاز داریم ، بنابراین از ویرایشگری به نام TinyMCE که رایگان و بهینه تر از RichTextBox است، استفاده میکنیم.

TinyMCE چیست؟

این پلت فرمی مستقل مبتنی بر وب جاوا اسکریپت، ویرایشگر HTML WYSIWYG که به عنوان منبع باز تحت LGPL منتشر شده است . این ویرایشگر ما را قادر به تبدیل text area HTML و یا دیگر عناصر HTML به یک

نمونه ویرایشگر میکند . این یک ویرایشگر جعبه متن جاوا اسکریپت غنی است.

ویژگی های این ویرایشگر:

Text Formatting

Tables Designer

Media File Editor

Cut, Copy, Paste Options.

List Designer

Hyperlink

نحوه استفاده از TinyMCE در پروژه

روی پروژه خود راست کلیک کرده و بروی Manage Nuget Package کلیک میکنیم . پس از آن یک کادر محاوره ای باز خواهد شد، TinyMCE را در آن جستجو و بسته "TinyMCE.Mvc" را دریافت کنید . "TinyMCE.Mvc" را نصب میکنیم.

پس از نصب TinyMCE.Mvc ، در داخل پوشه اسکریپت در پروژه ، یک پوشه با نام TinyMCE ایجاد میشود .

همانطور که در TinyMCE یک ویرایشگر جاوا اسکریپت است ، بنابراین در داخل پوشه TinyMCE بیشتر فایل های جاوا اسکریپت قرار میگیرند.

بعد از نصب TinyMCE در پروژه جدا از پوشه اسکریپت ، یک پوشه دیگر در داخل View > Shared "EditorTemplates" ایجاد میشود.

در داخل این editor Template ، دو فایل cshtml با نام tinymce\_full.cshtml و

tinymce\_full\_compressed.cshtml که هر دو محتویات فایل یکسانی دارند، اما یکی از نسخه ها فشرده و دیگری نسخه کامل است را دریافت میکنید . این فایل ها برای ما بسیار مهم است چرا که با این فایل ها ما می توانیم ویرایشگر خود را با ویژگی هایی که ما می خواهیم تغییر دهیم.

در حال حاضر برای استفاده از این ویرایشگر یک کلاس مدل با نام About ایجاد میکنیم که در آن فیلدهایی که

برای ذخیره در دیتابیس می‌خواهیم؛ را مشخص می‌کنیم. بنابراین کد زیر را در کلاس می‌نویسیم.

```
using System;
using System.ComponentModel.DataAnnotations;
using System.Web.Mvc;
namespace RTBEditor.Models
{
    public class About
    {
        public int UserID { get; set; }
        [AllowHtml]
        [UIHint("tinymce_full")]
        [Display(Name = "About")]
        public string AboutMe { get; set; }
    }
}
```

در اینجا از دو ویژگی استفاده می‌کنیم.

[AllowHtml]

این ویژگی بسیار مهم است. از آنجا که در هر جایی از، که بخواهید Text Editor می‌توانید کد HTML تایپ کنید، به همین دلایل و دلایل امنیتی MVC اجازه ارسال پست HTML را نمی‌دهد. توسط این ویژگی MVC، اجازه ارسال محتوای HTML برای پروپرتی About را می‌دهد.

[UIHint("tinymce\_full")]

این ویژگی اجازه استفاده از قالب tinymce\_fill.cshtml را می‌دهد. اگر شما قصد داریم از قالب فشرده استفاده کنید می‌توانید از ویژگی [UIHint("tinymce\_full\_compressed")] استفاده کنید. که به شما اجازه ی استفاده از قالب فشرده را می‌دهد.

چرا از ویژگی [UIHint("tinymce\_full")] استفاده می‌کنیم.

بدلیل اینکه تغییر چیزی در این قالب و قالب نسخه فشرده دشوار است، از این ویژگی استفاده می‌کنیم.

نحوه استفاده در : View برای نمایش ویرایشگر، کد زیر را در View وارد می‌کنیم.

```
@Html.EditorFor(model => model.AboutMe)
```

کدهای View

```
@model RTBEditor.Models.About
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Index</h2>
@using (Html.BeginForm())
```

```

{
@Html.AntiForgeryToken()
<div class="form-horizontal">
<h4>About</h4>
<hr />
@Html.ValidationSummary(true)
<div class="form-group">
@Html.LabelFor(model => model.AboutMe, new { @class = "col-md-12" })
<div class="col-md-12">
@Html.EditorFor(model => model.AboutMe)
@Html.ValidationMessageFor(model => model.AboutMe)
</div>
</div>
<div class="form-group">
<div class="col-md-offset-2 col-md-10">
<input type="submit" value="Create" class="btn btn-default" />
</div>
</div>
</div>
}
</div>
@Html.ActionLink("Back to List", "Index")
</div>
@section Scripts {
@Scripts.Render("~/bundles/jqueryval")
}

```

خروجی:

نحوه حذف گزینه های اضافی قالب در این ویرایشگر:

برای حذف برخی از ویژگی های این ویرایشگر شما می توانید `tinymce_full.cshtml` را تغییر دهید . هر توابعی

که بخواهید را می توانید ویرایش کامل و ویژگی های آن را تغییر دهید.

توجه داشته باشید `tinymce_full.cshtml` : تغییر خواهد کرد چرا که از ویژگی

`[UIHint("tinymce_full")]` در مدل استفاده

شده است . اگر شما از ویژگی `[UIHint("tinymce_full_compressed")]` استفاده کنید ، فایل

`tinymce_full_compressed.cshtml` تغییر میکند.

برای از بین بردن برخی ویژگی ها می توانید ، کد های مورد نظر را در `tinymce_full.cshtml` کامنت کنید.

// Theme options

```

theme_advanced_buttons1 :
"save,newdocument,|,bold,italic,underline,strikethrough,|,justifyleft,justifycenter,justif
yright,justifyfull,styleselect,formatselect,fontselect,fontsizeselect",
theme_advanced_buttons2 :
"cut,copy,paste,pastetext,pasteword,|,search,replace,|,bullist,numlist,|,outdent,indent,bl
ockquote,|,undo,redo,|,link,unlink,anchor,image,cleanup,help,code,|,insertdate,insertti
me,preview,|,forecolor,backcolor",
//theme_advanced_buttons3 :
"tablecontrols,|,hr,removeformat,visualaid,|,sub,sup,|,charmap,emotions,iespell,media,a
dvhr,|,print,|,ltr,rtl,|,fullscreen",
//theme_advanced_buttons4 :
"insertlayer,moveforward,movebackward,absolute,|,styleprops,|,cite,abbr,acronym,del,
ins,attribs,|,visualchars,nonbreaking,template,pagebreak,restoredraft,codehighlighting,
netadvimage",
theme_advanced_toolbar_location : "top",
theme_advanced_toolbar_align : "left",
theme_advanced_statusbar_location : "bottom",
theme_advanced_resizing : false,

```

همانطور که میبینید theme\_advanced\_buttons3 و theme\_advanced\_buttons4 را کامنت کرده ایم. خروجی پس از کامنت کردن مانند شکل زیر میشود.