|  |
| --- |
| **آموزش زبان برنامه نویسی** |
| Windows Form Application |
|  |
|  |
|  |
|  |
| **www.w3-farsi.com** |

## 

## 

## 

عنوان..............................................................................................................صفحه

[برنامهنویسیویژوال 2](#_Toc381395282)

[ایجادیکبرنامهویندوزیساده 3](#_Toc381395283)

[کنترلکنندهرویداد 12](#_Toc381395284)

[جداکردنمحیططراحیازمحیطکدنویسی 16](#_Toc381395285)

[کلاس MessageBox 17](#_Toc381395286)

[کنترلها 19](#_Toc381395287)

[نامگذاریکنترلها 27](#_Toc381395288)

[ویندوزفرم 28](#_Toc381395289)

[کنترلButton 36](#_Toc381395290)

[کنترل Label 38](#_Toc381395291)

[کنترلTextBox 39](#_Toc381395292)

[کنترلRichTextBox 42](#_Toc381395293)

[کنترل RadioButton 48](#_Toc381395294)

[کنترل CheckBox 49](#_Toc381395295)

[کنترلهای Panel و GroupBox 51](#_Toc381395296)

[کنترل ComboBox 53](#_Toc381395297)

[کنترل ListBox 56](#_Toc381395298)

[کنترل CheckedListBox 59](#_Toc381395299)

[کنترل NumericUpDown 63](#_Toc381395300)

[کنترل PictureBox 65](#_Toc381395301)

[کنترل LinkLable 67](#_Toc381395302)

[کنترل MonthCalendar 70](#_Toc381395303)

[کنترل DateTimePicker 73](#_Toc381395304)

[کنترل TabControl 77](#_Toc381395305)

[کنترل ToolTip 85](#_Toc381395306)

[کنترل TrackBar 88](#_Toc381395307)

[کنترل Timer 91](#_Toc381395308)

[طراحیفرمهایویندوزی 93](#_Toc381395309)

[خاصیت Anchor 100](#_Toc381395310)

[خاصیت Dock 105](#_Toc381395311)

[خاصیت TabIndex 108](#_Toc381395312)

[اضافهکردنمنوبهفرم 109](#_Toc381395313)

[ساختنوارابزار 117](#_Toc381395314)

[کنترل ToolStripContainer 129](#_Toc381395316)

[کادرهایمحاورهای 134](#_Toc381395317)

[کنترل ColorDialog 137](#_Toc381395318)

[کنترلFontDialog 140](#_Toc381395319)

[کنترل FolderBrowserDialog 143](#_Toc381395320)

[کنترل OpenFileDialog 147](#_Toc381395321)

[کنترل SaveFileDialog 152](#_Toc381395322)

[رویدادهایماوس 157](#_Toc381395323)

[رویدادهایکیبورد 163](#_Toc381395324)

## برنامه نویسی ویژوال

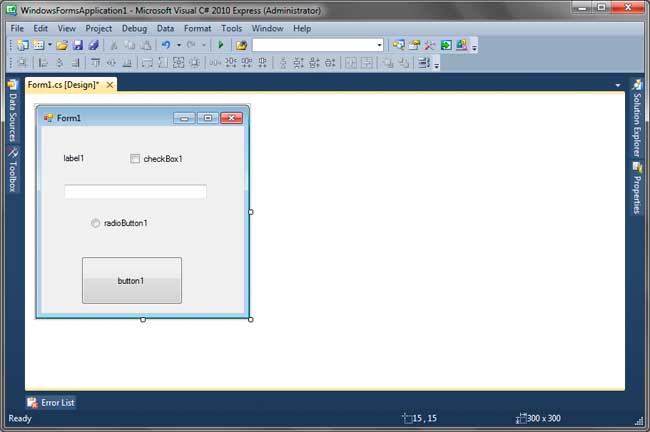
رابط گرافیکی کاربر  (GUI)به کاربر اجازه می دهد که با استفاده از اجزای بصری مختلف با برنامه ارتباط برقرار کند.در روزهای اولیه دنیای کامپیوتر برنامه ها  مبتنی بر متن بودند، بدین معنی که شما باید دستورات متنی زیادی برای ایجاد یک برنامه مفید تایپ می کردید و این کار مستلزم حفظ کردن یک لیست طولانی  از دستورات بود.برنامه های نرم افزاری امروزی دارای رابط گرافیکی هستند. این رابط گرافیکی تقریبا در همه برنامه هایی که امروزه با آنها سر و کار دارید به چشم می خورد. یک رابط گرافیکی حرفه ای باید جذاب و ساده باشد.ایجاد یک برنامه با رابط کاربری قبلا یک کار سخت و کسل کننده بود. مثلا برای ایجاد یک پنجره ساده که یک متن را نمایش دهد نیاز بود که تعداد زیادی کد تایپ شود.

اما با ورود ویژوال استودیو این کار راحت شد. برنامه نویسی ویژوال ایجاد برنامه های گرافیکی را راحت کرد، به طوری که شما می توانید محیط برنامه خود را با کشیدن کنترل های لازم از جعبه ابزار به نوعی “نقاشی” کنید. کنترل ها اجزای بصری هستند که GUI یا رابط گرافیکی را تشکیل می دهند.

نمونه ای از کنترل ها عبارتند از  buttons، text boxes،  labels، check boxes  و  radio buttons.

کلمه “visual” در# Visual C از مفهوم برنامه نویسی بصری گرفته شده است. مایکروسافت از کلمه Windows Forms برای نشان دادن هر پنجره در یک برنامه استفاده می کند.

برنامه ویژوال استودیو اجازه ایجاد هر چه راحت تر برنامه های ویندوزی را به شما می دهد. برای ایجاد و طراحی فرم ها می توانید از حالت Design استفاده کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/visual-programming-01.jpg)

شکل بالا حالت طراحی (Design) در Visual C# Express 2010 را نشان می دهد.

در این شکل فرم ویندوزی و کنترل هایی که بر روی آن کشیده شده اند نشان داده شده است.

در حالت طراحی، شما می توانید چگونگی به نظر رسیدن فرم را در حین اجرای برنامه مشاهده کنید.

کدهایی که باعث ایجاد و مقداردهی به کنترل ها می شوند از دید کاربر مخفی هستند، بنابراین شما می توانید بر روی کارکرد برنامه بیشتر تمرکز کنید.

همچنین می توانید از ابزارهای ویژوال استودیو مانند چپ چین یا راست چین کردن، تغییر اندازه و ….. برای طراحی کنترل ها استفاده کرد.

## ایجاد یک برنامه ویندوزی ساده

وقت آن رسیده است که برنامه نویسی ویژوال را تجربه کنید.

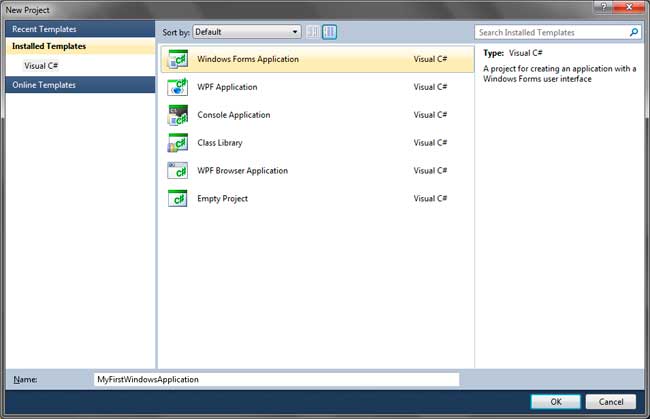
به این نکته توجه کنید که این درس به شما نحوه ایجاد یک برنامه ویندوزی که در آن از یک اداره کننده رویداد (event-handling) استفاده شده است، را نشان می دهد.من هر مرحله را به صورت گام به گام انجام داده و به طور مختصر در مورد هر کدام توضیح می دهم.مفاهیم فرم، کنترل، کنترل کننده رویداد (event-handling) و برخی قسمتهای Visual Studio در طراحی یک برنامه ویندوزی مورد استفاده قرار می گیرد و در مورد هر کدام از آنها در درسهای مربوطه شان بحث خواهد شد.

برنامه ویژوال سی شارپ را باز کنید و به مسیر File>New Project بروید.

سپس از لیست ظاهر شده مانند شکل زیر گزینه Windows Forms Application را انتخاب نمایید.

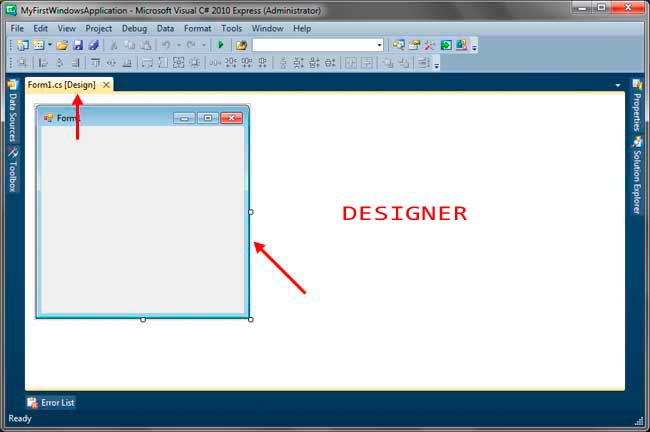
یک Windows Forms Application نوعی برنامه است که دارای یک رابط کاربر گرافیکی می باشد.

نام پروژه MyFirstWindowsApplication است.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/creating-simple-form-011.jpg)

بر روی دکمه OK کلیک کنید تا یک فرم خالی ظاهر شود.

تب انتخاب شده نشان می دهد که شما در حال تماشای فایل Form1.cs از محیط طراحی هستید.

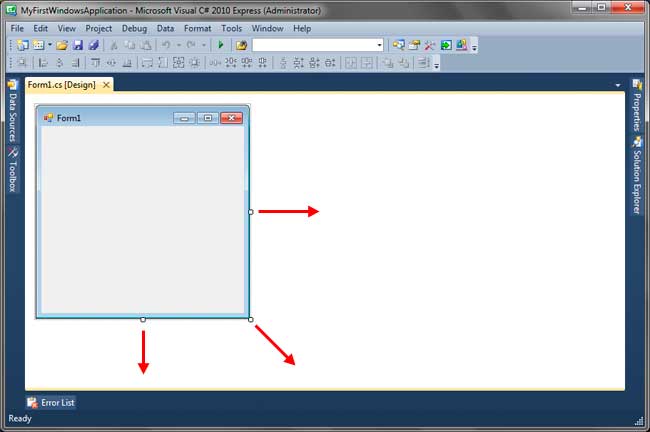
[](http://www.w3-farsi.com/wp-content/uploads/2013/09/creating-simple-form-02.jpg)

در حال حاضر بر روی کد فایلی که باعث فزوده شدن قابلیت به فرم می شود تمرکز می کنیم.

Form  را می توان در دو حالت مشاهده کرد، حالت طراحی و حالت کد.

در حالت طراحی می توانید فرم، کنترل های ویژوال و غیر ویژوال را مشاهده نمایید.

شکل زیر یک فرم را نشان می دهد که شما می توانید با استفاده از دستگیره های کناری (که با فلش نشان داده شده است)در محیط طراحی اندازه آنرا تغییر دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/creating-simple-form-03.jpg)

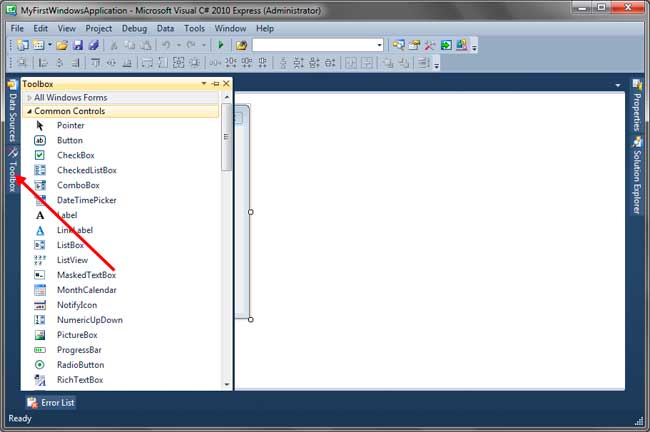
**اضافه کردن کنترل ها به فرم**

همه کنترل ها در جعبه ابزار یا Toolbox قرار دارند.

Toolbox از مسیر سربرگ جعبه ابزار که به صورت پیش فرض در سمت چپ محیط IDE قرار دارد قابل دسترسی است.

اگر جعبه ابزار را پیدا نکردید می توانید از مسیر View>Other Windows>Toolbox به آن دسترسی یابید.

برای نمایش جعبه ابزار کافیست با موس بر روی آن لحظه ای توقف کرده و یا کلیک نمایید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/creating-simple-form-04.jpg)

Toolbox

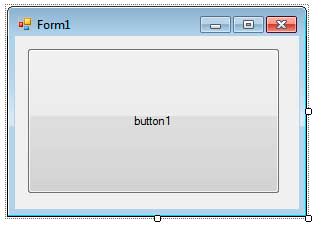
به قسمتهایی تقسیم شده است، مثلا کنترل های پر کاربرد در قسمت Common Controls قرار دارند. برای باز کردن یک دسته و نمایش کنترل های آن کافیست بر روی نام دسته کلیک کنید.  Toolbox در حالت پیشفرض مخفی می شود (خاصیت auto-hide).

اگر شما این حالت را دوست ندارید می توانید بر روی آیکون سنجاق شکل کنار دکمه close جعبه ابزار کلیک کنید تا Toolbox ثابت بماند و مخفی نشود.برای اضافه کردن یک کنترل به فرم می توانید بر روی آن دوبار کلیک کنید تا به فرم اضافه شود همچنین می توانید کنترل را از جعبه ابزار بر روی فرم کشیده و رها کنید (drag and drop).

به این نکته توجه کنید که شما فقط می توانید کنترل ها را به ناحیه کاربر فرم اضافه کنید.

ناحیه کاربر ناحیه خالی فرم است. برای حذف یک کنترل از روی فرم آن را انتخاب کنید و سپس دکمه Delete بر روی صفحه کلید را فشار دهید.

یک کنترل دکمه (button) را به فرم اضافه کنید.بیشتر کنترل ها قابلیت تغییر اندازه را دارند.حال مکان و اندازه کنترل ذکر شده را مانند شکل زیر تغییر دهید.

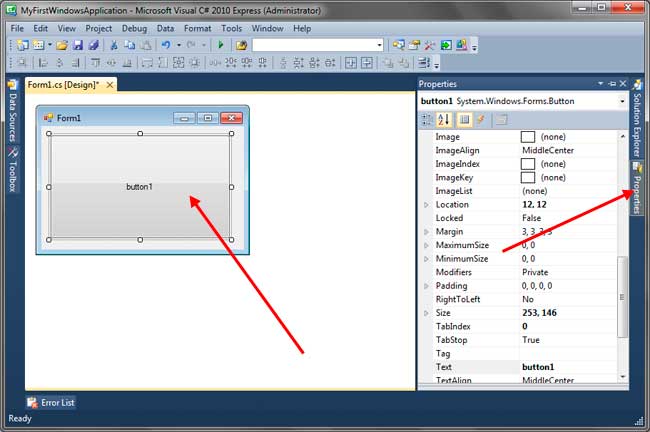
[](http://www.w3-farsi.com/wp-content/uploads/2013/09/creating-simple-form-05.jpg)

**تغییر خاصیت کنترل ها**

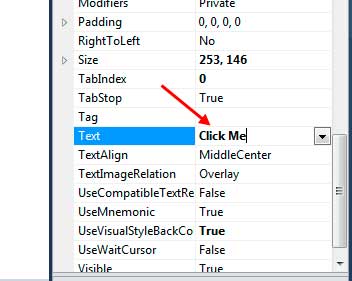
شما می توانید برخی از خواص فرم و کنترل ها را تغییر دهید.برای این کار از پنجره Properties برای مشاهده و تغییر مقادیر همه خواص در دسترس یک کنترل انتخاب شده در صفحه طراحی استفاده می کنیم.به این نکته توجه کنید که برخی از خواص در پنجره Properties نمایش داده نمی شوند و فقط از طریق کد می توان به آنها دست یافت.

یک کنترل را با استفاده از کلیک کردن بر روی آن در صفحه طراحی انتخاب کنید.

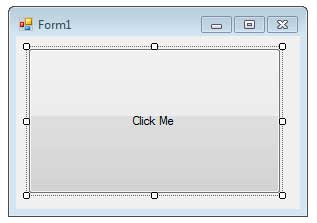
برای اینکه به شما نشان دهیم که چطور می توان خاصیت یک کنترل را تغییر داد ، بر روی دکمه ای در روی فرم کلیک کنید و سپس به پنجره Properties بروید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/creating-simple-form-06.jpg)

خاصیت Text را پیدا کنید و آن را به Click Me تغییر دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/creating-simple-form-07.jpg)

مشاهده می کنید که متن کنترل button تغییر می کند.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/creating-simple-form-08.jpg)

همچنین می توانید با کلیک بر روی فرم خاصیت Text آن را نیز تغییر دهید.

به این نکته توجه کنید که برای انتخاب فرم باید بر روی یک منقطه خالی از آن کلیک کنید نه یک کنترل که بر روی  آن قرار دارد.

**اضافه کردن یک کنترل کننده رویداد به کنترل**

در بخش آخر این آموزش قصد داریم که به شما نحوه اضافه کردن یک کنترل کننده رویداد را به کنترل آموزش دهیم.

کنترل کنند رویداد یا Event Handler ، قسمتی از برنامه است که مسئولیت کنترل رویدادها را بر عهده دارد.

رویداد وقتی به وقوع می پیوندد که یک اتقاف معین رخ دهد.

Event Handler هم برای کنترل کردن یک رویداد به کار می رود.

در مورد جزییات اداره کردن رویدادها در یک بخش جداگانه بحث خواهیم کرد.هر کنترل یک رویداد پیشفرض مخصوص به خود دارد.

به عنوان مثال رویداد پیشفرض کنترل دکمه، Click و رویداد پیشفرض کنترل فرم  Load می باشد.

کنترل کننده های رویداد متدهایی هستند که به رویداد وابسته اند و وقتی اجرا می شوند که رویدادها رخ دهند.

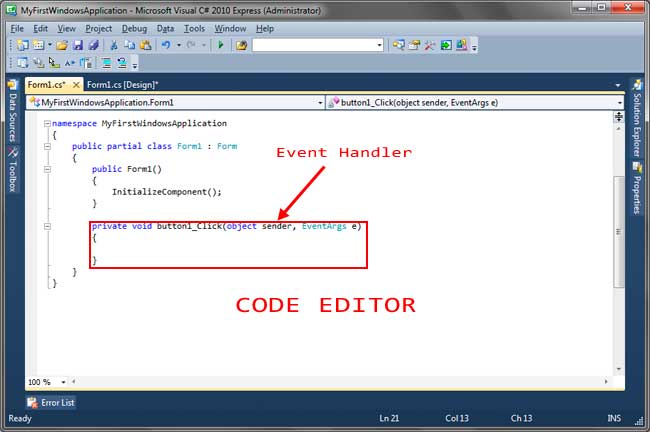
راه ساده برای اضافه کردن یک کنترل کننده رویداد دابل کلیک کردن بر روی کنترل در محیط طراحی می باشد.

ناگفته نماند  که این کار باعث اضافه شدن کنترل کننده رویداد به رویداد پیشفرض می شود. برای روشن شدن این مطلب بر روی دکمه button در محیط طراحی کلیک کنید.

ویژوال استودیو به صورت خودکار یک کنترل کننده رویداد ایجاد کرده و آن را به رویداد Click کنترل می چسباند.

همزمان با ایجاد کنترل کننده رویداد شما وارد قسمت کد نویسی می شوید و نشانگر موس نیز در داخل اداره کننده رویداد قرار می گیرد.

حال همه چیزی که شما نیاز دارید نوشتن کدی است که هنگام وقوع رویداد اجرا می شود.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/creating-simple-form-09.jpg)

در مورد سایر قسمت های کد در درس های آینده بحث خواهیم کرد.

کد MessageBox.Show(“You clicked the button!”); را در داخل کنترل کننده رویداد مربوط به رویداد Click به صورت زیر بنویسید.

private void button1\_Click(object sender, EventArgs e)

{

MessageBox.Show("You clicked the button!");

}

کلاس MessageBox به شما اجازه می دهد که یک جعبه متن را برای نمایش یک پیغام یا اطلاعات فوری به کاربر فراخوانی کنید.

متد Show باعث نمایش یک متن خاص در جعبه متن می شود.

در مورد کلاس MessageBox در درسهای آینده توضیح می دهیم.

برنامه را اجرا کرده و بر روی دکمه کلیک کنید.

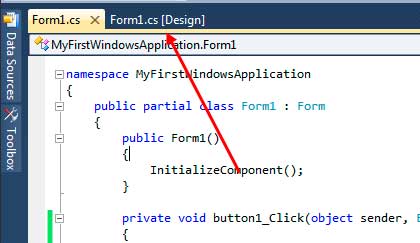
با اجرای برنامه یک پنجره پیغام که شامل پیامی است که به عنوان آرگومان به متد Show ارسال کرده اید نمایش داده خواهد شد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/creating-simple-form-10.jpg)

راه دیگر برای اضافه کردن یک کنترل کننده رویداد به رویداد کنترل ها مخصوصا به رویدادهای غیر پیشفرض استفاده از پنجره Properties است.

برای روشن شدن این مطلب اجازه دهید که رویداد Load را به فرم اضافه کنیم.

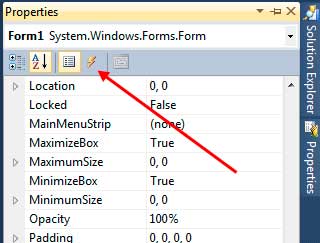
برای برگشتن به محیط طراحی هم می توانیم  بر روی تب Design کلیک کرده وهم از کلیدهای ترکیبی Shift + F7 استفاده کنیم.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/creating-simple-form-11.jpg)

کنترل form انتخاب کرده و سپس از  پنجره Properties رویداد های آن را پیدا کنید.

رویدادها در پنجره Properties به وسیله یک آیکون رعد و برق نمایش داده شده اند.

اگر این آیکون را نیافتید مطمئن شوید که کنترل در محیط طراحی انتخاب شده باشد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/creating-simple-form-12.jpg)

حال پنجره Properties لیستی از رویدادهای مربوط به کنترل انتخاب شده در محیط برنامه نویسی را نشان می دهد.

رویداد Load مربوط به کنترل form را پیدا کنید.

با کلیک بر روی منوی باز شونده (combo box) کنار آن، لیستی از متدهای موجود برای این رویداد نشان داده خواهد شد.

سپس می توانید یک متد برای الحاق به این رویداد انتخاب کنید.

همچنین می توان با دابل کلیک بر روی رویداد انتخاب شده در پنجره Properties یک کنترل کننده رویداد جدید ایجاد کرد که در این صورت به صورت خودکاروارد محیط کدنویسی شده و کنترل کننده رویداد مناسب برای شما ایجاد می شود.

کد پر رنگ شده زیر را در رویداد Load بنویسید.

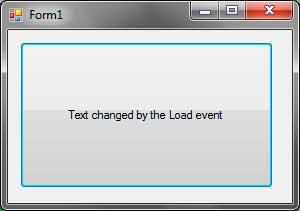
private void Form1\_Load(object sender, EventArgs e)

{

button1.Text = "Text changed by the Load event";

}

این دستور خاصیت Text مربوط به دکمه واقع بر روی فرم را تغییر می دهد. رویداد Load فرم وقتی روی می دهد که بارگذاری فرم تمام شود. بنابراین با هر بار اجرا ی برنامه ، متن داخل کنترل button تغییر می کند.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/creating-simple-form-13.jpg)

حال شما با موفقیت یک گرداننده رویداد را با استفاده از ابزارهای موجود در Visual Studio و Visual C# Express ایجاد کردید.

## کنترل کننده رویداد

رابط گرافیکی کاربر در دات نت و ویژوال سی شارپ از مکانیزم کنترل کننده رویداد برای کنترل رویدادها که در هنگام اجرای برنامه به وقوع می پیوندند استفاده می کند.رویدادها رفتارهایی یا اتفاقاتی هستند که هنگام اجرای برنامه به وقوع می پیوندند.کنترل رویداد فرایند نظارت بر وقوع یک رویداد مشخص می باشد.فرم های ویندوزی از کنترل کننده رویداد برای اضافه کردن یک قابلیت و پاسخ به کاربر استفاده می کنند..بدون کنترل کننده رویداد، فرم ها و رابط گرافیکی تا حد زیادی بدون استفاده هستند.در این درس فرض بر این است که شما بر مفاهیم  delegates و events (رویداد) تسلط دارید. رویدادها با استفاده از یک delegate به عنوان یک نوع تعریف می شوند.Delegate ها آدرس متدها را در خود ذخیره می کنند.

در مثال زیر یک delegate و یک event تعریف شده اند.

public delegate voidSampleEventHandler(int);

public eventSampleDelegate SampleEvent;

بر اساس تعریف بالا Delegate آدرس متدهایی را قبول می کند که دارای مقدار برگشتی نیستند و یک آرگومان ساده از نوع int قبول می کنند.سپس از این نوع Delegate برای ایجاد event مان استفاده می کنیم.حال یک کنترل کننده رویداد به رویداد اضافه می کنیم.کنترل کننده های رویداد متدهای متناظر با نوع Delegate رویداد هستند و هنگام وقوع رویداد اجرا می شوند.آنها (کنترل کننده های رویداد) چسبیده به یک event هستند و هنگامی که رویداد به وقع می پیوندد اجرا می شوند.می توان چندین کنترل کننده های رویداد را به یک event متصل کرد تا هنگام وقوع رویداد اجرا شوند. برای این کار ابتدا باید کنترل کننده رویداد را ایجاد کنید. بعد از ایجاد ،مطمئن شوید که امضای آن با delegate ی که رویداد از آن استفاده می کند مطابق باشد. به عنوان مثال به delegate مثال بالا توجه کنید که نوع برگشتی void یک پارامتر از نوع int است،پس کنترل کننده رویداد ما نیز باید دارای نوع برگشتی void یک پارامتر از نوع int باشد (به این نکته توجه کنید که سطح دسترسی مهم نیست).

public void ShowMessage(int number)

{

MessageBox.Show("Hello World");

}

سپس می توان با استفاده از عملگر =+ یک رویداد را متصل کرد :

SampleEvent += newSampleEventHandler(ShowMessage);

برای فعال کردن رویداد به همان روشی که متد ها را فراخوانی می کردیم ان را صدا زده و آرگومانی را که لازم دارد به آن ارسال می کنیم.

ShowMessage(3);

**کنترل کننده رویداد در فرم های ویندوزی**

برای شرح استفاده از رویدادها در فرم های ویندوزی، یک فرم ویندوزی جدید ایجاد کرده و نام آن را it EventHandling بگذارید. بر روی فرم دوبار کلیک کرده تا ویژوال استودیو به صورت خودکار یک کنترل کننده رویداد ایجاد کرده و ان را به رویداد Load فرم متصل کند. رویداد Load یک نوع delegate از کنترل کننده رویداد است. بیشتر رویدادهای کنترل ها یک نوع delegate از System.EventHandler دارند.

مثال زیر تعریفی از EventHandler delegate می باشد:

public delegate voidEventHandler(object sender, EventArgs e)

همانطور که مشاهده می کنید ،delegate مثال بالا دارای مقدار برگشتی نیست و دارای دو پارامتر است، یکی object و دیگری یک نمونه از EventArgs.

object sender نشان دهنده کنترلی است که رویداد را فعال می کند(بعدا توضیح داده می شود).

دومین آرگومان نمونه ای از کلاس EventArgs می باشد و میتوان آن را به عنوان آرگومان رویداد در نظر گرفت که حاوی دادهایی در مورد رویدادی که اتفاق افتاده است می باشد .

EventArgs در اصل یک کلاس پایه است و دارای اعضای مفیدی نمی باشد.برخی از رویداد ها  آرگومان هایی دارند که از کلاس EventArgs مشتق شده اند و دارای خاصیت های مفیدی هستند که می تواند توسط Event Handler یا کنترل کننده رویداد مورد استفاده قرار بگیرد.

به این نکته توجه کنید که امضای کنترل کننده رویدادی که به وسیله Visual Studio ایجاد می شود با امضای EventHandler یکسان است.

private void Form1\_Load(object sender, EventArgs e)

{

}

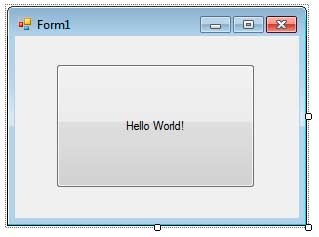
اگر می خواهید یک کنترل کننده رویداد به صورت دستی ایجاد کنید، باید مطمئن شوید امضای آن با امضای نماینده ی (Delegate) رویداد یکسان باشد.در ویژوال استودیو کنترل کننده های رویداد بر طبق یک قرار داد نامگذاری می شوند. هنگام نامگذاری آنها ابتدا نام کنترل (که مشخص کننده خاصیت Name کنترل است)، سپس علامت زیر خط و بعد از آن نام رویداد می آید مانند : Form1\_Load

می توان این قرارداد را زمانی که کنترل کننده رویداد به وسیله چند رویداد مورد استفاده قرار می گیرد، نادیده گرفت که در آینده توضیح داده خواهد شد.مشاهده کردید که نماینده EventHandler دارای ۲ پارامتر است ، که اولین پارامتر یک شی را نشان می دهد که همان کنترلی است که رویداد را ارسال کرده است( object e).

از آن جایی که نوع این پارامتر object است ، هر کنترلی می تواند منبع ارسال رویداد باشد چون هر کنترل یک شی است که از کلاس پایه object مشتق می شود.از آنجایی که ارسال کننده رویداد(کنترل) به نوع object تبدیل می شود ، برای استفاده مجدد از خصوصیات کنترل باید آن  را با استفاده از عمل cast تبدیل کنیم ( تولید کنیم).

برای روشن شدن مطلب به مثال زیر توجه کنید.

ابتدا یک کنترل Button را روی فرم قرار دهید و سپس خاصیت Text آن را به “Hello World!” تغییر دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/event-handling-01.jpg)

بر روی کنترل دوبار کلیک کنید تا یک کنترل کننده رویداد برای رویداد کلیک آن ایجاد شود.

مانند رویداد Load، رویداد Click نیز یک کنترل کننده رویداد دارد.کد زیر را در داخل کنترل کننده رویداد بنویسید :

private void button1\_Click(object sender, EventArgs e)

{

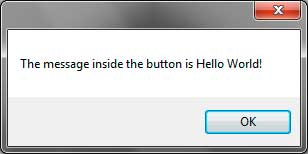
Button source = (Button)sender;

MessageBox.Show("The message inside the button is " + source.Text);

}

برنامه را اجرا کرده و بر روی دکمه کلیک کنید.

یک جعبه متن نمایش داده خواهد شد که به متن داخل کنترل دکمه می گوید که به رویداد Click  برود .

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/event-handling-02.jpg)

خط اول کنترل کننده رویداد شی sender را با استفاده از عمل cast  به یک دکمه تبدیل می کند بنابراین می توانیم به خاصیت Text دکمه دسترسی داشته باشیم.

سپس متد Show از کلاس MessageBox را برای نشان دادن مقدار خاصیت Text دکمه که به رویداد فرستاده شده است، فراخوانی می کنیم.

**آرگومان های رویداد**

دومین پارامتر که  بسته به نوع event متفاوت است یک آرگومان رویداد است.رایجترین نوع این پارامتر کلاس پایه EventArgs می باشد .

هیچ خاصیتی که شما بتوانید در کنترل کننده رویداد خود از آن استفاده کنید ندارد.

مثالی از یک رویداد که دارای آرگومان رویداد متفاوتی است ، رویداد MouseClick از کنترل button است که نسخه بهتری از رویداد Click می باشد. از آنجایی که این رویداد(MouseClick) رویداد پیشفرض نیست ، برای دسترسی به آن باید از بخش events از پنجره Properties اقدام نمایید. مطمئن شوید که کنترل button در قسمت طراحی فرم انتخاب شده است. بر روی آیکون events ( که به شکل رعد و برق است) کلیک کنید. ابتدا کنترل کننده رویداد Click را حذف کنید تا با این رویداد(MouseClick) تداخل پیدا نکند. رویداد Click را در پنجره events پیدا کنید و مقدار جلوی این رویداد را در پنجره حذف کنید.

سپس رویداد MouseClick را یافته و بر روی آن دوبار کلیک کنید. رویداد MouseClick یک نوع Delegate متفاوت از MouseEventHandler دارد.

این نماینده امضای مشابه ای با EventHandler دارد اما دارای آرگومان رویداد (event argument) متفاوتی با آن به نام MouseEventArgs می باشد.

 کلاس MouseEventArgs از کلاس پایه EventArgs مشتق شده است.

این کلاس دارای خواص مفیدی از جمله اینکه کدام دکمه موس فشار داده شده ، تعداد کلیک ها ، مقدار چرخش موس و مختصات مکانی که رویداد کلیک در آن اتفاق افتاده است می باشد.

از کد زیر به عنوان کنترل کننده رویداد MouseClick استفاده کنید.

private void button1\_MouseClick(object sender, MouseEventArgs e)

{

MessageBox.Show("You clicked at point (" + e.X + ", " + e.Y + ")");

}

کد بالا با استفاده از پارامتر event argument  به مختصات x و y  نقطه ای که با ماوس بر روی آن کلیک شده است دست می باید.

خروجی زیر بستگی به این دارد که شما بر روی چه نقطه ای از دکمه کلیک کنید.

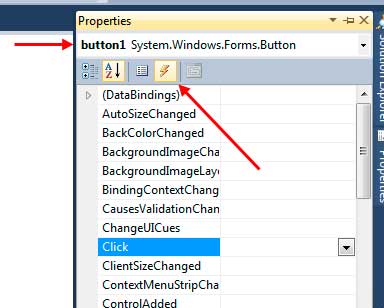
[](http://www.w3-farsi.com/wp-content/uploads/2013/09/event-handling-03.jpg)

انواع زیادی event argument وجود دارد که هر کدام خواص مفیدی برای رویدادی که اتفاق می افتد پیشنهاد می دهند.

**استفاده از پنجره خواص برای اضافه کردن کنترل کننده رویداد**

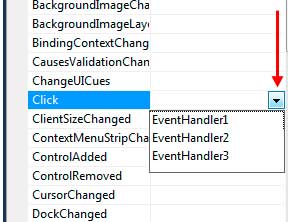
همانطور که می دانید صرف نظر از دوبار کلیک کردن بر روی کنترل ها برای ایجاد کنترل کننده رویداد می توانیم از پنجره Properties برای این کار نیز استفاده کنیم.

بر روی آیکونی که شبیه به یک جرقه است کلیک کنید تا وارد بخش رویداد های پنجره Properties شوید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/event-handling-04.jpg)

از منوی باز شونده ای که با فلش  قرمز در شکل بالا نشان داده شده است برای انتخاب کنترل ها در محیط طراحی استفاده می شود. استفاده از این منو زمانی مفید است که شما بخواهید یک کنترل غیر قابل رویت و یا بسیار کوچک را انتخاب کنید. رویداد مورد نظر را پیدا کنید. برای اضافه کردن کنترل کننده رویداد به آن می توانید بر روی نام رویداد دوبار کلیک کنید تا کنترل کننده رویداد مربوط به آن ایجاد شود.

همچنین اگر از قبل کنترل کننده های رویداد دیگری نیز ایجاد کرده اید می توانید مانند شکل زیر به آنها دسترسی داشته باشید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/event-handling-05.jpg)

**استفاده از یک کنترل کننده رویداد برای چندین رویداد**

به این نکته توجه کنید که می توان از یک کنترل کننده رویداد برای چندین رویداد که دارای نماینده (delegate) یکسانی هستند استفاده کرد. برای انجام این کار لازم است که یک کنترل کننده رویداد با یک امضای مناسب ایجاد کنید. سپس به بخش رویدادها واقع در پنجره Properties رفته و رویدادی را که می خواهید پیدا کنید. به جای دوبار کلیک بر روی آن کنترل کننده رویدادی را که ایجاد کرده اید انتخاب کنید. حال رویداد دیگر را که می خواهید انتخاب کرده و دوباره کنترل کننده رویدادی مذکور را برای این رویداد انتخاب کنید.

## 

## جدا کردن محیط طراحی از محیط کدنویسی

وقتی یک فرم ویندوزی جدید ایجاد می کنیم، یک کلاس که از System.Windows.Forms.Form ارث بری می کند تولید می شود.

این کلاس به دو فایل جدا  و وابسته به هم تقسیم می شود و یک کلاس تکه تکه (Partial class) را به وجود می آورد.

کلاسهای تکه تکه(Partial classes) به شما اجازه تعریف جداگانه یک کلاس در فایل های مختلف در داخل یک پروژه وفضای نام یکسان را می دهند.

با معرفی کلاس های partial در .NET 3.5 ویژوال استودیو قدرت جداسازی محیط طراحی از محیط کد نویسی را کسب کرد.

این قابلیت به برنامه نویس اجازه می دهد که بر روی بخش کدنویسی برنامه تمرکز کند.

به عنوان مثال نام فایلهای یک فرم به نام FormName در محیط های کدنویسی و طراحی به صورت زیر است:

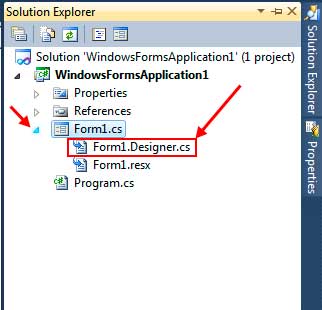
FormName.cs

FormName.Designer.cs

یک فایل سومی هم با پسوند .regx وجود دارد که در آینده در باره آن توضیح داده خواهد شد.

بعد از ایجاد یک فرم، اضافه کردن کنترل و دستکاری خواص همه کدها همه در یک فایل تا حدودی مخفی با پسوند  .Designer.cs نوشته می شوند.

برای مشاهده این فایل می توانید از Solution Explorer استفاده کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/design-functionality-01.jpg)

با دوبار کلیک بر روی این فایل به شما اجازه مشاهده همه کدهای ایجاد شده توسط ویژوال استودیو داده می شود.

این کدها شامل متدهایی برای مرتب سازی و مقداردهی به کنترل ها می باشند. شما می توانید کنترل هایی را که تعریف شده اند و خواصی  مانند محل و متن کنترل ، که شما در پنجره ی Properties به آنها مقادیری را اختصاص داده اید را مشاهده نمایید. همچنین قسمت Event handler را که به رویداد های  کنترل ها وصل شده است را مشاهده می کنید.

اگر شما نمیتوانید قسمت کد را ببینید، به طور پیش فرض مخفی است و به قسمت  “Windows Forms Designer generated code” متصل است برای نمایش آن بر روی آیکن + کلیک کنید تا  قسمت کد نمایش داده شود.

در آینده مشاهده خواهید کرد که مقدار دهی کنترل ها و خواص و رویدادهای مربوط به آنها در داخل متدی به نام InitializeComponent صورت می گیرد.

این متد در داخل سازنده فرم واقع در کدهای اصلی کلاس فرم فراخوانی می شود.

کد های نوشته شده در فایل Designer از چشم شما مخفی هستند چون Visual Studio می خواهد که شما به جای نوشتن کد به صورت دستی از پنجره Properties و محیط طراحی برای ایجاد یک برنامه ویندوزی استفاده نمایید.

## 

## کلاس MessageBox

System.Windows.Forms.MessageBox یک کلاس استاتیک است که از آن برای نشان دادن یک پیغام فوری، اطلاعات و یا یک هشدار به کاربران استفاده می شود.

برای نشان دادن یک پیغام به راحتی می توان از متد Show کلاس MessageBox استفاده نمایید. ساده ترین حالت متد Show این است که یک رشته متنی را به عنوان آرگومان قبول می کند و آن را نمایش می دهد.

MessageBox.Show("Hello World!");

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/message-box-12-19-10-001.jpg)

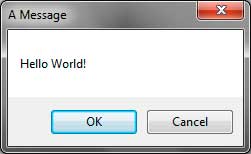
شما همچنین می توانید به راحتی و با استفاده از یکی دیگر از سربارگذاریهای متد Show  یک عنوان برای جعبه پیامتان بگذارید.

MessageBox.Show("Hello World!", "A Message");

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/message-box-12-19-10-01.jpg)

اگر نخواهید از دکمه OK پیشفرض جعبه متن استفاده کنید می توانید با استفاده از System.Windows.Forms.MessageBoxButtons آنرا تغییر دهید.

MessageBox.Show("Hello World!", "A Message", MessageBoxButtons.OKCancel);

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/message-box-12-19-10-02.jpg)

 جدول زیر اعضای MessageBoxButtons را نشان می دهد :

| عضو | دکمه هایی که نمایش می دهد |
| --- | --- |
| AbortRetryIgnore | Abort, Retry, Ignore |
| OK | OK |
| OKCancel | OK, Cancel |
| RetryCancel | Retry, Cancel |
| YesNo | Yes, No |
| YesNoCancel | Yes, No, Cancel |

متد Show() یک مقدار را از System.Windows.Forms.DialogResult بر می گرداند.

تشخیص اینکه چه دکمه ای توسط شما در جعبه متن فشار داده می شود مفید است.

به عنوان مثال اگر بر روی دکمه Yes در جعبه پیام کلیک کنید، متد Show() مقدار DialogResult.Yes را بر می گرداند.

DialogResult result;

result = MessageBox.Show("What is your choice?");

if (result == DialogResult.Yes)

{

//You pressed the Yes button

}

if (result == DialogResult.No)

{

//You pressed the No button

}

لطفا به این نکته توجه کنید که کلاس Form یک خاصیت DialogResult دارد و آن را با System.Windows.Forms.DialogResult اشتباه نگیرید.

می توان به جعبه پیام برای نشان دادن معنی و مفهوم آن یک آیکون اضافه کرد.

می توانید این کار را با استفاده از نوع شمارشی MessageBoxIcon انجام دهید.

MessageBox.Show("Hello World!", "A Message",

MessageBoxButtons.OK, MessageBoxIcon.Information);

در جدول زیر انواع آیکونها و کاربرد آنها در جعبه پیام نشان داده شده است :

| آیکون | عضو | استفاده |
| --- | --- | --- |
| [icon-information](http://www.w3-farsi.com/wp-content/uploads/2013/09/icon-information.jpg) | Asterisk Information | برای نشان دادن اطلاعات به کاربر |
| [icon-error](http://www.w3-farsi.com/wp-content/uploads/2013/09/icon-error.jpg) | Error Hand Stop | برای نشان دادن یک پیغام خطا |
| [icon-warning](http://www.w3-farsi.com/wp-content/uploads/2013/09/icon-warning.jpg) | Exclamation Warning | برای نشان دادن یک هشدار |
| [icon-question](http://www.w3-farsi.com/wp-content/uploads/2013/09/icon-question.jpg) | Question | برای سوال کردن از کاربر |

اگر بخواهید که جعبه متن هیچ گونه آیکونی نداشته باشد می توانید از MessageBoxIcon.None استفاده کنید.

نوع شمارشی MessageBoxDefaultButton دکمه پیشفرضی را که هنگام فشرده شدن دکمه Enter باید عمل کند را تعیین می کند و فقط دارای ۴ عضو است Button1، Button2، Button3 و Button4.

به عنوان مثال در جعبه متنی که دارای دکمه های OK و Cancel است استفاده از MessageBoxDefaultButton.Button1 باعث می شود که دکمه OK به صورت پیشفرض درآید.

 یعنی هنگامی که جعبه متن نشان داده شد با زدن دکمه Enter، دکمه OK فشرده می شود :

MessageBox.Show("Hello World!", "A Message", MessageBoxButtons.OKCancel, MessageBoxDefaultButton.

## کنترل ها

کنترل ها اجزای بصری هستند که به وسیله ی آنها محیط های گرافیکی (GUI)ساخته می شود. هرچیزی که شما در یک محیط گرافیکی می بینید یک کنترل است ، حتی محیط فرم نیز یک کنترل است. کنترل ها در دسته های مختلفی در قسمت نوار ابزار قرار دارند. بیشتر کنترل ها از کلاس پایه System.Windows.Forms.Control ارث بری می کنند که دارای خواصف متدها و رویدادهای متفاوتی برای آنها می باشد.

#### خواص کنترل ها

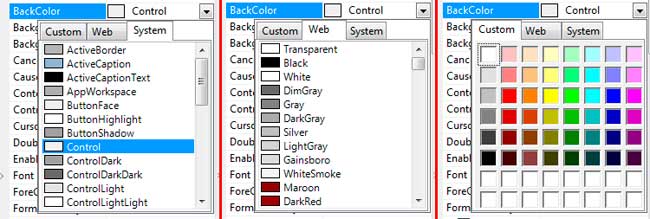
در قسمت زیر بعضی از مشخصه های مفید کلاس Control  را مشاهده می کنید.

| خاصیت | توضیح |
| --- | --- |
| Anchor | این گزینه مشخص می کند که کنترل ها وقتی که سایز فرم تغییر می کند چگونه محل و سایزشان تغییر کند. |
| AutoSize | وقتی که مقدار آن برابر با True باشد، سایز کنترل به صورت اتوماتیک براساس  محتوای آن تغییر می کند. |
| BackColor | رنگ قسمت زمینه ی کنترل را مشخص می کند. |
| BackgroundImage | به شما اجازه می دهد که یک عکس به زمینه ی کنترل خود اضافه کنید. |
| BackgroundImageLayout | حالت عکس زمینه ی کنترل را مشخص می کند. |
| ContextMenuStrip | به شما اجازه می دهد که یک توضیح را به کنترل خود در مورد آن اضافه کنید. |
| Controls | مجموعه ای از کنترل های مرتبط با کنترل مربوطه را نشان می دهد. |
| Dock | کنترل را به یکی از لبه های پنجره می چسباند. |
| Enabled | این قسمت مشخص می کند که کنترل فعال است یا نه. اگر مقدار آن False  باشد کنترل غیرفعال می شود. |
| ForeColor | رنگ پیش زمینه ی کنترل و همچنین رنگ قلم کنترل را مشخص می کند. |
| Height | طول کنترل را بر مبنای پیکسل مشخص می کند. |
| Location | محل کنترل بر اساس در بر دارنده ی آن را مشخص میکند. |
| Locked | مشخص می کند که کنترل در حالت Design می تواند جابجا شود و  یا تغییر  سایز دهد یا نه. |
| MaximumSize | حداکثر اندازه ای که کنترل میتواند داشته باشد را مشخص می کند. |
| Margin | فاصله ی بین لبه های کنترل مربوطه و لبه ی کنترل دیگر را مشخص می کند. |
| MinimumSize | کمترین سایز کنترل را مشخص می کند. |
| Name | نام کنترل را مشخص میکند.این نام برای اشاره به کنترل در زمان کد نویسی استفاده می شود. |
| Parent | والد کنترل را مشخص می کند. |
| Right | فاصله ی بین ضلع راست کنترل و ضلع چپ در بردارنده ی آن را مشخص می کند. |
| Size | سایز کنترل را مشخص می کند.این بخش شامل طول و عرض کنترل است. |
| TabIndex | با اختصاص یک عدد یه این خاصیت مشخص می کنیم که با چند بار زدن دکمه Tab، کنترل فوکوس بگیرد. |
| TabStop | مشخص می کند که یک کنترل می تواند توسط دکمه ی Tab مورد دستیابی قرار گیرد یا خیر. |
| Text | متنی که در داخل کنترل قرار می گیرد. |
| TextAlign | نحوه ی قرار گیری متن در داخل کنترل را مشخص می کند. |
| Top | فاصله ی بین ضلع بالایی کنترل و ضلع بالایی در بر دارنده ی آن را مشخص می کند. |
| Visible | نمایان بودن و نمایان نبودن کنترل را مشخص می کند. |
| Width | عرض کنترل براساس پیکسل را مشخص می کند. |

 مهمترین خاصیت  در جدول بالا خاصیت  Name است. این مشخصه به شما اجازه می دهد که از آن نام برای ارجاع به کنترل در قسمت کد نویسی استفاده کنید. در این قسمت به بحث در مورد بعضی از مشخصه هایی که در بیشتر کنترل ها وجود دارند میپردازیم.

#### تغییر رنگ زمینه ی کنترل

ما از مشخصه ی BackColor کنترل برای تغییر رنگ زمینه ی آن استفاده می کنیم. مشخصه ی BackColor را در پنجره ی  Properties پیدا کنید و بر روی نوار کرکره ای کنار آن کلیک کنید. شما یک پنجره با سه سربرگ مشاهده می کنید.هر سربرگ حاوی رنگ های مختلفی است.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/controls-01.jpg)

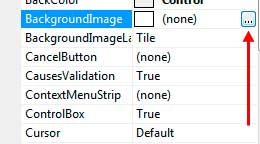
سربرگ System  شامل رنگ هایی است که سیستم عامل به طور پیش فرض برای رنگ آمیزی کنترل ها از آنها استفاده می کند.

سربرگ Web  شامل رنگ هایی است که برای استفاده در صفحات وب مناسب هستند.

سربرگCustom  شامل رنگهای بسیار زیادی است.در این سر برگ شما میتوانید مقادیر  RGB آن را در نوارها مربوطه وارد کنید.

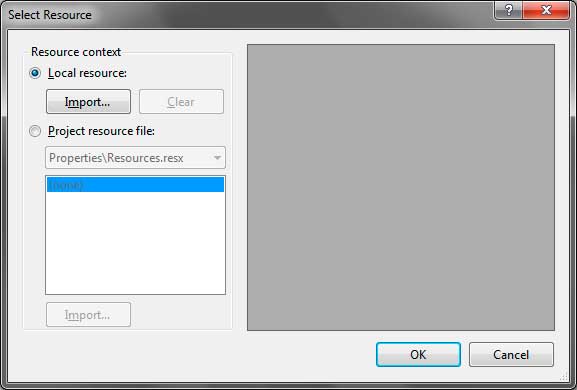
#### اضافه کردن یک عکس به زمینه ی کنترل

شما می توانید عکس زمینه ی  یک کنترل را به وسیله ی مشخصه ی BackgroundImage تغییر دهید. به عنوان مثال ما می خواهیم عکس زمینه ی یک فرم را تغییر دهیم.  برای اینکار ابتدا فرم را انتخاب کرده و سپس به پنجره ی Properties رفته و مشخصه ی  BackgroundImage را انتخاب می کنیم.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/controls-02.jpg)

 در این قسمت پنجره ای به ما نمایش داده می شود که به ما اجازه می دهد آدرس تصویر خود را مشخص کنیم.

 برای اینکار ابتدا قسمت Local resource را انتخاب کرده و سپس بر روی دکمه ی  Importکلیک می کنیم.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/controls-03.jpg)

 پس از اینکه عکس را انتخاب کردید دکمه Ok را برای تایید بزنید. حالا تصویر مورد نظر ما بر روی فرم نمایش داده می شود. حالت قرار گیری و سایز تصویر ممکن است چیزی نباشد که شما انتظار داشته اید.

برای تغییر آن یک مشخصه به نام BackgroundImageLayout  وجود دارد.

این مشخصه مقادیری از نوع شمارشی System.Windows.Forms.ImageLayout را می پذیرد.

این مقادیر به شرح زیر هستند:

| مقدار | توضیح |
| --- | --- |
| None | عکس بر اساس قسمت بالای چپ خود در محل قرار میگیرد و هیچ تغییر سایزی در آن صورت نمی گیرد. |
| Tile | اگر عکس از محیط کنترل کوچکتر باشد ، عکس به طور متناوب بر روی کنترل تکرار می شود. |
| Center | عکس در وسط کنترل قرار می گیرد. |
| Stretch | عکس در حد اندازه کنترل کوچک می شود و کل کنترل را در بر میگیرد. |
| Zoom | عکس در حد کنترل تغییر سایز می دهد بدون اینکه نسبت طول و عرض آن تغییر کند. |

در بیشتر مواقع Stretch  به خوبی عمل می کند.در فرم پایین عکس پس زمینه ی فرم در حالت Stretch قرار دارد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/controls-04.jpg)

#### خاصیت Text

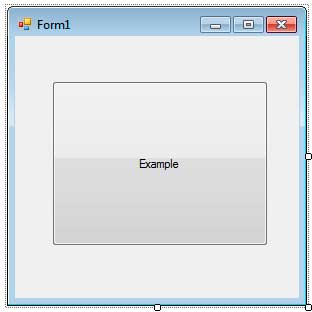
خاصیت Text یک متن را در داخل کنترل تعریف می کند. متن Text خاصیت در کنترل های گوناگون نمایش مختلفی دارد. به عنوان مثال خاصیت متن  خاصیت Text کنترل فرم در عنوان (Caption bar) آن قرار می گیرد. متن خاصیت Text یک دکمه ، در داخل آن دکمه نمایش داده می شود. متن خاصیت Text یک تکست باکس ، در داخل آن تکست باکس نمایش داده می شود. ما میتوانیم از خاصیت TextAlign برای تغییر مکان متن در داخل کنترل استفاده کنیم. اگر شما بر روی مشخصه ی TextAlign کلیک کنید یک نوار کرکره ای را در پنجره Properties مشاهده می کنید ، که به آسانی به شما اجازه  می دهد که  نحوه ی قرار گیری متن خود را در داخل کنترل تنظیم کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/controls-10.jpg)

#### تغییر قلم متن یک کنترل

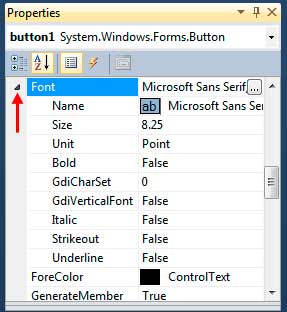
ما می توانیم نوع قلم ، رنگ ، سایز ، و حالت قلم یک کنترل را به وسیله ی خواص Fore Color و Font تغییر

دهیم.برای مثال ما یک دکمه را ایجاد می کنیم.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/controls-051.jpg)

برای تغییر نوع قلم این کنترل ابتدا می بایست خاصیت Font  را در پنجره ی Properties پیدا کنیم.

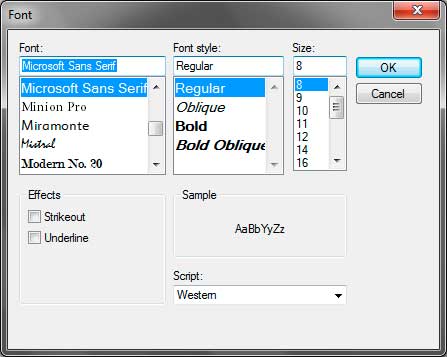
سپس در سمت چپ این خاصیت یک علامت فلش را مشاهده می کنیم.بر روی آن کلیک کرده تا تمامی زیر مجموعه های آنرا مشاهده کنیم.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/controls-06.jpg)

قسمت  Name که نوع قلم را تعریف می کند قسمت Size اندازه ی قلم را مشخص می کند

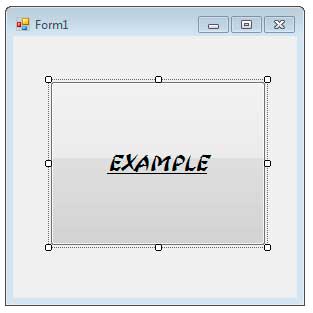
قسمت Bold برای ضخیم کردن متن و قسمت های  Italic و Strikeout برای کشیدن خط زیر متن کنترل به کار می روند.

همچنین ، شما می توانید از دکمه ای سمت راست خاصیت Font برای باز کردن پنجره ی فونت استفاده کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/controls-07.jpg)

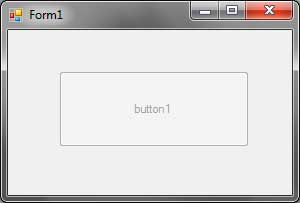
در اینجا شما می توانید نوع قلم ، حالت و سایز آنرا به طور دقیق انتخاب کنید.

همچنین در این پنجره شما می توانید یک پیش نمایش از نوع قلم خود را مشاهده کنید .به وسیله ی اینها شما میتوانید قلم کنترل خود را سفارشی کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/controls-08.jpg)

#### فعال کردن و غیر فعال کردن یک کنترل

می توان از خاصیت Enabled برای فعال کردن یا غیر فعال کردن یک کنترل استفاده کنیم. از مقدار False برای غیر فعال کردن و نیز از مقدار True برای فعال کردن یک کنترل استفاده می شود. وقتی یک کنترل را غیر فعال می شود ، ظاهر  آن تغییر می کند.برای مثال ، وقتی ما یک دکمه را غیر فعال می کنیم ( خاصیت Enabled   آنرا برابر False قرار می دهیم ) ، رنگ و ظاهر آن تغییر می کند.به این نکته توجه کنید که این تغییرات در هنگام اجرای برنامه قابل مشاهده هستند ، نه در محیط طراحی.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/controls-09.jpg)

وقتی که یک کنترل غیر فعال است ، هیچ فوکوس و رویدادی را قبول نمی کند. برای مثال ، اگر یک دکمه غیر فعال باشد ، شما نمیتوانید بر روی آن کلیک کنید.

#### مخفی کردن یک کنترل

شما میتوانید یک کنترل را به وسیله ی مقدار دادن False به خاصیت Visible آن به طور موقت مخفی کنید.

خاصیت Visible یک خاصیت بولی است که دو حالت دارد:

۱- کنترل قابل مشاهده است ۲- کنترل قابل مشاهده نیست

مقدار False آن ، کنترل را مخفی میکند و مقدار True کنترل را نمایش می دهد.

نکته: اثر این خاصیت فقط در هنگام اجرای برنامه قابل مشاهده است و در حالت طراح قابل مشاهده نیست !

شما حتی وقتی که مقدار این خاصیت را برابر False قرار بدهید می توانید آنرا در حالت طراحی انتخاب کنید.

در درسهای آینده به طور مفصل در مورد خواص و ویژگی های کنترل ها به بحث میپردازیم.

#### رویداد های کنترل

در جدول پایین شما برخی از رویداد های مفید که بین بیشتر کنترل های رایج هستند را مشاهده می کنید.

| رویداد | توضیح |
| --- | --- |
| BackColorChanged | وقتی که رنگ پس زمینه تغییر میکند این رویداد اتفاق می افتد. |
| BackgroundImageChanged | وقتی که یک تصویر پس زمینه اضافه می شود یا تغییر می کند این رویداد رخ می دهد. |
| Click | این رویداد وقتی رخ می دهد که بر روی کنترل با دکمه ی سمت چپ موس کلیک شود. |
| DoubleClick | زمانی که شما بر روی یک کنترل دابل کلیک میکنید رخ میدهد. به این نکته توجه داشته باشید که این رویداد میتواند به وسیله ی دستوراتی که کاربر نوشته نیز رخ دهد مثل ترکیب دو کلید ( کلید میانبر ) |
| DragDrop | زمانی که عملیات Drag&Drop بر روی کنترل انجام میشود این رویداد رخ می دهد. |
| EnabledChanged | زمانی که یک کنترل حالت Enable  و یا Disable را را به خود می گیرد ( به یکی از این دو حالت تغییر میکند ) این رویداد رخ می دهد. |
| Enter | زمانی که مکان نما به کنترل وارد شود.(بر روی آن قرار گیرد) |
| FontChanged | زمانی که خاصیت font  کنترل تغییر می کند این رویداد رخ می دهد. |
| ForeColorChanged | زمانی رخ میدهد که رنگ متن کنترل تغییر پیدا کند. |
| GotFocus | زمانی رخ میدهد که Focus روی کنترل قرار گیرد. |
| KeyDown | این رویداد زمانی رخ میدهد که focus بر روی کنترل باشد و کلیدی از صفحه کلید فشرده شود. |
| KeyPress | این رویداد زمانی رخ میدهد که focus بر روی کنترل باشد و کلیدی از روی صفحه کلید فشرده و رها شود.این رویداد قبل از رویداد Keydown رخ می دهد.  ۰۹۱۶۷۵۰۷۹۱۸ |
| KeyUp | این رویداد زمانی رخ میدهد که focus بر روی کنترل باشد و کلیدی که از روی صفحه کلید فشرده شده ، رها شود.این رویداد بعد از دو رویداد قبلی رخ می دهد. |
| Leave | این رویداد وقتی که مکان نما کنترل را ترک کند رخ میدهد. |
| LostFocus | وقتی که کنترل Focus خود را از دست بدهد این رویداد رخ میدهد. |
| MouseClick | یک حالت پیشرفته تر از رویداد کلیک است.کلیک کردن میتواند شامل فشردن کلید های صفحه ی کلید نیز باشد . ولی اگر شما نیاز به اطلاعاتی نظیر : تعداد کلیک  بر روی کنترل ، چرخیدن دکمه چرخنده ی موس و … باشید باید از این رویداد استفاده کنید |
| MouseDoubleClick | یک حالت پیشرفته تر از رویداد DoubleClick  است . |
| MouseDown | این رویداد زمانی رخ می دهد که دکمه ای از موس در داخل کنترل پایین نگه داشته شود. |
| MouseEnter | زمانی که مکان نما به یک کنترل وارد میشود این رویداد رخ می دهد. |
| MouseHover | این رویداد زمانی رخ می دهد که مکان نما روی یک کنترل منتظر بماند. ( حالت مکان نما به Wait  یا همان ساعت شنی تغییر یابد.) |
| MouseLeave | این رویداد زمانی رخ می دهد که مکان نما کنترل را ترک کند. |
| MouseMove | این رویداد زمانی رخ می دهد که مکان نما وقتی در داخل محدوده ی کنترل است حرکت کند. |
| MouseUp | این رویداد زمانی رخ می دهد که دکمه ای از موس را که قبلا فشرده ایم ، در داخل کنترل رها کنیم. |
| MouseWheel | این رویداد زمانی رخ می دهد که  بر روی کنترل  Focus باشد و دکمه چرخنده موس حرکت کند. |
| Move | این رویداد زمانی رخ می دهد که کنترل حرکت کند. |
| Paint | این رویداد زمانی رخ می دهد که کنترلی رسم شود. |
| ParentChanged | این رویداد زمانی رخ می دهد که خاصیت Parent کنترل تغییر کند. |
| Resize | این ویداد زمانی رخ می دهد که کنترل تغییر سایز دهد. |
| TextChanged | وقتی که خاصیت Text کنترل تغییر پیدا کند این رویداد رخ می دهد. |
| Validated | وقتی رخ می دهد که اعتبار سنجی کنترل پایان یابد. |
| Validating | وقتی رخ می دهد که کنترل اعتبار سنجی می شود. |
| VisibleChanged | وقتی که خاصیت نمایش کنترل تغییر پیدا کند. |

در مورد بیشتر رویداد های بالا در بخش های آینده صحبت خواهیم کرد.

#### متد های کنترل

متدهای زیر پرکابردترین متدهای کلاس کنترل هستند.

| متدها | توضیح |
| --- | --- |
| CreateControl | یک کنترل را ایجاد میکند و به کنترل جاری اضافه می کند. |
| FindForm | فرمی را که کنترل بر روی آن قرار دارد را بازیابی میکند. |
| Focus | Focus را بر روی کنترل جاری قرار می دهد. |
| Hide | کنترل را مخفی می کند. |
| Refresh | نوسازی کنترل را انجام می دهد. |
| Select | کنترل را فعال ( انتخاب ) می کند. |
| Show | کنترل مخفی شده را نمایش می دهد. |
| Update | محتویات کنترل را به روز می کند. |

همچنین کلاس کنترل به شما متدهایی را عرضه می کند که به شما اجازه می دهند به صورت دستی رویدادهای کنترل را راه اندازی کنید. برخی از متدها با کلمه ی On و سپس  نام یک رویداد شروع می شوند. برای مثال ، رویداد OnClick وقتی که رویداد Click فراخوانی می شود اتفاق می افتد.

## 

## نامگذاری کنترل ها

همیشه کنترل ها را نامگذاری کنید.ما از خاصیت Name برای نامگذاری کنترلها استفاده می کنیم.نامگذاری کنترلها از قواعد نامگذاری متغیر ها استفاده می کند ، برای مثال فاصله ها ، کاراکترهای ویژه ، و استفاده از کلمات کلیدی در نامگذاری کنترلها نیز ممنوع است.یک سری قرار داد در نامگذاری کنترلها وجود دارد که در ادامه به آنها اشاره می کنیم.

شما می توانید بسته به نوع استفاده ای که از کنترل می کنید آنرا نامگذاری نمایید.برای مثال ، وقتی که قرار است در داخل یک تکست باکس نام یک شخص یا کاربر وارد شود میتوانید نام آن تکست باکس را برای خوانایی بیشتر firstname قرار دهید.اما در نامگذاری کنترل ها بهتر است که نام واقعی آن کنترل قبل از نامی که برای آن انتخاب کرده ایم بیاید.برای مثال به جای نامگذاری تکست باکس به firstname ، میتوانید از TextBoxFirstName استفاده کنید.با توجه به این قرار دادها ، ما میتوانیم در داخل Intellisense ی که با آن کار می کنیم هر کنترلی را شناسایی کنیم.

تکنیک دیگری که برای نامگذاری کنترل ها به کار می رود ، خلاصه کردن نام آنها است.برای مثال ، به جای استفاده از کلمه ی TextBoxFirstName ، شما می توانید از txtFirstName استفاده کنید.کلمه ی txt فرم کوتاه شده ی کلمه ی textbox است.تعداد زیادی کلمه ی کوتاه شده برای هر کنترل وجود دارد و شما حتی میتوانید از کلمه ی کوتاه شده ای که خود برای آن کنترل انتخاب کرده اید استفاده کنید.البته این باید  گویا باشد . به طوری که برای کسانی که به کد شما نگاه می کنند واضح باشد.

تکنیک دیگری که برای نامگذاری کنترلها بکار می رود جابجا کردن نام اصلی کنترل با نامی که ما برای آن انتخاب کرده ایم و همچنین استفاده از روش کوهان شتری(Camel Casing) است.برای مثال یک تکست باکس داریم که نام کاربر یا شخصی را از ورودی می گیرد ، می توانید آنرا به صورت firstNameTextBox نامگذاری کنید و یا یک دکمه دارید که عملیات محاسباتی را انجام می دهد ، میتوانید نام آنرا calculateButton بگذارید.

در درسهای آینده از تکنیک سوم (Camel Casing) استفاده میکنیم.

نیازی به این نیست که شما تمامی کلمات اختصاری کنترل ها را حفظ کنید.وقتی که شما یک کنترل را ایجاد می کنید ، تنها کافیست عدد پسوندی آنرا حذف کنید و به جای آن ، نامی را با توجه به کاربرد کنترل به آن اضافه کنید.وقتی که شما در حال تایپ کردن هستید و قصد پیدا کردن کنترل خاصی را دارید کافیست براحتی نام کنترل را تایپ کرده تا پنجره ی intellisence نام آن کنترل و تمامی کنترل های مشابه آنرا را برای شما نمایش دهد.برای مثال : یک تکست باکس و تمامی کنترل های تکست باکس در intellisence برنامه نمایش داده می شوند ، فقط قسمتی که مربوط به نامی که شما برای آن کنترل انتخاب می کنید ممکن است طولانی باشد ، که شما با توجه به قواعدی که پیشتر ذکر شد میتوانید نسبت به نامگذاری و کوتاه کردن نام کنترلها اقدام کنید.

در زیر مثالهایی از نامگذاری کنترل ها بسته به کاربرد آنها ذکر شده است :

| کاربرد | نام پیشنهادی |
| --- | --- |
| کلیدی که برای نشان دادن یک پیغام به کار می رود. | buttonConfirm, confirmButton, btnConfirm |
| یک تکست باکس که یک ایمیل را از کاربر دریافت می کند. | textBoxAddress, addressTextBox, txtAddress |
| یک فرم که اطلاعات شخصی را از کاربر دریافت می کند. | formPersonalInformation, personalInformationForm, frmPersonalInformation |
| یک ComboBox که لیستی از محصولات را نمایش می دهد. | comboBoxProducts, productsComboBox, cmbProducts |
| یک RadioButton که نشان دهنده ی مذکر بودن یک کاربر است. | radioButtonMale, maleRadioButton, radMale |
| یک MenuItem برای ذخیره کردن یک فایل. | menuItemSave, saveMenuItem, mnuSave |
| یک CheckBox برای فرستادن یک نامه. | checkBoxSubscribe, subscribeCheckBox, chkSubscribe |

لازم نیست که تمامی کنترل های یک فرم را نام گذاری کنید.

 کنترلهایی که هیچوقت در کدنویسی مورد استفاده قرار نمی گیرند، می توانند از نام پیش فرض خود استفاده کنند.

برای مثال : اکثر Lable ها صرفا برای نامگذاری بکار می روند و در کد نویسی مورد استفاده قرار نمی گیرند.لذا لازم نیست خاصیت Name آنها را تغییر دهیم.

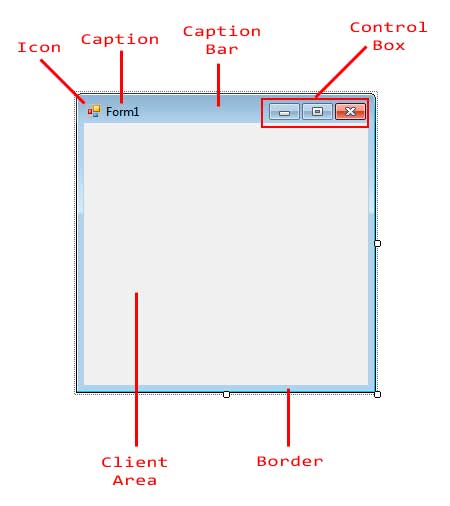
همیشه عادت کنید که کنترل های خود را نامگذاری کنید.

## 

## ویندوز فرم

ویندوز فرم ها ( یا همان فرم ها ) پنجره هایی هستند که شما در برنامه های تحت ویندوز می بینید. شما می توانید در یک برنامه بیش از یک فرم داشته باشید. هر فرم از خواص و متدهای کلاس System.Windows.Forms.Form ارث می برد. فضای نام System.Windows.Forms.Form تمامی اجزایی را که شما برای ساخت فرم ها و کنترل ها نیاز دارید را در بر دارد.

در شکل زیر شما نام قسمت های مختلف فرم را مشاهده می کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/windows-forms-01.jpg)

در قسمت بالا ، نوار عنوان (Caption Bar) قرار دارد. نوار عنوان از یک آیکن (Icon) ،یک عنوان (Caption) ، و یک جعبه ی کنترل (Control Box) تشکیل شده است. جعبه ی کنترل ( Control Box) دکمه های کوچکنمایی (Minimizing) ، بزرگنمایی (Maxmizing) ، و بستن (Closing) را در بر دارد. قسمت داخلی یا همان Client Area مکانی است که ما کنترل ها را در آن قرار می دهیم. قسمت حاشیه یا Border که شامل قسمت Caption Bar نیز می شود ، به شما اجازه می دهد که سایز فرم را تغییر دهید.

در جدول زیر برخی از خواص مفید کلاس فرم را مشاهده می کنید.

| خاصیت | توضیح |
| --- | --- |
| AcceptButton | نام یک دکمه در این خصوصیت تعیین می شود و هرگاه دکمه ی Enter فشار داده شود ، آن دکمه عمل خواهد کرد. |
| CancelButton | در این خصوصیت نیز مانند خصوصیت قبل نام یک دکمه در آن قرار می گیرد و با زدن دکمه Esc آن دکمه عمل می کند. |
| ControlBox | این خصوصیت مشخص می کند که قسمت جعبه ی کنترل (Control Box) نمایش داده شود یا خیر.جعبه ی کنترل (Control Box) قسمتی از فرم است که دکمه های کوچکنمایی ، بزرگنمایی و بستن فرم در آن قرار می گیرند. |
| DesktopBounds | به وسیله ی این خاصیت میتوانید به سایز و محل قرار گیری فرم در میز کار دسترسی داشته باشید. |
| Font | نوع قلمی که در فرم مورد استفاده قرار می گیرد.تمامی کنترل ها از این خاصیت استفاده می کند. |
| FormBorderStyle | این خاصیت نوع حاشیه ی فرم را مشخص می کند. |
| HelpButton | یک دکمه ی Help قبل از دکمه ی Close بالای فرم قرار می دهد.( دکمه های Minimize و Maximize غیر فعال می شوند ) |
| Icon | این خاصیت Icon فرم را مشخص می کند.(آیکنی که در قسمت Caption Bar وجود دارد) |
| Location | مختصات فرم را در صفحه ی نمایش مشخص می کند. |
| MainMenuStrip | یک منو را مشخص می کند تا در فرم به عنوان منوی اصلی استفاده شود. |
| MaximizeBox | مشخص می کند که دکمه ی Maximize در بالای فرم نمایش داده شود یا خیر. |
| MinimizeBox | مشخص می کند که دکمه ی Maximize در بالای فرم نمایش داده شود یا خیر. |
| Name | نام فرم که به عنوان مرجع برای استفاده در محیط کد نویسی انتخاب می شود. |
| ShowIcon | مشخص میکند که آیکن فرم نمایش داده شود یا خیر. |
| Size | سایز فرم را مشخص می کند. |
| StartPosition | این خاصیت محل نمایش فرم در صفحه را مشخص می کند. |
| Text | این خاصیت متنی را که در قسمت Caption Bar قرار می گیرد را مشخص می کند. |

در این قسمت برخی از متدهای مهم کلاس فرم را مشاهده می کنید.

| متد | توضیح |
| --- | --- |
| Activate | Focus را به فرم منتقل و فرم را فعال می کند . |
| CenterToScreen | فرم را به وسط صفحه منتقل می کند. |
| Close | فرم را می بندد. |
| Hide | فرم جاری را پنهان می کند. |
| OnLoad | وقتی که رویداد Load رخ می دهد این متد اجرا می شود. |
| Show | فرم را نمایش می دهد.(آشکار می سازد) |

در این قسمت رویداد های مربوط به فرم را مشاهده می کنید.

| رویداد | توضیح |
| --- | --- |
| Activated | زمانی رخ می دهد که فرم فعال باشد. |
| Click | زمانی رخ می دهد که بر روی فرم کلیک می شود. |
| Deactivated | این رویداد زمانی رخ می دهد که فرم حالت Focus خود را از دست بدهد. |
| FormClosed | زمانی رخ می دهد که فرم بسته شود. |
| FormClosing | زمانی رخ می دهد که فرم در حال بسته شدن است.به شما اجازه می دهد برای بسته شدن فرم مکث کنید. |
| HelpButtonClicked | زمانی رخ می دهد که بر روی دکمه ی Help کلیک شود. |
| KeyPress | این رویداد زمانی رخ می دهد که یک کلید بر روی صفحه کلید فشرده شود. |
| Load | زمانی رخ می دهد که بارگذاری فرم تمام شده است ولی هنوز نمایش داده نشده است. |
| MenuComplete | زمانی که یک منو از فرم حالت Focus خود را از دست می دهد. |
| MenuStart | زمانی که بر روی یک منو Focus می شود. |
| ResizeBegin | زمانی رخ می دهد که فرم شروع به تغییر سایز می کند. |
| ResizeEnd | زمانی رخ می دهد که تغییر سایز فرم تمام می شود. |
| Shown | زمانی رخ می دهد که فرم برای اولین بار نمایش داده می شود. |

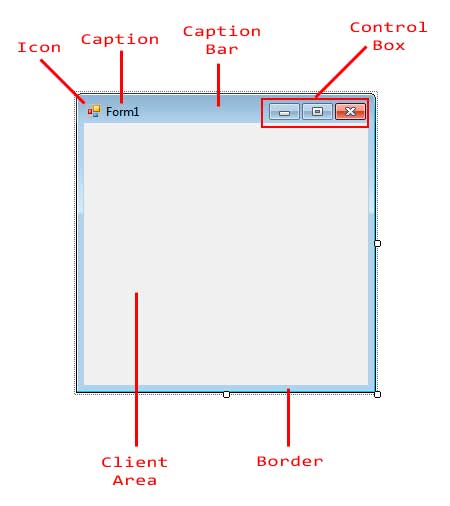
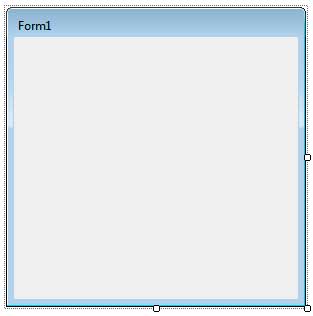
کلاس Form، فرزند کلاس اصلی System.Windows.Forms.Control است بنابراین متدها و خواص تعریف شده ی این کلاس توسط کلاس Form ارث بری می شوند.

#### تغییر دادن جعبه ی کنترل (Control Box)

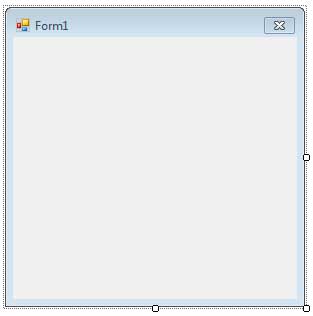
از خاصیت Control Box برای پنهان کردن و یا نمایش دادن قسمت جعبه ی کنترل (Control Box) استفاده می شود.

این خاصیت وقتی که شما می خواهید دکمه های Minimize و Maximize کنترل را غیر فعال کنید و یا از طریق کد نویسی فرم را ببنید بسیار کار آمد می شود.

در تصویر زیر مقدار خاصیت ControlBox برابر False قرار گرفته شده است.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/windows-forms-01.jpg) [](http://www.w3-farsi.com/wp-content/uploads/2013/09/windows-forms-02.jpg)

اگر شما می خواهید یکی از دکمه های Minimize  یا Maximize  را غیر فعال کنید باید از خواص MinimizeBox و MaximizeBox استفاده کنید و مقدار آنها را به False  تغییر دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/windows-forms-03.jpg)

در فرم بالا دکمه های Maximize و Minimize مخفی شده اند. متاسفانه شما نمی توانید فقط دکمه Close را غیر فعال کنید.

#### تغییر حالت حاشیه ی فرم

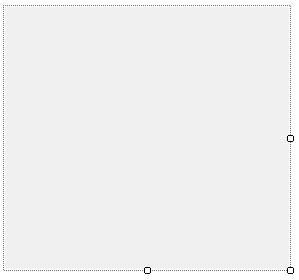
شما می توانید حالت حاشیه ی فرم را تغییر دهید.

برای مثال ، شما می خواهید که کاربر قادر به تغییر سایز فرم نباشد ، در صورتیکه در حالت پیشفرض حاشیه ی فرم به کاربر اجازه ی اینکار را می دهد.

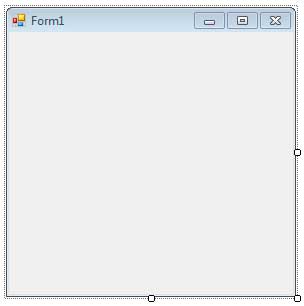
میتوانیم مقادیر مختلفی از نوع شمارشی System.Windows.Forms.FormBorderStyle  را به خاصیت FormBorderStyle بدهیم.

| مقدار | توضیح |
| --- | --- |
| None | فرم هیچ حاشیه ای ندارد. |
| FixedSingle | فرم یک حاشیه ی خطی بدون قابلیت تغییر سایز را دارد. |
| Fixed3D | فرم یک حاشیه ی سه بعدی بدون قابلیت تغییر سایز را دارد. |
| FixedDialog | در این حالت یک خط ضخیم دور فرم قرار دارد و سایز فرم قابل تغییر نیست ، همچنین دکمه های minimize  و Maximize در آن وجود ندارند. |
| Sizable | حالت پیشفرض است.در این حالت سایز فرم قابل تغییر است. |
| FixedToolWindow | فرم دارای حاشیه ای است که سایز آن غیر قابل تغییر می باشد و فقط دارای دکمه ی Close است.این حالت برای پنجره های ابزار بکار می رود. مانند پنجره ی ابزار در Office 2003 |
| SizableToolWindow | دقیقا مانند FixedToolWindow است ولی قابلیت تغییر سایز دارد. |

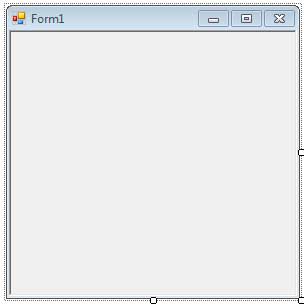
تصاویر زیر مربوط به حالات مختلف خاصیت FormBorderStyle است.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/windows-forms-04.jpg)

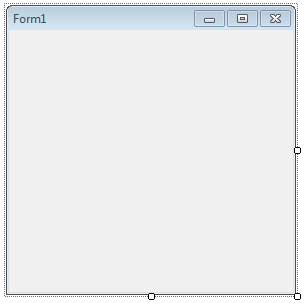
None

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/windows-forms-05.jpg)

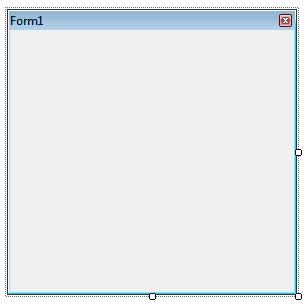
FixedSingle

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/windows-forms-06.jpg)

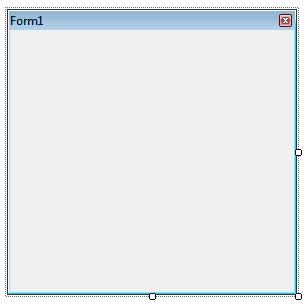
Fixed3D

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/windows-forms-07.jpg)

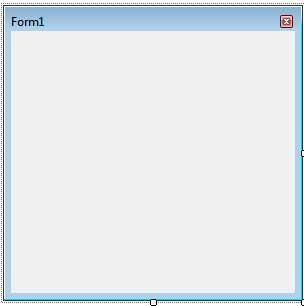
FixedDialog

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/windows-forms-09.jpg)

Sizable

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/windows-forms-091.jpg)

FixedToolWindow

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/windows-forms-10.jpg)

SizableToolWindow

#### آیکن های فرم

از خاصیت Icon برای تغییر آیکن بالای فرم استفاده می شود. بر روی دکمه ی Browse در کنار خاصیت Icon در پنجره ی Properties کلیک کنید و سپس فایلی را با پسوند .ico انتخاب کنید.(.ico پسوند فایلهای تصویری آیکن است) خاصیت ShowIcon به شما اجازه می دهد که آیکن را در نوارعنوان فرم (caption bar)  پنهان یا آشکار کنید.

#### دکمه های Accept و Cancel

شما می توانید یک دکمه را به فرم اضافه کنید که یکی از دو نوع Accept یا Cancel  باشد. برای این کار از خواص AcceptButton و CancelButton استفاده می شود. اگر دکمه ی شما از نوع Accept باشد ، زمانی که فرم فعال باشد و کاربر دکمه ی Enter را فشار دهد آنگاه رویداد Click دکمه رخ می دهد. دکمه ی Cancel زمانی رخ می دهد که دکمه ی Escape بر روی صفحه کلید فشرده شود. فقط به پنجره Properties رفته ، خاصیت مورد نظر را پیدا کنید و بر روی نوار کرکره ای آن کلیک کنید تا نام تمامی دکمه های فرم برای شما نمایش داده شود. دکمه مورد نظر را انتخاب کنید.برای مثال شما می خواهید یک فرم ورود (login) را ایجاد کنید. شما می توانید دکمه ی ورود را برای خاصیت AcceptButton انتخاب کنید. در اینصورت به آسانی ، وقتی کاربر پسورد خود را وارد کرد کلید Enter را برای ورود می فشارد. برای این خاصیت موارد زیادی وجود دارد که در درس های آینده به آنها می پردازیم.

## 

## کنترلButton

وقتی که بر روی کنترل Button کلیک می کنیم دستوراتی اجرا می شود، و این معمولترین استفاده ای است که از این کنترل می شود.

دکمه ها به نوعی برای تایید یا لغو یک عمل و یا باز کردن کادرهای محاوره ای به کار می روند.

کنترل Button دارای خواصی است که در جدول زیر تعدادی از این آنها ذکر شده است :

| خواص | توضیح |
| --- | --- |
| AutoEllipsis | اگر طول یک متن برای Button انتخاب می کنیم زیاد باشد تمام متن نمایش داده نشده و در انتهای آن سه نقطه ( … ) قرار داده می شود |
| AutoSize | اندازه کنترل را با متن داخل آن متناسب می کند |
| FlatStyle | برای تعیین شکل button به کار می رود.  Popup باعث صاف شدن کنترل می شود.در این حالت اگر با موس بر روی کنترل توقف کنید، کنترل به حالت برجسته در می آید.  Flat باعث صاف شدن کنترل می شود.در این حالت اگر نشانگر موس را بر روی کنترل ببریم، رنگ پس زمیه آن تغییر می کند. |
| Enabled | اگر مقدار این خاصیت false  باشد بر روی button نمی توان کلیک کرد. |
| Image | می توان یک تصویر اختیاری به کنترل اختصاص داد. |
| ImageAlign | برای تراز کردن عکس اختصاص داده شده به  button به کار می رود. |
| Text | عنوانی  است که به کنترل اختصاص داده می شود. |
| Visible | مشخص می کند که آیا کنترل button در روی فرم قابل رویت باشد یا نه |

فقط با ویرایش خواص یک button نمی توان از آن استفاده کرد. بلکه کنترل دکمه برای انجام برخی اعمال نیاز به واکنش به رویدادها دارد.

در زیر معمولترین رویدادهای کنترل button آمده است.

| رویداد | توضیح |
| --- | --- |
| Click | وقتی روی می دهد که بر روی دکمه کلیک کنید. |
| Enter | وقتی روی می دهد که کنترل، کنترل فعال روی فرم باشد. |
| Leave | وقتی روی می دهد که موس کنترل را ترک کند. |
| LocationChanged | وقتی روی می دهد که مکان دکمه تغییر کند. |
| MouseDown | وقتی روی می دهد که نشانگر موس بر روی کنترل باشد و دکمه موس رو به پایین فشار داده شود. |
| MouseEnter | وقتی روی می دهد که موس وارد کنترل دکمه می شود. |
| MouseHover | وقتی روی می دهد که موس بر روی کنترل دکمه برای لحظاتی توقف کند. |
| MouseUp | وقتی روی میدهد که دکمه موس فشار داده شده و رها شود. |
| MouseLeave | وقتی روی می دهد که نشانگر موش دکمه را ترک می کند. |

همانطور که قبلا مشاهده کردید رویداد پیشقرض کنترل دکمه رویداد Click می باشد.

اجازه دهید که با ایجاد برنامه هایی با رویدادهای دیگر نیز آشنا شویم.

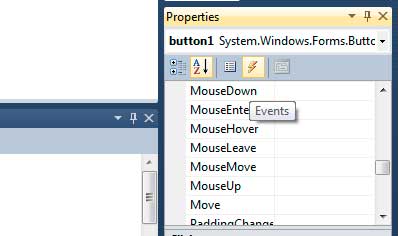
یک رفم جدید ایجاد کرده و یک کنترل button را از toolbox به داخل آن درگ کنید. لازم به تغییر خاصیت text  کنترل دکمه نمی باشد.

در پنجره properties، خاصیت Name دکمه را یافته و ان را به buttonSample تغییر دهید.

حال در هنگام کدنویسی می توانیم با استفاده از این نام به این کنترل مراجعه کنیم.

برنامه ما رویدادهای MouseEnter و MouseLeave را شرح می دهد.

برای دستیابی به این رویدادها کافیست که از پنجره properties بر روی دکمه ای که به صورت رعد و برق است کلیک کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/button-control-12-19-10-01.jpg)

ابتدا رویداد MouseEnter را یافته و بر روی آن دوبار کلیک کنید.

ویژوال استودیو یک کنترل کننده رویداد برای این رویداد ایجاد می کند.

کد پر رنگ شده زیر را در داخل متد بنویسید.

private void buttonSample\_MouseEnter(object sender, EventArgs e)

{

buttonSample.Text = "Mouse has entered!";

}

بعد از تایپ کد فوق به پنجره Properties برگشته و حال رویداد MouseLeave را انتخاب و بر روی آن دوبار کلیک کنید تا یک کنترل کننده رویداد نیز برای آن ایجاد شود :

کد پر رنگ شده زیر را نیز داخل متد کنترل کننده رویداد بنویسید .

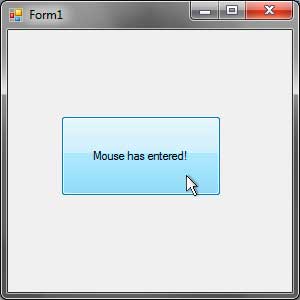
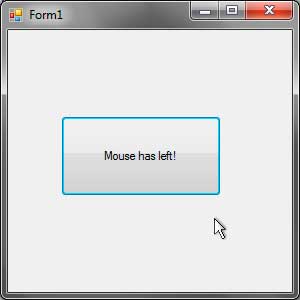
private void buttonSample\_MouseLeave(object sender, EventArgs e)

{

btnSample.Text = "Mouse has left!";

}

حال برنامه را اجرا کنید. مشاهده می کنید که با قرار گرفت اشاره گر موس بر روی کنترل button خاصیت text آن تغییر می کند. و از طرف دیگر وقتی که اشاره گر از کنترل دکمه دور می شود این خاصیت دوباره تغییر می کند.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/button-control-12-19-10-02.jpg)[](http://www.w3-farsi.com/wp-content/uploads/2013/09/button-control-12-19-10-03.jpg)

## کنترل Label

کنترل Label برای اضافه کردن یک متن به فرم بکار می رود. این متن می تواند حاوی یک پیام و یا توضیحی در باره سعملکرد سایر کنترل ها باشد.

یک کنترل Lable را از toolbox به فرم اضافه کنید.

در حالت پیشفرض این کنترل دارای یک متن است.

خواص زیر معمولترین خواص قابل تغییر این کنترل می باشند.

| خاصیت | توضیح |
| --- | --- |
| AutoSize | اگر مقدار این خاصیت true باشد اندازه کنترل بسته به متنی که به آن اختصاص می دهیم تغییر می کند |
| BorderStyle | کنترل را حاشیه دار می کند. |
| Font | برای تغییر فونت متن کنترل به کار می رود. |
| Text | متن کنترل را مشخص می کند. |
| TextAlign | جهت ترازبندی متن داخل کنترل به کار می رود. |

خاصیت Text مهمترین خاصیت آن می باشد چون هدف اصلی از استفاده از این کنترل نشان دادن متن در فرم است.

Label1.Text = "Hello World!";

می توان خاصیت Font را با تغییر نوع فونت، اندازه فونت و برخی دیگر از خواص آن اصلاح کرد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/label-control-12-20-10-01.jpg)

رویدادهایی هم برای این کنترل وجود دارد که اکثر مواقع به آن نیازی نیست و در نتیجه در مورد آنها بحث نمی کنیم.

## کنترلTextBox

کنترل TextBox ابتدایی ترین وسیله برای ورود اطلاعات در یک فرم ویندوزی می باشد.

این کار را با تایپ آنها (اطلاعات) در TextBox انجام می دهید.

متنی که شما تایپ می کنید به وسیله خاصیت Text این کنترل قابل دسترسی است.

در جدول زیر خاصیت های مفید این کنترل که شما می توانید از انها استفاده کنید آمده است :

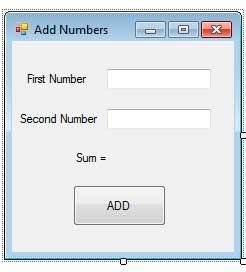
| خاصیت | توضیح |
| --- | --- |
| AcceptsReturn | تعیین می کند که در TextBox متن به صورت چند خطی نوشته شود یا نه. که در این صورت با زدن دکمه اینتر در داخل TextBox خط جدید ایجاد می شود. |
| AcceptsTab | در حالت چند خطی این خاصیت باعث می شود که مکان نما با زدن دکمه tab به جای اینکه به کنترل بعدی برود به tab بعد برود. |
| Enabled | اگر این خاصیت در حالت false قرار بگیرد TextBox  به صورت read-only در می آید و در نتیجه در حکم یک برچسب می شود. |
| Font | نوع فونتی که در جعبه متن مورد استفاده قرار می گیرد. |
| Lines | خطوط متن در یک textbox چند خطی |
| Multiline | اگر True باشد به شما اجازه تایپ چندین خط در TextBox   را می دهد. |
| Text | متن داخل جعبه متن را نشان می دهد. |
| PasswordChar | کاراکترهایی را که توسط کاربر وارد می شوند پنهان می کند. |
| ReadOnly | تعیین می کند که آیا متن قابل ویرایش باشد یا نه. |
| Visible | قابل رویت بودن یا نبودن textbox را تعیین می کند |
| WordWrap | در تکس باکس چند خطی کاربرد دارد.باعث می شود وقتی که نشانگر موس به انتهای تکس باکس رسید به صورت اتوماتیک به خط بعد منتقل شود. |

در مثال زیر نحوه استفاده از TextBox   نشان داده شده است.

یک پروژه ویندوزی ایجاد کنید.

برنامه از شما می خواهد که دو عدد را وارد کنید و با زدن دکمه جمع آنها را با استفاده از یک برچسب (label) به شما نشان می دهد.دو textbox را روی فرم بکشید و نام اولی را textBoxFirstNumber و دومی را textBoxSecondNumber بگذارید. سپس در کنار هر textbox یک کنترل label که نشان دهنده هدف آنهاست (textbox ها) قرار دهید. یک کنترل lable دیگر برای نشان دادن حاصل جمع دو عدد در فرم قرار داده و نام ان را labelSum بگذارید. یک دکمه Button را هم اضافه کرده و نام آن را buttonAdd بگذارید.

مکان قرار گیری و سایز کنترل ها را مانند شکل زیر تنظیم کنید:

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/textbox-control-1-9-11-01.jpg)

بر روی دکمه add دو بار کلیک کرده و در رویداد کلیک مربوط به آن کد زیر را وارد نمایید :

private void buttonAdd\_Click(object sender, EventArgs e)

{

int num1 = Int32.Parse(textBoxFirstNumber.Text);

int num2 = Int32.Parse(textBoxSecondNumber.Text);

int sum = num1 + num2;

labelSum.Text = "Sum = " + sum;

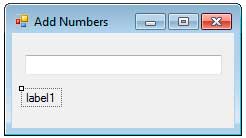
}

کدهای نوشته شده در داخلtextBoxFirstNumber  و  textBoxSecondNumberبا ساتفاده از متد Parse کلاس Int32 به نوع صحیح تبدیل شده و در متغیرهای مربوطه ذخیره می شوند. ما به این تبدیل نیاز داشتیم چون خاصیت Text از نوع رشته است. سپس جمع آنها محاسبه می شود. و در نهایت مقدار جمع به خاصیت Text برچسب labelSum نسبت داده می شود.

یکی از مفید ترین رویداد های تکست باکس ، رویداد TextChanged است.این رویداد زمانی رخ می دهد که متن داخل تکست باکس دستکاری شود.(تغییر کند)

برنامه ی زیر یک مثال از کاربرد این رویداد نشان می دهد.

یک برنامه ویندوزی  دیگر را ایجاد و یک تکست باکس (Text Box) و یک برچسب Label)) به آن اضافه کنید. می توانید نام آنها را تغییر ندهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/textbox-control-1-9-11-02.jpg)

متن برچسب را که در داخل خاصیت Text آن قرارگرفته است را پاک کنید.

نکته : اگر مقدار خاصیت AutoSize برچسب شما برابر با False باشد انتخاب کردن آن مشکل است.برای انتخاب کردن آن می توانید از نوار کرکره ای بالای پنجره ی خواص (Properties) استفاده کنید.

رویداد پیش فرض کنترل تکست باکس (Text Box) رویداد TextChanged است.پس با دوبار کلیک کردن بر روی تکست باکس یک کنترل کننده رویداد ( Event Handler) برای رویداد گفته شده ایجاد می کند.

کد روبرو را به Event Handler اضافه کنید.

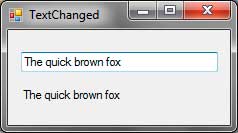
private void textBox1\_TextChanged(object sender, EventArgs e)

{

label1.Text = textBox1.Text;

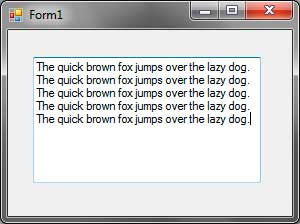
}

حالا برنامه را اجرا کرده و هر چیزی را که می خواهید درون تکست باکس بنویسید. مشاهده می کنید که هر چیزی که شما درون تکست باکس می نویسید دقیقا به همان صورت بر روی بر چسب نوشته می شود.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/textbox-control-1-9-11-03.jpg)

زمانی که متن تکست باکس دستکاری می شود ، Event Handler اجرا شده و هر متنی را که درون تکست باکس قرار دارد را دقیقا به همان صورت درون برچسب (label1) قرار می دهد.

به طور پیش فرض ، هر تکست باکس فقط می تواند یک خط را درون خود جای دهد.برای اینکه چند خط درون تکست باکس قرار بگیرد ، می توانید مقدار خاصیت Multiline آنرا برابر با True قرار دهید. وقتی که شما اینکار را انجام می دهید، باید توجه داشته باشید که شما نمی توانید طول تکست باکس را در قسمت طراحی تغییر دهید.

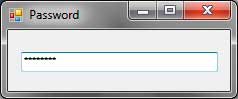
[](http://www.w3-farsi.com/wp-content/uploads/2013/09/textbox-control-1-9-11-05.jpg)

شما می توانید مقدار خاصیت WordWrap را برابر با True قرار دهید تا نشانگر موس زمانی که به انتهای قسمت راست تکست باکس رسید به طور اتوماتیک به خط بعد برود.

اگر مقدار آن برابر با False باشد ، متن در همان خط ادامه پیدا می کند.

شما می توانید مقدار خاصیت ScrollBar را به  Horizontal (افقی) ، Vertical (عمودی) و یا هر دو تغییر دهید.این کار باعث می شود که قسمت پیمایش به تکست باکس چند خطی اضافه شود.

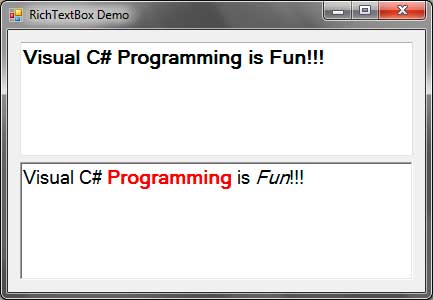
اگر تکست باکس شما می خواهد یک پسورد را بپذیرد،شما باید مقدار خاصیت PasswordChar را به چیزی مانند \* تغییر دهید(این کاراکتر بسته به سلیقه ی شما می تواند هر چیزی باشد).وقتی که کاربر یک حرف را وارد کند این کاراکتر(\*) به او نمایش داده می شود.به شکل زیر دقت کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/textbox-control-1-9-11-04.jpg)

## 

## کنترلRichTextBox

کنترل RichTextBox  شبیه کنترل TextBox است با این تفاوت که این کنترل به شما اجازه می دهد که قالب قسمت های مختلف متن آنرا تغییر دهید. کنترل Text Box به طور کلی برای پذیرفتن اطلاعات ورودی از کاربر بکار می رود ، در صورتیکه کنترل RichTextBox برای نمایش دادن متن های قالب بندی شده و ذخیره ی آن در قالب Rich Text Format (RTF) به کار می رود.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/richtextbox-control-01.jpg)

در تصویر بالا شما تفاوت بین TextBox (بالا) و RichTextBox (پایین) را مشاهده می کنید. اگرچه شما می توانید یک Text Box را قالب بندی کنید ولی این قالب بندی به کل متن آن اعمال می شود. کنترل RichTextBox به شما اجازه می دهد که فقط قسمتی از کل متن داخل کنترل را قالب بندی کنید. شما می توانید از کنترل RichTextBox  مانند کنترل TextBox برای گرفتن اطلاعات ورودی از کاربر استفاده کنید.هر دوی این کنترلها از کلاس TextBoxBase برگرفته می شوند بنابراین، بیشتر خواص آنها مثل خاصیت Text مشترک هستند.

در جدول زیر خواص کنترل RichTextBox را مشاهده می کنید.

| خاصیت | توضیح |
| --- | --- |
| AcceptsTab | مشخص می کند زمانی که دکمه ی Tab فشرده شد Focus بر روی آن قرار گیرد یا خیر. |
| CanRedo | مشخص می کند که تمامی کارهایی که در داخل RichTextBox انجام شد می توانند دوباره اعمال شوند یا خیر. |
| CanUndo | مشخص می کند که کاربر می تواند عملیات انجام شده در داخل RichTextBox را Undo کند یا خیر. |
| DetectUrls | مشخص می کند که وقتی یک URL در RichTextBox تایپ شود ، به طور خودکار قالب بندی شود یا خیر. |
| Lines | اطلاعات مربوط به کنترل RichTextBox را مشخص می کند.چنانچه بر روی دکمه ی کنار این خاصیت کلیک کنید می توانید مقادیر آنرا را مشاهده و دستکاری کنید. |
| Modified | مشخص می کند که محتوای RichTextBox پس از آخرین تغییرات دستکاری شده است یا خیر. |
| Multiline | مشخص می کند که کنترل RichTextBox می تواند حالت چند خطی داشته باشد یا خیر.(مقدار این خاصیت به طور پیش فرض برابر با True است) |
| ReadOnly | مشخص می کند که کنترل RichTextBox فقط خواندنی است یا خیر. |
| Rtf | متن کنترل RichTextBox را به همراه کدهای فرمت RichText (RTF) را در خود جای می دهد. |
| Scrollbars | برای تعیین نوع Scroll بکار می رود.این خاصیت نیز مانند خاصیت ScrollBars کنترل Text Box است. |
| SelectedText | متن داخل RichTextBox را در بر می گیرد. |
| SelectionBackColor | رنگ پشت زمینه ی متن انتخاب شده را مشخص می کند. |
| SelectionBullet | مشخص می کند که حالت گلوله ای به متن انتخاب شده اعمال شود یا خیر. |
| SelectionColor | رنگ متن انتخاب شده را مشخص می کند. |
| SelectionFont | نوع قلم متن انتخاب شده را مشخص می کند. |
| SelectionLength | تعداد کاراکترهای متن انتخاب شده را مشخص می کند. |
| SelectionRightIndent | فاصله بین ضلع راست کنترل RichTextBox و ضلع راست متن انتخاب شده یا محل درج جاری را مشخص می کند. |
| SelectionStart | محل شروع قسمت انتخاب شده یا محل درج را مشخص می کند. |
| ShowSelectionMargin | مشخص می کند که حاشیه ی انتخاب نمایش داده شود یا خیر. |
| Text | متنی را درون کنترل RichTextBox قرار می دهد. |
| WordWrap | متن درون کنترل RichTextBox را دسته بندی می کند. |

در این قسمت با رویداد هایی آشنا می شوید که از آنها می توانید برای کنترل RichTextBox استفاده کنید.

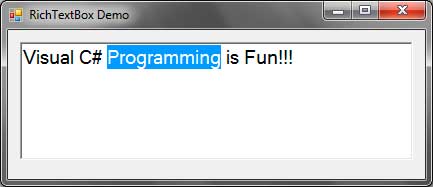
| رویداد | توضیح |
| --- | --- |
| LinkClicked | زمانی که بر روی یک لینک کلیک می شود این رویداد رخ می دهد. |
| Protected | زمانی رخ می دهد که کاربر بخواهد یک متن حفاظت شده را دستکاری کند. |
| TextChanged | زمانی رخ می دهد که متن داخل RichTextBox دستکاری شود. |
| SelectionChanged | زمانی رخ می دهد که متن انتخاب شده تغییر کند. |

**تغییر دادن قالب متن انتخاب شده**

تعداد زیادی از خواص کنترل RichTextBox  برای متن انتخاب شده استفاده می شوند.برای مثال ، خاصیت SelectedText متنی را که توسط کاربر انتخاب شده است را مشخص می کند.

همچنین شما می توانید از یک متد برای انتخاب یک متن استفاده کنید.

richTextBox1.Select(10, 11);

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/richtextbox-control-02.jpg)

مقدار اول، نقطه ی شروع انتخاب و مقدار دوم تعداد کاراکترهایی را که باید از نقطه ی شروع انتخاب شوند را مشخص می کند.

وقتی ما یک متن را انتخاب می کنیم، می توانیم متن انتخاب شده را به وسیله ی تعداد زیادی از خواص که بر روی متن انتخاب شده کار می کنند قالب بندی کنیم.

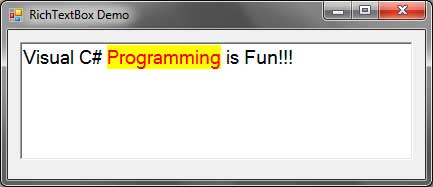
برای مثال،شما می توانید  برای تغییر رنگ متن انتخاب شده از خاصیت SelectionColor استفاده کنید.همچنین می توانید برای تغییر رنگ پس زمینه ی آن از خاصیت SelectionBackColor استفاده کنید.

richTextBox1.SelectionColor = Color.Red;

richTextBox1.SelectionBackColor = Color.Yellow;

شما با استفاده از متد DeselectAll می توانید متن انتخاب شده را با کلیک بر روی هر جایی از فرم از حالت انتخاب خارج کنید.

richTextBox1.DeselectAll();

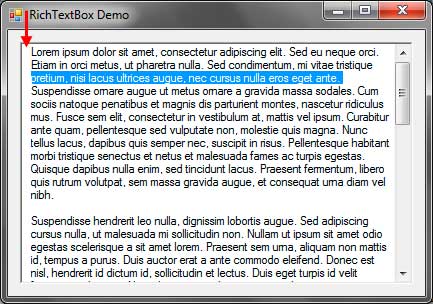
[](http://www.w3-farsi.com/wp-content/uploads/2013/09/richtextbox-control-03.jpg)

نکته: وقتی که هیچ متنی انتخاب نشده ، تاثیر خواص Selection  مثل SelectionColor ، از نقطه ی درج شروع می شود.

نقطه ی درج ،محلی  چشمک زن و به شکل حرف I است که مشخص می کند درج متن از آنجا شروع خواهد شد.هر کاراکتر جدیدی که شما تایپ می کنید قالب بندی جدید بر آنها اعمال می شود.

**اضافه کردن یک حاشیه به متن انتخاب شده**

حاشیه ی انتخاب، یک حاشیه ی کوچک است که در سمت چپ کنترل RichTextBox قرار می گیرد.(به شکل توجه کنید)

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/richtextbox-control-04.jpg)

حاشیه ی انتخاب (که با فلش مشخص شده است) برای انتخاب یک خط بکار می رود. برای مثال ، در تصویر بالا خط سوم توسط کلیک کردن بر روی حاشیه انتخاب کناری آن انتخاب شده است. ما می توانیم از خاصیت SelectionMargin برای نمایش دادن حاشیه ی انتخاب استفاده کنیم.

**اضافه کردن - Scroll نوار پیمایش**

برای اضافه کردن یک نوار پیمایش ، ما از خاصیت ScrollBars که مقادیری از نوع شمارشی RichTextBoxScrollBars را قبول می کند استفاده می کنیم. این مقادیر شمارشی  شامل مقادیری از جمله  Horizontal (افقی) است.این مقدار زمانی که طول خط ما زیاد باشد و مقدار خاصیت WordWrap نیز False باشد به کار می رود. مقدار بعدی Vertical (عمودی) است،و زمانی بکار می رود که تعداد خطوط متن از ارتفاع RichTextBox بیشتر باشد. مقدار Both نیز در مواقع لزوم به صورت اتوماتیک Scroll های عمودی و افقی را اضافه می کند. وقتی مقدار خاصیت SelectionMargin برابر false باشد، شما لازم است که از مقادیر شمارشی دیگری مانندForcedHorizontal، ForcedVertical و یا ForcedBoth استفاده نمایید.برای غیر فعال کردن نوار پیمایش به صورت کامل کافیست که مقدار خاصیت ScrollBars را برابر None قرار دهید.

**تغییر ترازبندی متن انتخاب شده**

ما می توانیم از خاصیت SelectionAlignment برای تغییر تراز بندی (Align) متن انتخاب شده استفاده کنیم. این خاصیت مقادیری از نوع شمارشی HorizontalAlignment دریافت می کند که شامل Left، Right و Center می باشد.

مهم است که بدانید پاراگراف ها چگونه در داخل کنترل ساخته می شوند.به طور کلی ، وقتی شما اولین کاراکتر خود را تایپ می کنید ،اولین پاراگراف در RichTextBox ساخته می شود. برای ساختن پاراگراف بعدی ، شما باید دکمه ی Enter را بزنید.

**Redo و Undo**

شما می توانید از خواص CanUndo و CanRedo برای اینکه کاربر بتواند عملیات انجام شده را Undo و یا Redo کند استفاده کنید.

Undo کردن یعنی شما تغییراتی را که در اجزای کنترل RichTextBox به وجود آورده اید را برگشت دهید ،

در صورتیکه Redo عملیات Undo را خنثی می کند و تغییراتی را که شما قبلا در اجزاء RichTextBox به وجود آورده اید را بر می گرداند.

خواص UndoActionName و RedoActionName برای محدود کردن عملیات Undo و Redo به کارهای مشخصی به کار می روند.

این خواص عملیات های زیر را به صورت رشته ای قبول می کنند.

| عمل | توضیح |
| --- | --- |
| Typing | عملیات تایپ |
| Delete | عملیات حذف |
| DragDrop | عملیات Drag & Drop |
| Cut | عملیات برش (Cut) |
| Paste | عملیات Paste |

برای مثال وقتی که شما کلمه ی “Delete” را به خاصیت UndoActionName نسبت می دهید.اینکار باعث می شود که کاربر فقط بتواند عملیات حذف کردن را برگشت (Undo) دهد.

richTextBox1.UndoActionName = "Delete";

به یک دلیل نامعلوم مایکروسافت برای این خواص نوع شمارشی ایجاد نکرده است بنابراین شما باید از یک رشته برای انجام این اعمال استفاده نمایید.

**شناسایی URL**

کنترل RichTextBox این قابلیت را داراست که بتواند URL هایی را که در آن تایپ و یا Paste می شوند را تشخیص دهد.

خاصیت DetectUrls به کنترل اجازه می دهد که URL ها را تشخیص داده و یک لینک به آن صفحه ایجاد کند.

مقدار True این خاصیت را فعال و مقدار False این خاصیت را غیر فعال می کند.

در تصویر زیر، شما یک URL را مشاهده می کنید که نشان دهنده ی یک لینک است.و زمانی که شما نشانگر موس را بر روی آن ببرید آیکن موس به شکل یک دست تغییر پیدا می کند.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/richtextbox-control-05.jpg)

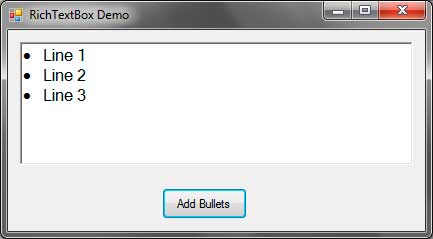
کلیک کردن بر روی Link آنرا در داخل مرورگر شما باز می کند.

**اضافه کردن بالت ها**

شما می توانید در داخل RichTextBox گلوله هایی را به وجود آورید. شما می توانید از خاصیت SelectionBullet و قرار دادن مقدار True برای نشانه گذاری متن انتخاب شده استفاده کنید.

خاصیت BulletIndent فاصله ی هر گلوله را با ابتدای متن مشخص می کند.

برنامه ی زیر یک دکمه دارد که در ابتدای هر خط انتخاب شده یا پارگراف یا جایی که محل درج قرار گرفته است یک گلوله اضافه می کند.

**[](http://www.w3-farsi.com/wp-content/uploads/2013/09/richtextbox-control-06.jpg)**

**فعال و غیر غیر فعال کردن کلیدهای میانبر**

خاصیت ShortcutsEnabled کلیدهای میانبری را که برای Cut کردن و Paste کردن متن بکار می روند فعال یا غیر فعال می کند.

وقتی که مقدار این خاصیت برابر با True باشد ، کلیدهای میانبر زیر توسط RichTextBox قبول می شوند.

| میانبر | عملکرد |
| --- | --- |
| Ctrl + Z | بازگردانی |
| Ctrl + E | وسط چین کردن متن |
| Ctrl + C | کپی کردن متن |
| Ctrl + Y | برعکس Undo |
| Ctrl + X | بریدن قسمتی از متن |
| Ctrl + Backspace | تمامی کلماتی که در سمت چپ نشانگر موس  قرار دارند را حذف می کند |
| Ctrl + V | Paste کردن |
| Ctrl + Delete | تمامی کلماتی را که در سمت راست نشانگر موس  قرار دارند را حذف می کند. |
| Ctrl + A | همه ی متن را انتخاب می کند. |
| Ctrl + L | چپ چین کردن |
| Ctrl + R | راست چین کردن |

وقتی که مقدار این خاصیت را برابر با False قرار دهید تمامی این کلیدهای میانبر در داخل RichTextBox غیر فعال می شوند.

**حفاظت از متن**

شما قابلیت محافظت از قسمتی از متن را با استفاده از خاصیت SelectionProtected را دارا هستید.وقتی که مقدار این خاصیت برابر با True باشد ، قسمتی از متن را که انتخاب کرده اید محافظت می شود.متنی که محافظت شده باشد را نمی توان دستکاری کرد. وقتی که کاربر می خواهد یک متن حفاظت شده را دستکاری کند ، رویداد Protect اتفاق می افتد. شما می توانید بر روی این رویداد مانور دهید، برای مثال ، می توانید یک پیغام را به کاربر نشان دهید مبنی بر اینکه او در تلاش برای دستکاری یک متن حفاظت شده است.

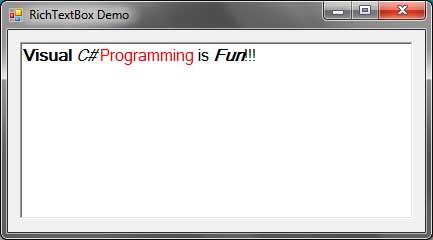
**Rich Text Format (RTF)**

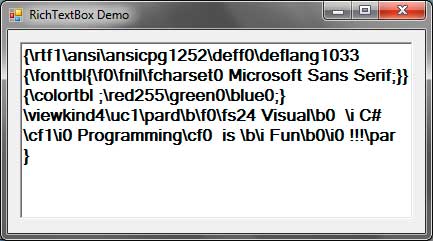
Rich Text Format فایلی است که اطلاعات قالب(فرمت) متن شما را در خود ذخیره می کند.

یک TextBox نمی تواند قالب (فرمت) متن ها در در خود ذخیره کند و فقط می تواند حاوی متن های ساده باشد.

RichTextBox حاوی خواص Text و RTF است که خاصیت Text آن متن ساده را در خود جای می دهد و خاصیت RTF آن، متن و کدهای RTF ی را که برای قالب بندی آن به کار رفته است را در خود جای می دهد.

تصویر زیر یک متن قالب بندی شده را به همراه کدهای RTF آن نشان می دهد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/richtextbox-control-07.jpg)

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/richtextbox-control-08.jpg)

کدهای RTF به برنامه های دیگر (نظیر Word) اجازه می دهد که قالب ها را خوانده و به همان صورت متن قالب بندی شده را به شما نمایش دهند. SelectRTF متن انتخاب شده را به همراه کد RTF آن انتخاب می کند.

این به شما اجازه می دهد که متن قالب بندی شده را به همراه قالب آن در هنگام Cut کردن و Paste  کردن با هم داشته باشید.

## 

## کنترل RadioButton

کنترل RadioButton دکمه ای است که دارای دو حالت خاموش و روشن می باشد. دکمه ی Radio یک دکمه ی دایره ای شکل به همراه یک برچسب است.  شما با کلیک کردن بر روی دکمه ی Radio می توانید آنرا از حالت خاموش به روشن ویا بلعکس تغییر دهید. وقتی که یک دکمه ی Radio روشن باشد ، یک نقطه درر وسط آن قرار می گیرد ، و زمانی که خاموش باشد ، دایره ی آن خالی است.

دکمه های Radio معمولا زمانی استفاده می شوند که یک کاربر می بایست از بین چند گزینه یکی از آنها را انتخاب کند.  برای مثال: زمانی که شما بخواهید جنسیت کاربر را مشخص کنید ، می توانید از دو دکمه ی Radio با نام های مرد و زن استفاده کنید. وقتی که شما از دکمه های Radio استفاده کردید ، فقط می توانید یکی از آن دو را انتخاب کنید.

در جدول زیر برخی از خواص کنترل Radio را مشاهده می کنید.

| خاصیت | توضیح |
| --- | --- |
| Appearance | دکمه ی Radio می تواند همانند یک دکمه ی معمولی و یا یک دکمه ی دایره ای همراه  با یک برچسب در کنار  آن نمایش داده شود. |
| CheckAlign | نوع چینش دکمه را مشخص می کند.حالت پیشفرض MiddleLeft است ، که دکمه را در سمت  برچسب آن نمایش می دهد. |
| Checked | وقتی که مقدار آن برابر با True  باشد دکمه ی Radio روشن می شود و یک نقطه در وسط  دکمه ی آن نمایش داده می شود. |
| Text | متن برچسب دکمه را مشخص می کند. |

دکمه ی Radio  یک رویداد CheckChanged و Click دارد. رویداد CheckChanged زمانی رخ می دهد که حالت دکمه تغییر پیدا کند. برای مثال وقتی که حالت دکمه ی Radio از خاموش به روشن تغییر کند رویداد CheckChanged اجرا می شود.

رویداد Click زمانی رخ می دهد که بر روی دکمه Radio کلیک شود. به طور پیش فرض اگر بر روی دکمه Radio کلیک کنید ، حالت آن تغییر می کند،که باعث می شود رویداد CheckChanged اجرا شود.

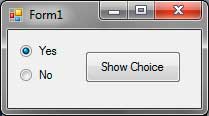
تفاوت بین رویداد Click و CheckChanged زمانی اتفاق می افتد که شما این حالت پیش فرض را تغییر دهید. در اینجا یک خاصیت در کنترل Radio به نام AutoCheck وجود دارد که اگر مقدار آن را به False تغییر کند ، کلیک کردن بر روی دکمه ی Radio دیگر حالت آن را تغییر نمی دهد ، اما باعث می شود که رویداد Click اتفاق بیافتد.

شما می توانید به صورت دستی مقدار خاصیت Checked دکمه ی Radio را  True قرار دهید.

رویداد پیشفرض برای دکمه ی Radio  رویداد CheckChanged است ، بنابراین ، دابل کلیک کردن بر روی این کنترل در حالت طراحی یک Event Handler برای رویداد گفته شده درست می کند.

مثال زیر کاربرد دکمه ی Radio را شرح می دهد.

دو عدد دکمه ی Radio را بر روی فرم قرار دهید .نام آنها را radioButtonYes و radioButtonNo بگذارید.یک کنترل Button  را نیز بسازید و نام آنرا buttonShow قرار دهید و مقدار خاصیت Text آنرا برابر Show Choice قرار دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/radio-button-control-12-20-10-01.jpg)

 بر روی buttonShow دوبار کلیک کنید تا یک handler برای رویداد Click آن ساخته شود.کد های روبرو را در قسمت کد آن وارد کنید.

private void buttonShow\_Click(object sender, EventArgs e)

{

if (radioButtonYes.Checked)

MessageBox.Show("You choosed yes!");

else

MessageBox.Show("You choosed no!");

}

وقتی که شما بر روی buttonShow کلیک می کنید ، برنامه تعیین می کند که کدام Radio انتخاب شده است.  شما می توانید این کار را به وسیله ی خاصیت Checked انجام دهید.

 ما از یک عبارت شرطی IF برای تعیین اینکه Radio انتخاب شده است یا خیر استفاده می کنیم. اگر آن انتخاب نشده باشد ، پس دکمه ی Radio ی دیگر انتخاب شده است . چون ما فقط دو دکمه ی Radio بر روی فرم دارم و نهایتا یکی از آنها انتخاب می شود.

## کنترل CheckBox

کنترل CheckBox (System.Windows.Forms.CheckBox) یک دکمه است و به شکل یک جعبه ی خالی به همراه یک برچسب در کنار آن نمایش داده می شود.

در حالت عادی ،زمانی که بر روی جعبه ی خالی کلیک شود ، یک تیک در داخل جعبه نمایان می شود که به ما می گوید کنترل CheckBox در حالت Checked قرار دارد.

برخلاف دکمه ی Radio که فقط اجازه ی انتخاب یکی از Radio های فرم را به ما می داد، شما می توانید چند عدد CheckBox و یا همه ی آنها را تیک بزنید.

CheckBox نیز خواصی را شبیه به خواص Radio در خود جای داده است. خواص زیر منحصرا مربوط  به کنترل CheckBox هستند.

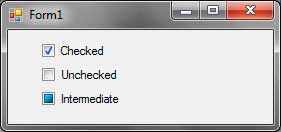
| خاصیت | توضیح |
| --- | --- |
| Checked | وقتی که CheckBox تیک خورده باشد مقدار آن True  می شود. |
| CheckState | مشخص می کند که  CheckBox آیا تیک خورده است یا خیر. |
| ThreeState | وقتی مقدار آن برابر با True باشد.CheckBox یک حالت دیگری را به نام InterMediate را قبول می کند. |

بر خلاف دکمه ی Radio ، کنترل CheckBox می تواند با دادن مقدار True به خاصیت ThreeState سه حالت داشته باشند.

آن حالات Checked ، Unchecked و Intermediate هستند

حالت میانی (Intermediate) نشان می دهد که checkbox بی اعتبار یا غیر قابل تشخیص است.

تصویر زیر شکل این سه حالت مختلف را نشان می دهد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/check-box-12-20-10-01.jpg)

شما با استفاده از خاصیت CheckState میتوانید حالت این کنترل را در قسمت کد نویسی تعیین کنید.

CheckState مقادیری از نوع شمارشی قبول می کند.

checkBox1.CheckState = CheckState.Checked;

checkBox2.CheckState = CheckState.Unchecked;

checkBox3.CheckState = CheckState.Intermediate;

اگر Checkbox فقط دو حالت On  یا Off  ( تیک خورده و تیک نخورده ) را قبول کند، آنگاه شما به سادگی می توانید از خاصیت Checked آن که دو مقدار True به معنی اینکه Checkbox تیک خورده و False به معنی اینکه CheckBox تیک نخورده است را قبول می کند استفاده کنید.

رویداد پیشفرض کنترل CheckBox ،رویداد CheckChanged است.

اما تفاوت بسیار کمی بین رویداد CheckChanged کنترل CheckBox و RadioButton وجود دارد.

برای مثال اگر مقدار خاصیت ThreeState کنترل Checkbox را True قرار دهید ، وقتی CheckBox حالتش را از Checked به InterMediate تغییر دهد ، رویداد CheckChanged آن اتفاق نمی افتد.

اگر می خواهید زمانی که CheckBox از Checked به Intermediate تغییر حالت می دهد یک رویداد اتفاق بی افتد ، می توانید از رویداد CheckStateChanged به جای CheckChanged استفاده کنید.

در مثال زیر نحوه ی کاربرد کنترل CheckBox را مشاهده می کنید.یک فرم را ساخته و یک Lable به آن اضافه کنید ، سپس سه عدد CheckBox و نیز یک Button به آن

اضافه کنید.متن کنترل ها و نحوه ی چینش آنها را مطابق تصویر قرار دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/check-box-12-20-10-02.jpg)

خاصیت Name ،  چک باکس ها را به checkBoxSoap ، checkBoxShampoo ، و checkBoxToothpaste تغییر دهید.

مشخصه ی Name،دکمه را به buttonCheckOut تغییر دهید. نام کنترل label را لازم نیست که تغییر دهید چون در کدنویسی از آن استفاده نمی شود.

string items = String.Empty;

if (checkBoxSoap.Checked)

items += "\n Soap";

if (checkBoxShampoo.Checked)

items += "\n Shampoo";

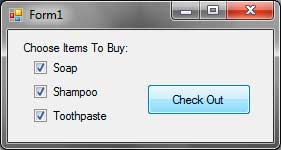
if (checkBoxToothpaste.Checked)

items += "\n Toothpaste.";

MessageBox.Show("You have bought: " + items);

بر روی دکمه( Button ) دوبار کلیک کرده و کدهای زیر را در داخل کنترل کننده ی رویداد (Event Handler) کپی کنید.

برنامه را اجرا کرده و برخی از CheckBox ها را تیک بزنید.وقتی که شما یک CheckBox را تیک می زنید ، و سپس بر روی دکمه Check Out کلیک می کنید ، آیتم هایی که قبلا تیک زده اید در یک پنجره نمایش داده می شوند.به شکل زیر توجه کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/check-box-12-20-10-03.jpg)[](http://www.w3-farsi.com/wp-content/uploads/2013/09/check-box-12-20-10-04.jpg)

در کد بالا ، یک متغیر از نوع رشته ای تعریف کرده ایم و آنرا با یک رشته ی تهی مقدار دهی کرده ایم ( با استفاده از خاصیت Empty از کلاس String).

سپس ما چک می کنیم که کدام یک از چک باکس ها تیک خورده اند  ،هر کدام از آنها که تیک خورده اند نامشان به وسیله ی عملگر += در رشته ای که قبلا به صورت تهی تعریف کرده ایم قرار می گیرد.

سپس به وسیله ی یک پیغام (Message Box) نتایج را در خروجی نمایش می دهیم.

## کنترل های Panel و GroupBox

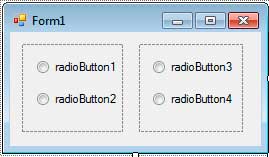
کنترل Panel برای گروه بندی کنترل های فرم بکار می رود.یکی از کاربردهای خوب Panel این است که شما می خواهید دکمه های Radio را گروه بندی کنید.

 با توجه به اینکه فقط یکی از دکمه های Radio در فرم می تواند فعال باشد ، شما می توانید با گروه بندی کردن آنها بیش از یک دکمه ی Radio فعال داشته باشید.

برای انجام اینکار،دو Panel را بر روی فرم قرار دهید.کنترل Panel در قسمت نوار ابزار(ToolBox) قرار دارد.

سایز آنها را تغییر داده ، و در داخل هر یک از آنها دو دکمه ی Radio قرار دهید.

فرم شما باید فرمی شبیه به شکل زیر داشته باشد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/panel-control-12-20-10-01.jpg)

وقتی که شما کنترلی را در داخل یک Panel قرار می دهید ، آن کنترل به فرزند Panel تبدیل می شود و Panel به والد کنترل تبدیل می شود.

بر درک این مساله کافیست ، Panel را حرکت دهید.

هر کنترلی که در داخل Panel قرار دارد به همراه آن حرکت می کند.این ارتباط به والد و فرزند اجازه می دهد که مقادیر خواص عمومی بین آنها به اشتراک گذاشته شود.

به عنوان مثال ، اگر مقدار خاصیت Enable ، پنل  را برابر با False قرار دهیم ، نه تنها خود Panel بلکه تمامی کنترل های داخل آن نیز به حالت غیر فعال در می آیند.

همانطور که می دانید Panel در حالت طراحی به شکل خط تیره نمایش داده می شود.

این خطوط در حالت اجرایی برنامه نمایش داده نمی شوند.اما شما این امکان را دارید که یک حاشیه به پنل اضافه کنید.

در جدول زیر تعدادی از خواص پنل را مشاهده می کنید.

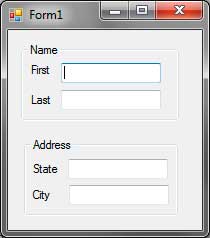
| خصوصیت | توضیح |
| --- | --- |
| BorderStyle | نوع حاشیه ی پنل را مشخص می کند. |
| Controls | مجموعه ی کنترلهای داخل پنل را نشان می دهد. |
| Enabled | به شما اجازه می دهد که پنل را فعال یا غیر فعال کنید. |

حالا برنامه را اجرا کنید.مشاهده می کنید که میتوانید از دسته های مختلف  یک دکمه ی Radio  را انتخاب کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/panel-control-12-20-10-02.jpg)

کنترل GroupBox یک کنترل مشابه است با این تفاوت که به شما اجازه می دهد که برای هر دسته یک عنوان قرار دهید.

شما می توانید با استفاده از خاصیت Text کنترل GroupBox یک عنوان برای آن قرار دهید.همچنین کنترل GroupBox در دور آن به طور پیشفرض یک حاشیه قرار دارد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/panel-control-12-20-10-03.jpg)

## کنترل ComboBox

کنترل ComboBox (نوار کرکره ای ) روش دیگری است که به کاربر اجازه دهید از بین گزینه های مختلف یکی را انتخاب کند. کنترل ComboBox شبیه یک کنترل TextBox است که در سمت راست آن یک دکمه قرار دارد. وقتی بر روی دکمه ی آن کلیک شود ، ComboBox یک نوار کرکره ای را که حاوی یک لیست از گزینه های مختلف است را نمایش می دهد. کاربر می تواند از بین این گزینه ها یکی را انتخاب کند.مورد انتخاب شده به متن داخل ComboBox تبدیل می شود.

در جدول زیر برخی از خواص این کنترل را مشاهده می کنید.

| خصوصیت | توضیح |
| --- | --- |
| DataSource | لیست داده هایی که این کنترل برای گرفتن داده هایش از آنها استفاده می کند. |
| DropDownHeight | ارتفاع نوار کرکره ای بر حسب پیکسل را مشخص می کند. |
| FormatString | قالبی که نحوه ی نمایش مقادیر را مشخص می کند. |
| Items | می توان آیتم هایی را برای نمایش در درون ComboBox را در داخل این خاصیت قرارداد. |
| Sorted | مشخص می کند که آیتم های ComboBox مرتب شوند یا خیر. |
| Text | متن پیشفرضی است که در حالتی که هیچ آیتمی انتخاب نشده است نمایش داده می شود. |
| SelectedIndex | شماره (index) آیتم انتخاب شده را مشخص می کند.هر آیتم یک شماره دارد که از  ۰ تا ( ۱ – تعداد آیتم ها) است. مقدار ۱- مشخص می کند که هیچ آیتمی انتخاب نشده است. |
| SelectedItem | آیتم انتخاب شده را بر می گرداند. |

در جدول زیر رویداد های مربوط به ComboBox را مشاهده می کنید.رویداد پیش فرض آن SelectedIndexChanged است.

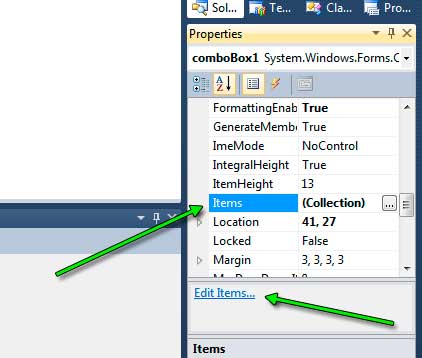
| رویداد | توضیح |
| --- | --- |
| Click | زمانی که بر روی یکی از اجزای آن کلیک شود این رویداد رخ می دهد. |
| DropDown | زمانی که نوار کرکره ای نمایش داده می شود این رویداد رخ می دهد. |
| DropDownClosed | زمانی که نوارکرکره ای بسته می شود این رویداد رخ می دهد. |
| SelectedIndexChanged | زمانی که مقدار خاصیت SelectedIndex تغییر می کند این رویداد رخ می دهد. |

در مثال زیر با کاربرد ComboBox آشنا می شوید.ابتدا در داخل فرم یک ComboBox قرار دهید.

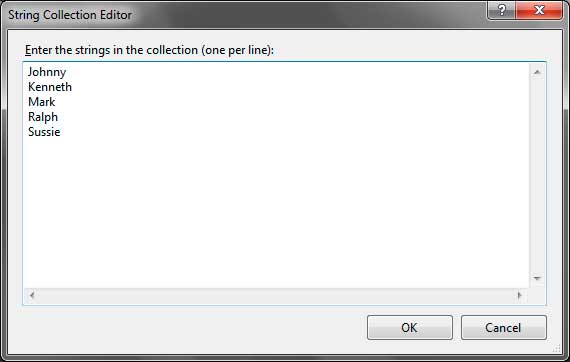
مقدار خاصیت Name آنرا به comboBoxNames تغییر دهید. به پنجره ی Properties رفته و خاصیت Items را پیدا کنید.

شما یک دکمه را به همراه سه نقطه مشاهده می کنید. بر روی آن کلیک کرده تا قسمت String Collection Editor باز شود.

همچنین شما می توانید بر روی Edit Items در قسمت پایین پنجره ی Properties  کلیک کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/combo-box-control-12-20-10-01.jpg)

در پنجره ی StringCollectionEditor ، نامهایی را که در تصویر زیر می بینید را وارد کرده و بر روی OK کلیک کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/combo-box-control-12-20-10-02.jpg)

مقدار مشخصه ی Text کنترل ComboBox را به Choose a name ” تغییر دهید بنابراین مقدار پیش فرض آن وقتی که هنوز آیتمی انتخاب نشده باشد “Choose a name” خواهد بود.

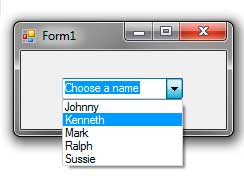
بر روی ComboBox دوبار کلیک کنید . VS/VCE یک کنترل کننده ی رویداد (Event Handler) برای رویداد SelectedIndexChanged می سازد.

کد زیر را در داخل کنترل کننده ی (Event Handler) رویداد وارد کنید .

string selectedName = comboBoxNames.SelectedItem.ToString();

MessageBox.Show("Hello " + selectedName);

برنامه را اجرا کرده و یک نام را انتخاب کنید.وقتی که شما نامی را انتخاب می کنید ،برنامه بلافاصله یک پیغام را به شما نشان می دهد که در آن به نام انتخاب شده خوش آمدگویی می کند.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/combo-box-control-12-20-10-03.jpg)[](http://www.w3-farsi.com/wp-content/uploads/2013/09/combo-box-control-12-20-10-04.jpg)

وقتی که شما یک نام را انتخاب می کنید ، رویداد SelectedIndexChanged رخ می دهد.

خاصیت SelectedItem تمامی داده های مربوط به آیتم انتخاب شده را در بر دارد.

از آنجایی که این خاصیت یک شی (Object) را بر می گرداند ، ما باید مقادیر آنرا به وسیله ی متد ToString() به رشته تبدیل کرده و در داخل یک متغیر String قرار دهیم.(هرچند شما می توانید از خاصیت Text استفاده کنید ولی ما برای این مثال از SelectedItem استفاده کرده ایم)

سپس ما یک متن را که حاوی پیغام خروجی است را نمایش می دهیم.

خاصیت SelectedIndex اندیس (Index) آیتم انتخاب شده را مشخص می کند.

اندیس آیتم اول ۰ است و این اندیس برای آیتم های بعدی به اندازه ی یک واحد اضافه می شود.به طوریکه اندیس هر آیتم از آیتم قبلی خودش یک واحد بیشتر است.

اندیس آخرین آیتم (۱- تعداد آیتم ها) است.بنابراین اگر ما ۱۰ آیتم داشته باشیم ، اندیس آخرین آیتم ۹ است. اگر هیچ آیتمی انتخاب نشده باشد ، اندیس آن برابر با  ۱-  خواهد بود.

اگر شما می خواهید در زمان اجرای برنامه آیتم هایی را اضافه کنید، می توانید از خاصیت Items کنترل ComboBox استفاده کنید.

خاصیت Items یک متد به نام Add دارد که به وسیله ی آن می توانید به ComboBox آیتم های جدیدی را اضافه کنید.

string[] names = { "Johnny", "Kenneth", "Mark", "Ralph", "Sussie" };

//Add each names from the array to the combo box

foreach(string name in names)

{

comboBoxNames.Items.Add(name);

}

شما می توانید لیستی از نام ها را در یک آرایه ی رشته ای قرار دهید. سپس با استفاده از یک حلقه ی Foreach آن نام ها را به ComboBox اضافه کنید.همچنین شما می توانید از خاصیت DataSource نیز استفاده کنید.

comboBoxNames.DataSource = names;

خاصیت DataSource می تواند یک مجموعه یا آرایه را قبول کرده و با آنها لیست آیتم های مربوط به خود را پر کند.

## 

## کنترل ListBox

کنترل ListBox برای نمایش لیستی از رشته ها که قابل انتخاب هستند استفاده می شود.به طور پیش فرض شما فقط می توانید یک آیتم را انتخاب کنید.

کنترل ListBox بهترین گزینه برای مواقعی است که شما می خواهید تعداد زیادی آیتم را نمایش دهید.در جدول زیر برخی از خواص کنترل ListBox را مشاهده می کنید.

| خواص | توضیح |
| --- | --- |
| ColumnWidth | وقتی که مقدار خاصیت MultiColumn برابر با True باشد ، این خاصیت پهنای هر ستون را مشخص می کند. |
| DataSource | منبع داده های را که در ListBox نمایش داده می شوند را مشخص می کند |
| Items | حاوی آیتم هایی است که در ListBox نمایش داده می شوند. |
| MultiColumn | مشخص می کند که ListBox حالت چند ستونی را پشتیبانی می کند یا خیر. |
| SelectedIndex | اندیس آیتم انتخاب شده را مشخص می کند. |
| SelectedIndices | حاوی اندیس آیتمهای انتخاب شده است. |
| SelectedItem | آیتم انتخاب شده را به عنوان یک شی بر می گرداند. |
| SelectedItems | مجموعه ای از اشیاء آیتم های انتخاب شده را بر می گرداند. |
| SelectionMode | تعداد آیتم هایی که شما در یک لحظه می توانید انتخاب کنید را مشخص می کند.  SelectionMode.None- هیچ آیتمی را نمی توانید انتخاب کنید.  SelectionMode.One- فقط یک آیتم را می توانید انتخاب کنید.  SelectionMode.MultiSimple- شما می توانید چند آیتم را به سادگی با کلیک کردن بر روی آنها انتخاب کنید.  SelectionMode.MultiExtended- شما می توانید آیتم های مختلفی را در با نگه داشتن کلید های  Ctrl ، Shift و همچنین کلید های مکان نما انتخاب کنید. |
| ScrollAlwaysVisible | این خاصیت بدون توجه به آیتم های انتخاب شده در ListBox ، Scroll Bar را همیشه نمایش می دهد. |
| Sorted | آیتم های ListBox را براساس حروف الفبا و یا بر اساس ترتیب صعودی مشخص می کند. |
| Text | اگر از یک مقدار رشته ای استفاده شود ، اولین آیتمی که با آن همخوانی داشته باشد انتخاب می شود.این خاصیت، متن اولین آیتم انتخاب شده را بر می گرداند. |

در جدول زیر متدهای مفیدی که شما می توانید آنها را بکار ببرید آورده شده است.

| متدها | توضیح |
| --- | --- |
| ClearSelected() | همه ی آیتم های انتخاب شده ی ListBox را از حالت انتخاب در می آورد. |
| FindString() | اولین آیتم ListBox را که با رشته ی مشخص شده شروع شده باشد را پیدا می کند.جستجو از اندیس مشخص شده شروع می شود. |
| FindStringExact() | اولین آیتمی از ListBox که دقیقا برابر با رشته ی مورد نظر باشد را انتخاب می کند. |
| GetSelected() | به ما می گوید که آیتمی که اندیس مشخص شده را داراست انتخاب شده است یا خیر. |
| SetSelected() | آیتمی را که اندیس مشخص شده را داراست انتخاب و یا از حالت انتخاب در می آورد. |

برای دستکاری آیتم های درون یک لیست باکس ، ما از خاصیت Items که از نوع ObjectCollection است استفاده می کنیم.

شما می توانید از مجموعه ای از متد  ها مثل Add ، Remove ، Clear استفاده کنید.

یک فرم را طراحی کرده و یک ListBox و TextBox به آن اضافه کنید.مقدار خاصیت Multiline کنترل TextBox را به True  تغییر دهید. از طرح بندی زیر استفاده کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/list-box-01.jpg)

مقدار خاصیت Name کنترل ListBox را به listBoxInventory و کنترل TextBox را به textBoxDescription تغییر دهید.بر روی فرم دوبار کلیک کرده تا یک کنترل کننده رویداد (Event Handler)  Load به آن اضافه شود.کدهای زیر را به آن اضافه کنید.

using System;

using System.Collections.Generic;

using System.Windows.Forms;

namespace WindowsFormTutorial

{

public partial classForm1 : Form

{

privateDictionary<string, string> products;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

products = newDictionary<string, string>();

products.Add("Shampoo", "Makes your hair beautiful and shiny.");

products.Add("Soap", "Removes the dirt and germs on your body.");

products.Add("Deodorant", "Prevents body odor.");

products.Add("Toothpaste", "Used to clean your teeth.");

products.Add("Mouthwash", "Fights bad breath.");

foreach (KeyValuePair<string, string> product in products)

{

listBoxInventory.Items.Add(product.Key);

}

}

}

}

ما مجموعه ای به نام Dictionary که یک کلید رشته ای و یک مقدار رشته ای دارد را ایجاد کرده ایم.

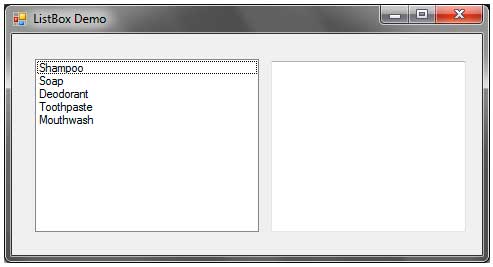
در داخل کنترل کننده ی رویداد Load فرم،ما تعدادی محصول را به همراه توضیحاتی در مورد آنها به این مجموعه اضافه کرده ایم.

با استفاده از حلقه ی Foreach ، ما نام هر محصول را به خاصیت Items لیست باکس اضافه می کنیم.

 به این نکته توجه کنید که هر آیتم در کلکسیون عمومی Dictionary از نوع KeyValuePari<TKey, TValue> می باشد.

وقتی برنامه را اجرا می کنید ، شما پنج محصول را در داخل لیست باکس مشاهده می کنید.

به یاد داشته باشید در صورتیکه ارتفاع لیست باکس شما برای نمایش آیتم های آن کافی نباشد ، یک نوار Scroll عمودی در سمت راست آن دیگه خواهد شد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/list-box-02.jpg)

حالا یک کنترل کننده ی رویداد به رویداد SelectedIndexChanged لیست باکس اضافه می کنیم.رویداد SelectedIndexChanged زمانی رخ می دهد که اندیس آیتم انتخاب شده تغییر کند.

 این رویداد ، رویداد پیشفرض لیست باکس است ، بنابراین با دوبار کلیک کردن بر روی لیست باکس بطور اتوماتیک یک کنترل کننده ی رویداد برای رویداد SelectedIndexChanged ایجاد می شود.کد زیر را به آن اضافه کنید.

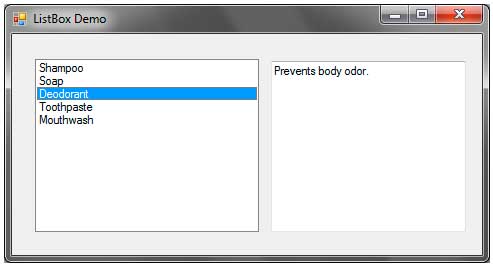
private void listBoxInventory\_SelectedIndexChanged(object sender, EventArgs e)

{

textBoxDescription.Text = products[listBoxInventory.Text];

}

حالا برنامه ی خود را اجرا کرده و یک محصول را انتخاب کنید.مشاهده می کنید که توضیحات مربوط به آن محصول در تکست باکس نمایش داده می شود.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/list-box-03.jpg)

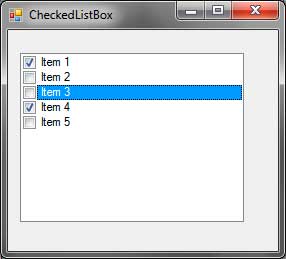
به این نکته نیز توجه داشته باشید که همچنین  می توانید از یک String Collections Editor که در درس آخر نمایش داده شده است برای اضافه کردن آیتم ها به لیست باکس

استفاده کنید.

## کنترل CheckedListBox

کنترل CheckedListBox شبیه کنترل ListBox است ، با این تفاوت که در کنار هر آیتم از آن یک کنترل CheckBox نیز وجود دارد.

 شما می توانید هر آیتم را مانند ListBox انتخاب کنید ، بعلاوه اینکه می توانید هر آیتم را توسط CheckBox کنار آن تیک بزنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/checked-list-box-01.jpg)

خواص و متدهای کنترل ListBox در کنترل CheckedListBox نیز وجود دارند.اما کنترل CheckedListBox دارای خواصی منحصر بفرد نیز می باشد.

| خاصیت | توضیحات |
| --- | --- |
| CheckedItems | مجموعه ای از آیتم های تیک خورده است. |
| CheckOnClick | مشخص می کند که آیا جعبه کنار آیتمی که انتخاب شده است تیک بخورد یا نه؟ |
| ThreeDCheckBoxes | حالت نمایش CheckBox را مشخص می کند ، این خاصیت دو مقدار دارد.  ۱-flat : ظاهر چک باکس دو بعدی است.  ۲-Normal : ظاهر چک باکس ۳ بعدی است. |

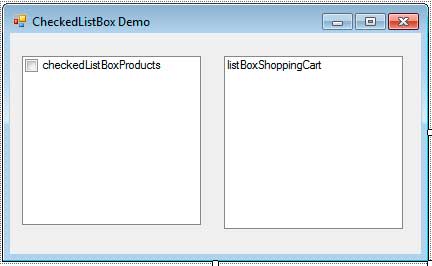
در زیر خواص منحصر به فرد کنترل CheckedListBox آمده است.

| متد | توضیحات |
| --- | --- |
| GetItemChecked() | به ما می گوید که یک آیتمی با یک اندیس مشخص چک خورده است یا خیر. |
| GetItemCheckState() | مقدار CheckState ،آیتمی با یک اندیس مشخص را بر می گرداند. |
| SetItemChecked() | آیتمی با اندیس مشخص را تیک می زند یا تیک آنرا بر می دارد. |
| SetItemCheckState() | حالت چک ، یک آیتم با اندیس مشخص را تعیین می کند. |

حالا برنامه ای می سازیم که در آن یک کنترل CheckedListBox بکار رفته است.

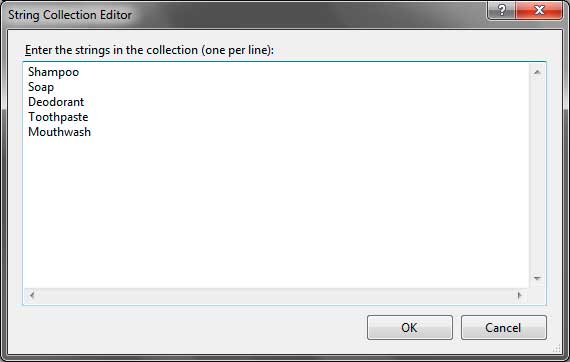
یک فرم را ساخته و یک کنترل   CheckedListBox بر روی آن قرار دهید و مقدار خاصییت Name آنرا به checkedListBoxProducts تغییر دهید.حالا یک ListBox را به فرم اضافه کرده و مقدار خاصیت نام آنرا به listBoxShoppingCart تغییر دهید.

فرم شما باید چیزی شبیه به این باشد.

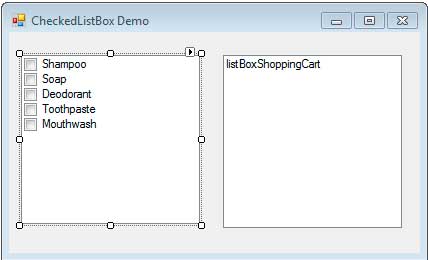
[](http://www.w3-farsi.com/wp-content/uploads/2013/09/checked-list-box-02.jpg)

در پنجره ی خواص (Properties) خاصیت Items کنترل CheckedListBox را پیدا کنید.

بر روی دکمه کنار آن کلیک کنید تا پنجره ی  String Collection Editor باز شود.مقادیر زیر را به آن اضافه کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/checked-list-box-03.jpg)

بر روی OK کلیک کنید.حالا شما آیتم هایی را که به آن اضافه کرده اید را مشاهده می کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/checked-list-box-04.jpg)

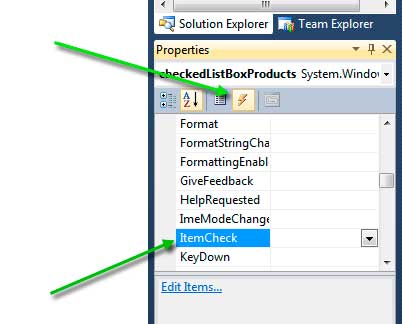
رویداد پیش فرض CheckedListBox رویداد SelectedIndexChanged است. که دقیقا شبیه رویدادی با همین نام در ListBox است.

حال ما می خواهیم آیتم های تیک خورده را به shopping cart اضافه کنیم.

کنترل CheckListBox یک رویداد به نام ItemCheck دارد، زمانی این رویداد رخ می دهد که حالت Check یکی از آیتم ها تغییر کند.

از آنجایی که این رویداد ، رویداد پیشفرض کنترل CheckedListBox نیست ، ما نمی توانیم به سادگی با دوبار کلیک کردن بر روی این کنترل ، این رویداد را ایجاد کنیم.

برای اینکار ما باید به بخش Events در پنجره ی خواص (Properties) رفته و رویداد ItemCheck را پیدا کنیم.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/checked-list-box-05.jpg)

بر روی ItemCheck دوبار کلیک کرده تا یک کنترل کننده ی رویداد برای آن ساخته شود.حالا این کدها را به آن اضافه کنید.

private void checkedListBoxProducts\_ItemCheck(object sender, ItemCheckEventArgs e)

{

if (e.NewValue == CheckState.Checked)

listBoxShoppingCart.Items.Add(checkedListBoxProducts.Items[e.Index]);

else if (e.NewValue == CheckState.Unchecked)

listBoxShoppingCart.Items.Remove(checkedListBoxProducts.Items[e.Index]);

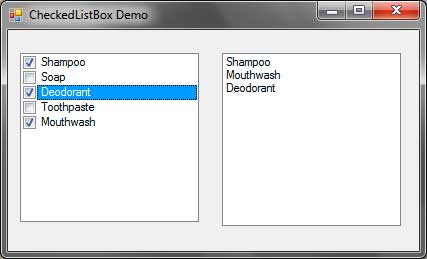
}

در داخل کنترل کننده ی رویداد ItemCheck یک پارامتر دومی از نوع ItemCheckEventArgs نیز وجود دارد که خواص CurrentValue (مقدار فعلی) ، NewValue (مقدار جدید) و Index (اندیس) را در بر می گیرد.

خاصیت CurrentValue مقداری از خاصیت CheckState آن آیتم است که رویداد ItemCheck  را قبل از تغییر CheckState راه اندازی می کند.

خاصیت NewValue مقداری جدیدی است که خاصیت CheckState آن آیتم به خود می گیرد.خاصیت Index ، اندیس آیتم بر مبنای صفر است .(یعنی اندیس ها از صفر شروع می شوند )

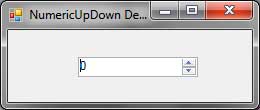
ما امتحان می کنیم اگر مقدار NewValue ی آیتم برابر با Checked بود ، آن آیتم به Shopping Cart اضافه شود و اگر مقدار NewValue برابر با Unchecked بود ، آیتم مورد نظر از Shopping Cart حذف شود.در تصویر زیر مشاهده می کنید که آیتم هایی که تیک زده شده اند به Shopping Cart اضافه شده اند.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/checked-list-box-06.jpg)

## کنترل NumericUpDown

کنترل NumericUpDown عموما برای دریافت اعداد از ورودی و محدود کردن کاربران برای وارد کردن مقادیر غیرعددی بکار می رود.

کنترل NumericUpDown از لحاظ شکل ظاهری شبیه به کنترل TextBox است با این تفاوت که دکمه هایی به شکل پیکان در سمت چپ یا راست آن برای افزایش و یا کاهش مقدار کنترل وجود دارند.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/numeric-updown-01.jpg)

مقدار عددی کنترل NumericUpDown می تواند توسط خاصیت Value مورد دستیابی و یا بازیابی قرار گیرد. نوع داده ی این خاصیت از نوع  Decimal است.

در جدول زیر برخی از خواص کنترل NumericUpDown را مشاهده می کنید.

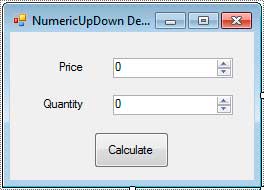
| خاصیت | توضیحات |
| --- | --- |
| DecimalPlaces | تعداد ارقام اعشار را مشخص می کند. |
| Hexadecimal | مشخص می کند که کنترل NumericUpDown مقدارش را به وسیله ی کدهای HexaDecimal نمایش دهد یا خیر. |
| Increment | گام  افزایش یا کاهش مقدار کنترل را مشخص می کند. |
| InterceptArrowKeys | اگر مقدار آن برابر با True باشد ، شما می توانید با استفاده از کلید های مکان نمای کیبورد مقدار را افزایش و یا کاهش دهید. |
| Maximum | بالاترین مقداری را که این کنترل می تواند در خود جای دهد را مشخص می کند. |
| Minimum | کمترین مقداری را که این کنترل می تواند در خود جای دهد را مشخص می کند. |
| ThousandsSeperator | برای جدا کردن ارقام هر رده (یکان,دهگان,صدگان, …) بکار می رود.( برای مثال ۱,۰۰۰) |
| UpDownAlign | مشخص می کند که دکمه های پیکانی در چه مکانی قرار بگیرند.اگر مقدار آن Right باشد ، آنها در سمت راست و اگر مقدار آن Left باشد ، دکمه های پیکانی در سمت چپ کنترل NumericUpDown قرار می گیرند. |
| Value | مقدار کنترل NumericUpDown control را مشخص می کند. |

رویداد پیش فرض کنترل NumericUpDown رویداد ValueChanged است ، این رویداد زمانی اتفاق می افتد که مقدار خاصیت Value این کنترل تغییر کند.

حال می خواهیم یک برنامه بسازیم که کنترل NumericUpDown در آن بکار رفته باشد.

یک فرم را ایجاد کرده و نام آنرا به NumericUpDown تغییر دهید.دو برچسب (lable) و دو کنترل  NumericUpDown بر روی آن قرار دهید.

مقدار خاصیت Text بر چسب ها را به “Price” و “Quantity” تغییر دهید. یک دکمه (Button) را به فرم اضافه کرده و عنوان (Caption) آنرا به “Calculate” تغییر دهید.کنترل ها را مانند شکل ۲ بر روی فرم جایگذاری کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/numeric-updown-02.jpg)

بر اساس جدول زیر خواص کنترل های خود را تغییر دهید.

| کنترل | خاصیت | مقدار |
| --- | --- | --- |
| button1 | Name | buttonCalculate |
| numericUpDown1 | Name | numericUpDownPrice |
| Decimal | 2 |
| Interval | 0.50 |
| Maximum | 10000 |
| numericUpDown2 | Name | numericUpDownQuantity |
| Maximum | 100 |

مقدار خاصیت Decimal کنترل numericUpDownPrice برابر با ۲ قرار داده شده ، بنابراین این کنترل می تواند ۲ رقم اعشار دقت داشته باشد.

مقدار خاصیت Interval نیز برابر با ۰٫۵۰ قرار داده ایم ، بنابراین افزایش و یا کاهش مقدار به اندازه ۰٫۵۰ واحد کم یا زیاد می شود.

مقدار خاصیت Maximum ، ۱۰۰۰۰ است ، پس قیمت (Price) به آن محدود می شود و کاربر نمی تواند رقمی بالاتر از آنرا وارد کند.

از آنجاییکه نهایت مقدار کنترل numericUpDownQuantity را در جدول بالا ۱۰۰ قرار داده ایم بنابراین شما می توانید فقط مقادیری نهایتا تا ۱۰۰ را در آن وارد کنید.

بر روی دکمه (Button) دوبار کلیک کنید تا یک کنترل کننده ی رویداد Click به آن اضافه شود.کدهای زیر را به آن اضافه کنید.

private void buttonCalculate\_Click(object sender, EventArgs e)

{

decimal price = numericUpDownPrice.Value;

int quantity = (int)numericUpDownQuantity.Value;

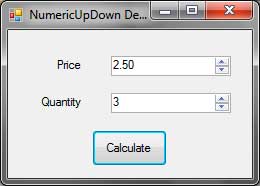
decimal total;

total = price \* quantity;

MessageBox.Show(String.Format("The total price is {0:C}", total));

}

برنامه را اجرا کنید.مقدار کنترل numericUpDownPrice را به وسیله ی دکمه های پیکانی کاهش یا افزایش دهید.مشاهده می کنید مقدار آن به اندازه ۰٫۵۰ واحد افزایش و یا کاهش پیدا می کند.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/numeric-updown-03.jpg)

بر روی دکمه ی Calculate کلی کنید ، این دکمه با ضرب کردن مقدار  Quantity در Price ، قیمت کل را بدست می آورد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/numeric-updown-041.jpg)

## کنترل PictureBox

به وسیله ی کنترل PictureBox می توانید یک تصویر بر روی فرم قرار دهید.

کار اصلی این کنترل نمایش دادن یک تصویر است.تمامی کاری که شما باید بکنید این است که عکس مورد نظر خود را انتخاب کنید تا Visual Studio/VCE آنرا به پروژه ی شما وارد کند.

شما می توانید از فرمت های گوناگونی مثل Jpeg ،PNG ،BMP ، GIF استفاده کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/picture-box-01.jpg)

در جدول زیر برخی از خواص مفید این کنترل را مشاهده می کنید.

| **خواص** | **توضیحات** |
| --- | --- |
| ErrorImage | تصویری که در موقع بالا نیامدن عکس مورد نظر به کاربر نمایش داده می شود. |
| Image | عکسی که توسط این کنترل نمایش داده می شود. |
| ImageLocation | مسیر تصویری که توسط PictureBox نمایش داده می شود. |
| InitialImage | تصویری که در هنگام بارگذاری (Load) تصویر اصلی به کاربر نمایش داده می شود. |
| SizeMode | به ما می گوید که تصویر چگونه نمایش داده خواهد شد.این خاصیت مقدار خود را از  System.Windows.Forms.PictureBoxSizeMode می گیرد. |
| WaitOnLoad | وقتی مقدار آن برابر با True باشد ، همه ی کارهای (تراکنش ها) فرم را تا زمانی که تصویر به صورت کامل بارگذاری شود مسدود می کند. |

برای نمایش یک تصویر در کنترل PictureBox ، چند روش وجود دارد.

شما می توانید به پنجره ی خواص (Properties) رفته و خاصیت Image را پیدا کنید.بر روی دکمه ی سمت راست آن کلیک کرده تا پنجره ی Select Resource  باز شود.

 در پنجره ی باز شده  شما دو انتخاب دارید ، یکی اینکه عکس مورد نظرتان را که در داخل یک درایو (Local resource) موجود است انتخاب کنید و دیگری اینکه عکس را قبلا در داخل فایل پروژه (project’s resouce file) قرار داده اید وارد برنامه کنید.

هر دوی آنها یک دکمه ی Import دارند که به شما اجازه می دهند که تصویر مورد نظر خود را به فرم اضافه کنید.وقتی که شما تصویر مورد نظر خود را انتخاب کردید، VS/VCE تصویر را به کنترل و یا منابع پروژه (Project Resource File ) اضافه می کند.

حال تصویر شما در داخل PictureBox نمایش داده می شود.به یاد داشته باشید که شما می توانید از خاصیت ImageLocation استفاده کنید که از شما مسیر تصویر مورد نظر در داخل هارد را درخواست می کند.

همچنین شما می توانید از مسیر یک تصویر که بر روی صفحات وب قرار دارد استفاده کنید.

ممکن است نمایش عکس با آن چیزی که شما انتظارش را دارید متفاوت باشد.

اگر تصویر بزرگتر از سایز کنترل PictureBox باشد ، فقط قسمتی از آن نمایش داده می شود.

شما می توانید از خاصیت SizeMode برای تغییر سایز یا مکان  تصویر در داخل کنترل استفاده کنید.

این خاصیت از مقادیر شمارشی System.Windows.Forms.PictureBoxSizeMode که در جدول زیر مشاهده می کنید استفاده می کند.

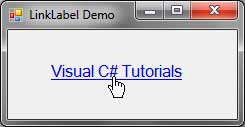
| **PictureBoxSizeMode** | **توضیحات** |
| --- | --- |
| Normal | تصویر در قسمت بالا- چپ جای می گیرد ، و اگر تصویر از اندازه ی PictureBox بیشتر باشد ، تصویر برش داده می شود.( فقط قسمتی از آن نمایش داده می شود ) |
| StretchImage | سایز تصویر را متناسب با اندازه PictureBox تغییر می دهد و تمام عکس را نمایش می دهد. |
| AutoSize | سایز PictureBox را متناسب با اندازه ی تصویر تغییر می دهد و قسمتی از عکس را نمایش می دهد. |
| CenterImage | تصویر در وسط PictureBox قرار می گیرد. اگر سایز تصویر از سایز PictureBox بیشتر باشد ، تصویر برش داده می شود.( فقط قسمتی از وسط عکس در آن نمایش داده می شود ) |
| Zoom | همه عکس را طوری نمایش می دهد که شکل آن در کنترل تغییر نکند مثلا کشیده نشود. |

رویداد پیشفرض این کنترل ، رویداد Click است ، و زمانی رخ می دهد که بر روی عکس یا کنترل کلیک شود.

## کنترل LinkLable

کنترل LinkLable شبیه کنترل Lable (برچسب) است ، با این تفاوت که این کنترل یک زیر خط دارد که شبیه به یک لینک در صفحات وب است.

کنترل LinkLable می تواند برای لینک کردن به فایل ها ، دایرکتوری ها ، و حتی صفحات وب استفاده شود.وقتی نشانگر موس را بر روی LinkLable ببرید ،آیکن نشانگر موس به یک دست تغییر می کند.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/link-label-01.jpg)

خواص این کنترل را در جدول زیر مشاهده می کنید.

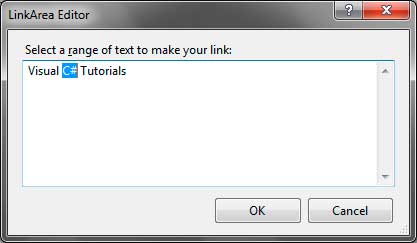
| خاصیت | توضیحات |
| --- | --- |
| BorderStyle | حالت حاشیه ی دور برچسب را مشخص می کند. |
| FlatStyle | ظاهر این کنترل را تعیین می کند.وقتی مقدار آن برابر با Popup باشد ، اگر  بر روی آن کمی منتظر بمانید دکمه ی آن برجسته می شود. |
| Links | به شما اجازه می دهد که در داخل یک متن چند لینک داشته باشید. |
| LinkArea | متن لینک مورد نظر را مشخص می کند. |
| LinkColor | رنگ لینک مشاهده نشده را مشخص می کند. |
| LinkVisited | وقتی مقدار آن برابر با True باشد ، رنگ لینک با رنگ خاصیت VisitedLinkColor عوض می شود. |
| TextAlign | مکان متن داخل کنترل را مشخص می کند. |
| VisitedLinkColor | رنگ لینک مشاهده شده را مشخص می کند. |

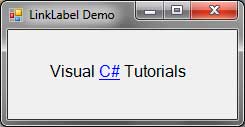
خاصیت LinkArea به شما اجازه می دهد که فقط قسمتی از متن را به عنوان لینک استفاده کنید. خاصیت LinkLabel به دو مقدار نیاز دارد :۱-اندیس شروع ۲- طول.

برای مثال شما می خواهید فقط قسمت “C#” به عنوان لینک نمایش داده شود ، برای اینکار باید مقدار اندیس را برابر با ۷ و طول آنرا برابر با ۲ قرار دهید.

همچنین شما می توانید بر روی دکمه ی سمت راست خاصیت LinkArea در پنجره ی خواص (Properties) کلیک کنید تا پنجره ی LinkArea Editor نمایش داده شود.

در پنجره ی باز شده به راحتی قسمتی را که می خواهید به عنوان لینک نمایش داده شود را انتخاب کرده و دکمه ی OK را بزنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/link-label-02.jpg)

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/link-label-03.jpg)

خاصیت Links به شما اجازه می دهد که در داخل یک متن چند لینک داشته باشید.

متاسفانه ، شما از طریق پنجره ی خواص (Properties) نمی توانید به این خاصیت دسترسی داشته باشید ، برای دسترسی به این قابلیت ، باید کدها و مقادیر آنها را به صورت دستی وارد کنید.شما می توانید کدهای زیر را در رویداد Load فرم قرار دهید.

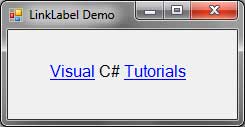
private void LinkLabelForm\_Load(object sender, EventArgs e)

{

linkLabel1.Links.Add(newLinkLabel.Link(0, 6));

linkLabel1.Links.Add(newLinkLabel.Link(10, 9));

}

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/link-label-04.jpg)

خاصیت Links مجموعه ای از اشیاء of LinkLabel.Link می باشد در نتیجه از متد Add و ایجاد یک نمونه از کلاس LinkLabel.Link استفاده کرده ایم (به این نکته توجه کنید که Link کلاسی است که در داخل کلاس LinkLable قرار دارد.).

کنترل LinkLable باید دارای یک کنترل کننده ی رویداد برای رویداد LinkClicked باشد.مکانی که قرار است با کلیک بر روی این کنترل به آنجا منتقل شوید در داخل کنترل کننده ی رویداد مشخص می شود.

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

System.Diagnostics.Process.Start("http://visualcsharptutorials.com");

linkLabel1.LinkVisited = true;

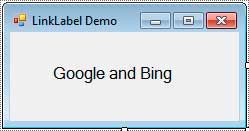
}

شما می توانید از متد System.Diagnostics.Process.Start برای باز کردن مرورگر و مرور صفحات وب استفاده کنید.

سپس مقدار خاصیت LinkVisited را به True تغییر دادیم تا زمانی که یک لینک مشاهده شد ، رنگ آن تغییر کند.

به یاد داشته باشید که شما همچنین می توانید یک مسیر را بر روی هارد انتخاب کنید.

اگر شما در داخل کنترل LinkLabel بیش از یک لینک دارید،شما باید از خاصیت Links از LinkLabelLinkClickedEventArgs استفاده کنید ، تا این رویداد بداند که  دقیقا بر روی کدام لینک کلیک شده است.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/link-label-05.jpg)

private void LinkLabelForm\_Load(object sender, EventArgs e)

{

LinkLabel.Link googleLink = newLinkLabel.Link(0, 6);

LinkLabel.Link bingLink = newLinkLabel.Link(11, 4);

googleLink.Name = "Google";

bingLink.Name = "Bing";

linkLabel1.Links.Add(googleLink);

linkLabel1.Links.Add(bingLink);

}

ما یک نمونه های دیگر از LinkLabel با مکانهای مربوط به خودشان ایجاد کردیم.

حال ما هر نام را به لینک مربوط به آن از طریق خاصیت Name وصل می کنیم.

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

switch (e.Link.Name)

{

case"Google":

System.Diagnostics.Process.Start("http://google.com");

break;

case"Bing":

System.Diagnostics.Process.Start("http://bing.com");

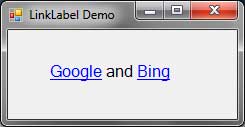
break;

}

}

در داخل کنترل کننده ی رویداد LinkClicked کنترل LinkLabel ، مکان لینکی که فشرده شده است با استفاده از خاصیت Link دریافت می کنیم.

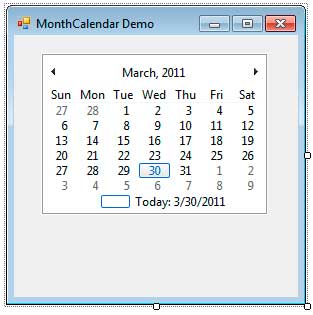
و در آخر تست می کنیم که بر روی نام کدام لینک کلیک شده است مرورگر سایت مربوط به آن را باز کند.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/link-label-06.jpg)

## کنترل MonthCalendar

کنترل MonthCalendar (System.Windows.Forms.MonthCalendar ) شبیه به یک تقویم است ، و یک ماه را به همراه روزهای آن نشان می دهد.

این کنترل به شما اجازه می دهد که یک ماه و یک تاریخ را انتخاب کنید ، برای انتخاب یک ماه بر روی پیکان های سمت راست و چپ این کنترل کلیک کنید تا به ماههای بعدی و قبلی ماه جاری دسترسی داشته باشید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/monthcalendar-control-01.jpg)

با کلیک بر روی نام ماه ، تمامی ماه های سال جاری نمایش داده می شوند.

اگر مجددا بر روی آن کلیک کنید ، تمامی سالهای دهه ی اخیر و اگر برای بار سوم بر روی آن کلیک کنید تمامی سالهای قرن حاضر را به شما نمایش می دهد.

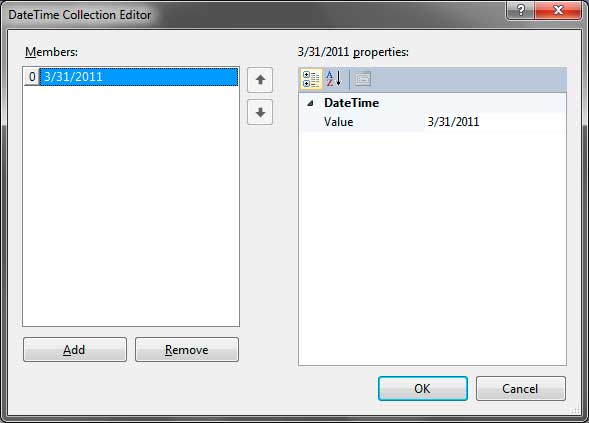
شما می توانید با استفاده از خواص این کنترل آنرا سفارشی کنید.

| **خاصیت** | **توضیحات** |
| --- | --- |
| AnnualyBoldedDates | مجموعه ای از تاریخ های مشخص یک سال ، در کنترل MonthCalendar به صورت ضخیم (Bold) نمایش داده می شوند. |
| BoldedDates | مجموعه ای از تاریخ های غیر تکراری را به صورت ضخیم نمایش می دهد. |
| CalendarDimensions | تعداد سطرها و ستونهای ماه هایی که نمایش داده می شوند را مشخص می کند. |
| FirstDayOfWeek | مشخص می کند که کدام روز اولین روز هفته باشد. |
| MaxDate | حداکثر تاریخ قابل قبول برای این کنترل را مشخص می کند. |
| MaxSelectionCount | حداکثر روزهایی را که کاربر می تواند به طور همزمان انتخاب کند را مشخص می کند. |
| MinDate | حداقل تاریخ قابل قبول برای این کنترل را مشخص می کند. |
| MonthlyBoldedDates | مجموعه ای از تاریخ های مشخص در یک ماه ، که به طور ماهانه به صورت ضخیم نمایش داده می شوند. |
| ScrollChange | تعداد ماه هایی را که با هر بار کلیک بر روی فلش ها حرکت می کنند. |
| SelectionEnd | وقتی کاربر یک بازه از تاریخ را انتخاب کند ( مثلا چند روز در یک ماه ) ، این خاصیت آخرین تاریخ در این بازه را مشخص می کند. |
| SelectionRange | اگر کاربر یک بازه از تاریخ را انتخاب کند ، این خاصیت تمامی تاریخ های درون این بازه را در خود جای می دهد. |
| SelectionStart | وقتی کاربر یک بازه از تاریخ را انتخاب کند ، این خاصیت اولین تاریخ در این بازه را مشخص می کند. |
| ShowToday | مشخص می کند که تاریخ امروز در قسمت پایین این کنترل نمایش داده شود یا خیر. |
| ShowTodayCircle | اگر مقدار آن برابر با True قرار گیرد ،تاریخ امروز در داخل یک کادر مربعی و یا یک کادر دایره ای قرار می گیرد. |
| ShowWeekNumbers | مشخص می کند که شماره ی هفته در سمت چپ هر سطر کنترل نمایش داده شود یا خیر. |
| TodayDate | تاریخی که توسط کنترل MonthCalendar به عنوان تاریخ امروز استفاده می شود. |

می توانید تاریخ های مشخصی را به صورت سالانه ضخیم کنید.برای مثال ، مشخص کنید که بیست و پنجمین روز دسامبر برای نشان دادن روز کریسمس ضخیم شود.

برای اینکار از خاصیت AnnuallyBoldedDates که مجموعه ای از تاریخ ها را قبول می کند استفاده می کنیم.به سادگی به پنجره ی خواص (Properties) رفته و این خاصیت را پیدا کرده و بر روی فلش نوار کرکره ای آن کلیک کنید.

پنجره ی DateTime Collection به شما نشان داده خواهد شد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/monthcalendar-control-02.jpg)

 به سادگی با کلیک کردن بر روی دکمه های Add و Remove تاریخ های مورد نظر خود را اضافه و یا حذف کنید. در این پنجره بر روی یک آیتم کلیک کرده و تاریخ آنرا به وسیله ی خاصیت Value در قسمت Properties در سمت راست آن مشخص کنید. با کلیک بر روی نوار کرکره ای یک انتخاب کننده ی تاریخ به شما نمایش داده می شود که شما به وسیله ی آن می توانید تاریخ مورد نظر خود را انتخاب کنید ، همچنین به جای اینکار شما به سادگی می توانید یک تاریخ را به دلخواه در داخل آن تایپ کنید.

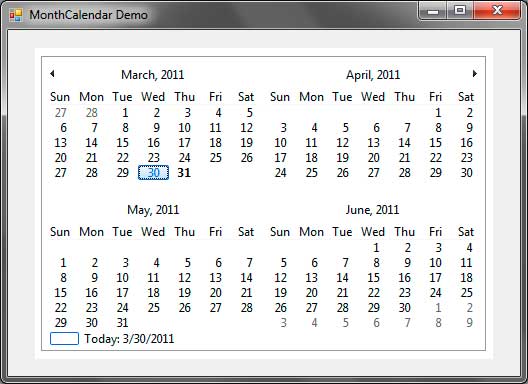
خاصیت MonthlyBoldedDates تاریخ هایی را به صورت ماهانه مشخص شده اند را به صورت ماهانه ضخیم می کند ، در حالیکه خاصیت BoldedDates یک تاریخ را مشخص می کند که فقط یکبار ضخیم می شود و به صورت ماهانه یا سالانه تکرار نمی شوند. به طور پیش فرض می توانید ۷ تاریخ پشت سر هم را انتخاب کنید.اینکار به وسیله ی دکمه ی Shift و انتخاب کردن اولین و آخرین تاریخ انجام می شود.

شما می توانید مقدار خاصیت MaxSelectionCount را تغییر دهید.همه ی تاریخ هایی که بین این دو تاریخ انتخاب شده وجود دارند ، انتخاب خواهند شد. اگر تعداد تاریخ های انتخاب شده از حد مقدار خاصیت MaxSelectionCount تجاوز کند،به طور اتوماتیک تعداد تاریخ های انتخاب شده برابر با مقدار این خاصیت قرار می گیرد.( برای مثال اگر شما ۸ تاریخ را انتخاب کنید ، و مقدار خاصیت

MaxSelectionCount برابر با ۷ باشد ، به طور اتوماتیک ۷ تاریخ انتخاب خواهند شد.) شما می توانید از خواص SelectionStart, SelectionEnd , SelectionRange برای برگرداندن تاریخهای انتخاب شده استفاده کنید. اگر فقط یک تاریخ انتخاب شده بود ، به سادگی می توانید از SelectionStart برای برگرداندن آن استفاده کنید.

به طور پیشفرض فقط یک ماه نمایش داده می شود ، که فقط یک سطر و یک ستون دارد.شما می توانید این حالت را به وسیله ی خاصیت CalendarDemensions تغییر دهید.

برای مثال، در تصویر زیر یک کنترل MonthCalendar  ۲ به ۲ را مشاهده می کنید که چهار ماه را به طور همزمان نمایش می دهد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/monthcalendar-control-03.jpg)

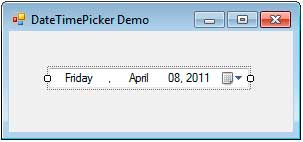
 شما می توانید رویداد های DateSelected و DateChanged را کنترل کنید. رویداد DateSelected زمانی که یک تاریخ یا بازه ای از تاریخ ها انتخاب می شود اتفاق می افتد. رویداد DateChanged زمانی که کاربر تاریخ و یا بازه ای از تاریخ های انتخاب شده را تغییر می دهد اتفاق می افتد.

## کنترل DateTimePicker

کنترل DateTimePicker (System.Windows.Forms.DateTimePicker) برای انتخاب یک تاریخ بکار می رود.

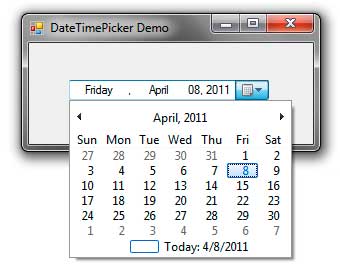
این کنترل به طور پیشفرض به شکل یک ComboBox که یک آیکن تاریخ در سمت راست آن قرار دارد است.

 شما می توانید هر کدام از اجزای تاریخ مانند ماه را انتخاب کرده و به وسیله ی کلیدهای مکان نما آنها تنظیم کنید.

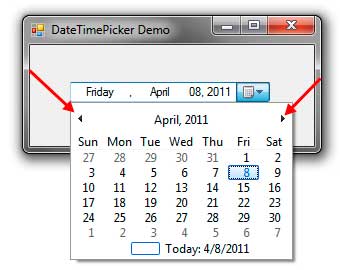
[](http://www.w3-farsi.com/wp-content/uploads/2013/09/datetimepicker-control-011.jpg)

به طور پیشفرض ، این کنترل تاریخ جاری یا تاریخ انتخاب شده را نمایش می دهد.

در زمان اجرای برنامه با کلیک بر روی آیکن تاریخ ، یک تقویم به شما نمایش داده می شود که شما از طریق آن می توانید یک تاریخ را انتخاب کنید.

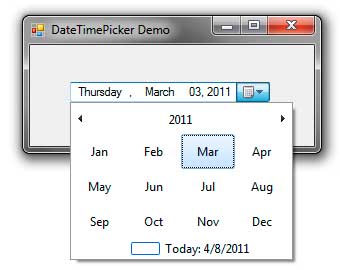
[](http://www.w3-farsi.com/wp-content/uploads/2013/09/datetimepicker-control-021.jpg)

شما از طریق این تقویم می توانید یک تاریخ مشخص را انتخاب کنید.همچنین از طریق پیکان های پیمایش در سمت چپ و راست آن می توانید به ماه بعد و قبل بروید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/datetimepicker-control-031.jpg)

نمای ابتدایی تقویم، تمامی روزهای ماه جاری را به همراه بعضی از روزهای ماه های قبل و بعد را نمایش می دهد.

با کلیک بر روی عنوانی که ماه و سال را نمایش می دهد ، می توانید نما را تغییر داده و همه ی ماههای سال جاری را نمایش دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/datetimepicker-control-04.jpg)

با یکبار کلیک کردن روی سال ، تمامی سالهای دهه ی جاری و با دوباره کلیک کردن روی آن به حالت اولیه باز می گردد.

در جدول زیر برخی از خواصی را که به شما اجازه می دهد که رفتار و ظاهر این کنترل را تغییر دهید را مشاهده می کنید.

| خاصیت | توضیحات |
| --- | --- |
| CalendarFont | نوع قلم تقویم را مشخص می کند. |
| CalendarForeColor | رنگ پیش زمینه ی تقویم را مشخص می کند. |
| CalendarMonthBackground | پشت زمینه ی ماه تقویم را مشخص می کند. |
| CalendarTitleBackColor | رنگ پشت زمینه ی عنوان تقویم را مشخص می کند. |
| CalendarTitleForeColor | رنگ پیش زمینه ی عنوان تقویم را مشخص می کند. |
| Checked | برای کار با این خاصیت باید مقدار خاصیت ShowCheckBox ، False باشد.وقتی مقدار این خاصیت برابر با True باشد ، تاریخ انتخاب شده می تواند تغییر و یا بهنگام سازی شود ، و اگر مقدار آن برابر با False باشد ، تاریخ انتخاب شده نمی تواند تغییر کند. |
| CustomFormat | یک قالب سفارشی رشته ای  که برای نمایش قالب تاریخ استفاده می شود را قبول می کند. |
| Format | نوع فرمت (قالب) تاریخی که توسط کنترل نمایش داده می شود را مشخص می کند. |
| MaxDate | حداکثر تاریخ و ساعتی که در کنترل می تواند انتخاب شود را مشخص می کند. |
| MinDate | پایین ترین تاریخ و ساعتی که در کنترل می تواند انتخاب شود را مشخص می کند. |
| ShowCheckBox | اگر مقدار آن برابر با True باشد ، یک تکست باکس در قسمت سمت چپ کنترل نمایش داده می شود.وقتی مقدار آن Checked باشد ، تاریخ انتخاب شده می تواند تغییر کند. و وقتی مقدار آن Uncheck  باشد تاریخ انتخاب شده نمی تواند تغییر کند. |
| ShowUpDown | وقتی مقدار آن True باشد، یک دکمه ی بالا-پایین به جای دکمه ی نوار کرکره ای قرار می گیرد.در این حالت شما نمی توانید به تقویم دسترسی داشته باشید.به جای آن ،باید اجزای تاریخ را انتخاب کرده و با استفاده از این دکمه ی بالا-پایین آنها را تنظیم کنید. |
| Value | تاریخ انتخاب شده ی جاری را نمایش می دهد. |

اگر از ویندوز Vista یا ۷ و نیز از تم هایی (Theme) مانند Aero استفاده کنید ، تاثیر خواصی که مربوط به دستکاری رنگ تقویم هستند از بین می رود.

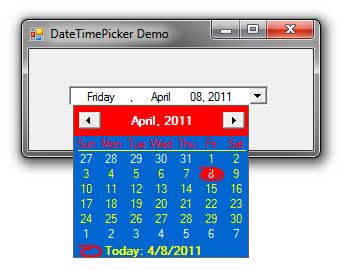
برای مشاهده ی نتیجه ی تغییر رنگ تقویم ، باید جلوه های ویژه (Visual Styles) را غیر فعال کنید.

برای درک این مطلب ، به پنجره ی Solution Explorer رفته و Program.cs را پیدا کرده و بر روی آن دوبار کلیک کنید و خط زیر را پاک کنید.

Application.EnableVisualStyles();

بیاد داشته باشید اگر این خط را پاک کنید ، کنترل ها حالت Classic ( قدیمی ) به خود می گیرند.

حال می توانید با استفاده خواص مختلفی که رنگ تقویم را تغییر می دهند ، رنگ آنرا تغییر دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/datetimepicker-control-05.jpg)

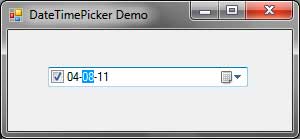
شما می توانید قالب نمایش تاریخ را با استفاده از خاصیت Format تغییر دهید.شما می توانید از قالب های Long ، Short ، Time ، Custom استفاده کنید.

زمانی که Custom را انتخاب کنید ، می توانید یک قالب رشته ای برای آن در خاصیت CustomFormat تعیین کنید.

برای مثال ، قرار دادن فرمت به MM-dd-yy تاریخی را نمایش می دهد که در آن شماره ی ماه ، روز و دو شماره ی آخر سال که به وسیله ی یک خط تیره از هم جدا شده اند را به کاربر نمایش می دهد.

برای مثال خروجی ۰۴-۰۸-۱۱ برای ما تولید خواهد شد.

خاصیت ShowCheckBox مشخص می کند که یک CheckBox در سمت چپ این کنترل قرار گیرد یا خیر.

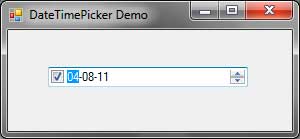
[](http://www.w3-farsi.com/wp-content/uploads/2013/09/datetimepicker-control-06.jpg)

اگر CheckBox تیک خورده باشد ، می توانید یک تاریخ یا زمانی را انتخاب کرده و با استفاده از کلید های مکان نما آن را تغییر دهید.

اگر چک باکس تیک نخورده باشد ، شما مجاز به انجام این کار نیستید.

با کلیک بر روی نوار کرکره ای ، checkbox  کنار تقویم تیک می خورد.حالت CheckBox با استفاده از خاصیت Checked مورد دستیابی قرار می گیرد.

خاصیت ShowUpDown دکمه ی نوار کرکره ای را به یک دکمه ی بالا-پایین تغییر داده و آیکن تقویم را مخفی می کند.در شکل زیر عملکرد آنرا مشاهده می کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/datetimepicker-control-07.jpg)

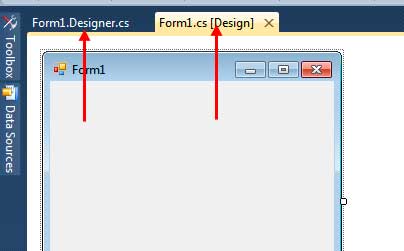
این خاصیت شما از دیدن تقویم منع کرده و شما را به استفاده از دکمه ی بالا- پایین محدود می کند.

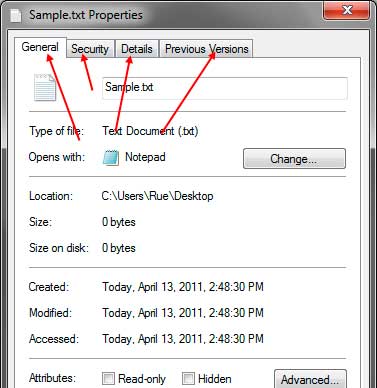
یکی از اجزای تاریخ و زمان را انتخاب کرده و با استفاده از دکمه ی بالا – پایین مقدار آن را تغییر دهید.

از خاصیت Value برای مقدار دادن به تاریخ انتخاب شده نیز می توانید استفاده کنید.با استفاده از رویداد ValueChanged این کنترل می توانید به تغییرات تاریخ واکنش نشان دهید.(یعنی براساس تغییراتی که کاربر در آن ایجاد می کند ، یک پیغام را نشان داده و یا یک کد را اجرا کنید )

## کنترل TabControl

کنترل TabContorl  (System.Windows.Forms.TabControl) به شما اجازه می دهد که برای پنجره های خود سربرگ (Tab) بسازید،شما در برنامه های زیادی این نوع پنجره ها را مشاهده کرده اید.برای مثال پنجره ی Properties فایلها و سربرگ های برنامه ی Visual Studio.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tabcontrol-01.jpg)

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tabcontrol-02.jpg)

TabControl می تواند حاوی کنترل های دیگر باشد. با کلیک بر روی هر سربرگ (Tab)  می توانید محتویات آنرا مشاهده کنید. ظاهر یک سربرگ فعال با دیگر سربرگ ها متفاوت است ، بنابراین شما می توانید متوجه شوید که کدام یک از سربرگ ها فعال است.زمانی که شما بر روی یک سربرگ کلیک می کنید ، کنترل هایی که به آن تعلق دارند نمایش داده می شوند. TabControl به شما اجازه می دهد که یک فرم را در داخل سربرگهای مختلف سازماندهی کنید ، به طوری که هر سربرگ یک دسته را نمایش دهد.  برای مثال،می توانید یک فرم که حاوی اطلاعات شخصی را در سربرگ Personal Info و پیش زمینه ی آموزشی را در سربرگ Educational Background قرار دهید.

برای اضافه کردن یک کنترل TabContorl ، به قسمت Containers در جعبه ابزار ( ToolBox) رفته و کنترل TabControl را بر روی فرم قرار دهید. می توانید سایز این کنترل را تغییر دهید اما بهترین راه استفاده از خاصیت Dock است ، که باعث می شود تمام فضای فرم اشغال شود. به پنجره ی خواص (Properties) رفته و خاصیت Dockرا پیدا کنید.بر روی نوار کرکره ای کلیک کرده و سپس دکمه ی وسط (Middle) را انتخاب کنید. این گزینه به این معناست که می تواند تمام  فرم را پر کند.(به شکل زیر توجه کنید)

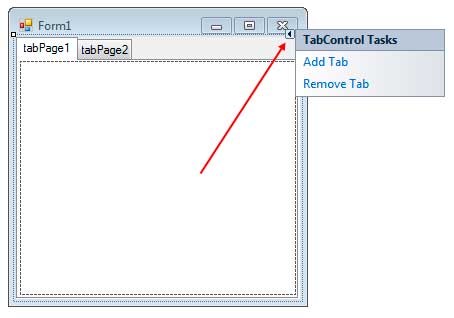
[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tabcontrol-03.jpg)

TabControl شامل کنترلهای TabPage است ، که هر کدام از آنها شامل سربرگ و اجزای مخصوص به خودشان هستند.این کنترل ها توسط خاصیت TabPages از کنترل TabControl مورد دستیابی قرار می گیرند.

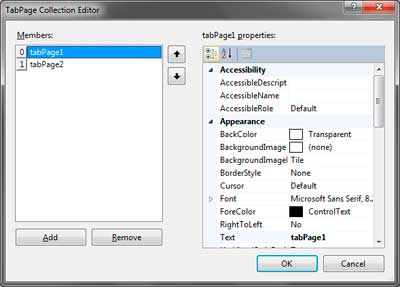
 در جدول زیر لیستی از خاصیت های مفید  کنترل TabPage را مشاهده می کنید.

| **خاصیت** | **توضیح** |
| --- | --- |
| Controls | کنترل هایی که در داخل TabPage وجود دارند. |
| ImageIndex | اندیس تصویری که برای Tabpage انتخاب می شود را تعیین می کند. |
| Text | متنی که در دکمه ی سربرگ نمایش داده می شود. |

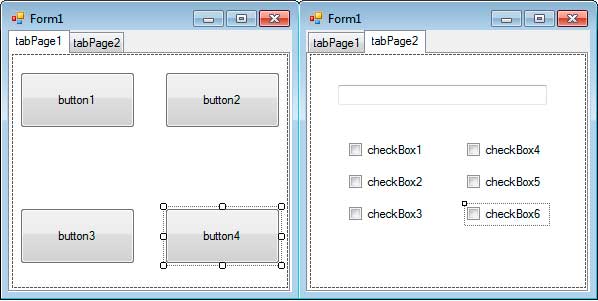
برای اضافه کردن یک TabPage ( سربرگ صفحه ای ) ، می توانید بر روی فلش کوچک که در قسمت بالا و راست TabControl وجود دارد کلیک کنید. با زدن این دکمه یک منو باز می شود که به وسیله ی آن شما می توانید Tab ها اضافه و یا حذف کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tabcontrol-04.jpg)

روش دیگر  ،  استفاده از  پنجره ی Properties رفته و خاصیت TabPages می باشد. برای باز کردن TabPages Collection Editor ابتدا مطمئن شوید که کنترل TabControl فعال است سپس بر روی دکمه در کنار خاصیت TabPages کلیک کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tabcontrol-05.jpg)

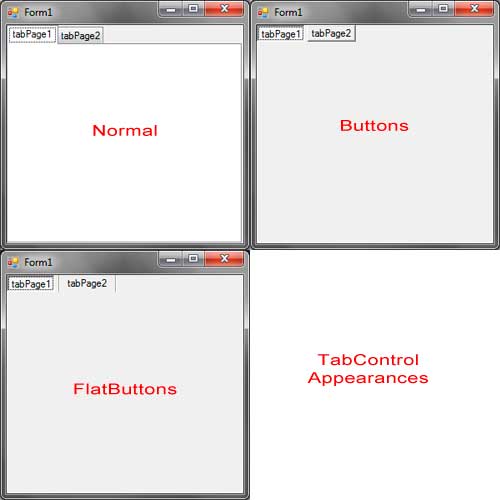
پنجره TabPage Collection Editor به شما اجازه می دهد که TabPage های مختلفی را ایجاد یا حذف کرده و خواص مربوط به هرکدام از آنها را تغییر دهید. هر TabPage می تواند در برگیرنده ی کنترل های مخصوص به خود باشد. به عنوان مثال چندین کنترل را به سربرگ اول مانند شکل زیر اضافه کنید سپس بر روی سربرگ دیگر کلیک کرده و کنترل های دیگری را به آن اضافه نمایید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tabcontrol-06.jpg)

حال اجازه دهید در مورد خود TabControl توضیح دهیم.در زیر  برخی از خواص مهم این کنترل ذکر شده اند :

| **خاصیت** | **توضیح** |
| --- | --- |
| Alignment | ناحیه ای که Tab ها بر اساس آن تراز بندی می شوند.مقدار پیشفرض آن top است. |
| Appearance | ظاهر Tab ها را تعیین می کند. |
| ImageList | شامل لیستی از عکس هایی است که می توانید برای هر سربرگ تعیین  کنید. |
| ItemSize | اندازه tab را تعیین می کند. |
| Multiline | به شما اجازه می دهد tab ها را در چندین سطر نمایش دهید.هنگامی کاربرد دارد که تعداد tab ها زیاد باشد. |
| SelectedIndex | اندیس tab جاری را تعیین می کند. |
| SelectedTab | Tab جاری را تعیین یا بر می گرداند. |
| TabCount | تعداد tab های کنترل را بر میگرداند. |
| TabPages | به شما اجازه مدیریت(حذف و اضافه کردن) tab ها را می دهد. |

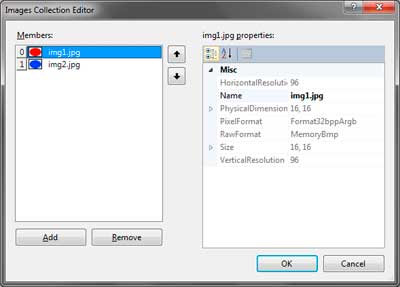
خاصیت Alignment مکان قرار گیری tab ها را تعیین می کند. در حالت پیشفرض مقدار این خاصیت برابر top است که باعث نمایش tab ها در بالای tabcontrol می شود.خاصیت Appearance شکل ظاهری tab ها را تعیین می کند.این خاصیت سه مقدار Normal(پیشفرض) ، Buttons(نمایش tab ها به شکل دکمه) ، FlatButtons(نمایش tab ها به شکل دکمه های مسطح) . حالت های مختلف این ۳ مقدار را در شکل زیر نمایش داده شده است.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tabcontrol-07.jpg)

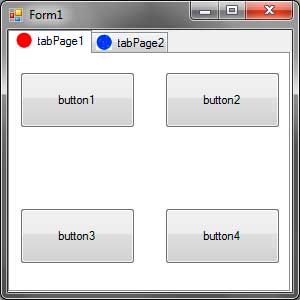
خاصیت ImageList به شما اجازه می دهد که برای هر tab آیکونی را مشخص کنید.این خاصیت به یک کنترل ImageList ارجاع دارد.کنترل ImageList در این خاصیت که به طور مشابه در کنترل های دیگری مانند TreeView هم و منو ها وجود دارد مورد استفاده قرار می گیرد.به طور خلاصه نحوه ی استفاده از این کنترل برای نمایش عکس در TabControl را توضیح می دهیم.یک کنترل ImageList را از جعبه ابزار بر روی فرم قرار دهید.این کنترل شمای ظاهری ندارد و در قسمت کنترل های غیر بصری(Component tray) قرار می گیرد.ImageList از دو عکس زیر استفاده می کند.

[img1](http://www.w3-farsi.com/wp-content/uploads/2013/09/img1.jpg)[img2](http://www.w3-farsi.com/wp-content/uploads/2013/09/img2.jpg)

این دو عکس را بر روی قسمتی از هارد خود  ذخیره و سپس بر روی کنترل ImageList کلیک کنید.خاصیت Images را از پنجره ی Properties پیدا کنید و بر روی دکمه کنار آن کلیک کنید.با کلیک بر روی این دکمه پنجره ی Image Collection Editor نمایش داده می شود.

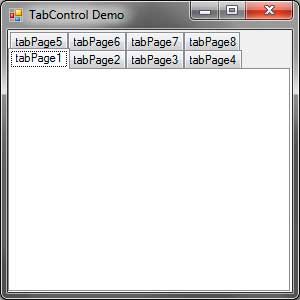
[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tabcontrol-08.jpg)

بر روی دکمه ی Add برای اضافه کردن یک عکس کلیک کنید.برای هر دو عکسی که قبلا بر روی هارد ذخیره کرده اید این مرحله را تکرار کنید و در آخر بر روی دکمه ی OK کلیک کنید.سپس خاصیت ImageList از کنترل TabControl را پیدا کرده و ارجاع این کنترل را به آن نسبت دهید. برای قرار دادن عکس ها در کنار هر tab روی tab مورد نظر کلیک کرده و سپس در پنجره ی Properties خاصیت ImageIndex آن را پیدا کنید.حال بر روی دکمه ی پایین افتادنی کنار آن کلیک و عکس مورد نظر خود را انتخاب نمایید.برای چند tab مختلف می توانید عکس مشابه ای را انتخاب کنید.

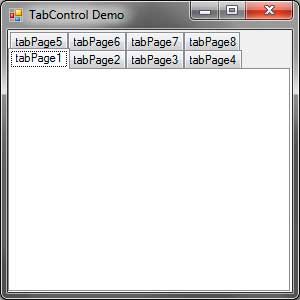
[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tabcontrol-09.jpg)

برای تغییر اندازه ی هر تب آن را انتخاب کنید و خاصیت ItemSize آنرا تغییر دهید.این کار زمانی مفید است که اندازه عکس های کناری tab بسیار بزرگ باشد.

اگر تعاد tab ها کنترل بسیار زیاد و از اندازه ی عرض فرم بیشتر باشد ، دکمه های فلش دار در سمت راست و بالای tabcontrol نمایش داده می شوند و شما می توانید با استفاده از آنها به تب های دیگر دسترسی یابید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tabcontrol-10.jpg) [](http://www.w3-farsi.com/wp-content/uploads/2013/09/tabcontrol-11.jpg)

برای نمایش tab ها در چند سطر می توانید از خاصیت MultiLine استفاده کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tabcontrol-10.jpg)

خاصیت TabPages به شما اجازه حذف یا اضافه کردن tab ها را می دهد.در زیر با کدنویسی tab ها مختلفی را ایجاد کرده و به کنترل اضافه می کنیم.

TabPage tab1 = newTabPage("Tab 1");

TabPage tab2 = newTabPage("Tab 2");

TabPage tab3 = newTabPage("Tab 3");

tabControl1.TabPages.Add(tab1);

tabControl1.TabPages.Add(tab2);

tabControl1.TabPages.Add(tab3);

خصوصیت TabPages دارای  نوع TabPageCollection است که مجموعه ای از اشیا TabPage می باشد.بنابراین شما می توانید از متد های عمومی مانند Add ، Remove ، AddRange استفاده کنید.

برای تعیین تب جاری می توان از خصوصیت SelectedIndex استفاده کرد.این خصوصیت اندیس tab جاری را تعیین می کند.

MessageBox.Show("The selected tab is " + tabControl1.TabPages[tabControl1.SelectedIndex].Text);

خصوصیت SelectedTab شی tab جاری را تعیین یا بر می گرداند.

TabPage selectedTab = tabControl1.SelectedTab;

MessageBox.Show("The selected tab is " + selectedTab.Text);

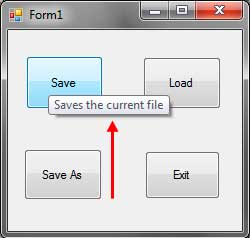
خصوصیت TabCount تعداد تب های کنترل را بر می گرداند.

وقتی که در بین tab های مختلف حرکت می کنید رویداد SelectedIndexChanged اتفاق می افتد.

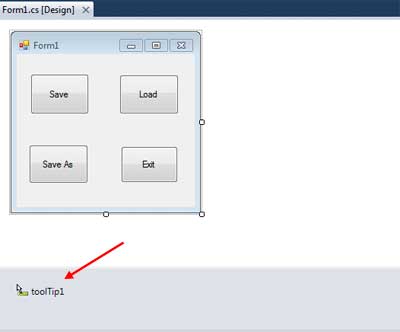
کنترل TabControl زمانی مفید است که فرم از قسمت های مختلفی تشکیل شده باشد و فضای کافی در اختیار نداشته باشید.به عنوان یک مثال زمانی که برنامه دارای قسمتی برای تعیین تنظیمات خود باشد می توانید از این کنترل استفاده کنید.

## کنترل ToolTip

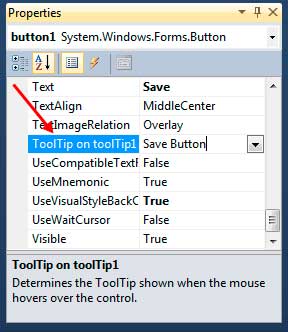
از کنترل ToolTip (System.Windows.Forms.ToolTip) برای نمایش توضیحاتی در مورد هر یک از کنترل های به کار برده شده در GUI استفاده می شود.   Tooltip ها یا همان توضیحات در مورد کنترل ها،  در بسیاری از برنامه های کاربردی دیده می شود و موقعی ظاهر می شوند که نشانگر موس را بر روی کنترل ها نگه داریم.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tooltip-control-01.jpg)

این کنترل در دسته common controls ویژوال استودیو قرار دارد. هنگامی که یک کنترل ToolTip را بر روی فرم می کشید مشاهده می کنید که به قسمتی که در شکل زیر نشان داده شده است منتقل می شود.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tooltip-control-02.jpg)

سپس می توان از آن در کنترل های مختلف استفاده کرد. هنگامیکه این کنترل به فرم اضافه شد، یک خاصیت جدید به نام ToolTip به هر کنترل در فرم اضافه می شود. این خاصیت را می توانید در پنجره properties مشاهده کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tooltip-control-03.jpg)

این خاصیت جدید تعیین کننده همان متنی است که اگر با موس بر روی کنترل مکث کنیم نمایش داده می شود. با اضافه کردن هر tooltip به فرم یک خاصیت tooltip به هر کنترل اضافه می شود. به عنوان مثال اگر شما دو کنترل  ToolTip به فرم اضافه کنید، هر کنترل موجود در فرم دارای دو خاصیت ToolTip می شود.

برخی از خواص کنترل ToolTip در جدول زیر آمده است :

| خاصیت | توضیح |
| --- | --- |
| Active | از این خاصیت برای فعال یا غیر فعال کردن tooltip استفاده می شود. |
| AutomaticDelay | مقدار این خاصیت بر روی سایر خصوصیات مربوط به تاخیر در نمایش، تاثیر می گذارد. |
| AutoPopDelay | مدت زمان نمایش tooltip را تعیین می کند. |
| InitialDelay | مدت زمانی که نشانگر باید بر روی کنترل باشد تا tooltip نمایش داده شود. |
| IsBalloon | مشخص می کند که آیا tooltip از پنجره ی به شکل بالون برای نمایش متن استفاده کند یا نه. |
| ReshowDelay | مدت زمان نمایش tooltip در هنگام جابه جایی نشانگر بین کنترل ها را تعیین می کند. |
| ShowAlways | اگر این گزینه را برابر true قرار دهید ، در صورت غیر فعال بودن کنترل در برگیرنده کنترل باز هم tooltip نمایش داده می شود. |
| ToolTipIcon | برای نمایش icon در tooltip به کار می رود. |
| ToolTipTitle | یک عنوان به متن داخل tooltip اضافه می کند. |
| UseFading | هنگامی که این خاصیت را برابر true قرار دهید ، پنجره tooltip با افکت محو(fade effect) نمایش داده می شود. |

برای اینکه یک Tooltip نمایش داده شود باید مقدار خاصیت Active آن true شود.

خاصیت AutoPopDelay مدت زمان نمایش Tooltip وقتی که ماوس بر روی کنترل قرار می گیرد را مشخص می کند.

خاصیت InitialDelay مشخص می کند که چه مدت زمان ماوس بر روی کنترل قرار بگیرد و سپس Tooltip نمایش داده شود.

خاصیت ReshowDelay مدت زمان دوباره نشان داده شدن Tooltip وقتی که از یک کنترل به کنترل دیگر می روید را مشخص می کند.

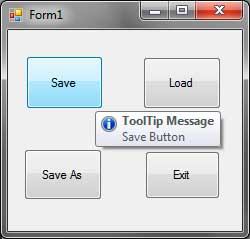
خاصیت AutomaticDelay به طور خودکار سه خاصیت قبل را متعادل می کند.

خاصیت AutoPopDelay ، ۱۰ برابر مقدار AutomaticDelay است.

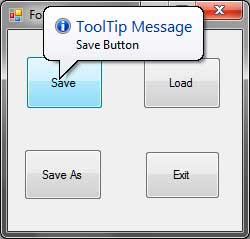
مقدار خاصیت InitialDelay برابر مقدار AutomaticDelay می شود و مقدار خاصیت ReshowDelay نصف مقدار خاصیت AutomaticDelay است.

به این نکته توجه کنید که این خاصیت ها مقداری از نوع صحیح دریافت می کنند و این مقدار بر حسب میلی ثانیه است.

همچنین شما با استفاده از خواص ToolTipTitle و ToolTipIcon می توانید برای tooltip عنوان و آیکن تعیین کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tooltip-control-04.jpg)

همچنین ظاهر tooltip را با استفاده از خاصیت IsBalloon تغییر داد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/tooltip-control-05.jpg)

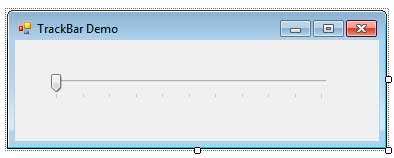
شما می توانید چندین کنترل tooltip را به فرم اضافه کرده که هر کنترل دارای tooltip ی با خواص متفاوت باشد.

همچنین یک کنترل می تواند از چندین tooltip به طور همزمان استفاده کند ، البته این روش توصیه نمی شود.

## کنترل TrackBar

کنترل TrackBar (System.Windows.Forms.TrackBar) شبیه یک نوار لغزنده با یک دستگیره است که با استفاده از این دستگیره می توان مقدار آن را تعیین کنید.

ناحیه ای که دستگیره به آن اشاره می کند نشان دهنده مقدار جاری کنترل است . شکل این کنترل به صورت زیر است :

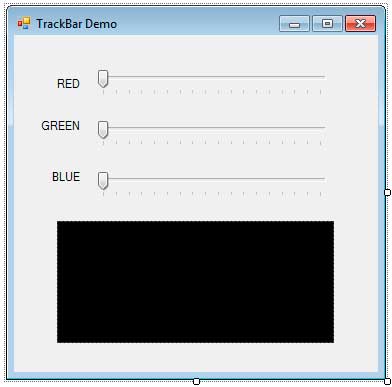
[](http://www.w3-farsi.com/wp-content/uploads/2013/09/trackbar-01.jpg)

کاربر می تواند دستگیره به در جهت افقی حرکت دهد و همچنین میتوان جهت حرکت را به صورت عمودی تغییر داد.

مقدار این کنترل در حالت افقی از سمت چپ به راست و در حالت عمودی از پایین به بالا افزایش می یابد. در زیر تعدادی از خواص این کنترل را مشاهده می کنید .

| خاصیت | توضیح |
| --- | --- |
| LargeChange | مقدار این خاصیت مشخص می کند که مقدار TrackBar با کلیک کردن روی آن یا با استفاده از دکمه های PageDown یا PageUp چه مقدار افزایش یابد. |
| Maximum | بیشترین مقداری که به TrackBar اختصاص داده ایم. |
| Minimum | کمترین مقداری که به TrackBar اختصاص داده ایم. |
| Orientation | نحوه  نمایش TrackBar را تعیین می کند(افقی یا عمودی) |
| SmallChange | مقداری که در هنگام فشار دادن کلید های جهت نما به TrackBar اضافه یا کم می شود را تعیین می کند. |
| TickFrequency | تعداد خطوط عمودی که در زیر TrackBar نمایش داده می شوند. |
| Value | مقدار جاری TrackBar را نشان می دهد. |

هنگامی که شما دستگیره کنترل را جابه جا می کنید رویداد Scroll اتفاق می افتد. در مثال زیر نحوه ی استفاده از کنترل TackBar به شما نشان داده می شود.یک پروژه Windows application ایجاد کرده و ۳ کنترل TackBar و ۳ Label به آن اضافه کنید. همچنین یک کنترل Panel به فرم اضافه کنید و خاصیت BackColor آن را به Black تغییر دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/trackbar-02.jpg)

خاصیت  Name آنها را به ترتیب به trackBarRed ، trackBarGreen و trackBarBlue و  خاصیت Maximum را برابر ۲۵۵ و خاصیت TickFrequency هر یک را به ۱۵ تغییر دهید. به محیط کد نویسی رفته و کنترل کننده رویداد زیر را برای استفاده سه TrackBar بنویسید.

private void trackBar\_Scroll(object source, EventArgs e)

{

int red = trackBarRed.Value;

int green = trackBarGreen.Value;

int blue = trackBarBlue.Value;

Color color = Color.FromArgb(red, green, blue);

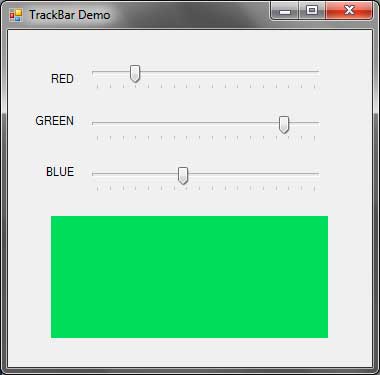
panelColor.BackColor = color;

}

به محیط طراحی برگردید و سه TrackBar را انتخاب و در قسمت Events مروبط به پنجره Properties رویداد scroll را یافته و متد trackBar\_Scroll را انتخاب کنید.

در داخل کنترل کننده رویداد ، مقادیر خاصیت Value از هرکدام از Trackbar ها را در سه متغیر ذخیره کرده ایم.مقادیر این سه متغیر نشان دهنده رنگ های قرمز ، سبز ، آبی (RGB values ) می باشد.

سپس یک شی از کلاس System.Drawing.Color را ساخته واز مقادیر TrackBar ها به عنوان پارامتر های متد FromArgb() استفاده می کنیم. سپس رنگ Panel را به رنگی که بر اساس مقادیر TrackBar ها ایجاد شده است تغییر می دهیم.برنامه را اجرا و دستگیره های TrackBar ها را جابه جا و نتیجه را مشاهده کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/09/trackbar-03.jpg)

## کنترل Timer

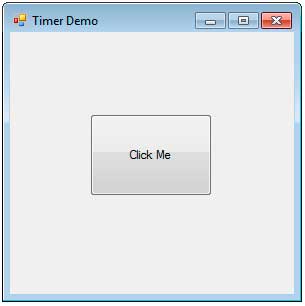
کنترل Timer (System.Windows.Forms.Timer) دستور یا دستورات خاصی را در بازه های زمانی خاصی که در خصوصیت interval مشخص می شود اجرا می کند. به عنوان مثال اگه بخواهید نامتان را هر ۵ ثانیه یکبار چاپ کنید می توانید از این کنترل استفاده نمایید. کنترل Timer از کنترل های غیر بصری است و در قسمت Component Tray قرار می گیرد. این کنترل خصوصیات زیادی ندارد.

دو خاصیت مهم آن عبارتند از Enabled و Interval .

وقتی که خصوصیت Enabled را برابر true قرار دهید کنترل شروع به کار و اگر برابر false قرار دهید متوقف می شود. خصوصیت interval مقدار زمانی که باید بگذرد تا دستور یا دستورات داخل رویداد tick فراخوانی شود را مشخص می کند.رویداد  tick رویداد پیشفرض کنترل تایمر است.

به این نکته توجه کنید که مقدار این خصوصیت(interval)  بر حسب میلی ثانیه مشخص می شود. به عنوان مثال قرار دادن مقدار ۵۰۰۰ در جلوی این خصوصیت به معنای این است که رویداد tick هر ۵ ثاینه یکبار فراخوانی شود. به یک نمونه برنامه که در آن از کنترل timer استفاده شده است توجه نمایید. برنامه شامل یک دکمه است که در جهت بالا و پایین فرم حرکت می کند.

یک پروژه Windows Application با نامTimerDemo ایجاد کنید. یک دکمه را در وسط فرم  مانند شکل زیر قرار دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/timer-control-01.jpg)

یک کنترل Timer را به روی فرم بکشید. این کنترل در قسمت Component  جعبه ابزار قرار دارد. همانطور که قبلا گفته شد این کنترل در قسمت Component Tray قرار می گیرد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/timer-control-02.jpg)

کنترل timer را انتخاب و خاصیت interval آن را برابر ۵۰ و خاصیت Enabled آنرا برابر true قرار دهید.این کار باعث می شود زمانی که برنامه اجرا شود کنترل timer شروع به کار کند. روی کنترل timer دابل کلیک کنید تا کنترل کننده رویداد پیشفرض آن که همان رویداد tick است ایجاد شود.به یاد داشته باشید دستورات داخل این کنترل کننده رویداد هر ۵۰ میلی ثانیه یکبار فراخوانی می شوند.یک نمونه متغیر به شکل زیر در کلاس فرم تعریف کنید.این متغیر برای نگهداری تعداد پیکسل هایی است که در هر بار فراخوانی رویداد tick پیمایش می شوند.

private int velocity = 5;

کد های زیر را به کنترل کننده رویداد tick اضافه کنید.

private void timer1\_Tick(object sender, EventArgs e)

{

if (button1.Top <= 0 || button1.Bottom >this.ClientSize.Height)

velocity = -velocity;

button1.Top += velocity;

}

وقتی که رویداد tick فراخوانی می شود دستورات بدنه آن اجرا می شوند. در دستور شرط ابتدا بررسی شده است که دکمه به پایین فرم رسیده است یا به بالای فرم.

برای اینکه مشخص کنیم که دکمه به بالای فرم چسبیده است از خاصیت Button.Top استفاده کرده ایم.این خصوصیت مقدار مختصات y گوشه ی بالا و چپ دکمه را مشخص می کند.

اگر مقدار این خصوصیت برابر یا کوچکتراز ۰ باشد به این معنی است که دکمه به بالای فرم رسیده است. به این نکته توجه کنید که مبدا مختصات در فرم گوشه بالای و سمت چپ فرم است.

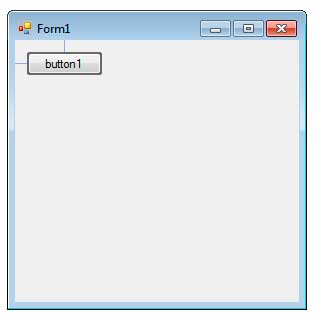
قسمت دوم شرط بررسی می کند که آیا دکمه به قسمت پایین فرم رسیده است یا نه? برای این کار از خاصیت Button.Bottom که مختصات y دکمه را بر می گرداند استفاده کرده ایم.سپس بررسی کرده ایم که این مقدار را با مقدار خصوصیت Height(ارتفاع) فرم مقایسه کرده ایم.برای بدست آوردن ارتفاع فرم از خصوصیت Height که خود در خاصیت Form.ClientSize قرار دارد استفاده کرده ایم.

اگر هر کدام از شرط ها درست باشند به راحتی با قرینه کردن مقدار (ضرب -۱ در مقدار فعلی)velocity جهت حرکت دکمه را تغییر می دهیم.اگر دکمه به سمت بالا فرم حرکت کند (به این خاطر که ۵ پیکسل به سمت بالا حرکت میکند) و به بالای فرم برسد جهت حرکت آن تغییر می کند.

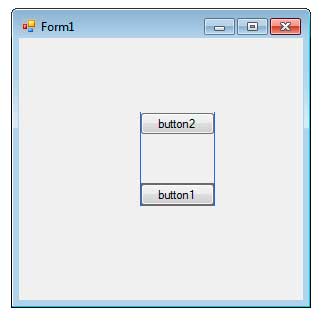
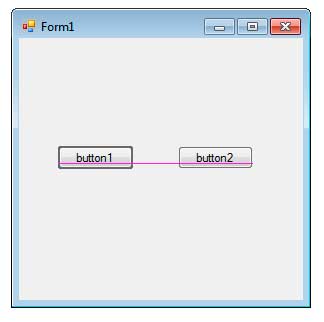
برنامه را اجرا کنید و نتیجه را مشاهده کنید.

## طراحی فرم های ویندوزی

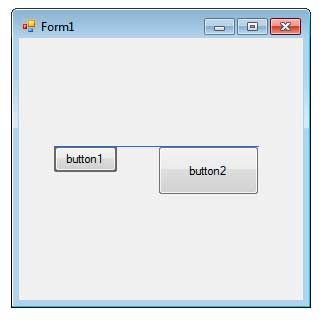
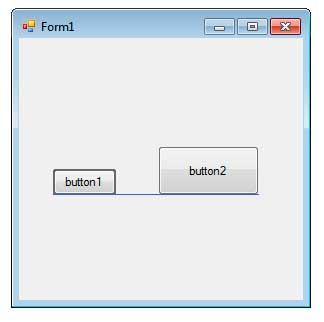
هنگام طراحی یک رابط گرافیکی، باید به فکر مکان، اندازه، تراز بندی و  رنگ کنترل ها یا اجزا گرافیکی باشید. خوشبختانه، ویژوال استودیو دارای ابزارهایی است که به شما در طراحی رابط گرافیکی کمک می کنند. برای توضیح این ابزارها یک پنجره ویندوزی ایجاد کنید. هنگامی که یک کنترل را بر روی فرم می کشید یا قرار می دهید متوجه ظاهر شدن خطوطی می شوید. این خطوط به شما اجازه می دهند که موقعیت کنترل را نسبت به کنترل های دیگر یا خود فرم تراز بندی کنید.

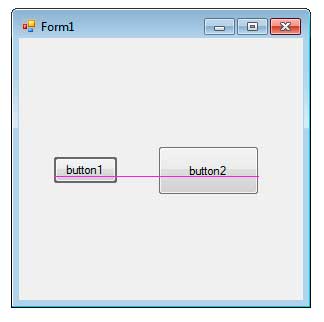
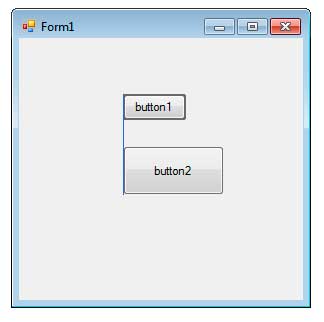
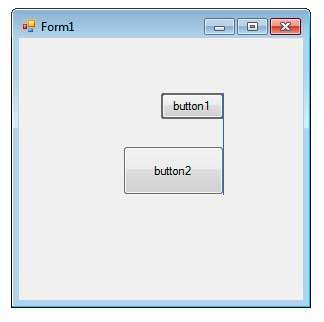
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-01.jpg)

می توان یک کنترل را در گوشه های مختلف فرم قرار داد.اگر کنترل های دیگری نیز در فرم قرار داشته باشند، کنترل را که شما بر روی فرم می کشید می تواند به طور خودکار خودش را نسبت به کنترل های دیگر تراز کند.

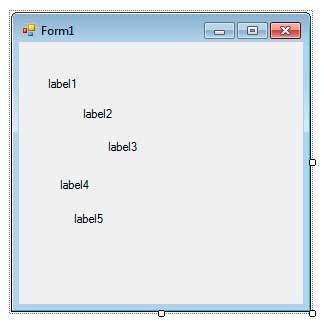
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-02.jpg) [](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-03.jpg)

می توان یک کنترل را نسبت به قسمت بالا، وسط، پایین، چپ یا راست یک کنترل بزرگتر تراز کرد.

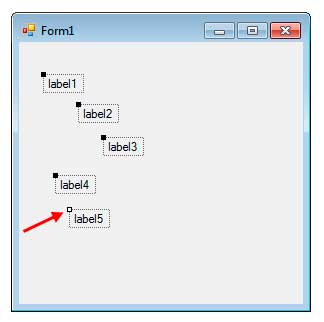
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-04.jpg)[](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-05.jpg)

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-06.jpg)[](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-07.jpg)[](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-08.jpg)

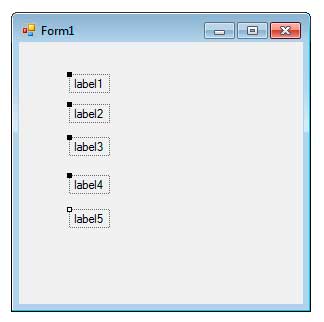
ویژوال استودیو همچنین دارای دستوراتی برای تراز بندی خودکار و یکنواخت کردن فضاها و اندازه کنترل ها دارد. به فرم زیر که دارای چندین کنترل است توجه کنید :

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-09.jpg)

می توانید هر کنترلی را که می خواهید به وسیله پایین نگه داشتن دکمه shift و کلیک بر روی کنترل تراز بندی کنید. همچنین می توان کنترل ها را با کشیدن ماوس به دور آنها انتخاب کرد.

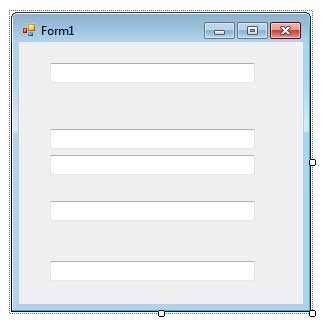
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-10.jpg)

وقتی که کنترل ها در حالت انتخاب هستند، کنترلی که در گوشه بالا و سمت چپ خود دارای یک مربع کوچک سفید است، کنترلی است که بقیه کنترل ها نسبت به آن تراز بندی می شوند. برای تراز بندی کنترل ها به مسیر Format>Align و سپس گوشه ای را که می خواهید تراز بندی بر راساس آن انجام شود انتخاب کنید. برای روشن شدن مطلب، سمت چپ را انتخاب کنید. کنترل های دیگر همگی بر اساس گوشه سمت چپ کنترل اصلی تراز می شوند.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-11.jpg)

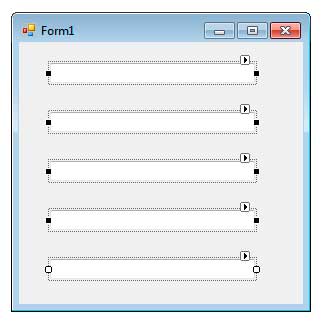
می توان یک کنترل را به صورت افقی یا عمودی در وسط یک فرم قرار داد. فقط کافیست به مسیر Format>Center Form و جهتی را که می خواهید کنترل در وسط داشته باشد انتخاب کنید.

از دستور دیگری برای مساوی کردن فضاها و فاصله های بین کنترل ها می توان استفاده کرد. به فرم زیر توجه کنید :

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-12.jpg)

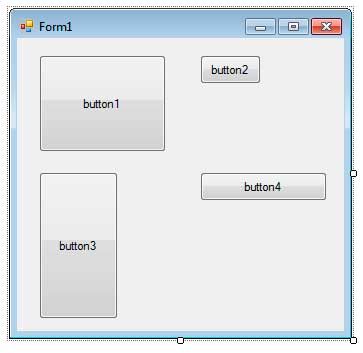
فرم بالا شامل کنترل هایی است که فاصله بین انها با هم برابر نیست. برای مساوی کردن این فاصله ها، همه کنترل ها را انتخاب کنید و از منوی Format  رفته و گزینه Vertical Spacing را برای تنظیم فاصله های پایین و بالای کنترل ها و یا Vertical Spacing را برای تنظیم فاصله های سمت چپ و راست کنترل ها انتخاب کنید. برای فرم مثالا بالا به مسیر

Format>Vertical Spacing رفته و بر روی  گزینه Make Equal کلیک کنید.

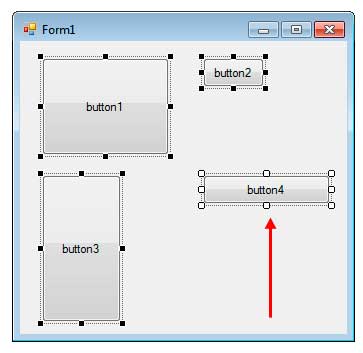
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-13.jpg)

مشاهده می کنید که فضای عمودی بین کنترل ها تراز می شود.

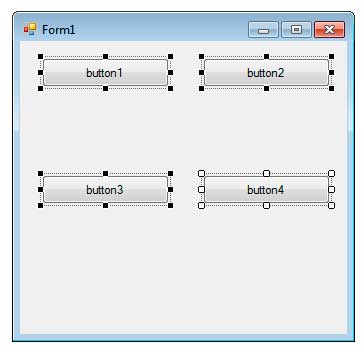
می توانید چندین کنترل را انتخاب و اندازه آنها را با هم برابر کنید. به فرم زیر توجه کنید :

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-14.jpg)

کنترلی که دارای یک دستگیره سفید است، کنترل مرجع می باشد و اندازه دیگر کنترل ها بر اساس اندازه آن تغییر می کند.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-15.jpg)

به مسیر Format>Make Same Size و سپس گزینه های widths، height یا both را انتخاب کرده تا اندازه کنترل ها از نظر عرض ، ارتفاع و یا هر دو یکسان شود.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/designing-forms-16.jpg)

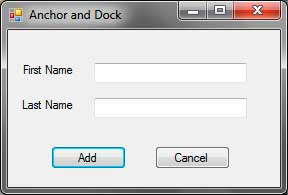
## خاصیت Anchor

یکی از مشکلات قرار دادن کنترل ها بر روی کنترل فرم این است که مکان آنها هنگام تغییر سایز فرم تغییر نمی کند و در جای قبلی می مانند.

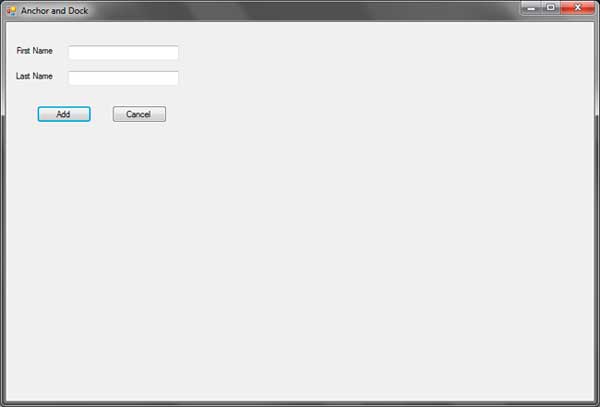
به مثال زیر توجه کنید :

یک فرم ساده همراه با چند کنترل در داخل آن ایجاد می کنیم.

وقتی برنامه را اجرا می کنیم، هیچ چیز اشتباهی به چشم نمی خورد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/anchor-dock-01.jpg)

اما وقتی فرم را به وسیله کشیدن گوشه های آن و یا با زدن دکمه بیشینه، بزرگ می کنیم، مشکل معلوم می شود.

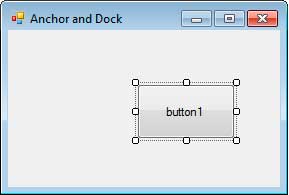
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/anchor-dock-02.jpg)

به عکس بالا که اندازه آن به اندازه صفحه دسکتاپ بزرگ کرده ایم توجه کنید. همانطور که مشاهده می کنید مکان کنترل ها با افزایش اندازه فرم تغییر نمی کند. برای برطرف کردن این مشکل می توان از خاصیت Anchor که در بیشتر کنترل ها در دسترس است، استفاده نمود. این خاصیت رفتار کنترل ها را در هنگام تغییر سایز فرم تعیین می کند. می توانیم مشخص کنیم که کنترل به کدام گوشه فرم بچسبد.همچنین تعیین کنیم که کنترل ها چگونه تغییر سایز دهند.

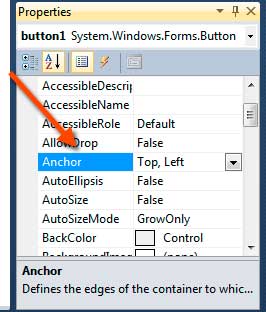
اجازه بدهید که خاصیت Anchor را به کنترل هایمان اختصاص دهیم تا رفتار طبیعی تری هنگام تغییر سایز فرم از خود نشان دهند. خاصیت Anchor یک مقدار از نوع شمارشی System.Windows.Forms.AnchorStyles قبول می کند.

| AnchorStyle | توضیح |
| --- | --- |
| Bottom | کنترل به قسمت پایینی مکان یا کنترلی که در آن قراردارد، می چسبد. |
| Left | کنترل به قسمت چپ مکان یا کنترلی که در آن قراردارد، می چسبد. |
| Right | کنترل به قسمت راست مکان یا کنترلی که در آن قراردارد، می چسبد.. |
| Top | کنترل به بالای مکان یا کنترلی که در آن قراردارد، می چسبد. |
| None | کنترل به هیچ قسمتی از مکان یا کنترلی که در آن قراردارد، نمی چسبد. |

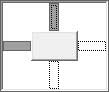
 این خاصیت در همه کنترل ها در حالت پیشفرض ترکیبی از مقادیر Top و Left است. وقتی مقادیر  AnchorStyles را به خاصیت Anchor تخصیص می دهیم، فاصله کنترل از گوشه خاصی از فرم با وجود تغییر اندازه فرم حفظ می شود. به عنوان مثال اجازه بدهید که کنترل دکمه زیر را به سمت راست فرم بچسبانیم.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/anchor-dock-05.jpg)

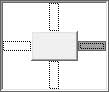
ویژوال استودیو و ویژوال  سی شارپ دارای ابزاری برای تعیین AnchorStyles می باشند. کنترلی را که می خواهید به یکی از گوشه ها بچسبانید، انتخاب کنید. به پنجره Properties رفته و خاصیت Anchor را پیدا کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/anchor-dock-03.jpg)

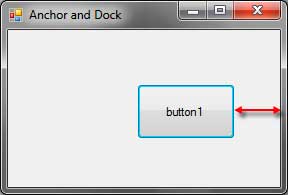
بر روی دکمه ای که به شکل یک فلش رو به پایین است (شکل بالا) کلیک کنید تا شکل زیر ظاهر شود:

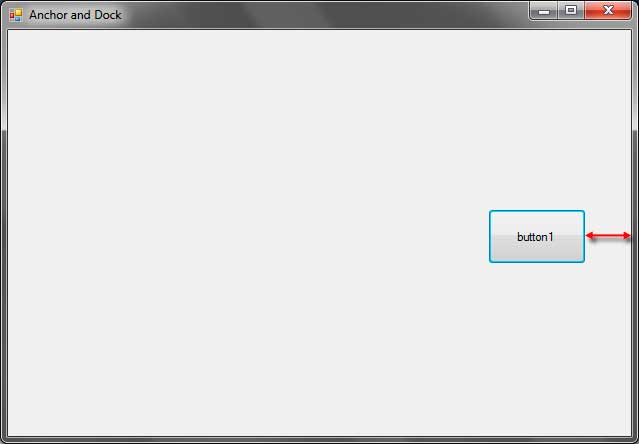
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/anchor-dock-04.jpg)

مربعی که در وسط شکل زیر قرار دارد نماینده کنترل است. مربعی که به رنگ خاکستری است گوشه ای را نشان می دهد که قرار است کنترل به آن بچسبد. شما به راحتی و با کلیک بر روی این مربع ها می توانید مقادیر  AnchorStyle مربوط به کنترل را اضافه یا حذف کنید. حال برای چسباندن دکمه مثال بالا به سمت چپ فرم بر روی مربع سمت راست کلیک می کنیم تا رنگ ان مانند شکل زیر خاکستری شود.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/anchor-dock-06.jpg)

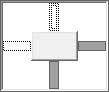
برنامه را اجرا کرده و اندازه فرم را تغییر می دهیم. اندازه فاصله دکمه از  گوشه راست فرم ، را در حالت عادی و در حالتی که فرم را تغییر اندازه می دهیم مقایسه می کنیم.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/anchor-dock-07.jpg)

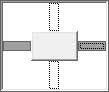
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/anchor-dock-08.jpg)

همانطور که مشاهده می کنید، این فاصله قبل و بعد از تغییر اندازه فرم یکسان است.

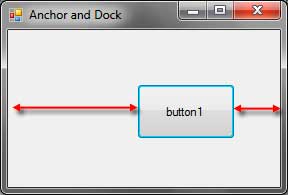
 حال اجازه بدهید که مقدار  Bottom را هم اضافه کنیم.

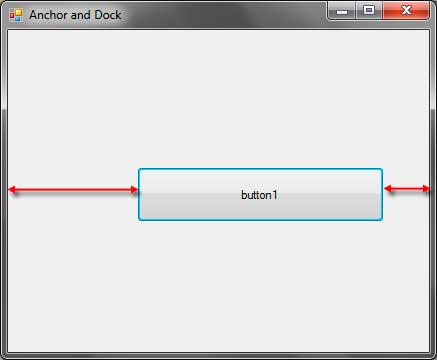
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/anchor-dock-09.jpg)

با ترکیب دو مقدار Right و Bottom، کنترل همیشه در فاصله یکسانی از گوشه سمت راست و چپ کنترل فرم قرار می گیرد. وقت آن رسیده که مقادیری را که مخالف همدیگر هستند را اضافه کنیم و مشاهده کنیم که چه اتفاقی برای کنترل می افتد. به عنوان مثال مقدار Left مخالف مقدار Right است. انتضار ما این است که کنترل با فاصله یکسانی از سمت چپ و راست فرم قرار بگیرد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/anchor-dock-10.jpg)

برنامه را اجرا کرده و فرم را تغییر اندازه دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/anchor-dock-11.jpg)

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/anchor-dock-12.jpg)

 اندازه دکمه در دو جهت مخالف هم تغییر می کند. اگر خاصیت Anchor کنترل را در چهار جهت مشخص کنیم اندازه دکمه به صورت افقی و عمودی در همه جهات تغییر می کند. اما وقتی مقدار خاصیت Anchor را برابر AnchorStyles.None قرار دهیم کنترل رفتار متفاوتی از خود نشان می دهد. در این حالت وقتی کنترل فرم را تغییر اندازه دهیم، کنترل به اندازه نصف تغیر اندازه فرم و در جهت تغییر اندازه آن حرکت می کند.

به عنوان مثال  اگر فرم را به اندازه ۱۰۰ پیکسل و در جهت راست تغییر اندازه دهیم، کنترل به اندازه ۵۰ پیکسل به سمت راست حرکت می کند و اگر فرم را به اندازه ۱۰۰ پیکسل در جهت راست و ۵۰ پیکسل در جهت پایین تغییر اندازه دهیم، کنترل ۵۰ پیکسل در جهت راست و ۲۵ پیکسل در جهت پایین حرکت می کند.

## 

## خاصیت Dock

خاصیت Dock به شما اجازه می دهد که کنترل مورد نظرتان را به هر گوشه ای از فرم یا کنترل در بر گیرنده آن بچسبانید. با استفاده از این روش کنترل ها در هنگام تغییر اندازه فرم دست نخورده باقی می مانند (یعنی ظاهر کلی فرم به هم نمی خورد و کنترل ها با تغییر اندازه فرم جا به جا می شوند). ااین خاصیت یک مقدار شمارشی از System.Windows.Forms.DockStyle قبول می کند.

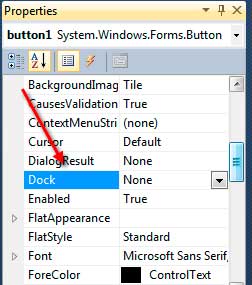
در جدول زیر مقادیر موجود این نوع  شمارشی ذکر شده اند :

| مقدار | توضیح |
| --- | --- |
| Bottom | کنترل به قسمت پایین کنترلی که در آن قرار دارد می چسبد. |
| Fill | کنترل تمام کنترلی را که در آن قرار دارد را پر می کند. |
| Left | کنترل به قسمت چپ کنترلی که در آن قرار دارد می چسبد. |
| Right | کنترل به قسمت راست کنترلی که در آن قرار دارد می چسبد. |
| Top | کنترل به قسمت بالای  کنترلی که در آن قرار دارد می چسبد. |

حال اجازه دهید که یک کنترل button را به گوشه های مختلف فرم بچسبانیم.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/docking-controls-01.jpg)

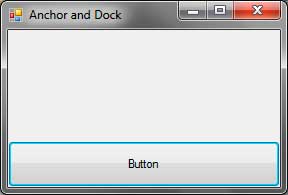
برای این کار، کنترل را انتخاب کرده و به پنجره Properties می رویم.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/docking-controls-02.jpg)

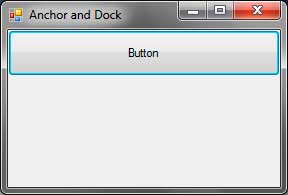
خاصیت Dock را یافته و بر روی دکمه بازشوند آن کلیک می کنیم. شکلی به صورت زیر نمایش داده می شود که شما با استفاده از آن تعیین می کنید که کنترل موردنظر به کدام قسمت فرم بچسبد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/docking-controls-03.jpg)

مربع مرکزی بدین معناست که شما می خواهید از مقدار DockStyle.Fill استفاده کنید. اشکال زیر تاثیر نحوه چسباندن کنترل به گوشه های مختلف را نشان می دهد :

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/docking-controls-04.jpg)

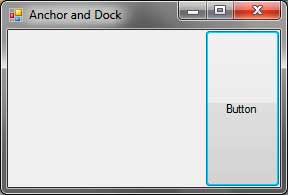
Bottom

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/docking-controls-05.jpg)

Top

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/docking-controls-06.jpg)

Left

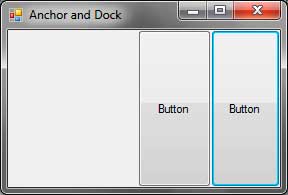
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/docking-controls-07.jpg)

Right

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/docking-controls-08.jpg)

Fill

اگر بخواهید از یک مقدار شمارشی یکسان برای چندین کنترل استفاده کنید، کنترل های مورد نظر یا در کنار هم و یا بر روی یکدیگر قرار می گیرند.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/docking-controls-09.jpg)

## 

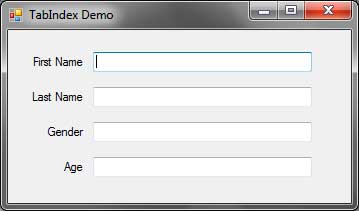
## خاصیت TabIndex

شما با استفاده از کلید TAB می توانید در بین کنترل های مختلف فرم حرکت کنید.

هر کنترل بصری در روی فرم دارای یک خصوصیت به نام TabIndex است که ترتیب فعال شدن آنها را در هنگام فشردن کلید tab مشخص می کند.

بهتر است بعد از طراحی فرم ، خصوصیت TabIndex تمامی کنترل های بصری فرم را بررسی کنید تا از ترتیب فعال شدن آنها در هنگام فشار دادن کلید Tab مطمئن شوید.

یک فرم با چهار جعبه متن در نظر بگیرید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/tab-index-01.jpg)

فرض کنید خاصیت TabIndex انها نا مرتب است و مقادیر آنها به صورت زیر است :

| TextBox | TabIndex |
| --- | --- |
| txtFirstName | 1 |
| txtLastName | 4 |
| txtGender | 3 |
| txtAge | 2 |

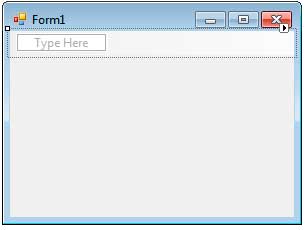
هنگامی که برنامه را اجرا می کنید ، کنترلی که کمترین مقدار را در این خصوصیت دارد که همان کنترل txtFirstName است ابتدا فعال می شود. اگر شما کلید Tab را فشار دهید دومین کنترل با نام txtAge (چهارمین کنترل در شکل بالا ) فعال می شود. یک کاربر با تجربه از خصوصیت TabIndex به درستی استفاده می کند. اگر از این خصوصیت به درستی استفاده نشود کاربر نمیتواند به درستی تشخیص دهد که کنترل بعدی که فعال می شود کدام است. تعیین ترتیب فعال شدن کنترل ها به مکان قرار گیری آنها بر روی فرم بستگی دارد.

از آنجایی که TextBox  ها در فرم  بالا به صورت عمودی قرار گرفته اند بهتر است که ترتیب فعال شدن آنها از بالا به پایین باشد یعنی خاصیت TabIndex اولی را برابر ۱، دومی را برابر ۲ و… قرار دهیم.  تنظیم کردن این خصوصیت در کنترل هایی که لازم نیست فوکوس را بگیرند الزامی نیست مانند کنترل Label .

## 

## اضافه کردن منو به فرم

نوار منو در اکثر برنامه به چشم می خورد. این نوار شامل دستورات مختلفی است که کاربر از آنها استفاده می کند و در ویژوال استودیو  با استفاده از کنترل MenuStrip می توان آن را به برنامه اضافه کرد. در شکل زیر یک کنترل MenuStrip را از جعبه ابزار بر روی فرم کشیده ایم :

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/adding-menu-bar-01.jpg)

MenuStrip همانند کنترل Timer جزو کنترلهای غیر بصری بوده و در قسمت components tray قرار می گیرد.

در زیر برخی از خواص پر کاربرد MenuStrip ذکر شده است :

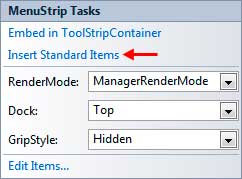
| **خاصیت** | **توضیح** |
| --- | --- |
| Dock | مشخص می کند که کنترل منو به کدام گوشه فرم بچسبد.مقدار پیشفرض آن top است |
| GripStyle | به شما اجازه می دهد که مکان منو را تغییر دهید. |
| Items | کلکسیونی از منوی های اصلی را در بر می گیرد. |
| Stretch | تعیین می کند که کنترل منو در سرتاسر در برگیرنده خود کشیده شود. |

**اضافه کردن منو های استاندارد**

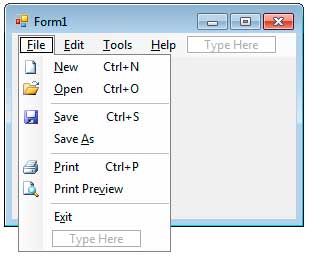
ویژوال استودیو برای اضافه کردن آیتم های استاندارد به MenuStrip روشی را پیشنهاد می دهد. برای انجام این کار، مانند شکل زیر بر روی فلش کلیک کرده تا صفحه ای ظاهر شود :

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/adding-menu-bar-02.jpg)

از صفحه باز شده بروی گزینه Insert Standard Items کلیک کنید :

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/adding-menu-bar-03.jpg)

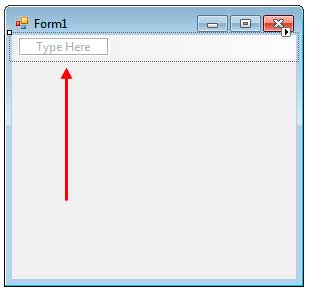
با انجام این کار ویژوال استودیو به صورت خودکار گزینه ها و آیکونها مربوط به آنها را مانند شکل زیر به منو اضافه می کند :

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/adding-menu-bar-04.jpg)

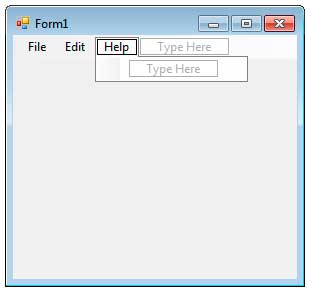
**اضافه کردن منو های دلخواه**

ممکن است در بعضی از شرایط  به منوهای خاص نیاز نداشته باشید . شما می توانید با استفاده از ابزارهای ویژوال استودیو منو ی دلخواهی را ایجاد کنید.

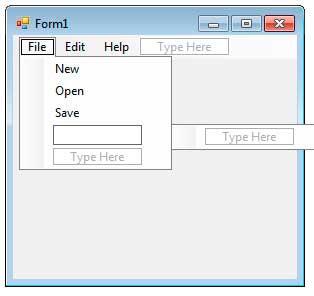
ابتدا یک کنترل MenuStrip را به فرم اضافه کرده و با کلیلک بر روی قسمتی که با فلش نشان داده شده است یک منو اضافه و عنوان آنرا تایپ کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/adding-menu-bar-05.jpg)

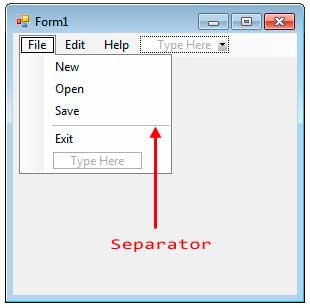
به این نکته توجه نمایید که در هنگام تایپ عنوان منو جعبه دیگری در کنار آن ظاهر می شود و شما می توانید منو های دیگری را اضافه نمایید به عنوان مثال منوهایی مانند File ، Edit ، Help در کنترل MenuStrip ایجاد کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/adding-menu-bar-06.jpg)

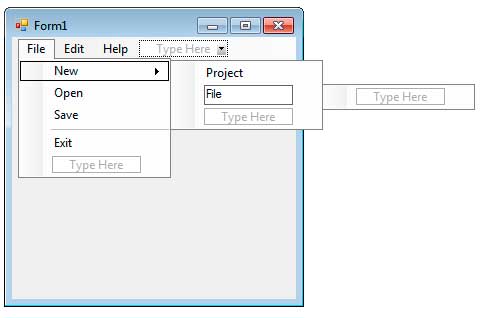
برای اضافه کردن زیر منو بر روی یکی از منو ها کلیک کرده تا جعبه ای برای اضافه کردن زیرمنو به آن ظاهر شود .

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/adding-menu-bar-07.jpg)

برای ایجاد یک جدا کننده یک کاراکتر خط تیره ( – ) را تایپ کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/adding-menu-bar-08.jpg)

همچنین شما می توانید برای هر زیر منو مانند شکل زیر ، زیرمنوهای دیگری را تعریف نمایید

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/adding-menu-bar-09.jpg)

هر کدام از منوهایی که اضافه می کنید از نوع ToolStripMenuItem هستند. همچنین نوع جدا کننده ToolStripSeparator هست از ToolStripSeparator فقط برای تقسیم بندی منو های مرتبط با هم استفاده می شود.

هر ToolStripMenuItem به خاصیت  Items کنترل MenuStrip و هر زیر منو به خاصیت DropDownItems از ToolStripMenuItem اضافه می شود.

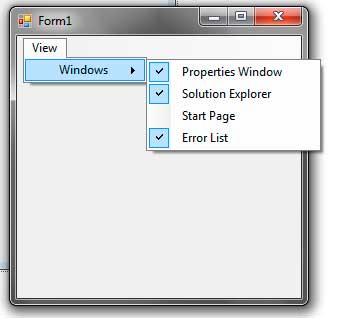
ویژوال استودیو به طور خودکار بر اساس متنی که برای هر یک از ToolStripMenuItem انتخاب کرده اید نامی به آنها اختصاص می کند.

در زیر برخی از خواص ToolStripMenuItem را مشاهده می کنید.

| **خاصیت** | **توضیح** |
| --- | --- |
| Checked | تعیین می کند که آیا آیتم تیک خورده است یا نه |
| CheckOnClick | تعیین می کند که ایتم قابلیت تیک خوردن داشته باشد یا نه |
| CheckState | تعیین می کند که به طور پیشفرض آیتم تیک داشته باشد یا نه |
| DropDownItems | زیر منوهای آیتم در این خاصیت قرار می گیرند |
| Enabled | آیتم را فعال یا غیر فعال می کند. |
| Image | عکسی را به آیتم اختصاص می دهد. |
| ShortcutKeys | کلید میانبر آیتم را تعیین می کند. |
| ShowShortcutKeys | تعیین می کند که کلید میانبر در کنار آیتم نشان داده شود یا نه |
| Text | عنوان آیتم را تعیین می کند |
| ToolTipText | وقتی که موس را بر روی آیتم قرار می دهید ، متنی که برای این خاصیت تعیین کرده اید نمایش داده می شود. |

**منوهای تیک دار و غیر تیک دار**

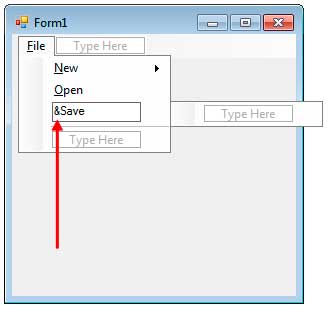
خاصیت CheckOnClick به هر منو این قابلیت را می دهد که قابل تیک خوردن باشد یا نه. به عنوان مثال ، شما می توانید منویی به شکل زیر ایجاد کنید. برای اینکار باید خاصیت CheckOnClick زیر منوهای آن را برابر true قرار دهید. با کلیک بر روی هر آیتم یک علامت تیک در سمت چپ آن نمایش داده می شود.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/adding-menu-bar-10.jpg)

با استفاده از خاصیت های Checked و CheckState می توانید تعیین کنید که یک آیتم تیک داشته باشد یا خیر.

**اضافه کرده کلید های میانبر به منو**

ساده ترین را برای انجام این کار اضافه کرده کاراکتر & در ابتدای نام آیتم است.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/adding-menu-bar-11.jpg)

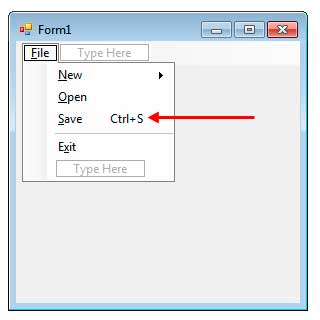
حرف بعد از کاراکتر & به عنوان کلید میانبر در نظر گرفته می شود.  به عنوان مثال در منوی , &Save حرف S و در E&xit حرف x کلید میانبر است. در قسمت طراحی کلید های میانبر به صورت زیر خط دار نمایش داده  می شوند در زمان اجرا علامت زیر خط با فشار دادن کلید Alt نمایان می شود. براس استفاده از این نوع کلید های میانبر باید کلید Alt را ابتدا فشار دهید.

برای فعال کردن یک منو شما باید کلید های ترکیبی (کلید میانبر + Alt ) را فشار دهید.به عنوان مثال برای استفاده از منوی File از کلید های ترکیبی Alt + F استفاده کنید. برای ایجاد کلید های میانبر پیچیده تر باید از خاصیت ShortcutKeys استفاده نمایید. یک آیتم را انتخاب و در پنجره Properties خصوصیت ShortcutKeys را پیدا کنید.

بر روی فلش کوچکی که در کنار این خصوصیت قرار دارد کلیک کرده تا پنجره ای به شکل زیر نمایان شود.

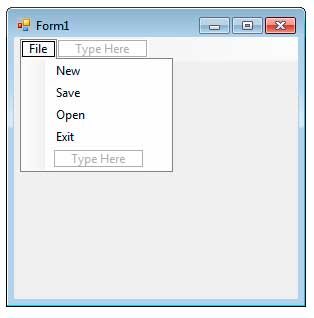
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/adding-menu-bar-12.jpg)

هنگامی که یک کلید ترکیبی را انتخاب می کنید ، عنوان آن در صورتی در کنار منو ( در سمت راست) نمایش داده می شود که خاصیت ShowShortcutKeys را برابر true قرار دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/adding-menu-bar-13.jpg)

**اضافه کردن آیکن به منو ها**

شما می توانید برای هر منو آیتم یک تصویر کوچک که در سمت چپ آن نمایش داده می شود را انتخاب کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/adding-menu-bar-14.jpg)

برای اینکار از خاصیت Image استفاده می شود.  بر روی دکمه کوچک کنار خاصیت Image در پنجره Properties کلیک کنید. منبعی که عکس در آن قرار دارد را انتخاب و سپس مسیر آنرا مشخص نمایید. اگر سایز عکس بیش از اندازه باشد ، به صورت خودکار بر اساس مقدار خاصیت ImageScaling اندازه آن تغییر می کند.

**اضافه کردن عمکلرد به منو ها**

هنگامی که بر روی یک منو آیتم کلیک می کنید رویداد Click آن اتفاق می افتد. برای اضافه کردن قابلیتی مشخص به منو ایتم باید کد های خود را در این رویداد قرار دهیم.

برای اضافه کردن کنترل کننده رویداد کلیک به هر منو ایتم ، در محیط طراحی بر روی منو آیتم مورد نظر دوبار کلیک کرده تا ویژوال استودیو به صورت خودکار آن را ایجاد کند.

برای منو آیتم هایی که مقدار خاصیت CheckOnClick آن برابر true است ، می توانید از رویداد های CheckedChange و CheckStateChange آنها را استفاده کنید.مانند شکل زیر یک فرم دیگر ساخته و یک کنترل منو به آن اضافه کنید.

بر روی آیتم Exit دوبار کلیک کنید تا ویژوال استودیو کنترل کننده رویداد پیشفرض که همان کلیک است را تولید کند.سپس کد های زیر را در آن قرار دهید.

private void exitToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

متد استاتیک Exit از کلاس Application به سادگی برنامه را می بندد.

## ساخت نوار ابزار

نوار ابزار شامل دکمه ها و اجزای مفیدی برای راحتی کار کاربر می باشد. مانند منو، نوار ابزار هم در تعداد زیادی از نرم افزارهای مشهور مانند Microsoft Office 2003 به چشم می خورد. در زیر نوار ابزار Microsoft Office 2003 را مشاهده می کنید :

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1001.jpg)

**کنترل ToolStrip**

دکمه ها و اجزای یک نوار ابزار در داخل کنترل ToolStrip قرار می گیرند. برای جا به جا کردن کنترل ToolStrip از نقطه چین های عمودی که در سمت چپ آن قرار دارند استفاده می شود.

در زیر برخی از خواص مهم کنترل ToolStrip توضیح داده شده اند :

| **خصوصیت** | **توضیح** |
| --- | --- |
| CanOverflow | آیتم هایی که داخل کنترل toolstrip فضای کافی برای نمایش ندارند در یک قسمت مخفی قرار می گیرند که این فضا یا یک دکمه در دسترس قرار می گیرد. |
| GripStyle | تعیین می کند دستگیره ی جابه جایی toolstrip نمایش داده شود یا خیر |
| Items | آیتم هایی که در کنترل toolstrip قرار می دهید در داخل این خصوصیت قرار می گیرند |
| LayoutStyle | نحوه ی نمایش آیتم ها را تعیین می کند |
| ShowItemToolTips | تعیین می کند tool tip برای آیتم نمایش داده شود یا نه |
| Stretch | جهت نمایش ToolStripPanel را تعیین می کند |

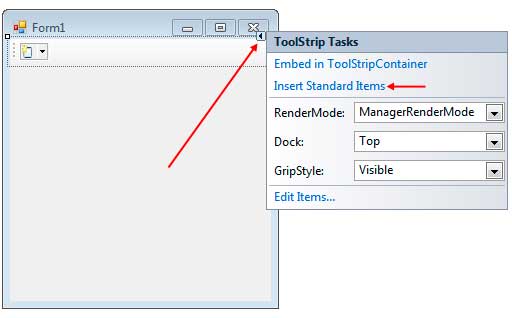
**اضافه کردن آیتم های استاندارد به نوار ابزار**

قبل از ایجاد یک نوار ابزار دلخواه، اجازه دهید ببینیم که ویژوال استودیو چگونه به طور خودکار  آیتم های استاندارد که شامل دکمه هایی مانند Save، Open، Copy و Paste می باشند  را ایجاد می کند. یک برنامه ویندوزی جدید ایجاد کرده و نام آنرا ToolBarDemo بگذارید.

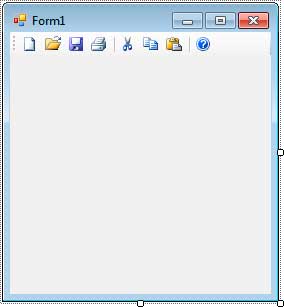
از قسمت Menus&Toolbars نوار ابزار  به صورت زیر یک کنترل ToolStrip را بر روی فرم قرار دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1002.jpg)

بر روی فلش کوچک واقع در سمت راست کنترل ToolStrip کلیک کرده تا پنجره ای به شکل زیر نمایش داده شود:

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-10031.jpg)

در پنجره باز شده بر روی گزینه Insert Standard Items کلیک کرده تا ویژوال استودیو به طور خودکار دکمه های استاندارد را به صورت زیر ایجاد کند :

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1004.jpg)

**خاصیت Items**

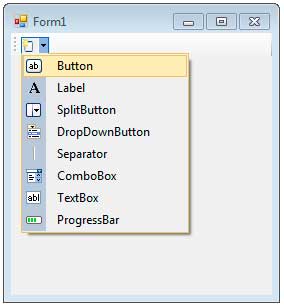
کنترل های که داخل ToolStrip اضافه می کنید در داخل این خصوصیت قرار می گیرند. انواع مختلفی از آیتم ها را می توان در داخل کنترل ToolStrip قرار داد و همگی آنها از کلاس ToolStripItem ارث بری می کنند.

| **نوع آیتم** | **توضیح** |
| --- | --- |
| ToolStripButton | یک آیتم که نمایش دهنده یک دکمه هست . با کلیک بر روی این آیتم می توان عملیات مختلفی را اجرا کرد. |
| ToolStripLabel | یک آیتم که نمایش دهنده یک متن غیر قابل تغییر است.از این آیتم می توان برای نمایش عکس هم استفاده کرد |
| ToolStripSplitButton | یک آیتم که نمایش دهنده دکمه و فلش رو به پایین در کنار آن است.با کلیک بر روی این دکمه یک منوی باز شونده نمایش داده می شود.دکمه خود می تواند عملیات جداگانه ای را انجام دهد. |
| ToolStripDropDownButton | شبیه به آیتم ToolStripSplitButton ، این آیتم نیز یک دکمه با فلش رو به پایین را نمایش می دهد ولی در این آیتم دکمه کار خاصی را انجام نمی دهد و صرفا منو را نمایش می دهد. |
| ToolStripComboBox | این آیتم یک کامبو باکس را نمایش می دهد. |
| ToolStripProgressBar | یک نوار پیشرفت به کنترل ToolStrip اضافه می کند. |
| ToolStripSeparator | یک جدا کننده افقی یا عمودی را نمایش می دهد. |
| ToolStripTextBox | یک کنترل TextBox را نمایش می دهد. |

در قسمت های بعدی نحوه ی استفاده از آیتم ها ی بالا توضیح داده شده است.

**آیتم ToolStripButton**

این آیتم رایج ترین آیتم ی است که شما در یک نوار ابزار می بینید. زمانی که بر روی این آیتم کلیک می کنید رویداد Click آن همچون یک دکمه معمولی اتفاق می افتد. کنترل ToolStrip قبلی را حذف و یک نمونه جدید از آن را روی فرم خود قرار دهید. اگر شما کنترل ToolStrip را انتخاب کنید یک آیکن با فلش کوچک رو پایین در کنار آن،نمایان می شود.با کلیک بر روی این آیکن لیستی از آیتم هایی که می توانید به کنترل اضافه کنید ، نمایش داده می شود.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1005.jpg)

بر روی Button کلیک کنید تا یک کنترل ToolStripButton به ToolStrip اضافه شود.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1006.jpg)

این کنترل از یک آیکن پیشفرض استفاده می کند. برای تغییر دادن این آیکن به خاصیت Image از پنجره Properties رفته و عکس دلخواه خود را انتخاب کنید.

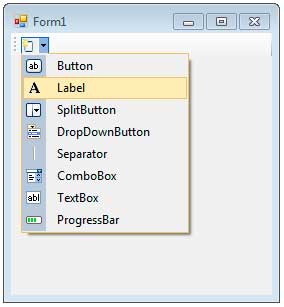
در قسمت طراحی فرم ، بر روی دکمه دوبار کلیک کنید تا کنترل کننده رویداد کلیک آن به طور خودکار توسط ویژوال استودیو ایجاد شود. کد زیر را به کنترل کننده رویداد اضافه کنید :

toolStripProgressBar1.PerformStep();

کد بالا باعث می شود هنگامی که بر روی دکمه کلیک کنید یک پیغام ساده نمایش داده شود. در واقع ، کدهایی که وظیفه خاصی را انجام می دهند مانند تغییر استایل فونت جعبه متن و…  در این رویداد قرار می دهید.

**آیتم ToolStripLabel**

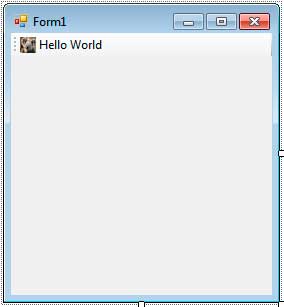
این آیتم به شما اجازه می دهد که یک متن غیر قابل تغییر را به ToolStrip اضافه می کند.  همچنین به شما اجازه می دهد که یک آیکن کوچک را در کنار آن نمایش دهید. برای اضافه کردن یک ToolStripLabel ، بر روی آیکن کوچک کلیک کرده و سپس  Label را انتخاب کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1007.jpg)

هنگامی که این ایتم را اضافه کردید ، می توانید از خاصیت Text آن برای تغییر عنوان آن استفاده کنید.

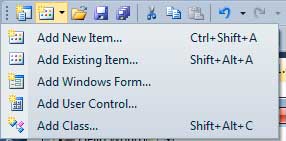
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1008.jpg)

با استفاده از خاصیت Image می توان یک عکس را در کنار آن نمایش داد.

**[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1009.jpg)**

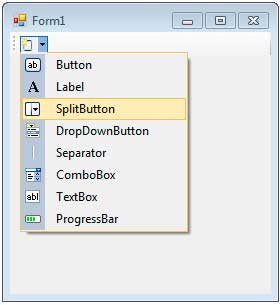
**آیتم ToolStripSplitButton**

این آیتم یک دکمه که قابل کلیک کردن است را نمایش می دهد.  به علاوه اینکه یک فلش رو به پایین در کنار آن قرار دارد که با کلیک بر روی آن یک منوی باز شونده شامل دستوراتی که مرتبط با دکمه هستند نمایان می شود.  یک نمونه از این آیتم ، دکمه Add New Item در نوار ابزار Visual Studio است.

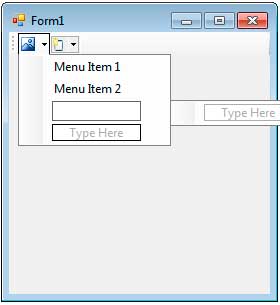
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1010.jpg)

هنگامی که بر روی دکمه اصلی کلیک می کنید پنجره ی Add New Item به جای منو نمایان می  شود. برای اضافه  کردن کنترل ToolStripSplitButton بر روی new item  در ToolStrip کلیک کرده و  SplitButton را انتخاب  کنید.

برای اضافه کردن منو به این کنترل ، بر روی فلش کوچک کنار آن کلیک کنید و لیست منوها را بنویسید.

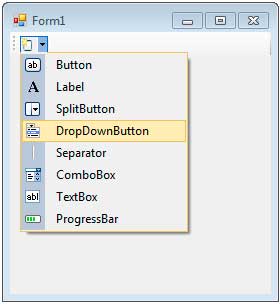
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1011.jpg)

دکمه و منو آیتم های مرتبط با آن می توانند عملیات مختلفی را هنگام کلیک بر روی آنها، انجام دهند. به این نکته توجه فرمایید ، که کنترل کننده رویداد دکمه با کنترل کننده منوهای زیر مجموعه آن متفاوت است.  همانند کنترل  ToolStripButton، می توانید برای این کنترل آیکن کوچکی را با استفاده از خاصیت Image تعیین کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1012.jpg)

**آیتم ToolStripDropDownButton**

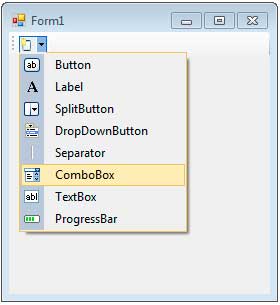
این آیتم بسیار شبیه به آیتم ToolStripSplitButton است ، از این جهت که هنگامی که بر روی آن کلیک می کنید یک منوی بازشونده نمایش داده می شود.  تنها تفاوت بین آنها این است که DropDownButton یک تصویر ساده است ، و هنگامی که بر روی آن کلیک می کنید تنها منو را نمایش می دهد. پس هدف DropDownButton فقط نمایش منو است. برای اضافه کردن این ایتم در لیست بازشونده بر روی DropDownButton کلیک کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1013.jpg)

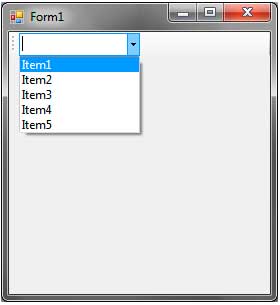
شیوه اضافه کردن منو بب این آیتم شبیه به کنترل ToolStripSplitButton است. همچنین می توانید به روش مشابه با سایر کنترل ها برای این آیتم یک آیکن کوچک انتخاب کنید.

**آیتم  ToolStripComboBox**

این آیتم به شما اجازه می دهد که یک کامبو باکس را به ToolStrip اضافه کنید. این آیتم شبیه به کنترل ComboBox است. برای اضافه کردن بخش هایی به این کنترل از خاصیت Items استفاده کنید.همچنین می توانید از رویداد SelectedIndexChanged استفاده کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1014.jpg)

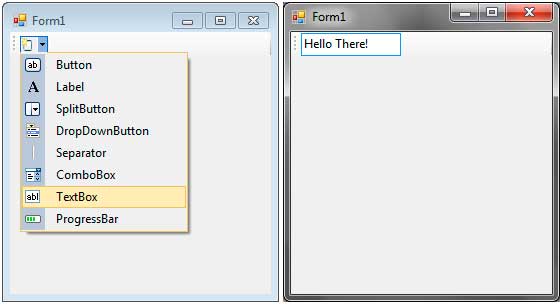
در شکل زیر ،  یک کامبو باکس با تعدادی آیتم داخل آن نمایش داده شده است.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1015.jpg)

رویداد پیش فرض این کنترل کلیک است ، پس برای استفاده از رویداد SelectedIndexChanged به قسمت رویداد ها در پنجره Properties رفته و بر روی قسمت جلوی نام این رویداد دوبار کلیک کنید تا کنترل کننده رویداد آن به صورت خودکار برای شما نوشته شود.

**آیتم ToolStripTextBox**

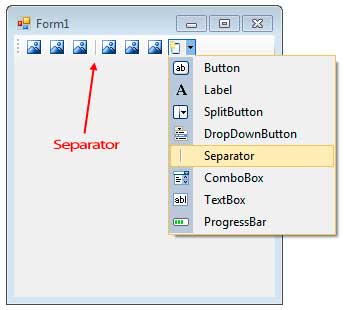
این آیتم نمایانگر یک جعبه متن است. این کنترل شبیه به کنترل TextBox است و خاصیت بسیار مهم آن Text است که هنگام تغییر آن رویداد TextChanged اتفاق می افتد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1017.jpg)

به این نکته توجه کنید که رویداد پیشفرض این آیتم کلیک است. اگر می خواهید یک کنترل کننده رویداد به آن اضافه کنید به پنجره Properties رفته و در قسمت رویدادها ، رویداد TextChanged را پیدا و بر روی آن دوبار کلیک کنید تا کنترل کننده رویداد آن اضافه شود.

**آیتم ToolStripSeparator**

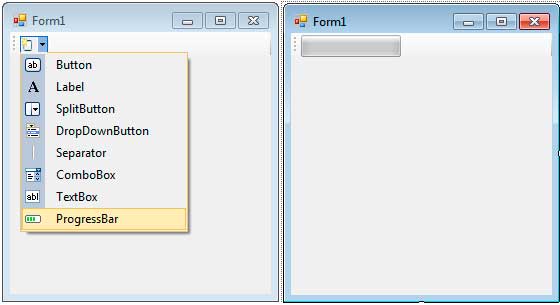
تنها هدف این آیتم جدا کردن آیتم ها، به قسمت های مرتبط با هم است.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1016.jpg)

همانطور که در شکل بالا می بینید ، این آیتم کنترل ToolStrip را به قسمت های مختلفی تقسیم می کند.

**آیتم ToolStripProgressBar**

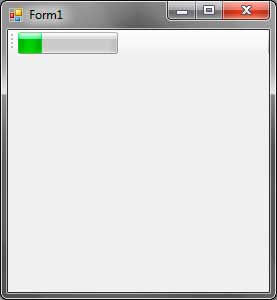
از این کنترل برای نمایش روند پیشرفت یک عملیات مانند ذخیره کردن ، حذف کردن یا هر عملیاتی که کاربر نیاز به صبر کردن دارد استفاده  می شود. با استفاده از این آیتم یک نوار پیشرفت به کاربر نمایش داده می شود.بدون وجود این کنترل کاربر ممکن است تصور کند که برنامه قفل کرده یا کار نمی کند .

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1018.jpg)

دو خاصیت مهم ToolStripProgressBar عبارتند از: Step و Value. خاصیت Value تعیین کننده وضعیت جاری نوار پیشرفت است و خاصیت Step تعیین می کند که با فراخوانی متد PerformStep() چه مقدار به خاصیت Value اضافه شود.

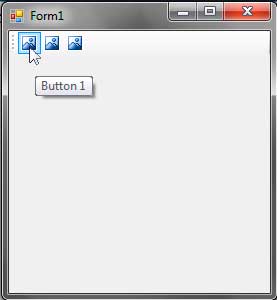
به عنوان مثال ، مقدار این خاصیت را برابر ۱ قرار داده و یک کنترل Timer از قسمت Components جعبه ابزار بر روی فرم قرار دهید. مقدار خاصیت Enabled تایمر را به true تغییر داده و بر روی آن دوبار کلیک کنید تا کنترل کننده رویداد Tick ایجاد شود.سپس کد زیر را درون آن قرار دهید  :

کد بالا باعث می شود هر ۱۰۰ میلی ثانیه یک بار(این مقدار برابر خاصیت interval تایمر است)  مقدار خاصیت value کنترل ToolStripProgressBar (که برابر خاصیت Step است ) یک واحد افزایش یابد. برنامه را اجرا و نتیجه رامشاهده کنید.

**[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1019.jpg)**

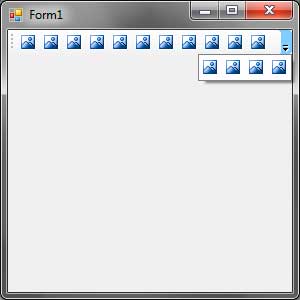
**اضافه کردن ToolTip به آیتم های ToolStrip**

هر آیتم دارای خاصیتی به نام ToolTipText است که مشخص کننده متنی است که هنگام قرار دادن موس بر روی آیتم نمایان می شود.  به این نکته توجه کنید که ابتدا مقدار خاصیت ShowItemToolTips کنترل ToolStrip را به true تغییر دهید.

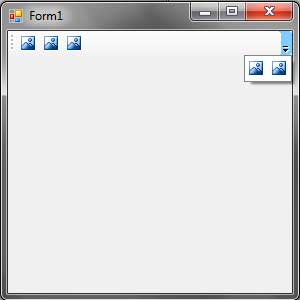
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1020.jpg)

**استفاده از خاصیت های و Overflow**

هنگامی که شما تعداد زیادی آیتم به کنترل ToolStrip اضافه کنید برای آیتم های بعدی چه اتفاقی افتد؟ به عبارت دگیر برای آیتم های بعدی فضای کافی وجود ندارد.به طور پیشفرض مقدار خاصیت CanOverflow برابر true است. این کار سبب می شود که هنگامی که فضای کافی در کنترل ToolStip وجود ندارد ایتم ها در منوی مخفی که با یک دکمه کوچک در دسترس است قرار گیرند

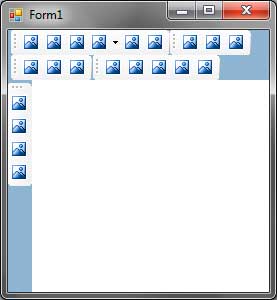
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1021.jpg)

همچنین می توانید مقدار خاصیت Overflow از هر ایتم را برابر Always قرار دهید . این کار سبب می شود که حتی زمانی که فضای کافی در کنترل وجود داشته باشد آیتم در منوی مخفی قرار گیرد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolbars-1022.jpg)

## کنترل ToolStripContainer

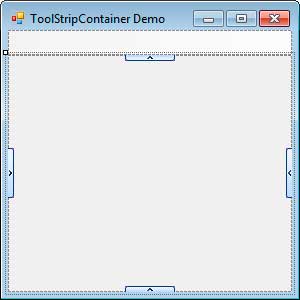
از این کنترل (System.Windows.Forms.ToolStripContainer) به عنوان یک دربرگیرنده برای نوار ابزارها استفاده می شود. می توانید آرایش و مکان نوار ابزارها را داخل این کنترل تعیین کنید. این کنترل متشکل از تعدادی ToolStripPanel است که کنترل های ToolStrip مختلف داخل آنها قرار می گیرند. به طور پیشفرض ، ۴ پنل در هر طرف ToolStripContainer وجود دارد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolstrip-container-control-1001.jpg)

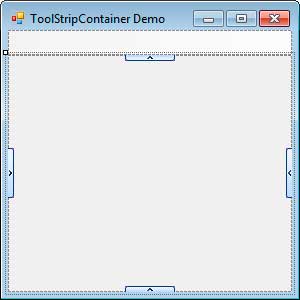
در زیر تعدادی از خصوصیت های مهم کنترل توضیح داده شده است.

| خاصیت | توضیح |
| --- | --- |
| BottomToolStripPanel | کنترل ToolStripPanel پایینی را بر می گرداند |
| BottomToolStripPanelVisible | تعیین می کند که ToolStripPanel پایینی نمایش داده شود یا نه |
| ContentPanel | پنل مرکزی کنترل را بر می گرداند |
| LeftToolStripPanel | ToolStripPanel سمت چپ را بر می گرداند |
| LeftToolStripPanelVisible | تعیین می کند که ToolStripPanel چپ نمایش داده شود یا نه |
| RightToolStripPanel | کنترل ToolStripPanel سمت راست را بر می گرداند |
| RightToolStripPanelVisible | مشخص می کند که ToolStripPanel سمت راست نمایش داده شود یا نه |
| TopToolStripPanel | کنترل ToolStripPanel بالایی را بر می گرداند |
| TopToolStripPanelVisible | مشخص می کند که ToolStripPanel بالایی نمایش داده شود یا نه |

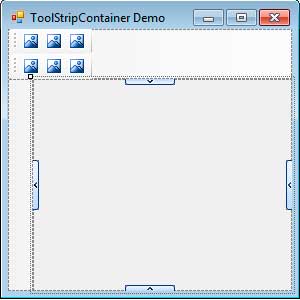
اجازه دهید در عمل از این کنترل استفاده کنیم. یک پروژه Windows Forms Application ایجاد کنید و نام آن را ToolStripContainerDemoبگذارید. یک ToolStripContainer از قسمت Menus&Toolbars جعبه ابزار بر روی فرم قرار دهید. مقدار خاصیت Dock کنترل را برابر Fill قرار دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolstrip-container-control-1022.jpg)

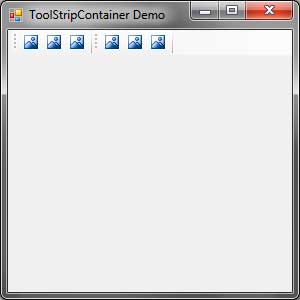
در حالت پیشفرض فقط ToolStripPanel بالایی نمایش داده می شود. شما می توانید  ToolStrip های مختلفی را در داخل این پنل قرار دهید. فلش های کناری به شما این امکان را می دهند که پنل های مختلفی را اضافه کنید. به عنوان مثال با کلیک بر روی فلش سمت چپ یک ToolStripPanel جدید ایجاد می شود که می توانید ToolStrip های مختلفی را در آن قرار دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolstrip-container-control-1022.jpg)

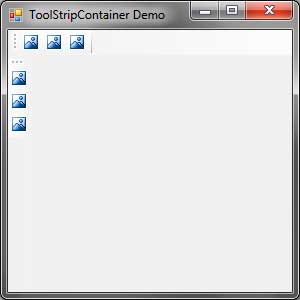
اجازه دهید دو ToolStrip را در پنل بالایی قرار دهیم.همچنین به هر کدام از آنها سه ToolStripButtons اضافه کنیم.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolstrip-container-control-1004.jpg)

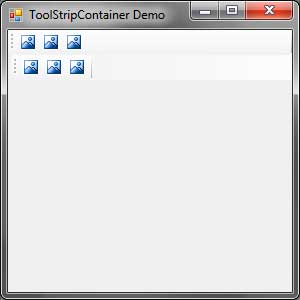
به این نکته توجه کنید که هنگامی که یک ToolStrip جدید اضافه می کنید ، کنترل جدید در زیر یا جلوی کنترل قبلی قرار می گیرد.همچنین سایز پنل برای جای دادن ToolStrip ها به طور خودکار تغییر می کند. برنامه را اجرا و نوار ابزاها را مشاهده کنید. با کلیک کردن یا کشیدن دستگیره جابه جایی هر ToolStrip، می توان مکان یا آرایش آن را تغییر داد. به عنوان مثال می توانید دومین ToolStrip را به بالای اولین ToolStrip کشیده تا در کنار آن قرار گیرد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolstrip-container-control-1005.jpg)

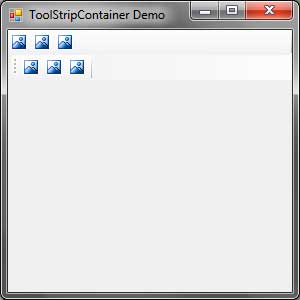
یاد آوری می کنیم که یک ToolStripPanel به ناحیه سمت چپ ToolStripContainer اضافه کرده ایم. اگر این ناحیه خالی باشد در زمان اجرا مخفی است.ولی شما می توانید یه کنترل ToolStrip را به مانند شکل زیر در آن قرار دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolstrip-container-control-1006.jpg)

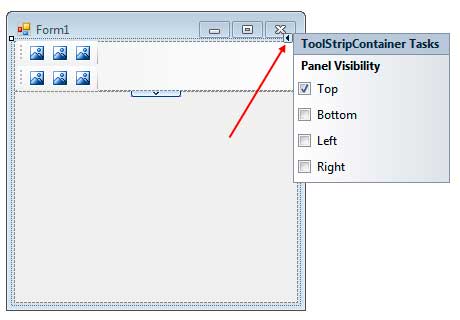
به قسمت طراحی فرم برگردید.اگر می خواهید که کنترل ToolStrip تمامی سطر یا ستون را اشغال کند مقدار خاصیت Stretch ان را برابر true قرار دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolstrip-container-control-1007.jpg)

برای غیر فعال کردن امکان تغییر مکان ToolStrip باید خاصیت GripStyle آنرا برابر Hidden قرار دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolstrip-container-control-1008.jpg)

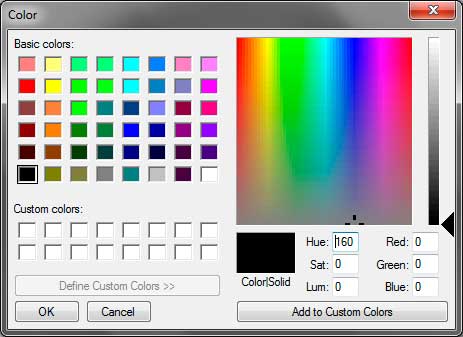
همچنین می توانید ToolStripPanels های که نیاز ندارید را غیر فعال کنید.برای این کار ToolStripContainer را انتخاب ، روی دکمه کوچک بالا سمت راست آن کلیک کنید تا پنجره Smart Task نمایان شود .تیک پنل هایی که نیاز ندارید را بر دارید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/toolstrip-container-control-1009.jpg)

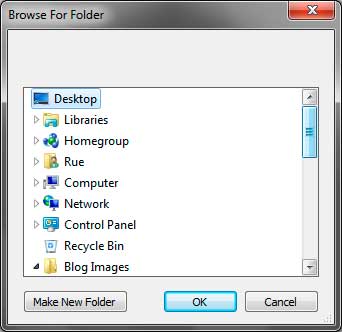
## 

## کادرهای محاوره ای

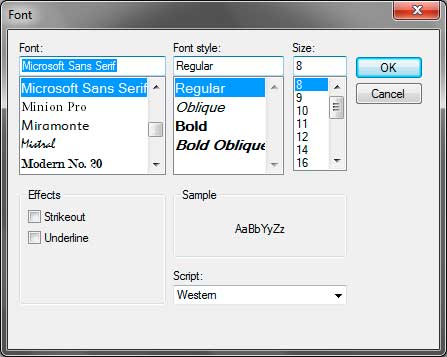
کادرهای محاوره ای پنجره هایی هستند که دارای وظایف مشخصی مانند ذخیره یا باز کردن یک فایل، انتخاب یک رنگ و یا چاپ یک سند می باشند. دات نت فریم ورک دارای کنترل هایی  می باشد که به شما اجازه می دهند با فراخوانی کادرهای محاوره ای امور خاصی را انجام دهید. کادرهای محاوره ای را می توان با استفاده از  خواص این کنترل ها سفارشی کرد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/dialogs-01.jpg)

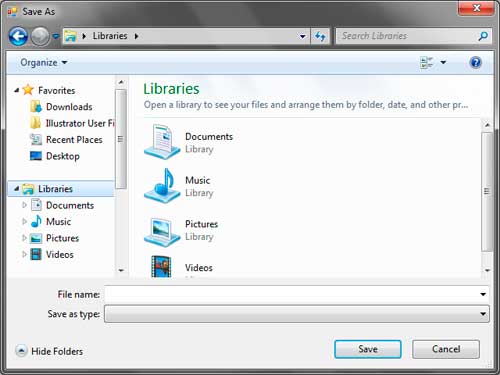
کادر محاوره ای انتخاب رنگ

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/dialogs-02.jpg)

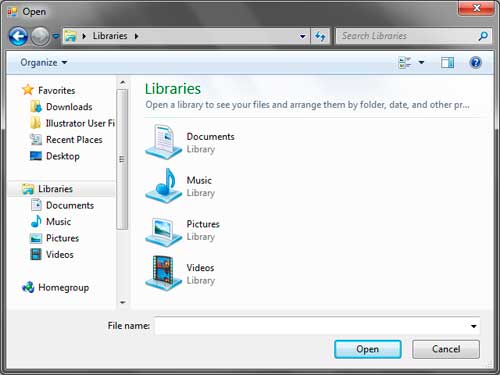
کادر محاوره ای جستجوی پوشه ها

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/dialogs-03.jpg)

کادر محاوره ای انتخاب سبک فونت

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/dialogs-04.jpg)

کادر محاوره ای ذخیره فایل ها

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/dialogs-05.jpg)

کادر محاوره ای باز کردن فایل ها

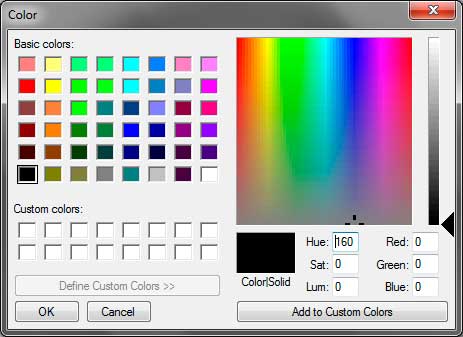
بنابراین می توانید با اسنتفاده از  این کادرهای محاوره ای برنامه های تحت ویندوز استانداردتری ایجاد کنید. این کادرها معمولا در تمام برنامه های ویندوزی مورد استفاده قرار می گیرند.

به این نکته توجه کنید که ظاهر کادرهای محاوره ای در نسخه های مختلف ویندوز متفاوت است.

عکس های بالا ظاهر این کادرها را در ویندوز ۷ نشان می دهد.

## کنترل ColorDialog

از کنترل ColorDialog وقتی استفاده می شود که شما بخواهید یک رنگ انتخاب کنید. به عنوان مثال وقتی بخواهید رنگ یک فونت یا رنگ پس زمینه فرم را تغییر دهید می توانید از این کنترل استفاده کنید.

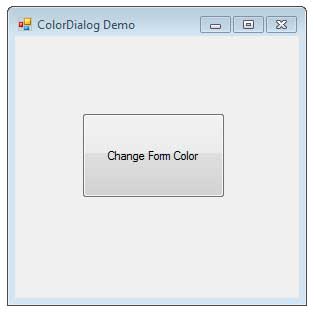
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/dialogs-011.jpg)

برخی خواص مفید کنترل ColorDialog در زیر ذکر شده است.

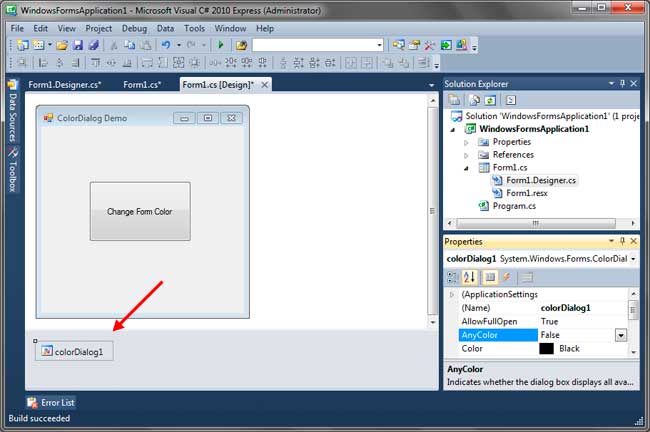
| خاصیت | خاصیت |
| --- | --- |
| AllowFullOpen | مشخص می کند که آیا کاربر می تواند یک رنگ سفارشی انتخاب کند یا نه |
| Color | رنگی که کاربر انتخاب می کند را بر می گرداند. |
| CustomColors | کلکسیونی از رنگ هایی که کاربر انتخاب کرده است را نشان می دهد. |
| FullOpen | Specifies whether the part used to pick custom colors are automatically open. |

پنجره ColorDialog از رنگ های از پیش تعریف شده تشکیل شده است. شما می توانید با کیلک بر روی دکمه  Define Custom Colors تعداد بیشتری رنگ مشاهده کنیید به طوری که در پنجره ظاهر شده هر رنگی که دوست دارید را انتخاب نمایید. همچنین می توانید از گزینه های Hue، Sat، Lum و همچنین RGB برای انتخاب رنگ مورد نظر استفاده کرد. فلش ساه رنگی که در سمت راست پنجره اصلی قرار دارد به شما اجازه تنظیم شفافیت یک رنگ خاص را می دهد. با استفاده از دکمه Add Custom Colors رنگ مورد نظرتان را به پالت Custom Colors برای استفاده های بعدی اضافه نمایید.

برای نشان دادن عملکرد کادر محاوره ای ColorDialog یک فرم ساده به صورت زیر ایجاد می کنیم :

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/color-dialog-01.jpg)

یک کنترل ColorDialog به فرم اضافه می کنیم. همانطور که مشاهده می کنید این کنترل به قسمت پایینی بخش طراحی اضافه می شود.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/color-dialog-02.jpg)

برای تغییر خواص کنترل ColorDialog بر روی ان کلیک کرده و به پنجره Properties بروید. بر روی کنترل  Button دوبار کلیک کنید و یک کنترل کننده رویداد برای رویداد Click آن ایجاد نمایید.

کد زیر را برای کنترل کننده رویداد بنویسید :

private void button1\_Click(object sender, EventArgs e)

{

DialogResult result = colorDialog1.ShowDialog();

if (result == DialogResult.OK)

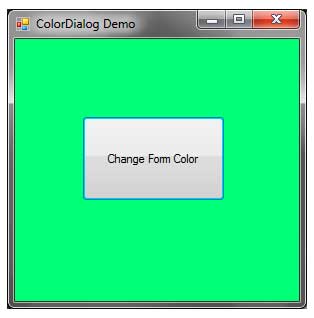
{

this.BackColor = colorDialog1.Color;

}

}

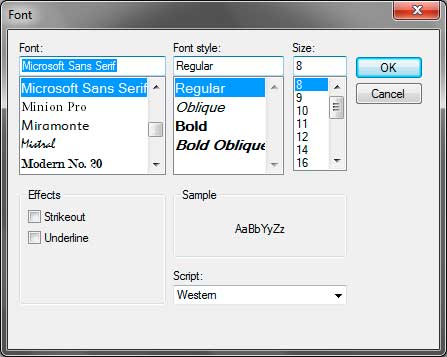
خط اول (خط ۲) متد استاتیک ShowDialog مربوط به نوع شمارشی  ColorDialog را فراخوانی می کند. این متد باعث ظاهر شدن کادر محاوره ای می شود در نتیجه کاربر می تواند یک رنگ را انتخاب کند. این متد همچنین یک مقدار System.Windows.Forms.DialogResult بر می گرداند که نشان می دهد آیا کاربر بر روی دکمه OK کلیک کرده است یا دکمه Cancel. اگر کاربر بر روی یک رنگ کلیک کرده و دکمه OK را بزند کادر بسته می شود و رنگی که کاربر انتخاب کرده در خاصیت Color کنترل ذخیره می شود. با آزمایش مقدار result می توان تشخیص داد که کاربر بر روی دکمه OK کلیک کرده است. در صورتی که دکمه OK توسط کاربر فشار داده شود رنگ پس زمینه فرم به رنگی در می آید که توسط کاربر در خاصیت Color انتخاب شده است.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/color-dialog-03.jpg)

برنامه را اجرا کرده و بر روی دکمه کلیک کنید تا کادر محاوره ای باز شود. یک رنگ را انتخاب کرده و بر روی ok کلیک کنید.مشاهده می کنید که رنگ فرم به رنگی که شما انتخاب کرده اید تغییر می کند.

## کنترلFontDialog

کنترل FontDialog کنترلی است که از آن برای انتخاب انواع مختلف فونت و خواص وابسته به فونت استفاده می شود.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/dialogs-031.jpg)

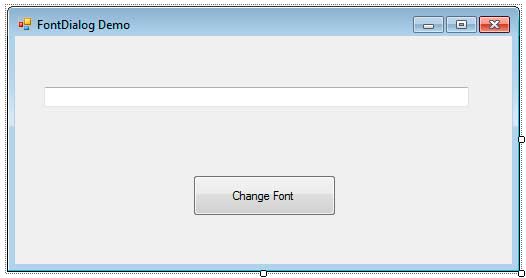
با استفاده از FontDialog می توانید نوع، اندازه و رنگ فونت را تغییر دهید. همچنین می توانید قبل از اعمال تغیر یک پیش نمایش از تغییرات را مشاهده نمایید.

برخی از خواص مفید این کنترل در جدول زیر آمده است.

| خاصیت | توضیح |
| --- | --- |
| Color | رنگ انتخاب شده توسط کاربر را نشان می دهد |
| Font | برای تعیین نوع فونت به کار می رود. |
| MaxSize | حداکثر اندازه فونت را مشخص می کند. |
| MinSize | حداقل اندازه فونت را مشخص می کند. |
| ShowApply | نشان می دهد که آیا کادر محاوره ای شامل دکمه Apply باشد یا نه |
| ShowColor | نشان می دهد که آیا کادر محاوره ای دارای کادری برای انتخاب رنگ باشد یا نه |
| ShowEffects | نشان می دهد که آیا کادر محاوره ای دارای کادری برای زیر خط دار کردن فونت باشد یا نه |

به این نکته توجه کنید که بخش انتخاب رنگ در کادر محاوره ای FontDialog در حالت پیشفرض مخفی می باشد. برای اینکه به کاربر اجازه انتخاب رنگ بدهید مقدار خاصیت ShowColor رابرابر true کنید. خاصیت ShowApply اجازه اضافه کردن دکمه Apply را به کادر FontDialog می دهد. در این صورت شما با اجرای رویداد Apply تغییرات را بر روی فونت مورد نظر انجام می دهید بدون اینکه کادر محاوره ای بسته شود.  به مثال ساده زیر در مورد استفاده از این کنترل توجه کنید :

یک فرم شبیه به فرم زیر ایجاد کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/font-dialog-01.jpg)

ما می خواهیم فونت داخل کنترل textbox را با استفاده از کنترل Fontdilog تغییر دهیم. کنترل FontDialog را برروی فرم می کشیم. از آنجاییکه کنترل FontDialog یک کنترل غیر بصری است در قسمت پایینی بخش طراحی قرار می گیرد.

بر روی دکمه دوبار کلیک کنید و در  کنترل کننده رویداد ایجاد شده کد زیر را وارد کنید.

private void button1\_Click(object sender, EventArgs e)

{

DialogResult result = fontDialog1.ShowDialog();

if (result == DialogResult.OK)

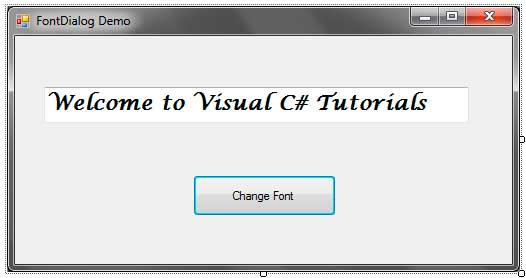
{

textBox1.Font = fontDialog1.Font;

}

}

برای نشان داده کادر محاوره ای به کاربر ابتدا متد استاتیک ShowDialog این کنترل FontDialog) ) را فراخوانی می کنیم. کاربر اکنون می تواند فونت های مختلفی را انتخاب کند و سپس دکمه OK را فشار دهد. با استفاده از مقدار برگشتی از متد ShowDialog تست می کنیم که آیا کاربر دکمه OK را فشار داده است یا نه. اگر کاربر دکمه OK را فشار داده باشد فونت داخل کنترل textbox تغییر خواهد کرد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/font-dialog-02.jpg)

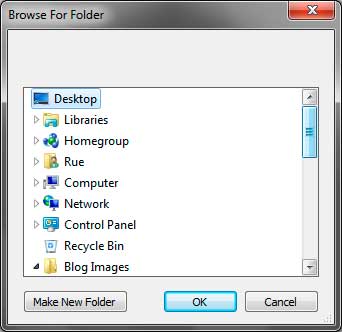
برنامه را اجرا کرده و یک متن را در داخل textbox بنویسید. بر روی دکمه کلیک کنید تا پنجره Font Dialog باز شود و سپس یک فونت را به دلخواه انتخاب کنید.

در نهایت بر روی دکمه OK کلیک کنید تا فونت textbox تغییر کند.

## 

## کنترل FolderBrowserDialog

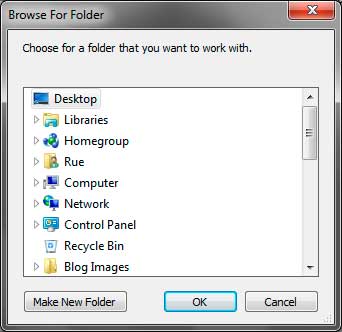
کنترل FolderBrowserDialog به شما اجازه می دهد که به دنبال یک فایل یا پوشه در داخل سیستم بگردید. این کنترل از یک نمایش درختی برای نمایش پوشه ها استفاده می کند. مرورگر این کنترل در حالت پیشفرض دسکتاپ و محتویات آن را نمایش می دهد. می توانید با کلیک بر روی فلش کوچکی که در کنار هر کدام از زیر مجموعه های دسک تاپ قرار دارد به زیرپوشه ها و فایل های آنها دست یافت(شکل زیر). همچنین می توان یک پوشه جدید در داخل پوشه انتخاب شده با استفاده از دکمه Make New Folder ایجاد کرد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/dialogs-021.jpg)

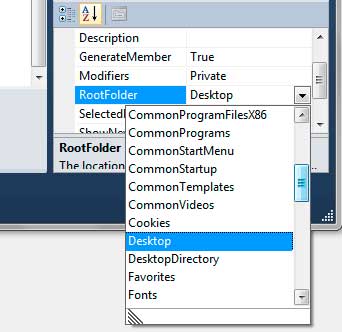
در زیر برخی از خواص مفید کنترل FolderBrowserDialog آمده است.

| خاصیت | توضیح |
| --- | --- |
| Description | به شما اجازه می دهد که یک متن توضیحی به بالای نمایش درختی FolderBrowserDialog اضافه کنید |
| RootFolder | یک پوشه را به عنوان پوشه اصلی کنترل نمایش می دهد.مثلا دسکتاپ، پوشه اصلی و بقیه زیر پوشه. |
| SelectedPath | پوشه یا مسیری که کاربر انتخاب کرده است. |
| ShowNewFolderButton | نشان دادن یا مخفی کردن دکمه Make New Folder. |

خاصیت Descriptionبه شما اجازه می دهد که یک توضیح به قسمت بالای نمایش درختی کادر محاوره ای کنترل اضافه کنید.

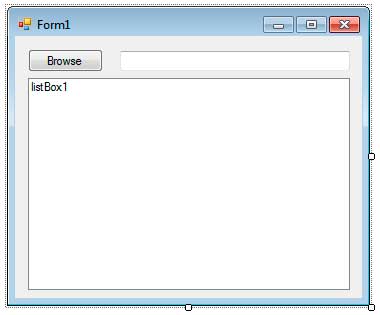
[](http://www.w3-farsi.com/wp-content/uploads/2013/10/folderbrowserdialog-01.jpg)

اگر نمی خواهید از دسک تاپ به عنوان ریشه اصلی استفاده نمایید (و بقیه پوشه ها زیر مجموعه آن باشند) می توانید از خاصیت RootFolder استفاده نمایید. در پنجره properties خاصیت RootFolder را یافته و بر روی دکمه drop down کناری آن کلیک کنید. در این صورت مسیرهای از پیش تعریف شده ای که می توانید آنها را انتخاب کنید به شما نمایش داده می شود.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/folderbrowserdialog-02.jpg)

با استفاده از خاصیت SelectedPath می توان پوشه یا مسیر انتخاب شده را بدست آورد. مسیر به عنوان یک رشته برگردانده می شود. می خواهیم با یک مثال کارکرد کنترل FolderBrowserDialog را به شما نشان دهیم. برنامه ای که به عنوان مثال در نظر گرفته ایم به کاربر اجازه می دهد که یک پوشه را انتخاب کرده و سپس مسیر ان را در داخل جعبه متن (text box) نمایش دهد. محتویات پوشه هم در داخل یک لیست باکس فهرست می شوند.

یک فرم جدید مانند شکل زیر ایجاد کنید. و کنترل های button، text box و listbox را در داخل آن قرار دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/folderbrowserdialog-03.jpg)

از گروه Dialogs واقع در Toolbox کنترل FolderBrowserDialog را اضافه کنید. مشاهده می کنید که کنترل در قسمت پایینی محیط طراحی قرار می گیرد.

بر روی دکمه Browse دو بار کلیک کرده و در داخل کنترل کننده رویداد آن کد زیر را بنویسید.

private void button1\_Click(object sender, EventArgs e)

{

DialogResult result = folderBrowserDialog1.ShowDialog();

if (result == DialogResult.OK)

{

//Show the path using the text box

textBox1.Text = folderBrowserDialog1.SelectedPath;

//Obtain information about the path

DirectoryInfo selectedPath = newDirectoryInfo(textBox1.Text);

//Clear the list box first

listBox1.Items.Clear();

//Check if there are directories then add a label

if (selectedPath.GetDirectories().Length > 0)

listBox1.Items.Add("== Directories ==");

//Show all the directories using the ListBox control

foreach (DirectoryInfo dir in selectedPath.GetDirectories())

{

//Show only the name of the directory

listBox1.Items.Add(dir.Name);

}

//Check if there are files then add a label

if (selectedPath.GetFiles().Length > 0)

listBox1.Items.Add("== Files ==");

//Show all the directories using the ListBox control

foreach (FileInfo file in selectedPath.GetFiles())

{

listBox1.Items.Add(file.Name);

}

}

}

این نکته فراموش نشود که لازم است در بالای کد فضای نام System.IO را وارد کنیم :

using System.IO;

ابتدا با استفاده از متد ShowDialog ، FolderBrowserDialog را فراخوانی می کنیم. حال کاربر می تواند یک پوشه را انتخاب کرده و دکمه OK را بزند.

می توانید تست کنید که آیا کاربر دکمه OK را زده است یا نه. برای تشخیص اینکار می توانید یک مقدار را به وسیله متد ShowDialog برگشت دهید (خط ۵).

اگر کاربر دکمه OK را فشار داد سپس مسیر انتخاب شده توسط و در داخل جعبه متن نمایش داده شود.

DirectoryInfo selectedPath = newDirectoryInfo(textBox1.Text);

این خط با استفاده از مسیر انتخاب شده توسط کاربر یک شی از DirectoryInfo ایجاد می کند. شی DirectoryInfo در داخل فضای نام System.IO قرار دارد و شامل خواصی است که مشخص می کنند که مسیر پوشه انتخاب شده کجاست و محتویات آن چیست.

چون با انتخاب یک مسیر یا پوشه زیر مجموعه های آن در داخل لیست باکس فهرست می شوند لازم است برای ورود آیتم های جدید لیست باکس پاک شود این کار را با استفاده از کد زیر انجام می دهیم.

//Check if there are directories then add a label

if (selectedPath.GetDirectories().Length > 0)

listBox1.Items.Add("== Directories ==");

//Show all the directories using the ListBox control

foreach (DirectoryInfo dir in selectedPath.GetDirectories())

{

//Show only the name of the directory

listBox1.Items.Add(dir.Name);

}

سپس چک می کنیم که آیا در داخل پوشه انتخاب شده پوشه های دیگری وجود دارند یا نه؟ این کار را با استفاده از خاصیت Length کلاس DirectoryInfo[] و برگرداندن مقدار بوسیله فراخوانی متد GetDirectories انجام می دهیم. اگر وجود داشت یک کنترل Lable ایجاد کرده و آن را به لیست باکس مان اضافه می کنیم.

با استفاده از یک حلقه foreach و متد GetDirectories همه زیر پوشه های، پوشه جاری را می شماریم. سپس نام هر یک از زیر پوشه ها را به لیست باکس اضافه می کنیم.

//Check if there are files then add a label

if (selectedPath.GetFiles().Length > 0)

listBox1.Items.Add("== Files ==");

//Show all the directories using the ListBox control

foreach (FileInfo file in selectedPath.GetFiles())

{

listBox1.Items.Add(file.Name);

}

از تکنیک مشابهی برای نوشتن نام همه فایل های داخل یک پوشه انتخاب شده استفاده می کنیم. اما در اینجا ما از متد GetFiles و اشیا کلاس FileInfo استفاده می کنیم. شی FileInfo شامل اطلاعات خاصی در باره یک فایل مشخص است.

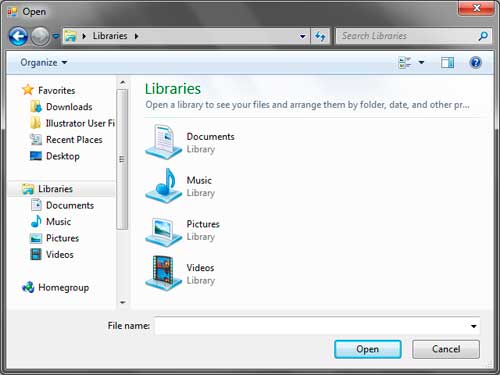
حال برنامه را اجرا کرده و یک پوشه انتخاب می کنیم. بر روی دکمه OK کلیک کرده تا برنامه، پوشه هایی را که در داخل پوشه انتخاب شده وجود دارند را به شما نشان دهد.

## 

## کنترل OpenFileDialog

کنترل System.Windows.Forms.OpenFileDialog به شما اجازه می دهید که یک فایل را انتخاب کرده و بخوانید مثلا یک فایل متنی را باز کرده و محتویات آن را بخوانید. این دیالوگ به شما امکان پیمایش فایل ها موجود بر روی  سیستم را می دهد.

شما می توانید نام یک فایل را در مسیر جاری یا اینکه مسیر کامل آن را در جعبه متنی که در جلوی متن File Name و پایین پنجره قرار دارد بنویسید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/dialogs-051.jpg)

در جدول زیر لیستی از خاصیت های مهم این کنترل توضیح داده شده است.

| خاصیت | توضیح |
| --- | --- |
| AddExtention | تعیین می کند که هنگامی که کاربر نام یک فایل بدون پسوند را تایپ می کند یک پسوند به طور خودکار به آن اضافه شود یا نه؟ |
| CheckFileExists | تعیین می کند زمانی که کاربر نام یک فایل را که وجود ندارد می نویسد یک پنجره ی اخطار به او نمایش داده شود یا نه |
| CheckPathExists | تعیین می کند زمانی که کاربر مسیری که وجود ندارد را می نویسد یک پنجره ی اخطار به او نمایش داده شود یا نه |
| DefaultExt | پسوند پیشفرض . زمانی که کاربر نام فایل را بدون پسوند بنویسد به طور خودکار به آخر فایل اضافه می شود. |
| FileName | فایلی که توسط کاربر انتخاب شده است. |
| FileNames | مجموعه ای از فایل ها که توسط کاربر انتخاب می شود. |
| Filter | با استفاده از این خاصیت می توانید لیست فایل هایی (پسوند هایی) که کاربر می تواند انتخاب کند را مشخص کنید. |
| FilterIndex | اگر پسوندهای متفاوتی در دسترس باشد ، با استفاده از این خصوصیت می توانید تعیین کنید که کدام پسوند به طور پیشفرض هنگام باز شدن پنجره نمایش داده شود. |
| InitialDirectory | دایرکتوری پیشفرضی که هنگام نمایش پنجره در آن قرار داریم را تعیین می کند. |
| Multiselect | تعیین می کند که امکان انتخاب چندین فایل توسط کاربر وجود دارد یا نه. |
| Title | عنوان پنجره را تعیین می کند. |

خاصیت AddExtension تعیین می کند زمانی که کاربر پسوند فایل را نمی نویسد یک پسوند که در خاصیت DefaultExt مشخص شده است به آن اضافه کند یا نه؟

توصیه می شود که مقدار دو خاصیت CheckFileExists و CheckPathExists را برابر true قرار دهید تا زمانی که فایل یا مسیر مشخصی وجود ندارد ، به کاربر اخطاری نمایش داده شود.

خصوصیت InitialDirectory تعیین کننده پوشه پیشفرضی است که هنگام نمایش دیالوگ در آن قرار داریم.

خصوصیت Title نمایانگر متنی است که در عنوان دیالوگ قرار دارد.

خصوصیت FileName مسیر فایلی که کاربر توسط دیالوگ انتخاب کرده است را بر می گرداند.

همچنین با استفاده از خاصیت MultiSelect امکان انتخاب چندین فایل در دیالوگ را می دهید.لیست فایل های انتخاب شده در خصوصیت FileNames قرار می گیرد.

**فیلتر کردن فایل ها**

شما می توانید فقط فایل هایی که دارای نوع مشخصی هستند را در پنجره نمایش دهید. برای این کار باید یک الگو که یک رشته است را در خاصیت Filter مشخص کنید.

به عنوان مثال ، می توانیم فقط فایل هایی که دارای پسوند .txt هستند را در پنجره نمایش دهیم.

خاصیت Filter به یک الگوی خاص نیاز دارد.

Description1|FilterPattern1|Description2|FilterPattern2|...DescriptionN|FilterPatternN

ابتدا باید شرحی از نوع فایل را مشخص کنیم.

سپس کاراکتر |  و الگوی مشخصی  را بعد از آن قرار می دهیم.

به عنوان مثال الگوی زیر فقط لیست فایل های متنی را نمایش می دهد.

Text Files|\*.txt

شرح نوع فایل Text Files و الگوی آن \*.txt است.

کاراکتر \* یک کاراکتر جایگزین است که بیانگر هر نامی است.

قسمت .txt مشخص کننده یک پسوند خاص است.

الگو تعیین می کند که فایل های با پسوند .txt در پنجره نمایش داده شود.

همچنین از کاراکتر های جایگزین برای بسیاری از هدف های دیگر نیز استفاده کنید.

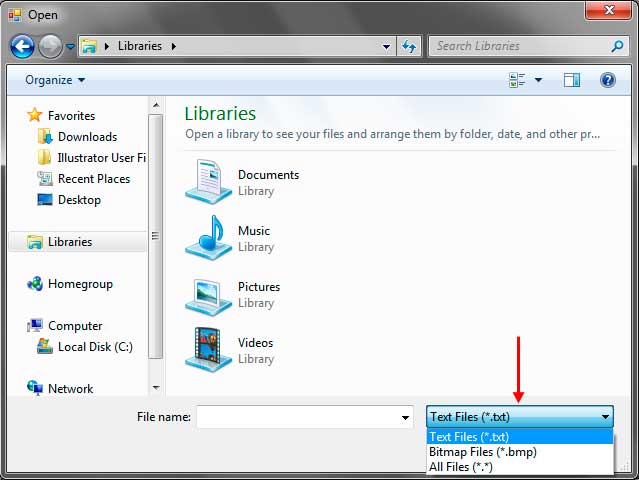
به عنوان مثال ، الگوی m\*.txt تمامی فایل های متنی که با حرف m شروع می شوند را تطبیق میدهد . الگوی \*r.txt تمامی فایل های متنی که نام آنها به حرف r ختم می شود را تطبیق می دهد. الگوی \*.\* تمامی انواع فایل ها را تطبیق می دهد.الگوی \*.t\* تمامی فایل های که پسوند انها با حرف r شروع می شود را تطبیق می دهد.

شما می توانید چندین فیلتر را مشخص کنید.

به عنوان مثال الگوی زیر متشکل از چندین فیلتر است.

Text Files|\*.txt|Bitmap Files|\*.bmp|All Files|\*.\*

در جلوی جعبه متنی که نام فایل را می نویسید یک کامبو باکس وجود دارد که لیست فیلتر های مختلف در آن قرار دارد.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/file-open-dialog-01.jpg)

همچنین این امکان وجود دارد که برای چندین پسوند یک توضیح نوشته شود.

به عنوان مثال، فایل های تصویری دارای پسوند های مختلفی از جمله bmp ، jpeg ، png هستند.

شما به راحتی می توانید پسوند های مختلف را با سمی کالن از یکدیگر جدا کنید.

Image Files|\*.bmp;\*.jpeg;\*.png;\*.gif

هنگامی که این فیلتر را انتخاب می کنید تمامی فایل هایی که حداقل با یکی از الگوها ی آن مطابقت داشته باشد نشان داده می شود.

**مثالی از کاربرد OpenFileDialog**

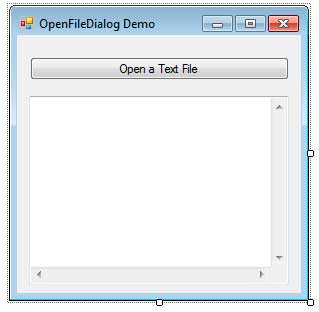
در ادامه یک پروژه جدید از نوع Windows Application درست می کنیم که از قابلیت های پایه ای کنترل OpenFileDialog استفاده می کند.

کاربر با استفاده از برنامه یک فایل متنی را انتخاب می کند سپس برنامه محتوای آن فایل را در یک جعبه متن چند خطی نشان می دهد.

ابتدا به این نکته توجه کنید که فضای نامی System.IO را به برنامه اضافه کنید.

using System.IO;

یک فرم به شکل زیر طراحی کنید. از جعبه متنی استفاده کنید که خاصیت MultiLine آن برابر true باشد. مقادیر خاصیت های Scrollbars و WordWrap را به ترتیب برابر both و false  تغییر دهید.یک کنترل OpenFileDialog را ازجعبه ابزار بر روی فرم قرار دهید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/file-open-dialog-02.jpg)

بر روی دکمه دو بار کلیک کنید تا کنترل کننده رویداد Click ایجاد شود. دوباره یادآوری می شود که فضای نامی System.IO را به لیست فضای نامی کلاس اضافه کنید.

از کد زیر در داخل کنترل کننده رویداد استفاده کنید.

private void button1\_Click(object sender, EventArgs e)

{

//Filter to only text files

openFileDialog1.Filter = "Text Files|\*.txt";

//No initial file selected

openFileDialog1.FileName = String.Empty;

//Open file dialog and store the returned value

DialogResult result = openFileDialog1.ShowDialog();

//If Open Button was pressed

if (result == DialogResult.OK)

{

//Create a stream which points to the file

Stream fs = openFileDialog1.OpenFile();

//Create a reader using the stream

StreamReader reader = newStreamReader(fs);

//Read Contents

textBox1.Text = reader.ReadToEnd();

//Close the reader and the stream

reader.Close();

}

}

در خط اول با استفاده از خاصیت Filter یک فیلتر را اضافه می کنیم. با استفاده از الگوی آن مشخص می کنیم که در پنجره فقط لیست فایل های متنی نمایان شود.

در خط دوم به خاصیت FileName مقدار empty را نسبت می دهیم تا در ابتدا هیچ فایلی انتخاب نشود.

سپس با استفاده از متد ShowDialog() پنجره را نمایش می دهیم.این متد خروجی از نوع DialogResult دارد.

هنگامی که کاربر یک فایل مجاز را انتخاب و بر روی دکمه Open کلیک می کنید خروجی متد برابر DialogResult.OK می شود. با استفاده از دستور شرط خروجی متد را بررسی می کنیم.

خروجی  متد OpenFile() از کنترل OpenFileDialog را در یک شی از نوع Stream ذخیره می کنیم.

شی Stream به فایل انتخاب شده ارجاع دارد ، از این شی در ساخت شی ای از کلاس StreamReader که کار خواندن از فایل را به عهده دارد استفاده می کنیم.

با استفاده از متد ReadToEnd کلاس StreamReader محتوای کامل فایل را خوانده و در خاصیت Text کنترل textBox1 نمایش می دهیم.

برنامه را اجرا و بر روی دکمه کلیک کنید. یک فایل متنی را از پنجره نمایش داده شده انتخاب کنید.

هنگامی که فایلی که مسیر آن توسط کاربر نوشته شده موجود نباشد، یک پیغام خطا به او نمایش داده می شود.فراموش نکنید که ابتدا خصوصیات  CheckFileExists و CheckPathExists را برابر true قرار دهید.

اگر فایل انتخاب شده موجود و مجاز باشد محتوای کامل آن در جعبه متنی قرار می گیرد.

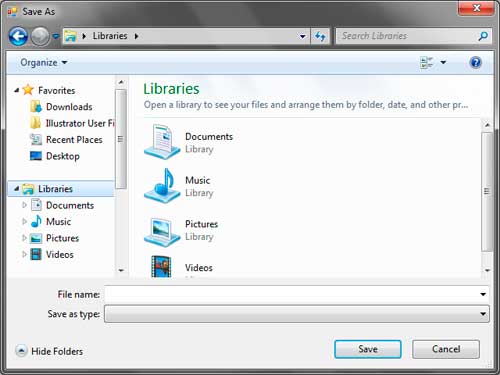
## کنترل SaveFileDialog

کنترل System.Windows.Forms.SaveFileDialog به شما اجازه می دهد که یک فایل را ذخیره یا متنی را در یک فایل بنویسید.

این کنترل همچنین به شما اجازه می دهد که در بین پوشه های موجود در سیستم خود یک پوشه را انتخاب کرده ، سپس نام فایلی که قصد دارید اطلاعاتی را در آن قرار دهید را بنویسید.

زمانی که  نام فایلی که در پوشه جاری پنجره قرار دارد را بنویسید یک پیغام خطا نمایش داده می شود مبنی بر اینکه اطلاعات موجود در فایل دوباره نویسی می شود(overwrite) .

همچنین می توانید مسیر کامل فایل را در داخل جعبه متنی که در قسمت پایین آن قرار دارد بنویسید.

[](http://www.w3-farsi.com/wp-content/uploads/2014/02/dialogs-04.jpg)

در زیر تعدادی از خاصیت های مفید این کنترل را مشاهده می کنید.

| خاصیت | توضیح |
| --- | --- |
| AddExtention | تعیین می کند که هنگامی که کاربر نام یک فایل بدون پسوند را تایپ می کند یک پسوند به طور خودکار به ان اضافه شود یا نه |
| CheckFileExists | تعیین می کند زمانی که کاربر نام یک فایل را که وجود ندارد می نویسد یک پنجره ی اخطار به او نمایش داده شود یا نه |
| CheckPathExists | تعیین می کند زمانی که کاربر مسیری که وجود ندارد را می نویسد یک پنجره ی اخطار به او نمایش داده شود یا نه |
| DefaultExt | پسوند پیشفرض . زمانی که کاربر نام فایل را بدون پسوند بنویسد به طور خودکار به نام فایل اضافه می شود. |
| FileName | فایلی که توسط کاربر انتخاب شده است. |
| Filter | با استفاده از این خاصیت می توانید لیست فایل های داخل پنجره را فیلتر کنید. |
| FilterIndex | اگر فیلتر های متفاوتی در دسترس باشد ، با استفاده از این خصوصیت می توانید تعیین کنید که کدام فیلتر به پنجره اعمال شود |
| InitialDirectory | دایرکتوری پیشفرضی که هنگام نمایش پنجره در آن قرار داریم را تعیین می کند. |
| OverwritePrompt | زمانی که فایل مورد نظر شما در پنجره موجود باشد یک پیغام به شما نمایش داده  می شود |
| Title | عنوان پنجره |

خاصیت AddExtension به طور خودکار زمانی که کاربر فایل بدون پسوندی را بنویسد ، پسوندی که در خاصیت DefaultExt قرار دارد را به آن اضافه می کند.

توصیه می شود که مقدار دو خاصیت CheckFileExists و CheckPathExists را برابر true قرار دهید تا زمانی که فایل یا مسیر مشخصی وجود ندارد ، به کاربر اخطاری نمایش داده شود.

خصوصیت InitialDirectory تعیین کننده پوشه پیشفرضی است که هنگام نمایش دیالوگ در آن قرار داریم.

 خصوصیت Title نمایانگر متنی است که در عنوان دیالوگ قرار دارد.

**مشخص کردن نوع فایل**

می توان نوع فایل را تعیین کرد.

برای این کار از خاصیت Filter استفاده می شود.این خاصیت یک رشته حاوی الگوی خاص را به عنوان مقدار می پذیرد

به عنوان مثال ، تعیین می کنیم که فایلی که قصد ذخیره آن را داریم از نوع متنی می باشد . خاصیت Filter باید دارای یک الگوی خاص باشد.

Description1|Extension1|Description2|Extention2|...DescriptionN|ExtentionN

ابتدا برای نوع فایل یک توضیح می نویسیم.

در ادامه یک کاراکتر |  و سپس نوع فایل را می نویسیم.

به عنوان مثال الگوی زیر به شما اجازه می دهد که فایل را فقط به صورت متنی ذخیره کنید.

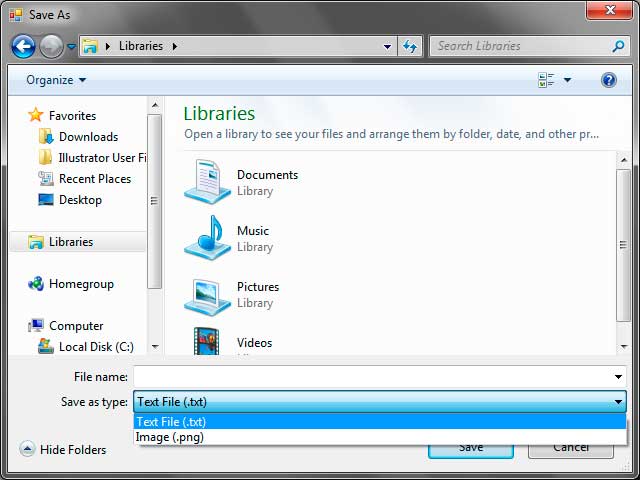
Text Files|.txt

شرح فایل در مثال بالا برابر Text Files و پسوند آن برابر .txtاست.

این امکان وجود دارد که چندین پسوند را مشخص کنید.به عنوان نمونه الگوی زیر به شما اجازه می دهد که یک فایل را به متنی با پسوند .txt با به صورت تصویری با پسوند .png ذخیره کنید.

Text Files|\*.txt|Image|\*.png

نوع فایل را از کامبو باکسی که در زیر جعبه متن نام فایل قرار دارد انتخاب می کنید.

[](http://www.w3-farsi.com/wp-content/uploads/2014/02/save-file-dialog-011.jpg)

**یک مثال از کاربرد کنترل SaveFileDialog**

اجازه دهید یک برنامه نمونه که در آن از قابلیت های کنترل SaveFileDialog استفاده شده است را طراحی کنیم.

برنامه به کاربر اجازه می دهد که محتوای یک جعبه متن چند خطی را در یک فایل متنی که به وسیله این کنترل انتخاب می شود قرار دهد.

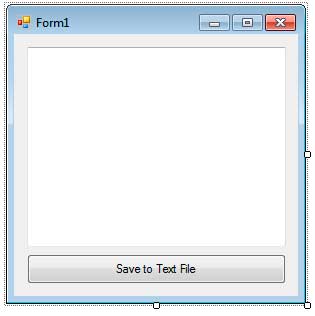
ابتدا توجه کنید که کد زیر را به لیست فضای نامی های کلاس اضافه کنید.

using System.IO;

یک فرم شبیه فرم زیر طراحی کنید.

از جعبه متنی که خاصیت mutiline آن برابر true است استفاده کنید.

یک کنترل  SaveFileDialog را از قسمت Dialogs جعبه ابزار بر روی بکشید.

[](http://www.w3-farsi.com/wp-content/uploads/2014/02/save-file-dialog-02.jpg)

بر روی دکمه دوبار کلیک کنید تا کنترل کننده رویداد کلیک آن ایجاد شود.

به شکل زیر کد های مشابه را در کنترل کننده رویداد قرار دهید.

private void button1\_Click(object sender, EventArgs e)

{

//Specify the extensions allowed

saveFileDialog1.Filter = "Text File|.txt";

//Empty the FileName text box of the dialog

saveFileDialog1.FileName = String.Empty;

//Set default extension as .txt

saveFileDialog1.DefaultExt = ".txt";

//Open the dialog and determine which button was pressed

DialogResult result = saveFileDialog1.ShowDialog();

//If the user presses the Save button

if (result == DialogResult.OK)

{

//Create a file stream using the file name

FileStream fs = newFileStream(saveFileDialog1.FileName, FileMode.Create);

//Create a writer that will write to the stream

StreamWriter writer = newStreamWriter(fs);

//Write the contents of the text box to the stream

writer.Write(textBox1.Text);

//Close the writer and the stream

writer.Close();

}

}

خط اول تعیین می کند که کاربرد فقط می تواند محتوای جعبه متنی را در یک فایل متنی قرار دهد.

خط دوم مقدار empty را به خصوصیت FileName نسبت می دهد تا در زمان نمایش هیچ فایلی به طور پیشفرض انتخاب نشود.

مقدار خاصیت DefaultExt را برابر .txt قرار داده ایم تا اگر کاربر پسوند فایل مورد نظر خود را فراموش کند برنامه به طور خودکار این پسوند را به انتهای نام فایل اضافه کند.

سپس با استفاده از متد ShowDialog پنجره را نمایش می دهیم.خروجی این متد از نوع DialogResult است.

هنگامی که کاربر بر روی دکمه Save کلیک می کند خروجی متد برابر DialogResult.OK می شود.

با استفاده از دستور if خروجی متد را بررسی می کنیم.

یک شی از کلاس FileStream ساخته و خاصیت های FileName و FileMode آنرا تنظیم کرده ایم.

سپس یک شی از کلاسStreamWriter ساخته و شی FileStream قبلی را به آن انتقال می دهیم.

با استفاده از متد Write متن موجود در جعبه متن را در استریم نوشته و سپس با استفاده از متد close استریم را می بندیم.

برنامه را اجرا و متنی را داخل جعبه متن بنویسید.

بر روی دکمه کلیک کنید و یک پوشه را انتخاب کنید.سپس نام یک فایل را بنویسید و بر روی دکمه Save پنجره کلیک کنید.

اگر فایلی با همان نام نوشته شده توسط شما موجود باشد یک پیغام به شما نمایش داده می شود.در صورت تایید پیغام، محتوای آن فایل کاملا  حذف و محتوای جدید به جای آن قرار می گیرد. اگر مقدار خاصیت OverwritePrmpt را برابر false قرار دهید پیغام خطا نمایش داده نمی شود .

## 

## رویدادهای ماوس

می توان از چندین رویداد خاص برای واکنش نشان دادن به اعمالی که با ماوس انجام می شود استفاده نمود.

همانطور که می دانید رویداد کلیک زمانی اتفاق می افتد که بر روی ماوس کلیک شود.

چندین رویداد پیشرفته برای اداره اعمال ماوس وجود دارد.

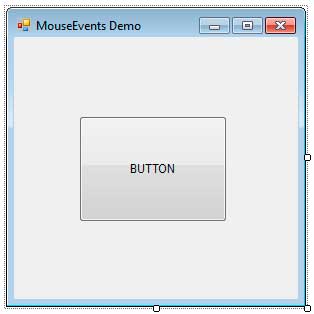
برخی از این رویدادها دارای یک نماینده (delegate) از نوع MouseEventHandler  می باشند که آن نیز به نوبه خود دارای پارامتر MouseEventArgs که شامل جزییاتی در مورد اعمال ماوس است، می باشد.

در زیر رویدادهای ماوس که می توانید از آنها استفاده کنید آمده است :

| رویداد | توضیح |
| --- | --- |
| MouseClick | وقتی روی می دهد که با ماوس بر روی کنترل کلیک شود. |
| MouseDoubleClick | وقتی روی می دهد که با ماوس بر روی کنترل دو بار کلیک شود. |
| MouseDown | وقتی روی می دهد که اشاره گر ماوس بر روی کنترل قرار گرفته و دکمه ماوس به سمت پایین فشار داده شود. |
| MouseEnter | وقتی روی می دهد که ماوس وارد کنترل شود. |
| MouseHover | وقتی روی می دهد که با ماوس بر روی کنترل مکث کنیم. |
| MouseLeave | وقتی روی می دهد که ماوس کنترل را ترک کند. |
| MouseMove | وقتی روی می دهد که موس بر روی کنترل حرکت کند. |
| MouseUp | وقتی روی می دهد که ماوس بر روی کنترل قرار دارد و دکمه آن رها می شود. |
| MouseWheel | وقتی روی می دهد که دکمه وسط ماوس که با شکل چرخ است به سمت بالا یا پایین حرکت کند. |

حال اجازه بدهید با برخی از این رویدادها بیشتر آشنا شویم.

یک فرم  ویندوزی جدید ایجاد کرده و یک کنترل دکمه (button) را در وسط آن قرار دهید (شکل زیر).

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/mouse-events-01.jpg)

دو کنترل کننده رویداد را به رویدادهای MouseEnter و MouseLeave دکمه اضافه کنید.

می خواهیم کاری کنیم که با ورود ماوس به محدوده کنترل دکمه، اندازه دکمه بزرگ شده و با دور شدن از آن اندازه دکمه به حالت اول برگردد.

کنترل دکمه را انتخاب کرده و سپس به پنجره Properties رفته و بر روی آیکون رعد و برق که پنجره رویدادهای کنترل مذکور است کلیک می کنیم.

در پنجره رویدادها، رویداد MouseEnter را پیدا کرده و بر روی آن دو بار کلیک می کنیم تا کنترل کننده رویدادی برای آن ایجاد شود. کد زیر را وارد می کنیم.

private void button1\_MouseEnter(object sender, EventArgs e)

{

button1.Height += 30;

button1.Width += 30;

button1.Top -= 15;

button1.Left -= 15;

}

وقتی که ماوس وارد کنترل می شود طول و عرض کنترل به اندازه ۳۰ پیکسل افزایش می یابد.

در این حالت برنامه را اجرا کنید و در حالت اجرا چندین بار با ماوس بر روی کنترل بروید مشاهده می کنید که کنترل فقط بزرگ و بزرگتر می شود و به حالت اول بر نمی گردد.

اگر فقط خطوط ۳ و ۴ را درج می کردیم  دکمه با هر بار تغییر اندازه می دهد دیگر در وسط فرم قرار نمی گرفت ولی اضافه کردن خطوط ۵ و ۶ باعث می شود که این کنترل با هر بار تغییر اندازه در وسط فرم قرار بگیرد.

حال نوبت آن رسیده که با خروج ماوس از کنترل اندازه آن به حالت اول برگردد.

در پنجره رویدادها، رویداد MouseLeave را پیدا کرده و بر روی آن دو بار کلیک می کنیم تا کنترل کننده رویدادی برای آن ایجاد شود. کد زیر را وارد می کنیم.

private void button1\_MouseLeave(object sender, EventArgs e)

{

button1.Height -= 30;

button1.Width -= 30;

button1.Top += 15;

button1.Left += 15;

}

کد بالا برعکس رویداد MouseEnter عمل می کند.

رویداد MouseClick نسخه بهبود یافته رویداد Click است.

این رویداد به شما اجازه می دهد جزییاتی در مورد رویداد click از جمله مکان کنترلی که با ماوس بر روی آن کلیک شده است را به دست آورید.

اجازه بدهید که یک کنترل کننده رویداد به رویداد MouseClick کنترل دکمه اضافه کنیم.

ابتدا از پنجره Properties به قسمت رویدادها رفته و رویداد MouseClick پیدا کنید و بر روی آن دو بار کلیک کنید.

با استفاده از کد زیر طول و عرض مکانی که شما بر روی آن کلیک می کنید نسبت به کوشه سمت چپ بالای کنترل در اختیار شما قرار داده می شود :

private void button1\_MouseClick(object sender, MouseEventArgs e)

{

MessageBox.Show(String.Format("Clicked at point ({0}, {1})", e.X, e.Y));

}

کنترل کننده رویداد با استفاده از شی MouseEventArgs به مختصات X و Y نقطه ای که شما بر روی آن کلیک کرده اید دست می یابد.

حال با هر بار کلیک بر رو دکمه یک پیغام که نشان دهنده مختصات مکانی است که کلیک کرده اید به شما نشان داده می شود.

رویداد MouseDown زمانی اتفاق می افتد که دکمه ماوس بر روی کنترل فشار داده شود.

رویداد MouseUp زمانی اتفاق می افتد که دکمه ماوس که بر روی کنترل فشار داده شده رها شود.

برای روشن شدن عملکرد این رویداد یک کنترل کننده رویداد به رویداد MouseDown اضافه می کنیم.

فقط به یاد داشته باشید که رویداد MouseClick با رویداد MouseDown تداخل دارد پس رویداد MouseClick را یا حذف کنید و یا به حالت توضیحات در آورید.

کد زیر را به کنترل کننده رویداد MouseDown اضافه کنید :

private void button1\_MouseDown(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

MessageBox.Show("Left Click");

else if (e.Button == MouseButtons.Right)

MessageBox.Show("Right Click");

else

MessageBox.Show("Middle Click");

}

کد بالا از شی MouseEventArgs برای دسترسی به خاصیت های کنترل Button استفاده می کند و شامل جزییاتی است که نشان می دهد کاربر بر روی کدام دکمه ماوس کلیک کرده است، راست یا چپ ؟

نوع شمارشی MouseButton دارای سه مقدار Left،  Right و Middle که نشان دهنده سه دکمه ماوس هستند می باشد.

از رویداد MouseWheel برای تشخیص اینکه آیا دکمه وسط ماوس (که غالبا با شکل یک چرخ با سمت بالا و پایین حرکت می کند ) حرکت کرده است یا نه استفاده می شود.

این رویداد در پنجره Properties موجود نیست و برای دسترسی به آن باید کد نوشت.

در سازنده فرم، درست زیر متد InitializeComponent کد زیر را تایپ کنید.

دقیقا بعد از تایپ += دو بار کلید tab را فشار دهید تا کنترل کننده رویداد مربوط به MouseWheel  برای شما تولید شود.

public Form1()

{

InitializeComponent();

button1.MouseWheel += newMouseEventHandler(button1\_MouseWheel);

}

کد زیر را نیز برای رویداد MouseWheel بنویسید :

void button1\_MouseWheel(object sender, MouseEventArgs e)

{

button1.Height += e.Delta / 60;

button1.Width += e.Delta / 60;

button1.Top -= e.Delta / 120;

button1.Left -= e.Delta / 120;

}

در کد بالا از خاصیت Delta از کلاس MouseEventArgs استفاده کرده ایم.

فرض کنید که کلیک وسط ماوس دارای چرخ دنده است که به سمت عقب و جلو حرکت می کنند و مقدار هر یک از آنها برابر با ثابت WHEEL\_DATA و ۱۲۰ است .

ما مقدار Delta را بر ۶۰ تقسیم کرده یام بنابراین با هر بار حرکت دکمه وسط ماوس، طول و عرص کنترل به جای ۱۲۰ پیکسل، ۲ پیکسل افزایش می یابد. با حرکت دکمه ماوس به بالا مقادیر مثبت و با حرکت به پایین آن مقادیر منفی اعمال می شوند. دو خط آخر هم برای این است که مکان دکمه در وسط صفحه حفظ شود. در این دو خط مقدار Delta بر ۱۲۰ تقسیم شده است در نتیجه با هر بار چرخش  چرخ دنده ها ، قسمت بالا و چپ دکمه ۱ پیکسل حرکت می کند.

برنامه را اجرا کنید، از آنجاییکه فوکوس بر روی کنترل button است با حرکت دکمه وسط ماوس به جول اندازه کنترل بزرگ و با حرکت آن به سمت عقب اندازه آن کوچک می شود.

کل کد :

using System;

using System.Windows.Forms;

namespace WindowsFormsApplication4

{

public partial classForm1 : Form

{

public Form1()

{

InitializeComponent();

button1.MouseWheel+=new MouseEventHandler(button1\_MouseWheel);

}

void button1\_MouseWheel(object sender, MouseEventArgs e)

{

button1.Height += e.Delta / 60;

button1.Width += e.Delta / 60;

button1.Top -= e.Delta / 120;

button1.Left -= e.Delta / 120;

}

private voidbutton1\_MouseEnter(object sender, EventArgs e)

{

button1.Height += 30;

button1.Width += 30;

button1.Top -= 15;

button1.Left -= 15;

}

private voidbutton1\_MouseLeave(object sender, EventArgs e)

{

button1.Height -= 30;

button1.Width -= 30;

button1.Top += 15;

button1.Left += 15;

}

private void button1\_MouseDown(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

MessageBox.Show("Left Click");

else if (e.Button == MouseButtons.Right)

MessageBox.Show("Right Click");

else

MessageBox.Show("Middle Click");

}

private void button1\_MouseClick(object sender, MouseEventArgs e)

{

MessageBox.Show(String.Format("Clicked at point ({0}, {1})", e.X, e.Y));

}

}

}

اگر بخواهید یکی از رویدادهای مربوط به فشرده شده دکمه های کیبورد را کنترل کنید باید رویدادهای KeyPress، KeyDown و KeyUp را کنترل کنید. رویداد KeyDown زمانی اتفاق می افتد که یکی از دکمه های کیبورد به سمت پایین فشار داده شود و رویداد KeyUp زمانی اتفاق می افتد که دکمه فشرده شده رها می شود.

رویداد KeyPress تلفیقی از دو رویداد بالا است.

به عنوان مثال یک رویداد KeyPress را به فرم اضافه می کنیم به طوریکه هر زمانی که یکی از دکمه های کیبورد فشرده شد مقدار آن با خاصیت Text کنترل Lable اضافه شود.

یک برنامه ویندوزی جدید ایجاد کنید و نام آن را KeyBoardEvents بگذارید.سپس یک کنترل Label به آن اضافه نمایید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/keyboard-events-01.jpg)

خاصیت Text کنترل Lable را پاک کنید. بر روی کنترل form کلیک کرده و از پنجره Properties رویداد KeyPress را یافته و بر روی آن دو بار کلیک کنید تا یک کنترل کننده رویداد ایجاد شود.

در داخل کنترل کننده رویداد Form1\_KeyPress کد زیر را وارد کنید:

۱:using System;

2:using System.Windows.Forms;

3:

4:namespace KeyBoardEvents

5: {

6:public partial classForm1 : Form

7: {

8://Variable to count letters for simple word wrap

9:private int charCount = 0;

10:

11:public Form1()

12: {

13: InitializeComponent();

14: }

15:

16:private void Form1\_KeyPress(object sender, KeyPressEventArgs e)

17: {

18: charCount++;

19:

20://Go to next line after the line's 30th character

21:if (charCount > 30)

22: {

23: label1.Text += "\r\n";

24: charCount = 0;

25: }

26:else

27: {

28://Append the pressed keyboard key to the label using KeyChar property

29: label1.Text += e.KeyChar;

30: }

31: }

32: }

33: }

رویداد KeyPress هر وقت که یکی از دکمه های کیبورد فشرده شود اتفاق می افتد.

در خط ۱۳ یک متغیر به نام charCount تعریف و مقدار دهی اولیه شده است و از آن برای تشخیص تعداد کاراکترهای خط جاری و رفتن به خط بعد استفاده می شود.

کنترل کننده رویداد تعداد کاراکترهای تایپ شده را به وسیله افزایش افزایش یک واحدی charCount در خط ۱۷ تشخیص می دهد.

در خط ۲۱ تست می شود که آیا مقدار charCount از ۳۰ بیشتر است یا نه؟

اگر بیشتر بود، یک خط جدید با استفاده از دستور “\r\n” ایجاد و بقیه کاراکترها تایپ شده در آن نمایش داده می شوند.

سپس در خط ۲۳ بار دیگر مقدار charCount را به دلیل اینکه در ابتدای یک خط جدید هستیم، صفر می کنیم .

اگر مقدار charCount کوچکتر یا مساوی ۳۰ باشد، به سادگی کاراکتر تایپ شده توسط کاربر با استفاده از خاصیت KeyPressEventArgs.KeyChar به متن اضافه می شود.

حال برنامه را اجرا کنید. مشاهده می کنید که با زدن هر دکمه کیبورد، متن کنترل lable تغییر کرده و کاراکترهایی را که شما وارد می کنید به آن اضافه می شود.  وقتی که رویدادهای KeyDown و KeyUp را کنترل می کنید، می توانید از آرگومان KeyEventArgs که شامل خاصیت های زیادی درباره دکمه فشرده شده است استفاده نمایید.

خواص این آرگومان به شرح زیر است :

| خاصیت | توضیح |
| --- | --- |
| Alt | تشخیص می دهد که آیا دکمه Alt فشرده شده است یا نه؟ |
| Control | تشخیص می دهد که آیا دکمه Control  فشرده شده است یا نه؟ |
| KeyCode | کد اسکی دکمه فشرده شده را می گیرد. و با استفاده از آن تشخیص می دهد که دکمه خاصی فشرده شده است یا نه؟ |
| KeyData | بسیار شبیه به خاصیت KeyCode است با این تفاوت که فشرده شدن دکمه های ترکیبی را نیز تشخیص می دهد. |
| KeyValue | مقدار عددی دکمه فشرده شده را بر می گرداند. |
| Modifier | تشخیص می دهد که کدام دکمه های ترکیبی  (SHIFT, CTRL, ALT) فشرده شده اند؟ |
| Shift | می گوید که آیا دکمه Shift  فشرده شده است یا نه؟ |
| SuppressKeyPress | به شما اجازه می دهد که از ورود اطلاعات از طریق کیبورد توسط کاربر جلوگیری کنید. |

به عنوان مثال در کد زیر از خاصیت SupressKeyPress استفاده شده است که در آن فقط اجازه ورود اعداد به کاربر داده شده است (کاربر نمی تواند هیچ کاراکتر و یا یا نشانه ای را وارد کند).

یک کنترل text box را به فرمتان اضافه کرده و یک کنترل کننده رویداد را به رویداد KeyDown اضافه نمایید.

private void textBox1\_KeyDown(object sender, KeyEventArgs e)

{

if (!(e.KeyCode >= Keys.D0 && e.KeyCode <= Keys.D9 && !e.Shift))

{

e.SuppressKeyPress = true;

}

}

شرط داخل دستور if می گوید که اگر دکمه تایپ شده توسط کاربر یک دکمه غیر عددی بود یا کلید shift فشرده شد، توسط خاصیت SuppressKeyPress نادیده گرفته شود. خاصیت KeyCode که از آن در شرط استفاده کرده ایم شامل مقادیری از نوع شمارشی Keys است. دکمه های عددی به وسیله مقایر D0 تا D9 نمایش داده شده اند.

در ضمن لازم است که چک کنیم که  آیا دکمه Shift فشرده شده است یا نه؟ از آنجاییکه نتیجه فشردن دکمه Shift همرا با یک عدد ممکن است منجر به تولید یک علامت مثلا @ (ترکیب دکمه Shift و عدد ۲ ) شود بنابراین باید از خاصیت Shift کلاس KeyEventArgs استفاده کنیم.

## رویدادهای کیبورد

اگر بخواهید یکی از رویدادهای مربوط به فشرده شده دکمه های کیبورد را کنترل کنید باید رویدادهای KeyPress، KeyDown و KeyUp را کنترل کنید. رویداد KeyDown زمانی اتفاق می افتد که یکی از دکمه های کیبورد به سمت پایین فشار داده شود و رویداد KeyUp زمانی اتفاق می افتد که دکمه فشرده شده رها می شود.

رویداد KeyPress تلفیقی از دو رویداد بالا است.

به عنوان مثال یک رویداد KeyPress را به فرم اضافه می کنیم به طوریکه هر زمانی که یکی از دکمه های کیبورد فشرده شد مقدار آن با خاصیت Text کنترل Lable اضافه شود.

یک برنامه ویندوزی جدید ایجاد کنید و نام آن را KeyBoardEvents بگذارید.سپس یک کنترل Label به آن اضافه نمایید.

[](http://www.w3-farsi.com/wp-content/uploads/2013/10/keyboard-events-01.jpg)

خاصیت Text کنترل Lable را پاک کنید. بر روی کنترل form کلیک کرده و از پنجره Properties رویداد KeyPress را یافته و بر روی آن دو بار کلیک کنید تا یک کنترل کننده رویداد ایجاد شود.

در داخل کنترل کننده رویداد Form1\_KeyPress کد زیر را وارد کنید:

۱:using System;

2:using System.Windows.Forms;

3:

4:namespace KeyBoardEvents

5: {

6:public partial classForm1 : Form

7: {

8://Variable to count letters for simple word wrap

9:private int charCount = 0;

10:

11:public Form1()

12: {

13: InitializeComponent();

14: }

15:

16:private void Form1\_KeyPress(object sender, KeyPressEventArgs e)

17: {

18: charCount++;

19:

20://Go to next line after the line's 30th character

21:if (charCount > 30)

22: {

23: label1.Text += "\r\n";

24: charCount = 0;

25: }

26:else

27: {

28://Append the pressed keyboard key to the label using KeyChar property

29: label1.Text += e.KeyChar;

30: }

31: }

32: }

33: }

رویداد KeyPress هر وقت که یکی از دکمه های کیبورد فشرده شود اتفاق می افتد.

در خط ۱۳ یک متغیر به نام charCount تعریف و مقدار دهی اولیه شده است و از آن برای تشخیص تعداد کاراکترهای خط جاری و رفتن به خط بعد استفاده می شود.

کنترل کننده رویداد تعداد کاراکترهای تایپ شده را به وسیله افزایش افزایش یک واحدی charCount در خط ۱۷ تشخیص می دهد.

در خط ۲۱ تست می شود که آیا مقدار charCount از ۳۰ بیشتر است یا نه؟

اگر بیشتر بود، یک خط جدید با استفاده از دستور “\r\n” ایجاد و بقیه کاراکترها تایپ شده در آن نمایش داده می شوند.

سپس در خط ۲۳ بار دیگر مقدار charCount را به دلیل اینکه در ابتدای یک خط جدید هستیم، صفر می کنیم .

اگر مقدار charCount کوچکتر یا مساوی ۳۰ باشد، به سادگی کاراکتر تایپ شده توسط کاربر با استفاده از خاصیت KeyPressEventArgs.KeyChar به متن اضافه می شود.

حال برنامه را اجرا کنید. مشاهده می کنید که با زدن هر دکمه کیبورد، متن کنترل lable تغییر کرده و کاراکترهایی را که شما وارد می کنید به آن اضافه می شود.  وقتی که رویدادهای KeyDown و KeyUp را کنترل می کنید، می توانید از آرگومان KeyEventArgs که شامل خاصیت های زیادی درباره دکمه فشرده شده است استفاده نمایید.

خواص این آرگومان به شرح زیر است :

| خاصیت | توضیح |
| --- | --- |
| Alt | تشخیص می دهد که آیا دکمه Alt فشرده شده است یا نه؟ |
| Control | تشخیص می دهد که آیا دکمه Control  فشرده شده است یا نه؟ |
| KeyCode | کد اسکی دکمه فشرده شده را می گیرد. و با استفاده از آن تشخیص می دهد که دکمه خاصی فشرده شده است یا نه؟ |
| KeyData | بسیار شبیه به خاصیت KeyCode است با این تفاوت که فشرده شدن دکمه های ترکیبی را نیز تشخیص می دهد. |
| KeyValue | مقدار عددی دکمه فشرده شده را بر می گرداند. |
| Modifier | تشخیص می دهد که کدام دکمه های ترکیبی  (SHIFT, CTRL, ALT) فشرده شده اند؟ |
| Shift | می گوید که آیا دکمه Shift  فشرده شده است یا نه؟ |
| SuppressKeyPress | به شما اجازه می دهد که از ورود اطلاعات از طریق کیبورد توسط کاربر جلوگیری کنید. |

به عنوان مثال در کد زیر از خاصیت SupressKeyPress استفاده شده است که در آن فقط اجازه ورود اعداد به کاربر داده شده است (کاربر نمی تواند هیچ کاراکتر و یا یا نشانه ای را وارد کند).

یک کنترل text box را به فرمتان اضافه کرده و یک کنترل کننده رویداد را به رویداد KeyDown اضافه نمایید.

private void textBox1\_KeyDown(object sender, KeyEventArgs e)

{

if (!(e.KeyCode >= Keys.D0 && e.KeyCode <= Keys.D9 && !e.Shift))

{

e.SuppressKeyPress = true;

}

}

شرط داخل دستور if می گوید که اگر دکمه تایپ شده توسط کاربر یک دکمه غیر عددی بود یا کلید shift فشرده شد، توسط خاصیت SuppressKeyPress نادیده گرفته شود. خاصیت KeyCode که از آن در شرط استفاده کرده ایم شامل مقادیری از نوع شمارشی Keys است. دکمه های عددی به وسیله مقایر D0 تا D9 نمایش داده شده اند.

در ضمن لازم است که چک کنیم که  آیا دکمه Shift فشرده شده است یا نه؟ از آنجاییکه نتیجه فشردن دکمه Shift همرا با یک عدد ممکن است منجر به تولید یک علامت مثلا @ (ترکیب دکمه Shift و عدد ۲ ) شود بنابراین باید از خاصیت Shift کلاس KeyEventArgs استفاده کنیم.