

## فصل چهارم : کنترل روند اجرای برنامه

## • دستور شرطی if

شکل کلی استفاده از این دستور به صورت زیر است :

```

(شرط ۱) if
{
مجموعه دستور اول
}
(شرط ۲) else if
{
مجموعه دستور دوم
}
.....
else
{
مجموعه دستور آخر
}

```

مثال ۱: برنامه ای ایجاد کنید که معدل یک دانشجو را بگیرد و رتبه ی وی را چاپ کند.

رتبه	معدل دانشجو
<b>E</b>	زیر ۱۰
<b>D</b>	بین ۱۰ تا ۱۲
<b>C</b>	بین ۱۲ تا ۱۵
<b>B</b>	بین ۱۵ تا ۱۷
<b>A</b>	بین ۱۷ تا ۲۰

با دو روش حل کنید: یکبار بدون استفاده از عملگرهای && و || و یک بار هم با استفاده از این عملگرها درون شرط های if.

## مقایسه ی رشته ها

معمولا برای مقایسه ی رشته ها با استفاده از دستور if به مشکل برخورد خواهید خورد، چراکه دستور if نسبت به بزرگ یا کوچک بودن حروف حساس است.

اگر شما بخواهید بزرگ یا کوچک بودن حروف را نادیده بگیرید، باید از تابع Compare که زیر مجموعه ی توابع String می باشد، استفاده کنید.

شکل کلی استفاده از این تابع به صورت زیر است :

```
int string.Compare(string strA, string strB, bool ignoreCase);
```

همانطور که از تعریف فوق پیداست، تابع `String.Compare()` تابعی است که یک خروجی از نوع `int` دارد. از طرفی این تابع دارای سه ورودی است. ورودی اول و دوم دو متغیر رشته ای هستند که قرار است محتوای آنها با یکدیگر مقایسه شود. ورودی سوم مقداری منطقی یا بولی است که می تواند `true` یا `false` باشد. همانطور که از نام این عنصر پیداست `ignoreCase` (نادیده گرفتن حالت حروف)، اگر مقدار آن را `true` قرار دهیم، حالت حروف یا بزرگ و کوچکی حروف را نادیده می گیرد و اگر مقدار آن را `false` قرار دهیم، نادیده نمی گیرد.

### تفسیر خروجی تابع:

همانطور که گفتیم خروجی این تابع یک عدد صحیح است که این عدد از تفریق دو رشته ی مورد مقایسه از یکدیگر به دست آمده. به این ترتیب :

- اگر خروجی تابع برابر با **صفر** شد، یعنی دو رشته با یکدیگر **برابرند**،
- اگر خروجی تابع، عددی **مثبت** شد، یعنی رشته ی اول از رشته ی دوم بزرگتر است،
- اگر خروجی تابع، عددی **منفی** شد، یعنی رشته ی اول از رشته ی دوم کوچکتر است.

**مثال :** یک پنجره ی `Login` بسازید که شامل دو کنترل جعبه متن، یکی برای نام کاربری و دیگری برای رمز ورود و دو دکمه برای `Login` کردن باشد. یک دکمه را حساس به بزرگ و کوچک بودن حروف قرار دهید و دیگری را غیرحساس. (`userName="Mahdi" , Password="57"`)

```
private void button1_Click(object sender, EventArgs e)
{
    string un, pass;
    un = textBox1.Text;
    pass = textBox2.Text;
    if (string.Compare(un, "mahdi", true) == 0 && pass == "57")
        MessageBox.Show("wellcome");
    else
        MessageBox.Show("invalid un or pass");
}

private void button2_Click(object sender, EventArgs e)
{
    string un, pass;
    un = textBox1.Text;
    pass = textBox2.Text;
    if (string.Compare(un, "mahdi", false) == 0 && pass == "57")
        MessageBox.Show("wellcome");
    else
        MessageBox.Show("invalid un or pass");
}
```

## معرفی کنترل **ToolTip** :

با استفاده از این ابزار، میتوان توضیحاتی را به بخش های مختلف برنامه اضافه نمود. روش کار به این صورت است که یک نمونه از این ابزار را به فرم اضافه کرده و سپس در بخش صفات مربوط به سایر ابزارها، صفتی به نام **ToolTip on Tooltip1** اضافه میشود که میتوان متن مورد نظر را در این بخش قرار داد.

## معرفی کنترل جعبه لیست یا **ListBox**

برای نگهداری لیستی از اشیاء از این ابزار استفاده میشود.

### چند خاصیت مهم از لیست باکس:

**name** : معمولاً با سه حرف **lst** شروع میشود. مثلاً **lstStudent**.

**BackColor** , **ForeColor** , **Enabled** , **Font** , **Visible** , **RightToLeft** , **TabIndex** و ... مانند قبل.

**Items** : آرایه ای است که شامل مواردیست که در لیست باکس نمایش داده میشوند. این خاصیت را در آینده با استفاده از دستورات برنامه نویسی تغییر خواهیم داد.

**Dock** : وضعیت های مختلف لیست باکس نسبت به صفحه برنامه را تعیین میکند.

**Sorted** : تعیین میکند که آیا موارد محتوی جعبه لیست مرتب باشند یا خیر.

**نکته بسیار مهم:** زمانی که کاربر روی هر کدام از آیتم ها (محتویات) لیست باکس کلیک میکند، اندیس عنصر انتخاب شده در خاصیت **SelectedIndex** ذخیره میشود که میتوان با استفاده از آن تشخیص داد که در حال حاضر کدام آیتم انتخاب شده است. مثلاً با استفاده از دستور زیر میتوان آن عنصر را چاپ کرد :

```
int x = listBox1.SelectedIndex;
MessageBox.Show( listBox1.Items[x].ToString() );
```

**نکته:** هر گاه بخواهیم عنصری از آرایه ی **items** را چاپ کنیم، باید ابتدا آن را با استفاده از **ToString()** به رشته تبدیل نماییم.

**معرفی تابع **Items.Add()** :** از این تابع برای اضافه کردن یک عنصر جدید به لیست استفاده میشود.

**معرفی تابع **Items.Clear()** :** این تابع تمام محتوی لیست باکس را حذف میکند.

**معرفی تابع **Remove** :** این تابع نام یک عنصر را گرفته و از لیست حذف میکند.

**معرفی تابع **RemoveAt** :** این تابع اندیس یک عنصر را گرفته و از لیست حذف میکند.

**مثال:** برنامه ای بنویسید که تمامی توابع فوق را درون آن بکار برده باشید.

## معرفی دستور Switch در سی شارپ :

از این دستور زمانی استفاده میشود که انتخابهای زیادی برای یک متغیر خاص داشته باشیم و استفاده از if خوانایی برنامه را کاهش دهد.

از طرفی دستور switch به خاطر ساختار داخلی خود، سرعت بسیار بالاتری از دستور if دارد. شکل کلی استفاده از آن به صورت زیر است :

```
switch ( variable )
{
case value1 :
    statement 1 ;
    break;
case value2 :
    statement 2 ;
    break;
....
default:
    statement n ;
    break;
}
```

در دستور فوق، variable همان متغیری است که قرار است نسبت به مقدار آن تصمیم گیری صورت گیرد.

value 1 تا value 2 و ... مقادیری هستند که اگر متغیر فوق، برابر با آنها باشد، statement 1 الی آخر متناظر با آنها اجرا می شوند.

اگر مقدار متغیر مورد بحث، برابر با هر کدام از مقادیر جلوی case ها باشد، دستورات داخل همان case اجرا شده و برنامه با استفاده از دستور Break، کلاً از دستور Switch خارج خواهد شد.

**نکته:** دستورات قسمت default در صورتی اجرا میشوند که هیچکدام از case های قبلی صادق نباشند و به عنوان بخشی اختیاری در این دستور استفاده میشود.

**مثال ۱:** در یک لیست باکس نام پنج خودرو را قرار دهید. به طوریکه وقتی کاربر روی نام هر خودرو کلیک میکند، نام شرکت سازنده ی آن خودرو در یک کادر پیغام نمایش داده شود.

```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    int x = listBox1.SelectedIndex;
    string car = listBox1.Items[x].ToString();
    string manufacture;
```

```
switch (car)
{
    case "پراید":
        manufacture = "سایپا";
        break;
    case "پیکان":
        manufacture = "خودرو ایران";
        break;
    case "زانتیا":
        manufacture = "سیتروئن";
        break;
    case "پارس":
        manufacture = "پژو";
        break;
    default:
        manufacture = "ناشناس";
        break;
}
MessageBox.Show(manufacture);
}
```