

**نکته:**

تعداد عناصر موجود در لیست باکس، در خصوصیت `listBox1.Items.Count` قرار میگیرند.

**حلقه ی تکرار for :**

برای مواقعی که تعداد تکرار دستورات از قبل مشخص است.

شکل کلی آن به صورت زیر است :

```
for ( گام حرکت ; شرط ادامه حلقه ; مقدار اولیه = شمارنده حلقه )
{
    دستورات تکرار شونده
}
```

تا زمانیکه شرط ادامه برقرار باشد، حلقه تکرار خواهد شد.

مثال : برنامه ای ایجاد کنید که در اثر کلیک روی یک دکمه، اعداد از ۱۰ تا ۲۰ در یک لیست نمایش داده شوند.

حل :

```
int i;
for ( i=10 ; i<=20 ; i++)
{
    lstNumber.Items.Add(i);
}
```

تغییر در گام حرکت :

مثال ۳ : برنامه ای ایجاد کنید که در اثر کلیک روی یک دکمه، اعداد مضارب ۳ دو رقمی در یک لیست باکس نمایش داده شوند.

روش اول:

```
for (i=10;i<=99;i++)
{
    if(i%3==0)
        lst.Items.Add(i);
}
```

روش دوم:

```
for (i=12;i<=99;i+=3)
{
    lst.Items.Add(i);
}
```

مثال ۴: برنامه ای ایجاد کنید که از دو جعبه متن، دو عدد گرفته و مضارب پنج بین این دو عدد را چاپ کند. (ابتدا بررسی کنید که کدام عدد بزرگتر و کدام کوچکتر است.)

```
int x, y, temp, i;
listBox1.Items.Clear();
x = int.Parse(textBox2.Text);
y = int.Parse(textBox3.Text);
//ebteda bayad tashkhis dahim kodam koochehtar ast..
// mikhahim koochehtar ra dar x gharar dahim
if (y < x)
{
    temp = x;
    x = y;
    y = temp;
}
// در حال حاضر مقدار کوچکتر در متغیر x قرار گرفته است
listBox1.Items.Add("+y) و "+x+" عدد دو بین 5 مضارب کردن چاپ");
for (i = x; i <= y; i++)
{
    if (i % 5 == 0)
        listBox1.Items.Add(i);
}
```

### حلقه ی for کاهششی:

میتوان حلقه for را به صورت کاهششی به صورت زیر به کار برد:

```
for (i=10 ; i>=0; i--)
{
    .....
}
```

### حلقه ی تکرار foreach :

معمولاً زمانی که مجموعه ای از اشیاء را در اختیار داریم و بخواهیم بین این اشیاء حرکت کرده و موضوعی را بررسی کنیم، از این نوع حلقه استفاده می کنیم. معمولاً اشیاء مورد نظر در یک آرایه ذخیره میشوند. مثال: فرض کنید که student آرایه ای از نامهای دانشجویان است، لیست اسامی آنها را در یک لیست باکس نمایش دهید:

```
foreach ( string s in student)
{
    lst.Items.Add(s);
}
```

مثال : برنامه ای بنویسید که تمام پوشه های موجود در درایو d کامپیوتر شما را در یک لیست باکس نمایش دهد.

```
foreach (string s in System.IO.Directory.GetDirectories("d:\\"))
{
    lst1.Items.Add(str);
}
```

مثال : برنامه ای بنویسید که در یک جعبه متن، از کاربر متنی را دریافت کرده و با فشردن یک دکمه، طول متن را به او نمایش دهد. این کار را با استفاده از متد انجام دهید.

```
private void button1_Click(object sender, EventArgs e)
{
    int a;
    string str;
    str = textBox1.Text;
    a = tooleMatn(str);
    MessageBox.Show(a + " کاراکتر");
}

int tooleMatn(string text)
{
    int x;
    x = text.Length;
    return x;
}
```

### معرفی تابع Split از کلاس String

این تابع یک کاراکتر را دریافت کرده و بر اساس آن، متن گرفته شده را تکه تکه میکند و هر تکه را در خانه ای از آرایه قرار میدهد. روش استفاده ی آن به صورت زیر است:

```
string [] Split("کاراکتر جداکننده");
```

نام متغیر رشته ای مورد نظر جهت تکه تکه شدن = نام آرایه ی جواب

مثال : برنامه ای بنویسید که در یک جعبه متن، از کاربر متنی را دریافت کرده و با فشردن یک دکمه، تعداد کلمه های متن را به او نمایش دهد. این کار را با استفاده از متد انجام دهید.

```
private void button2_Click(object sender, EventArgs e)
{
    string text;
    text = textBox1.Text;
    int a;
    a = CountWords(text);
    MessageBox.Show(a + " کلمه");
}
```

```
private int CountWords(string s)
{
    string[] words = s.Split(' ');
    int c = 0;
    foreach (string str in words)
    {
        if (str == "")
            c++;
    }
    // تعداد خانه های پوچ درون متغیر c قرار گرفته است
    return words.Length - c;
}
```

حلقه های while , do..while , do..until

تعریف کلاس Random:

کلاسی است که به ما امکان ایجاد یک عدد تصادفی را میدهد.

برای ایجاد یک عدد تصادفی، باید ابتدا یک شیء از نوع این کلاس تعریف کنیم.

روش کلی تعریف یک شیء از نوع یک کلاس به صورت زیر است:

() نام کلاس new = نام شیء نام کلاس

پس خواهیم داشت :

```
Random rdm = new Random();
```

حال شیء ساخته شده را برای ایجاد یک عدد تصادفی جدید فراخوانی میکنیم. به این منظور باید از تابعی به نام

تابع next استفاده نمود.

**مدلهای مختلف استفاده از تابع next :**

سه مدل میتوان از تابع next استفاده کرد :

۱- تولید یک عدد نامنفی `int x = rdm.next();`

۲- تولید عددی کوچکتر از max `int x=rdm.next(int max);`

۳- تولید عددی بزرگتر از min و کوچکتر از max `int x=rdm.next(int min , int max);`

تکلیف : برنامه ای ایجاد کنید که **تعدادی** عدد تصادفی (بین صفر تا ۵۵) تولید کرده و در یک لیست باکس قرار

دهد. آخرین عدد تولید شده ۵۰ می باشد.

حلقه while