

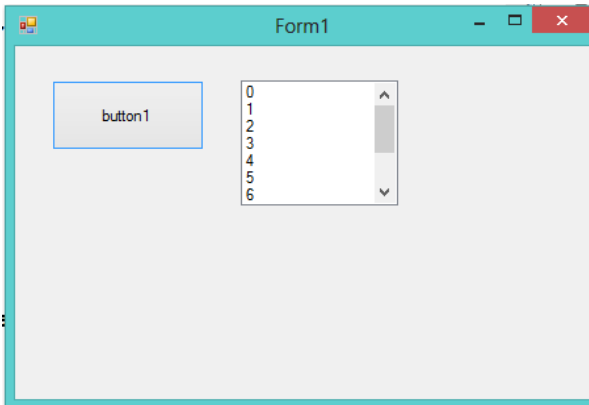
İNTERNET TABANLI PROGRAMLAMA

DÖNGÜLER (FOR, WHILE, DO-WHILE)

C# da döngüleri dört çeşit olarak sayabiliriz. Bunlar

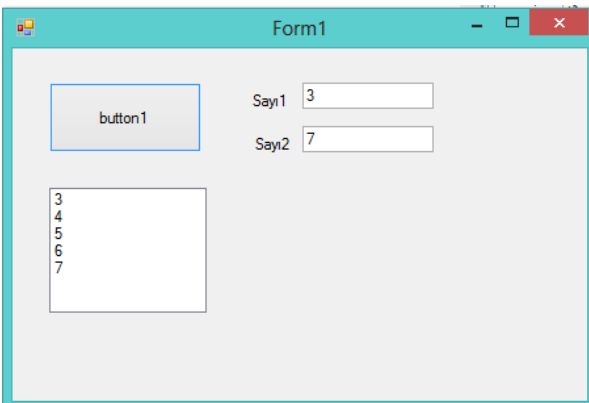
- For döngüsü:** Bu döngüde başlangıç ve bitiş sınırları verilerek kullanılır. Döngü tanım kısmında içinde bir de artan bir sayaç kullanılır.
- While Döngüsü:** Sonsuz döngüdür. Şart gerçekleştiği müddetçe döner. Döngüye daha girmeden şart başlangıçta kontrol edilir.
- Do-While Döngüsü:** Sonsuz döngüdür. Döngüye ilk girişte şart kontrol edilmez. Döngüden çıkarken şart kontrol edilir.
- For-Each Döngüsü:** Dizilerde kullanılan döngü tipidir. Dizi içinde kaç tane eleman varsa her biri için bir işlem yapacaksak bu döngü kullanılır. Bu döngü bir sonraki ders olan Diziler konusu içinde anlatılacaktır.

ÖRNEK



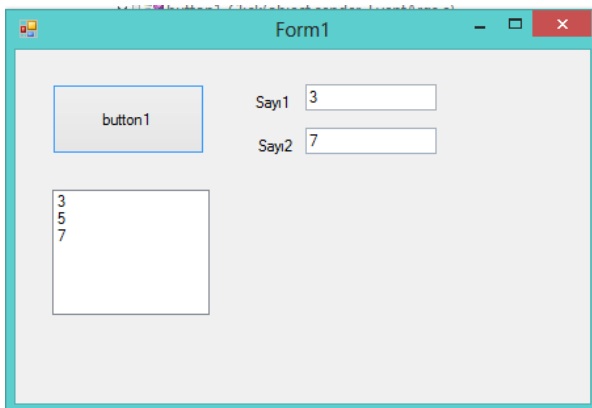
```
private void button1_Click(object sender, EventArgs e)
{
    for (int i = 0; i <= 10; i++)
    {
        listBox1.Items.Add(i.ToString());
    }
}
```

ÖRNEK



```
private void button1_Click(object sender, EventArgs e)
{
    int Sayi1, Sayi2;
    Sayi1 = Convert.ToInt32 (textBox1.Text);
    Sayi2 = Convert.ToInt32 (textBox2.Text);

    for (int i = Sayi1; i <= Sayi2; i++)
    {
        listBox1.Items.Add(i.ToString());
    }
}
```

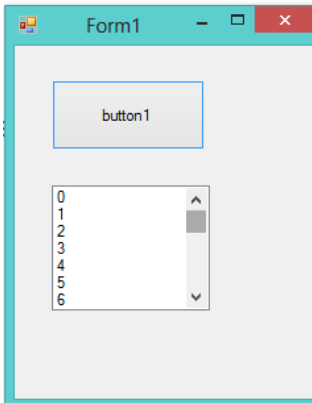
ÖRNEK

```
private void button1_Click(object sender, EventArgs e)
{
    int Sayi1, Sayi2;
    Sayi1 = Convert.ToInt32 (textBox1.Text);
    Sayi2 = Convert.ToInt32 (textBox2.Text);

    listBox1.Items.Clear();

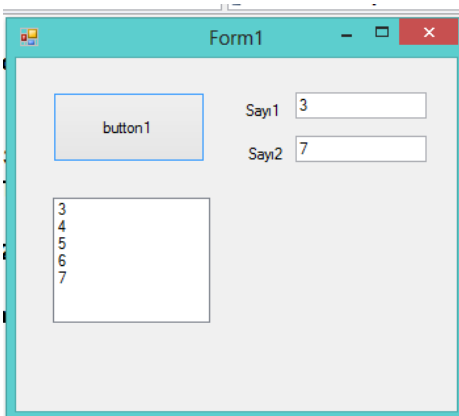
    for (int i = Sayi1; i <= Sayi2; i = i + 2) // i++ =>
i=i+1 , i=i+2
    {
        listBox1.Items.Add(i.ToString());
    }
}
```

ÖRNEK



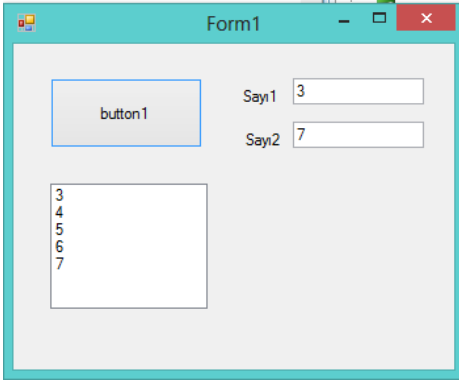
```
private void button1_Click(object sender, EventArgs e)
{
    int i=0;
    while (i <= 100)
    {
        listBox1.Items.Add(i.ToString());
        i++;
    }
}
```

ÖRNEK



```
private void button1_Click(object sender, EventArgs e)
{
    int i=0, Sayı2;
    i = Convert.ToInt32(textBox1.Text);
    Sayı2 = Convert.ToInt32(textBox2.Text);

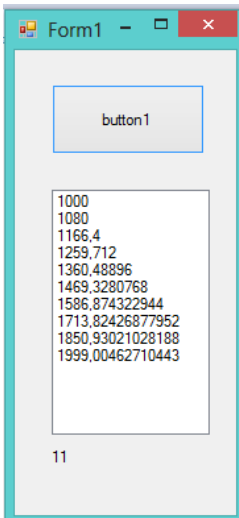
    while (i<=Sayı2)
    {
        listBox1.Items.Add(i.ToString());
        i++;
    }
}
```

ÖRNEK

```
private void button1_Click(object sender, EventArgs e)
{
    int i=0, Sayi2;
    i = Convert.ToInt32(textBox1.Text);
    Sayi2 = Convert.ToInt32(textBox2.Text);

    do
    {
        listBox1.Items.Add(i.ToString());
        i++;
    } while (i <= Sayi2);
}
```

NOT : Bir programı çalıştırırken değişkenlerin hangi değerleri aldığını görmek için F9 ile durmak istediğimiz yere bir çentik (durdurma işareti) atarız. Daha sonra F5 ile programı çalıştırdığımızda program oraya gelince duracaktır. Bu durumda mouse ile değişkenin üzerine geldiğimizde değişkenin içerisindeki değeri bize gösterecektir. Programı devam ettirmek için F11 ile adım adım devam edebiliriz. Yada sona kadar çalışmasını istersek F5 ile çalıştırabiliriz.

ÖRNEK:

```

private void button1_Click(object sender, EventArgs e)
{
    double maas = 1000;
    int yil=1;

    while (maas <=2000)
    {
        listBox1.Items.Add(maas.ToString());
        yil++;
        maas = maas + maas * 0.08;
    }
    label3.Text = yil.ToString();
}

```

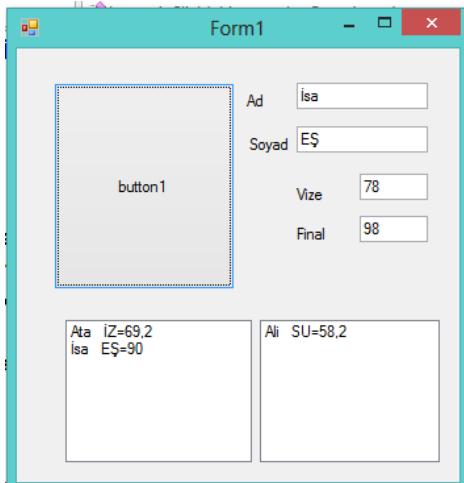
ÖRNEK

```

private void button1_Click(object sender, EventArgs e)
{
    string Ad, Soyad;
    Ad = textBox1.Text;
    Soyad = textBox2.Text;

    if (radioButton1.Checked == true)
    {
        listBox1.Items.Add(Ad + " " + Soyad);
    }
    else if (radioButton2.Checked == true)
    {
        listBox2.Items.Add(Ad + " " + Soyad);
    }
}

```

ÖRNEK

```

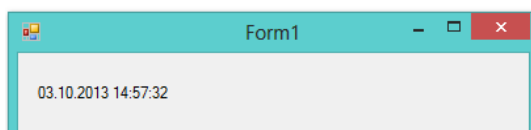
private void button1_Click(object sender, EventArgs e)
{
    string Ad, Soyad;
    Ad = textBox1.Text;
    Soyad = textBox2.Text;

    double Vize, Final, Ortalama;
    Vize = Convert.ToDouble(txtVize.Text);
    Final = Convert.ToDouble(txtFinal.Text);

    Ortalama = Vize * 0.4 + Final * 0.6;

    if (Ortalama >= 60)
    {
        listBox1.Items.Add(Ad + " " + Soyad + "=" +
Ortalama);
    }
    else
    {
        listBox2.Items.Add(Ad + " " + Soyad + "=" +
Ortalama);
    }
}

```

TARİH VE SAAT FONKSİYONLARI

```

private void Form1_Load(object sender, EventArgs e)
{

```

```
label7.Text = DateTime.Now.ToString();
```

```
}
```

Örnek Kodlar

Aşağıdaki kodları deneyin yeni tarih saat formatları ve kullanımları bulun bunları deneyin.

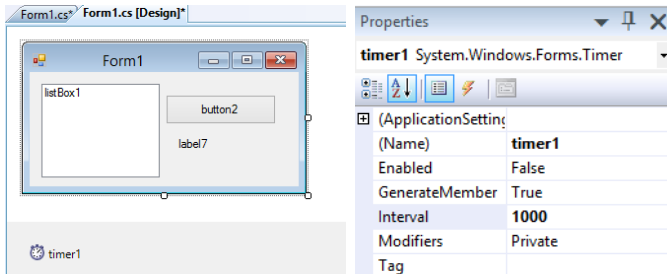
```
DateTime Tarih = new DateTime(2008, 8, 23);
string YeniFormat = String.Format("{0:d}", Tarih);
Label1.Text = YeniFormat;
```

```
DateTime YeniTarih = DateTime.Today.Add(new TimeSpan(3, 12, 24)); //Yeni
saat,dak,saniye.
```

```
string Dakika = DateTime.Now.Minute.ToString(); //Şimdiki zamanın daikası
string Saniye = DateTime.Now.Second.ToString(); //Şimdiki zamanın saniyesi
```

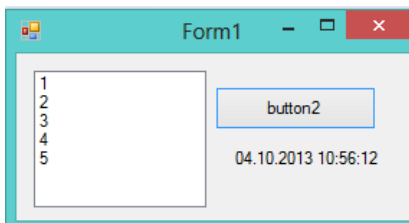
TIMER NESNESİ

Masaüstü program yazarken bilgisayarın belli aralıklar bir işi sürekli yapması istendiğinde bu nesne kullanılır. Nesnenin Properties penceresinden Interval özelliği süreyi belirler. Örneğin burada 1000 yazıyorsa her saniyede bir içerisindeki kodları çalıştıracak demektir. Bu nesne forma eklenirken arka planda kod olarak eklenir. Formün üzerinde gözükmez altındaki boş alanda kullanıcıya gösterilir.



Örnek:

Ekranda sistem saatini sürekli olarak her saniye göseren ve bu esnada bir listbox a geçen saniyeleri yazan bir program yazın.



```
int Sayac = 0; //global de tanımlanmalı

private void button2_Click(object sender, EventArgs e)
{
```

```
        timer1.Enabled = true;
    }

    private void timer1_Tick(object sender, EventArgs e)
    {

        label7.Text = DateTime.Now.ToString();

        Sayac++;
        listBox1.Items.Add(Sayac.ToString());

    }
```