



Microsoft®
Small Basic

Programlamaya Giriş

Programlama

- Bilgisayar programlaması, programlama dilleri kullanılarak, bilgisayar yazılımlarının oluşturulması sürecidir.



Programlama



- Tıpkı bizim Türkçe'yi veya İngilizce'yi konuşup anlamamız gibi, bilgisayarlar da belirli dillerde yazılmış programları anlayabilirler. Bunlar programlama dilleri olarak adlandırılır.



Small Basic ve Programlama

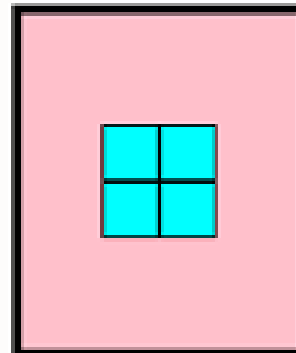
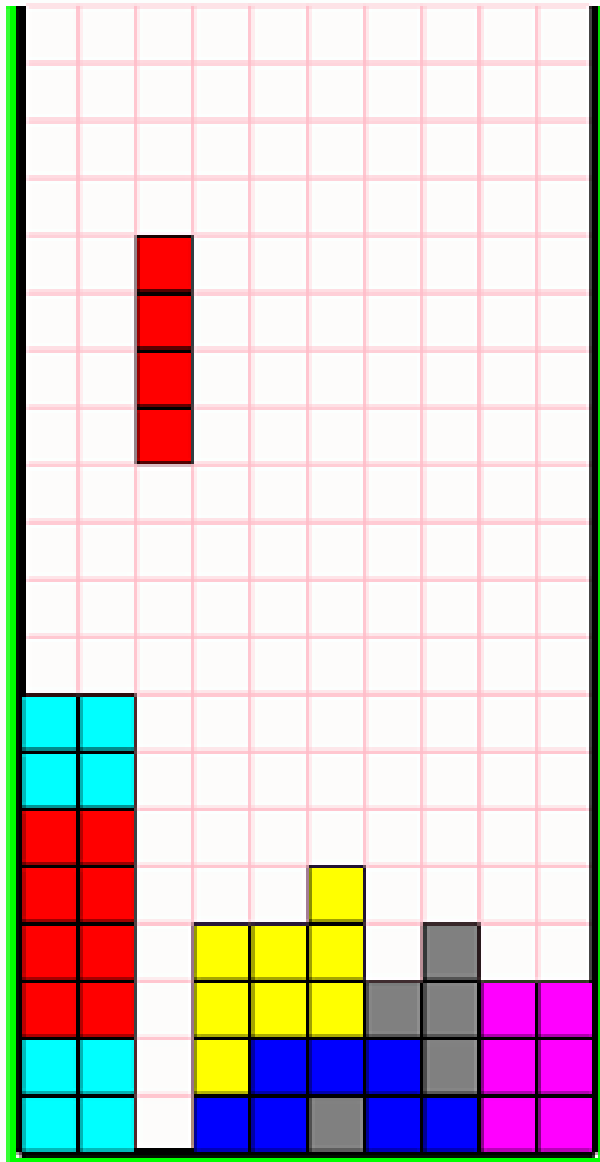
- ▶ Small Basic, programlamayı yeni başlayanlar için son derece kolay, anlaşılır ve eğlenceli hale getirmek üzere tasarlanmış olan bir programlama dilidir.



SMALL BASIC

Programlamayı yeni başlayanlar için kolay, anlaşılır ve eğlenceli hale getirmek üzere tasarlanmış olan bir programlama dilidir.





00000113

Game control keys:
Left Arrow = Move piece left
Right Arrow = Move piece right
Up Arrow = Rotate piece
Down Arrow = Drop piece

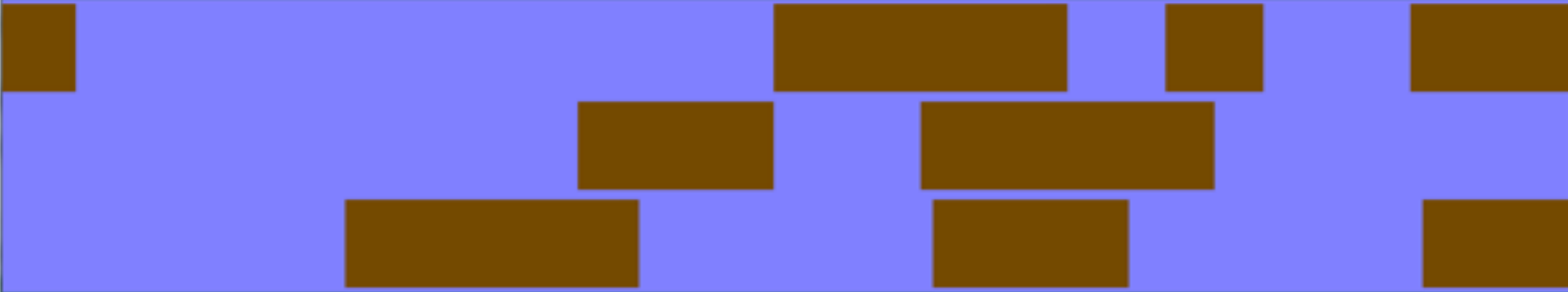
Press to stop game

Small Basic Tetris

ver.0.1

Score 7071

High Score 7071

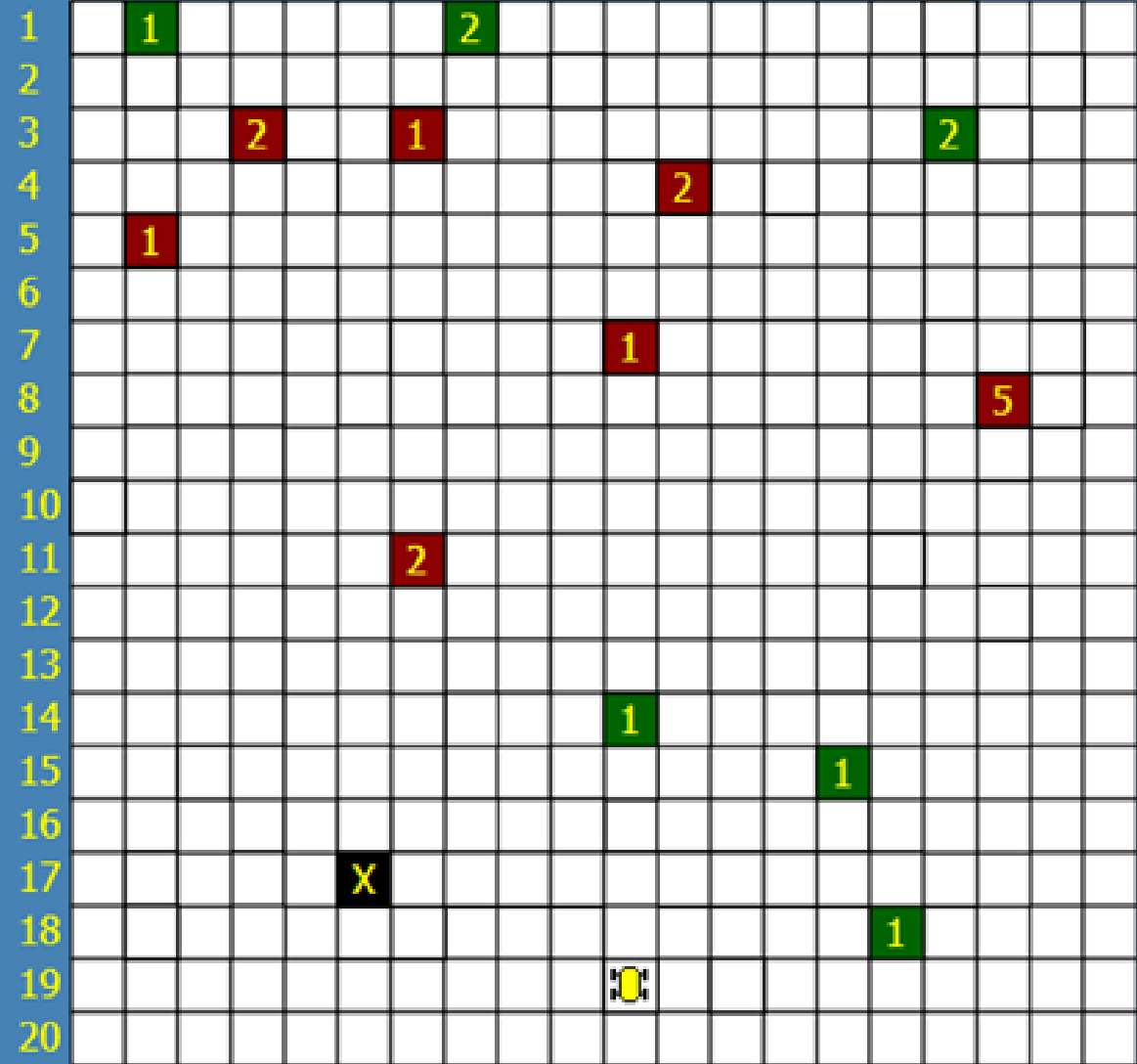


Pause Game

Stop Game

High Scores

A B C D E F G H I J K L M N O P Q R S T



Time	Sales
9:04	\$480.00

- 8:14 | 4 L -> 2 pizza(s)
- 8:20 | 8 R -> 5 pizza(s)
- 8:22 | 3 G -> 1 pizza(s)
- 8:24 | 3 D -> 2 pizza(s)
- 8:27 | 5 B -> 1 pizza(s)

Car at 19 K
No Pizza Wanted

Oven Off

Add Pizza Bake Pizza

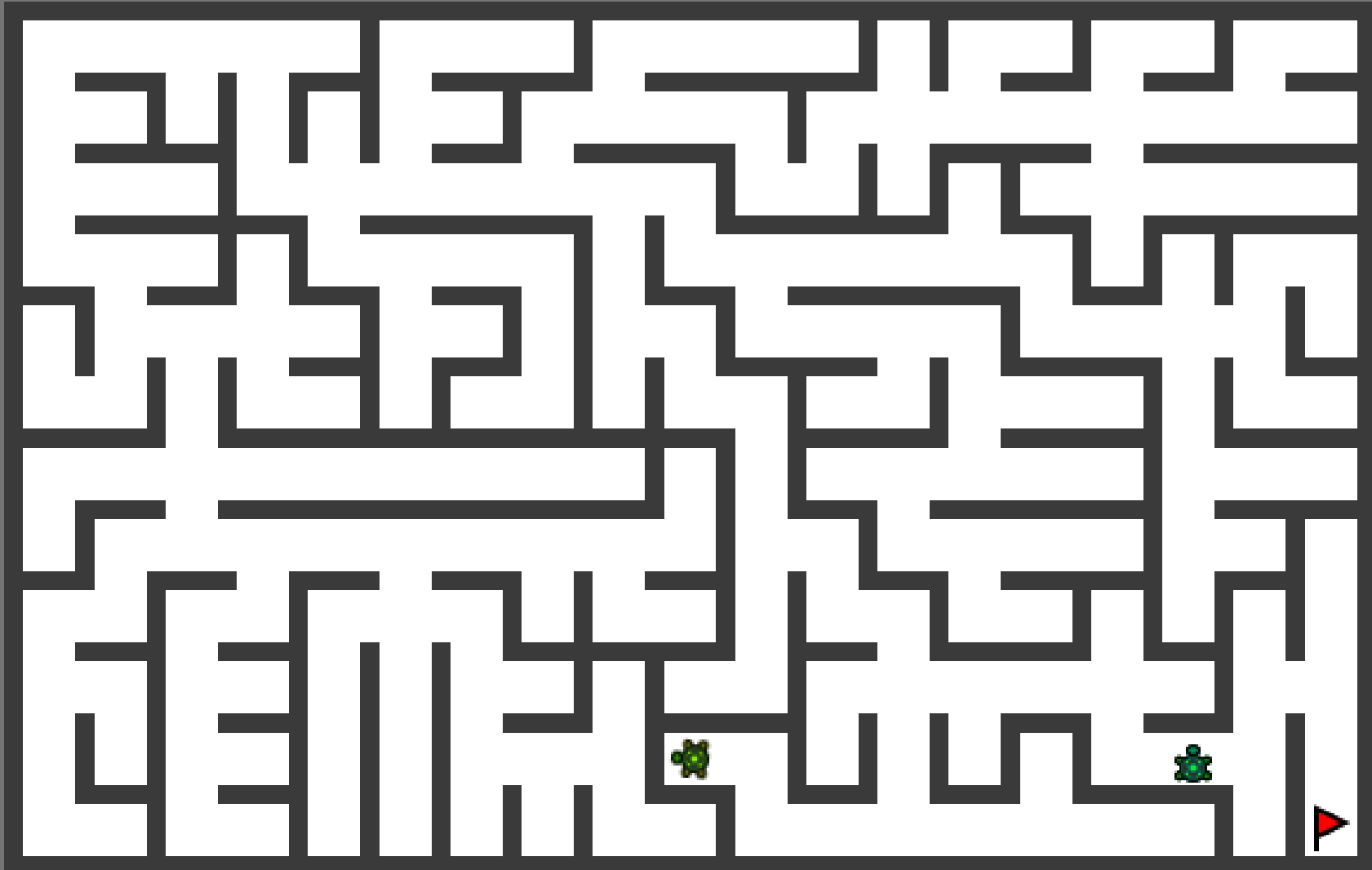
Baked	Pizzas in Car
0	8

Pause Game Stop Game

Maze Game 1.56

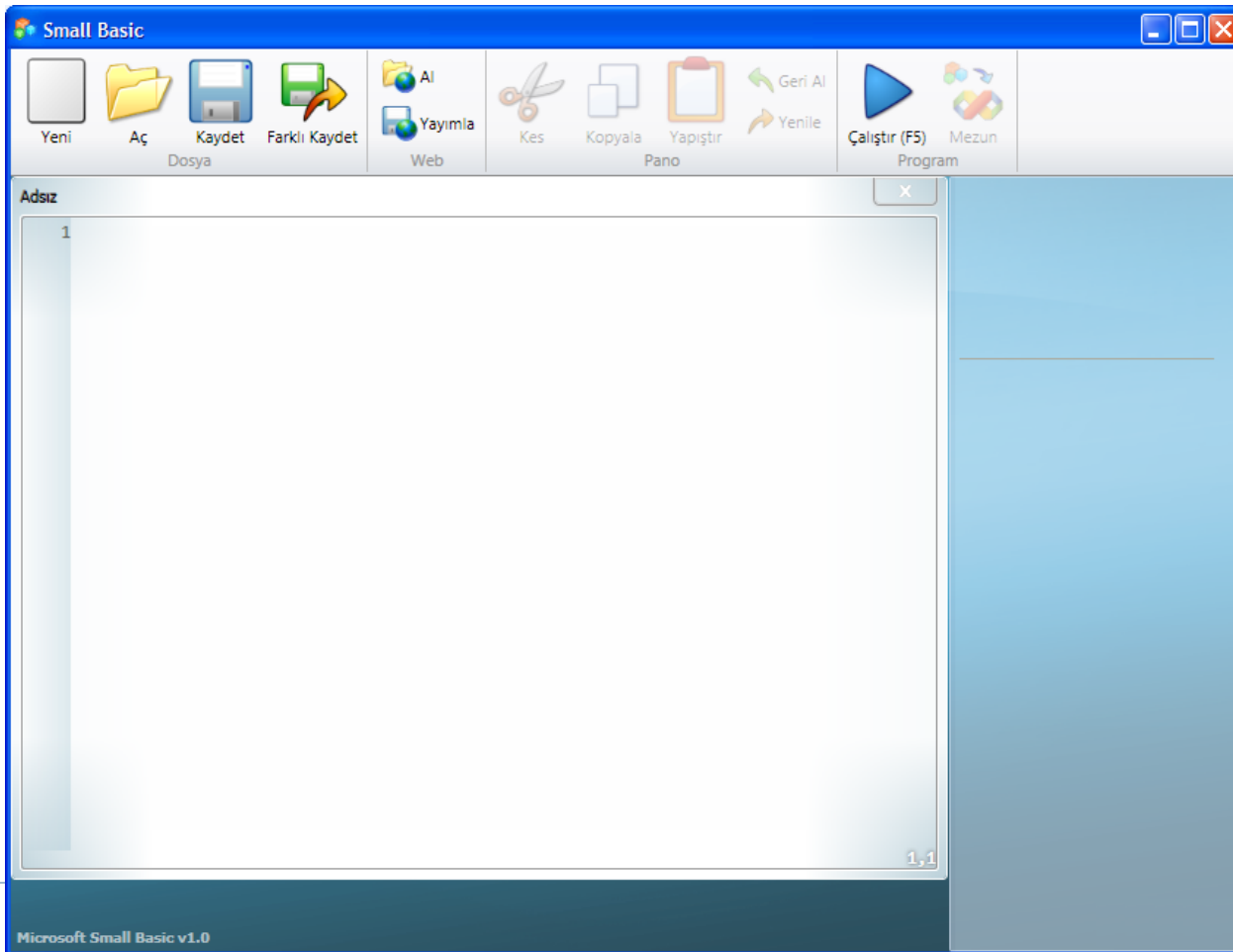


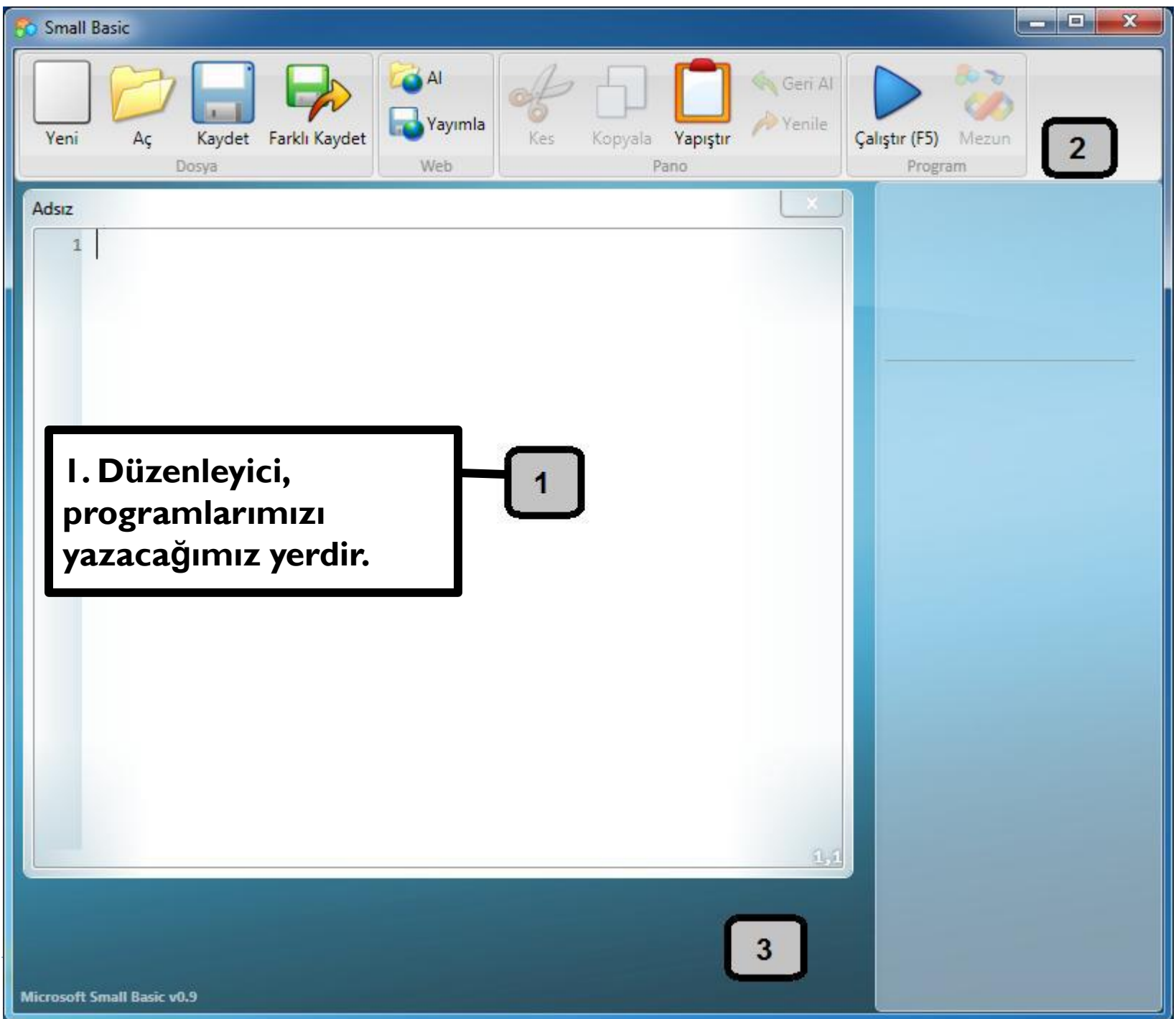
High Score: 9.292[s] by You

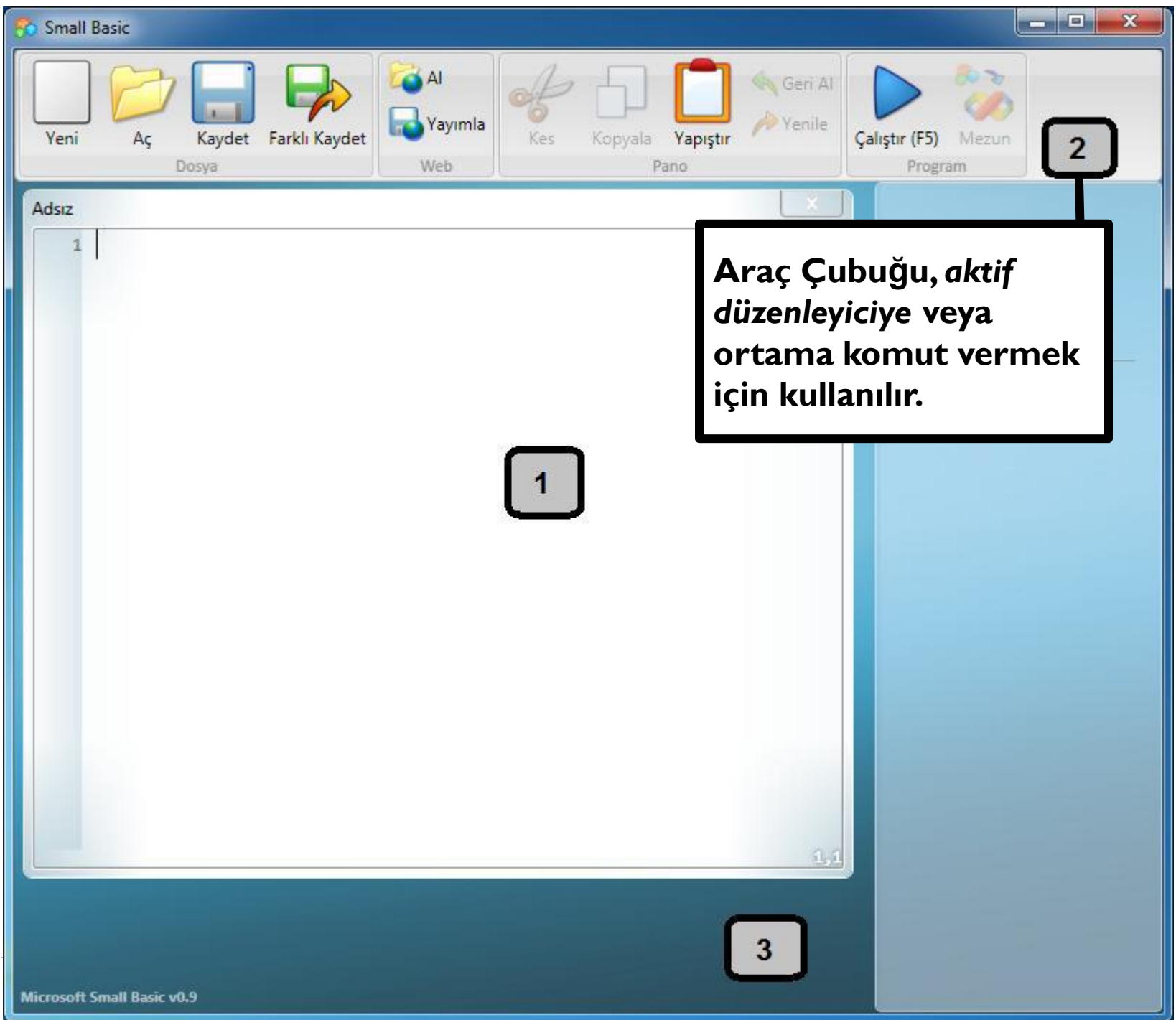


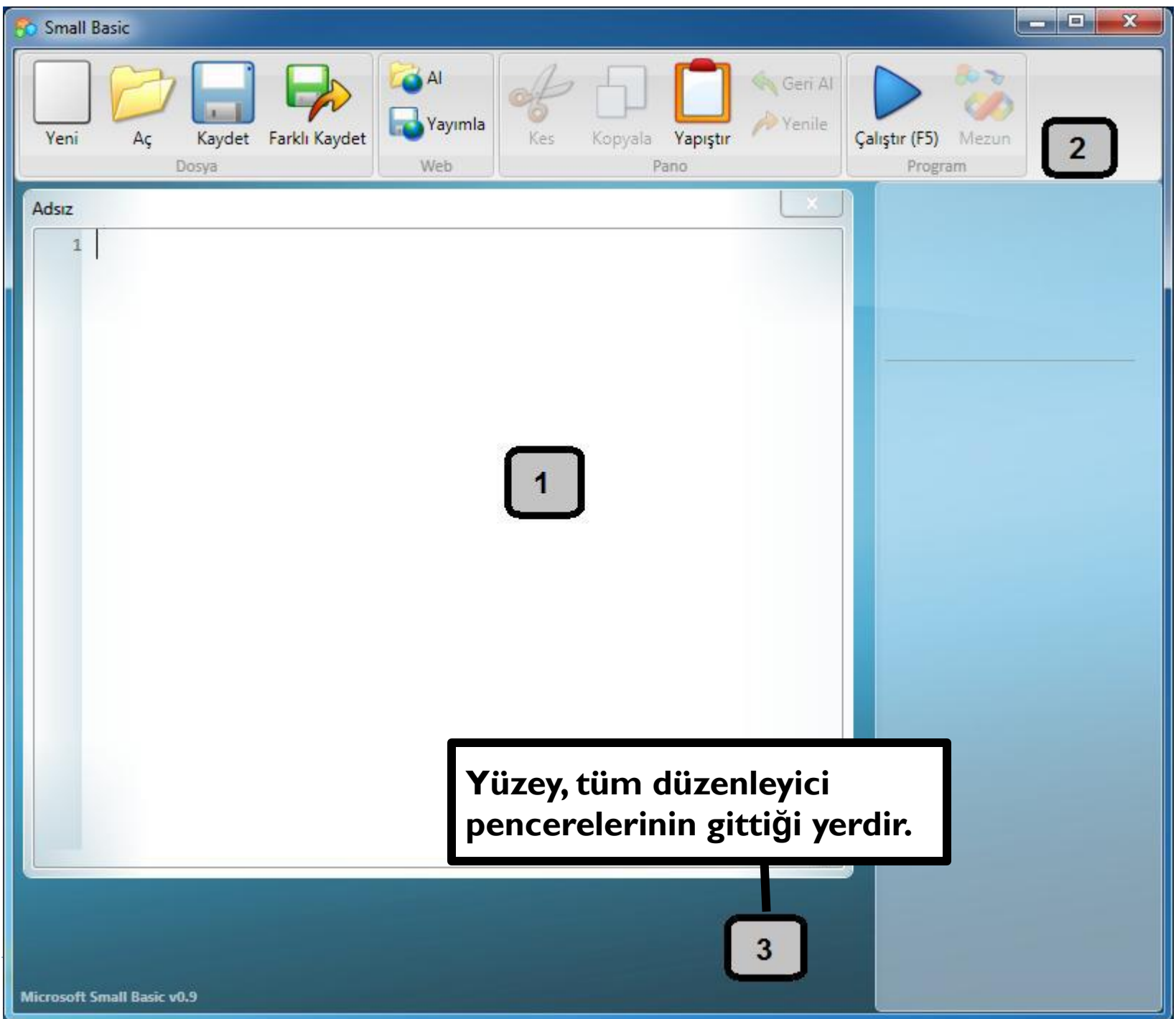
Small Basic Ortamı

Small Basic'i çalıştırdığınızda, programları yazıp çalıştıracağınız Small Basic ortamını göreceksiniz.









1

2

Yüzey, tüm düzenleyici pencerelerinin gittiği yerdir.

3

İlk Programımız

- ▶ Artık Small Basic ortamı ile tanıştığınıza göre, şimdi programlama yapmaya başlayabilirsiniz. Bunun için, önce aşağıdaki satırı düzenleyiciye yazın.



```
TextWindow.WriteLine("Merhaba Dünya")
```



İlk Programımız:

```
TextWindow.WriteLine("Merhaba Dünya")
```



Bu bizim ilk programımız. Eğer doğru yazdıysanız, aşağıdaki şekle benzer görünmelidir.

The image shows the Microsoft Small Basic v0.9 interface. The window title is "Small Basic". The menu bar includes "Yeni", "Aç", "Kaydet", "Farklı Kaydet", "Web", "Pano", and "Program". The code editor shows the following code:

```
1 TextWindow.WriteLine("Merhaba Dünya")
```

The help panel on the right is titled "İşlem" and features a yellow gear icon. It contains the following text:

WriteLine

Metin penceresine metin ya da sayı yazar. Yeni satır karakteri çıktıya eklenir, bu yüzden sonraki metin penceresine yazımlarda yazılan metin yeni satıra eklenir.

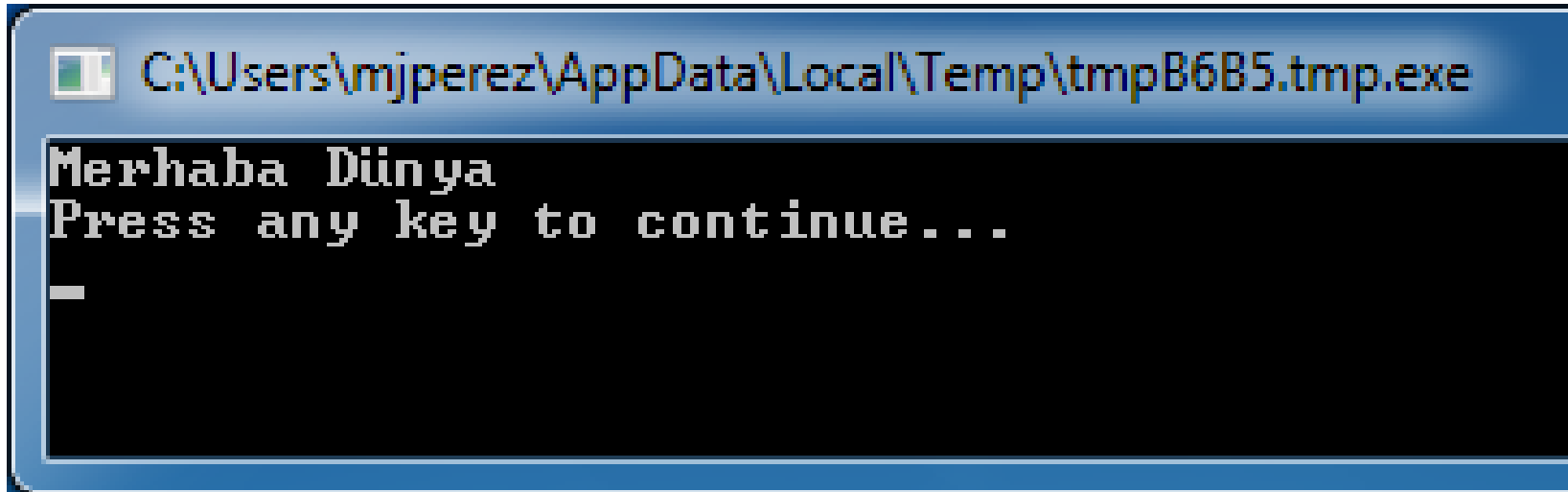
TextWindow.
WriteLine(data)

data
Metin penceresine yazılacak metin ya da sayı için kullanılır.

Microsoft Small Basic v0.9

İlk Programımız

- ▶ Programımızı araç çubuğu üzerindeki *Run (Çalıştır)* düğmesi ile veya klavyedeki F5 tuşu ile çalıştırabiliriz. Her şey yolundaysa, programımız aşağıdaki sonucu verecektir.



```
C:\Users\mjperrez\AppData\Local\Temp\tmpB6B5.tmp.exe  
Merhaba Dünya  
Press any key to continue...  
-
```



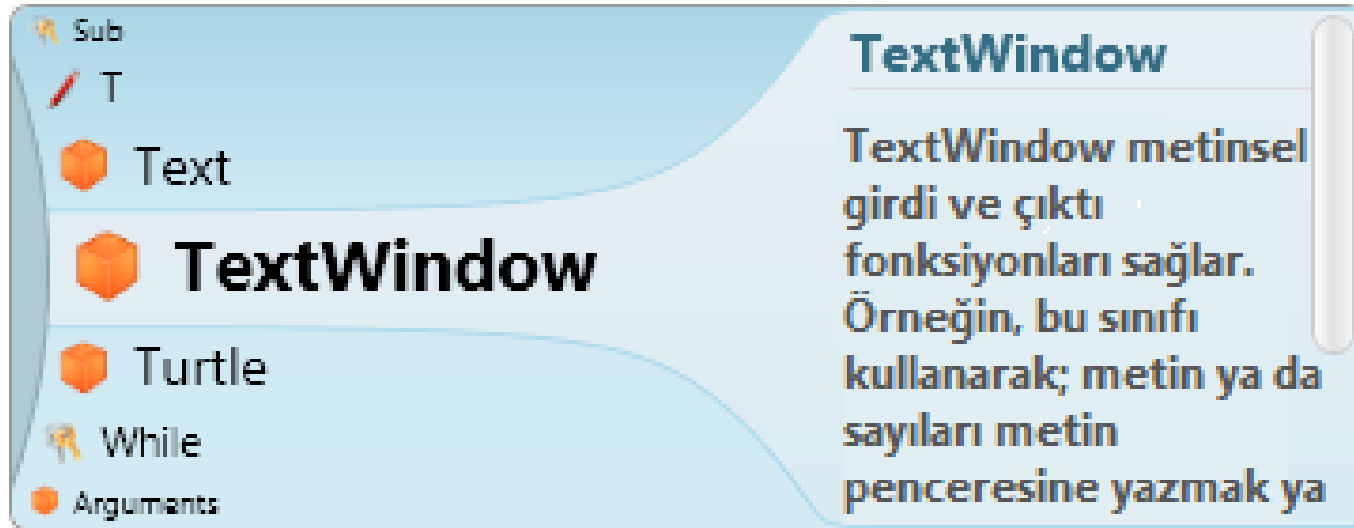
İlk Programımız

- Tebrikler! İlk Small Basic programınızı yazdınız ve çalıştırdınız. Şimdi, bunu anlayabilmek için, yazdığımız programı analiz edeceğiz.



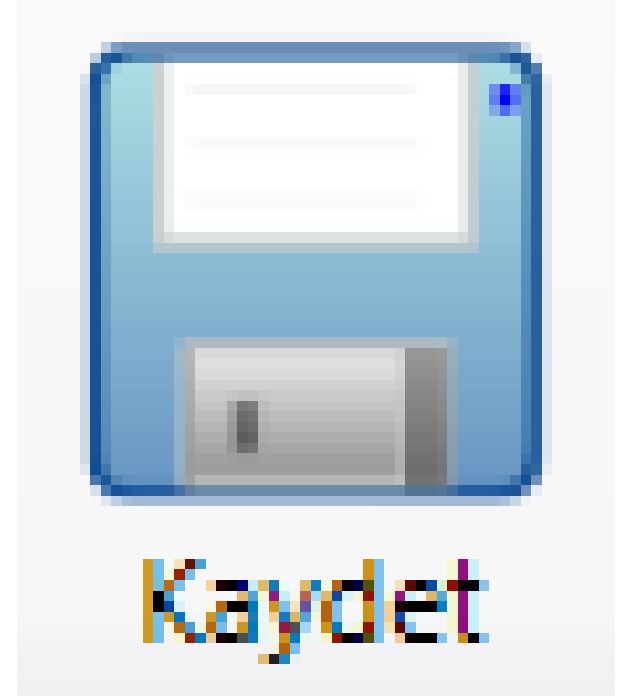
İlk Programımız

- ▶ Program yazarken, içinde öğeler listesi bulunan bir pencere açıldı. Bu, **akıllı algılamadır** ve programı daha hızlı yazmanıza yardımcı olur. Yukarı/Aşağı ok tuşlarıyla listeyi kaydırabilir ve istediğinizi bulduğunuzda, Enter tuşuna basarak öğeyi programa ekleyebilirsiniz.



Programımızı Kaydetmek

- ▶ Small Basic'i kapattıktan sonra aynı program üzerinde tekrar çalışmanız veya elektrik kesintisi durumunda bilgilerinizi kaybetmemeniz için, programı kaydetmelisiniz. Bunun için araç çubuğundaki "Kaydet" ikonuna basabilir veya "Ctrl+S" tuşlarını kullanabilirsiniz.





Microsoft®
Small Basic

İlk Programımızı Anlamak

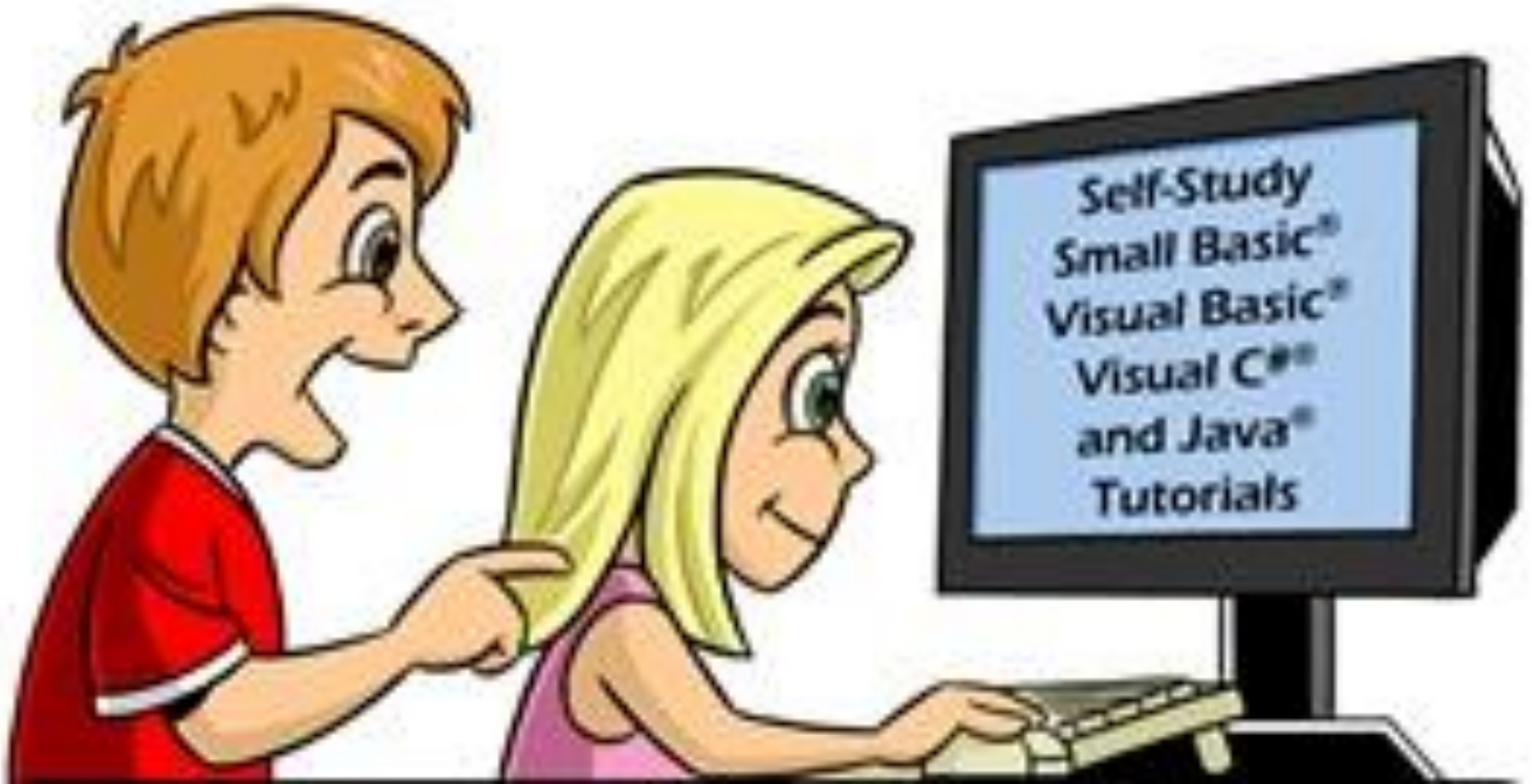
Bir bilgisayar programı aslında nedir?

- ▶ Bir program, bilgisayar için bir talimatlar dizisidir. Bu talimatlar bilgisayara tam olarak ne yapacağını söyler ve bilgisayar da bu talimatları izler. Bilgisayarın anlayabileceği pek çok dil vardır ve **Small Basic** de bunlardan birisidir.



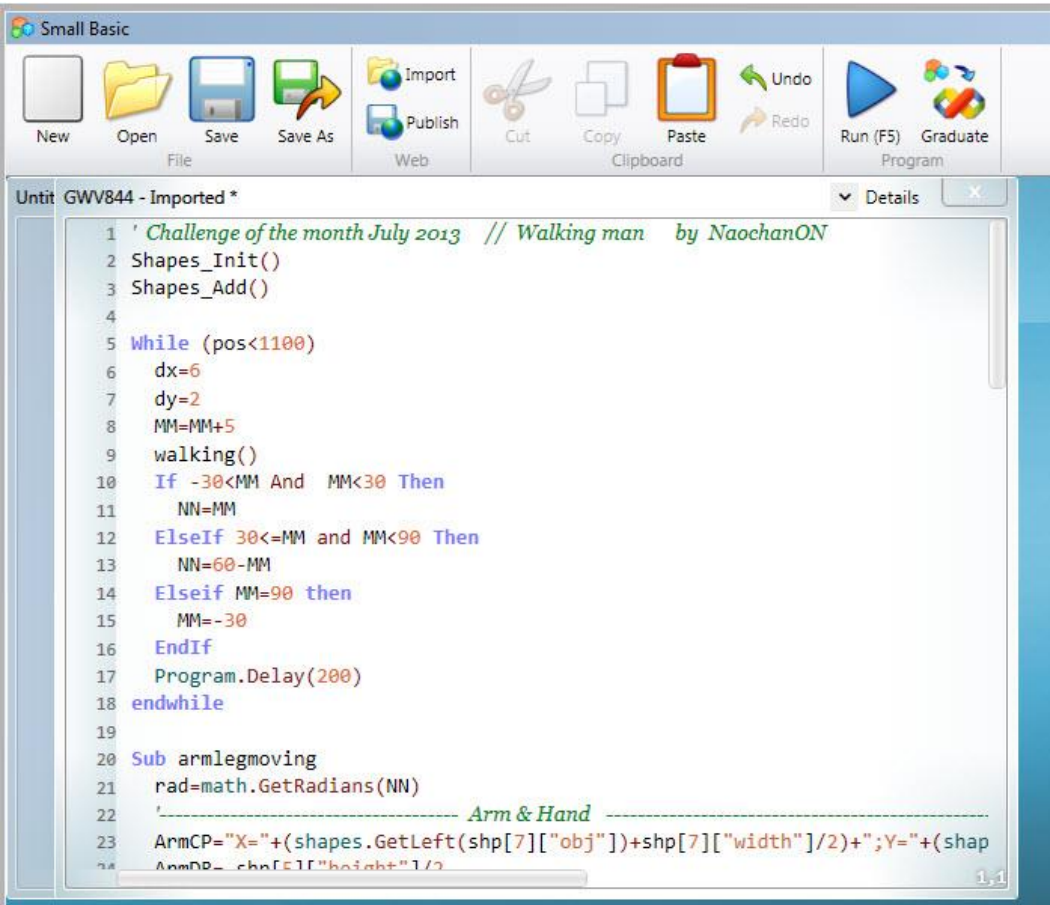
Bir bilgisayar programı aslında nedir?

- ▶ Java, C++, Python, Visual Basic vs. dillerinin tümü, basitten karmaşık yazılım programlarına kadar çeşitli programlar geliştirmek için kullanılacak, güçlü modern bilgisayar dilleridir.



Small Basic Programları

- ▶ Bir programın her bir satırı bilgisayar için birer talimattır. Bilgisayardan bir programı uygulamasını istediğimizde, bilgisayarı önce ilk ifadeyi okur ve uygular. Sonra ikinci, üçüncü satır derken programın sonuna ulaşınca kadar bunu yapmaya devam eder ve program tamamlanır.



The screenshot shows the Small Basic IDE interface. The title bar reads "Small Basic". The menu bar includes "File", "Web", "Clipboard", and "Program". The toolbar contains icons for "New", "Open", "Save", "Save As", "Import", "Publish", "Cut", "Copy", "Paste", "Undo", "Redo", "Run (F5)", and "Graduate Program". The main window displays a code editor with the following code:

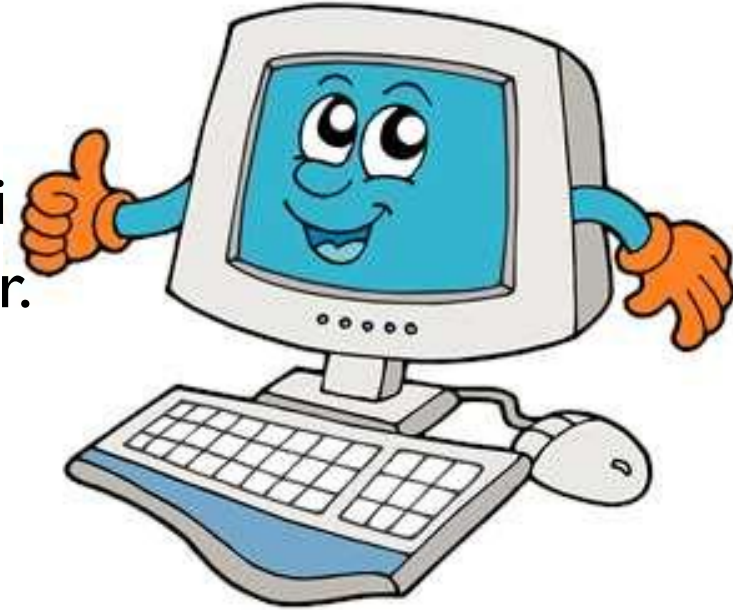
```
1 ' Challenge of the month July 2013 // Walking man by NaochanON
2 Shapes_Init()
3 Shapes_Add()
4
5 While (pos<1100)
6   dx=6
7   dy=2
8   MM=MM+5
9   walking()
10  If -30<MM And MM<30 Then
11    NN=MM
12  ElseIf 30<=MM and MM<90 Then
13    NN=60-MM
14  ElseIf MM=90 then
15    MM=-30
16  EndIf
17  Program.Delay(200)
18 endwhile
19
20 Sub armlegmoving
21   rad=math.GetRadians(NN)
22   '----- Arm & Hand -----
23   ArmCP="X="+{shapes.GetLeft(shp[7]["obj"])+shp[7]["width"]/2};"Y="+{shap
24   ArmDP=shp[5]["height"]/2
```


İlk Programımıza Geri Dönelim

- ▶ İşte yazdığımız ilk program:

```
TextWindow.WriteLine("Merhaba Dünya")
```

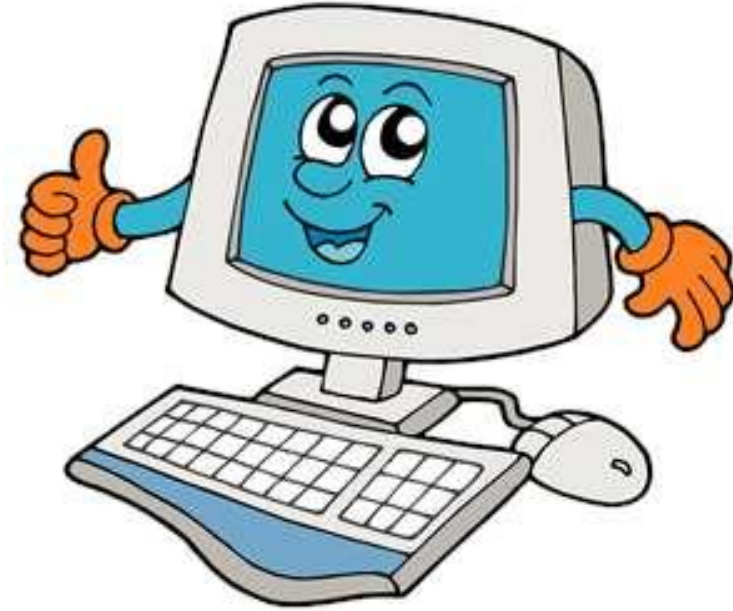
- ▶ Bu tek bir ifadeden oluşan program, bilgisayara **Merhaba Dünya** metnini Metin Penceresine yazmasını söylüyor.



İlk Programımıza Geri Dönelim

`TextWindow.WriteLine("Merhaba Dünya")`

- ▶ Bu ifadede, 3 ayrı bölüm bulunuyor:
 - a) `TextWindow`
 - b) `WriteLine`
 - c) "Merhaba Dünya"



İlk Programımıza Geri Dönelim

```
TextWindow.WriteLine("Merhaba Dünya")
```

- ▶ Nokta, parantezler ve tırnak işaretlerinin tümü, bilgisayarın niyetimizi anlaması için, doğru yerlere yerleştirilmesi gereken noktalama işaretleri.



TextWindow

İlk programımızı çalıştırdığımızda çıkan siyah pencere, TextWindow veya bazen Konsol olarak adlandırılır. Bu, programın sonucunun gideceği yerdir.



- BackgroundColor
- CursorLeft
- CursorTop
- ForegroundColor
- Left
- Title
- Top
- Clear
- Hide
- Pause
- PauseIfVisible
- PauseWithoutMessage
- Read
- ReadKey
- ReadNumber
- Show
- Write
- WriteLine

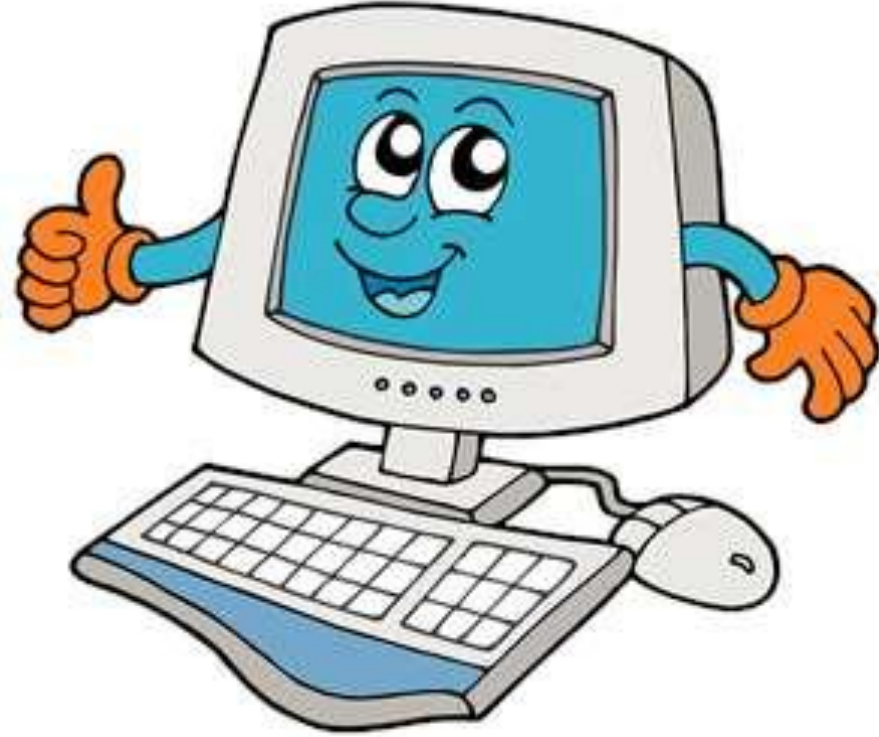
TextWindow:

Metin ya da sayıları
metin penceresine
yazmak ya da okumak
için kullanılır.



WriteLine

Programımızda *WriteLine* işlemini kullandık. Bu işlemi, tırnak işareti içerisinde **Merhaba Dünya** metni takip etti. Bu metin, *WriteLine* işlemine bir girdi olarak geçilmiştir, bu da daha sonra kullanıcı için yazdırılır.



WriteLine:

Metin penceresine
metin ya da sayı yazar
ve yeni satır eklenir.



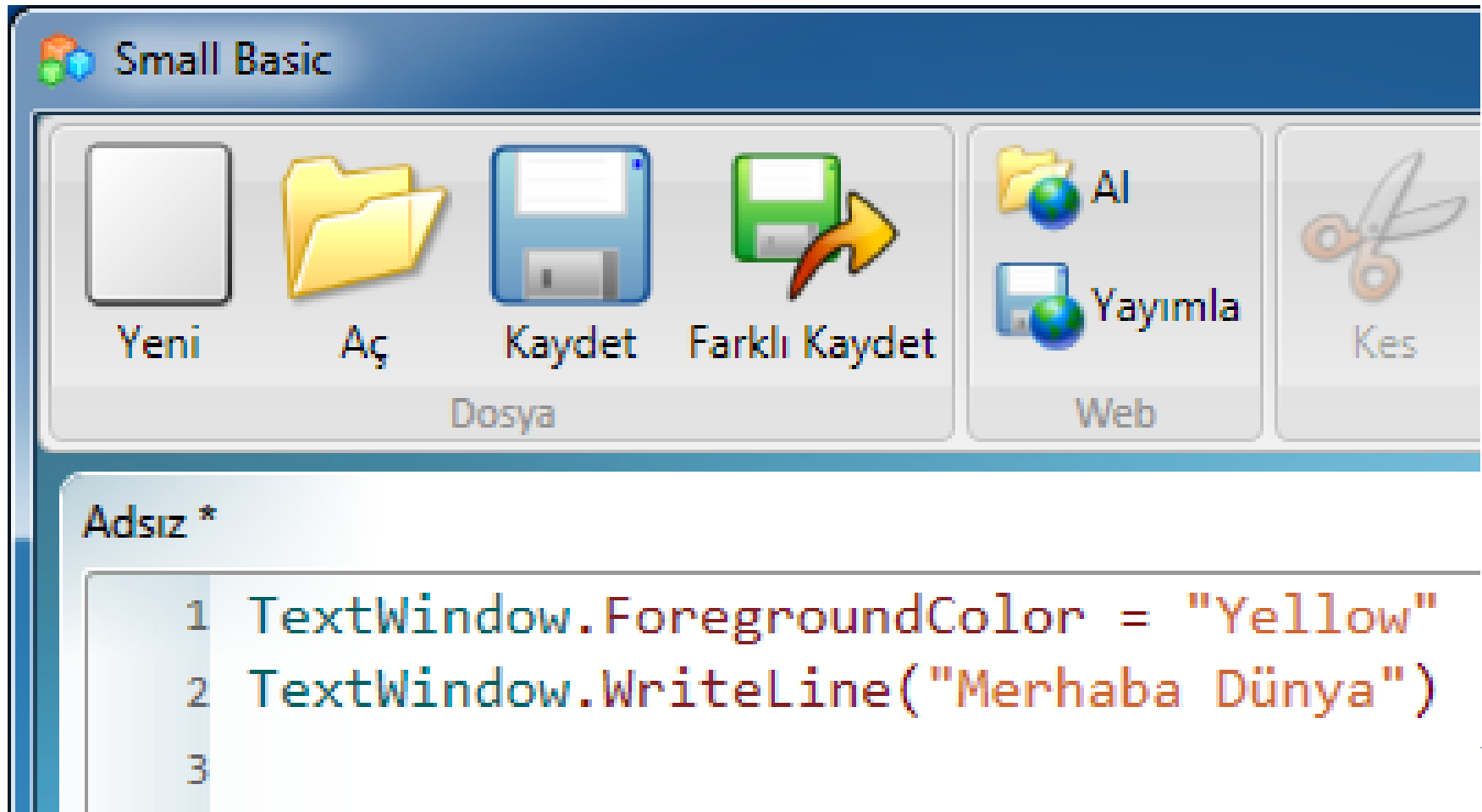
Noktalama İşaretleri

- ▶ Tırnak işaretleri, boşluklar ve parantezler gibi noktalama işaretleri, bir bilgisayar programında son derece önemlidirler.
- ▶ Yerlerine ve adetlerine bağlı olarak, ifade edilen anlamı değiştirebilirler.



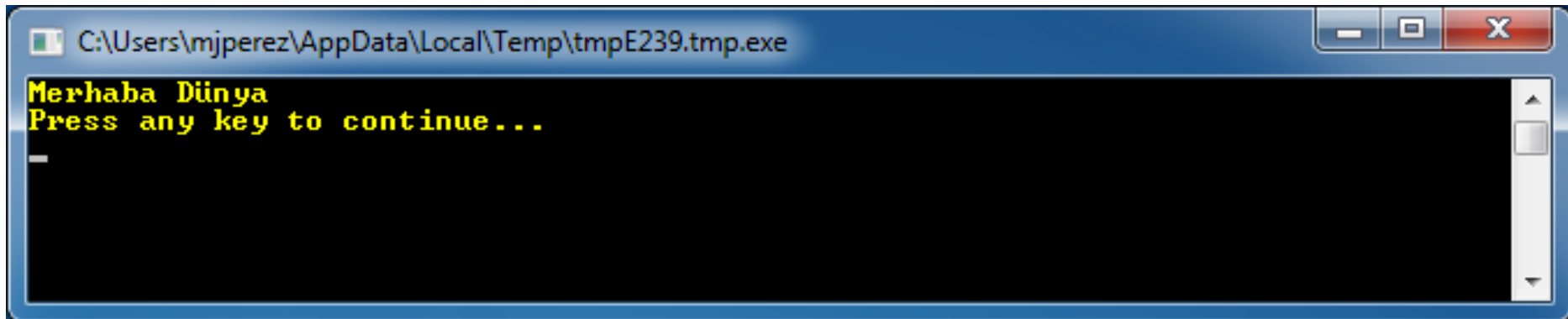
İkinci Programımız

- ▶ Artık ilk programımızı anladığınızınza göre, buna bazı renkler ekleyerek daha süslü hale getirelim.



İkinci Programımız

- ▶ Bu programı çalıştırdığınızda, TextWindow'un yine "Merhaba Dünya" sözcük grubunu yazdığını göreceksiniz, ancak bu kez sarı renkte yazar.



A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\mjperrez\AppData\Local\Temp\tmpE239.tmp.exe. The window contains the following text in yellow font: Merhaba Dünya, Press any key to continue..., and a small horizontal line on the next line. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
C:\Users\mjperrez\AppData\Local\Temp\tmpE239.tmp.exe  
Merhaba Dünya  
Press any key to continue...  
-
```



Örnek: Ekrana sarı renkte
“Merhaba Dünya” yazdırmak için
aşağıdaki programı yazarız.

```
TextWindow.ForegroundColor = "Yellow"  
TextWindow.WriteLine("Merhaba Dünya")
```



ForegroundColor

- ▶ Metnin çıktısı olarak metin penceresinde görünecek rengini belirtir.
- ▶ `ForegroundColor`, herhangi bir paranteze ihtiyaç duymaz. Onun yerine, bunu bir eşittir sembolü ve bir kelime takip eder.



ForegroundColor:

Metin penceresinde
görünecek yazının
rengini belirtir.



ForegroundColor

- ▶ İşte, ForegroundColor özelliği için geçerli olan değerlerin bir listesi. “Yellow”u bunlardan birisiyle değiştirmeyi deneyin ve sonuçları görün. Tırnak işaretlerini unutmayın.

Black

Blue

Cyan

Gray

Green

Magenta

Red

White

Yellow

DarkBlue

DarkCyan

DarkGray

DarkGreen

DarkMagenta

DarkRed

DarkYellow

▶ Diğer renkler için [tıklayın](#)

ForegroundColor için kullanılan renkler:

Black

Cyan

Green

Red

Yellow

DarkCyan

DarkGreen

DarkRed

Blue

Gray

Magenta

White

DarkBlue

DarkGray

DarkMagenta

DarkYellow





Microsoft®
Small Basic

Değişkenlerin Eklenmesi

Değişkenlerin Kullanılması

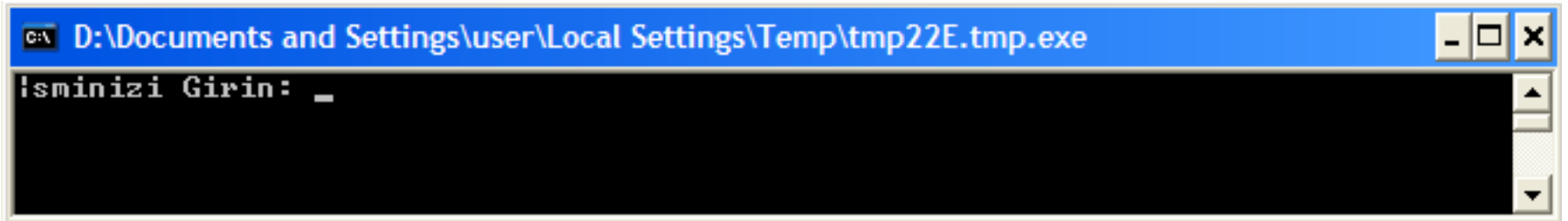
- ▶ Yeni programımız, “Merhaba Dünya” yerine kullanıcının ismiyle birlikte “Merhaba” desin.
- ▶ Bunun için önce kullanıcıya isminin sorulması ve sonra da bunu bir yerde saklayarak, kullanıcının ismiyle birlikte “Merhaba” metninin yazılması gerekir.



```
1 TextWindow.Write("İsminizi Girin: ")
2 ad = TextWindow.Read()
3 TextWindow.WriteLine("Merhaba " + ad)
```

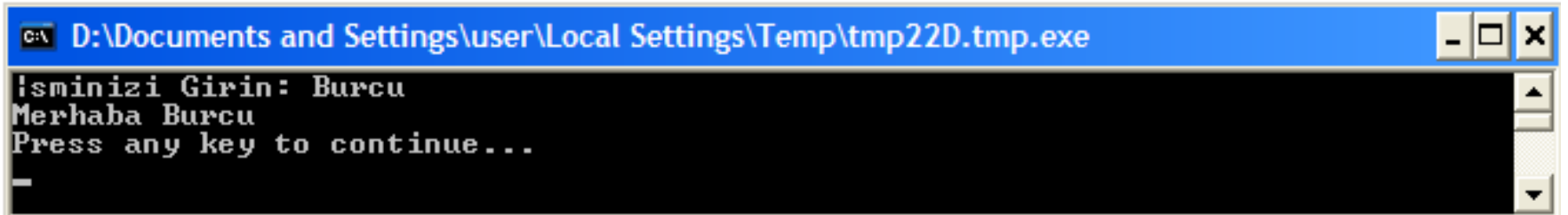
Değişkenlerin Kullanılması

- ▶ Bu programı yazıp çalıştırdığınızda, aşağıdaki çıktıyı göreceksiniz:



```
C:\> D:\Documents and Settings\user\Local Settings\Temp\tmp22E.tmp.exe
İsminizi Girin: _
```

- ▶ Ve, isminizi girip ENTER tuşuna bastığınızda, aşağıdaki çıktıyı göreceksiniz:



```
C:\> D:\Documents and Settings\user\Local Settings\Temp\tmp22D.tmp.exe
İsminizi Girin: Burcu
Merhaba Burcu
Press any key to continue...
_
```



Değişkenlerin Kullanılması

- Şimdi, programı tekrar çalıştırırsanız, size aynı soru tekrar sorulacaktır. Farklı bir isim girebilirsiniz ve bilgisayar size o isimle “Merhaba” diyecektir.



Merhaba

DEĞİŞKENLERİN KULLANILMASI



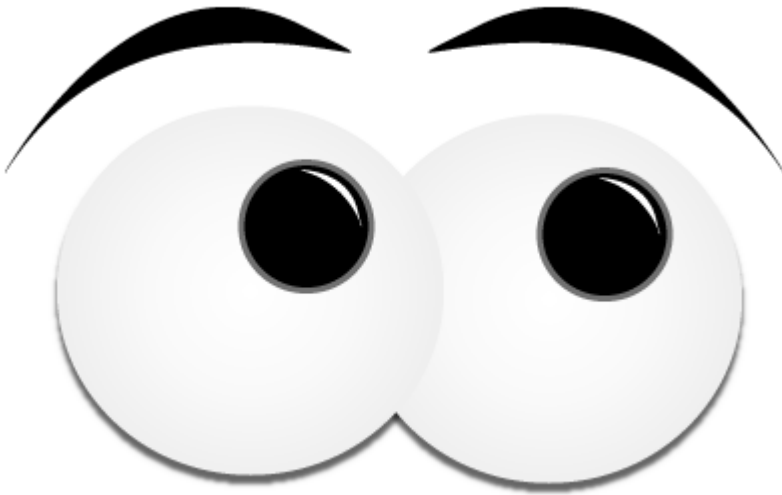
Örnek: Kullanıcının girdiği isimle birlikte “Merhaba” diyen bir program yazalım.

```
TextWindow.Write("İsminizi girin: ")  
ad = TextWindow.Read()  
TextWindow.WriteLine("Merhaba " + ad)
```

Programın Analizi

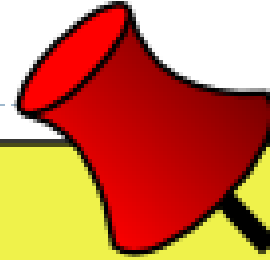
- ▶ Biraz önce çalıştırdığınız programda, dikkatinizi çekmiş olabilecek satır şudur:

```
ad= TextWindow.Read()
```



Read

- ▶ Read() işlemi, bilgisayara kullanıcının bir metin girmesini ve Enter tuşuna basmayı beklemesini söyler. Kullanıcı Enter'a bastığında, girilen metni alır ve programa geri döner. Kullanıcının girdiği metin, **isimli** bir değişkende saklanır.



Bir **değişken**, değerleri geçici olarak saklayabildiğiniz ve sonra kullanabildiğiniz bir yer olarak tanımlanır. Az önceki satırda, kullanıcının ismini saklamak için, **ad** kullanılmıştır.

Değişken:

Değerleri geçici olarak saklayabildiğiniz ve sonra kullanabildiğiniz bir yerdir.



Değişkenlerin İsimlendirilmesi ile İlgili Kurallar

Değişkenlerin onlarla bağlantılı isimleri vardır ve onları bu şekilde tanırırsınız. Bu değişkenlerin isimlendirilmeleriyle ilgili belirli basit kurallar vardır. Bunlar:

1. İsim bir harfle başlamalıdır.
2. **if, for, then, vs.** gibi kelimelerle çakışmamalıdır.
3. Bir isim, harflerin, sayıların ve altçizgilerin herhangi bir kombinasyonundan oluşabilir.
4. Değişkenleri anlamlı bir şekilde isimlendirmek faydalıdır. Değişkenler istendiği kadar uzun olabileceği için, amaçlarını açıklayan değişken isimleri kullanın.



Değişkenlerin İsimlendirilmeleri ile İlgili Kurallar:

1. İsim bir harfle başlamalıdır.
2. if, for, then, vs. gibi kelimelerle çakışmamalıdır.
3. Bir isim; harfler, sayılar ve altçizgilerden oluşabilir.
4. Değişkenler istendiği kadar uzun olabileceği için, anlamlı bir şekilde isimlendirilmelidir.



Read:

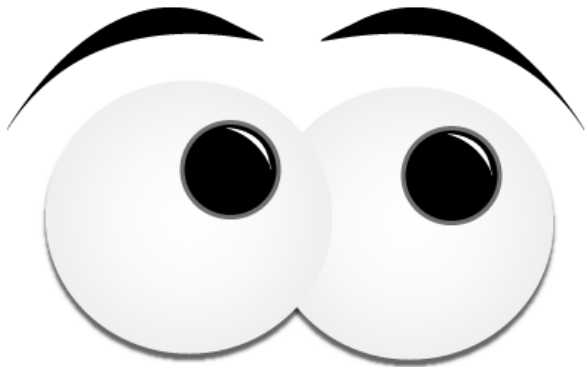
Metin penceresinden kullanıcının girdiđi metni okur ve Enter tuşuna basılana kadar bekler.



Programın Analizi

```
TextWindow.WriteLine("Merhaba " + ad)
```

- ▶ Bu, **ad** değişkenimizde sakladığımız değeri kullandığımız yerdir. **ad** bölümünde yazan değeri alıyoruz, bunu “Merhaba”ya ekliyoruz ve TextWindow’a yazıyoruz.



Programı Geliştirme

- ▶ Bir değişken girildiğinde, bunu istediğiniz zaman tekrar kullanabilirsiniz. Örneğin; şunları yapabilirsiniz:

```
1 TextWindow.Write("İsminizi Girin: ")
2 ad = TextWindow.Read()
3 TextWindow.Write("Merhaba " + ad + ". ")
4 TextWindow.WriteLine("Nasilsin " + ad + "?")
5
```

- ▶ Şu çıktıyı göreceksiniz:

```
İsminizi Girin: Burcu
Merhaba Burcu.  Nasilsin Burcu?
Press any key to continue...
```

—



Write

- ▶ Tıpkı WriteLine gibi, Write da TextWindow'da (Konsol) bir diğer işlemdir. Write, Konsola bir şey yazmanıza izin verir, ancak bundan sonra gelen metnin mevcut metinle aynı satırda olmasını sağlar.



Write:

Metin penceresine metin
ya da sayı yazar ve yeni
satır eklenmez.



Sayılarla Oynamak

- Biraz önce kullanıcının ismini saklamak için deęişkenleri nasıl kullanabileceğimizi gördük. Bundan sonraki birkaç programda, deęişkenlerde sayıları nasıl saklayabileceğimizi ve işleyebileceğimizi göreceğiz.

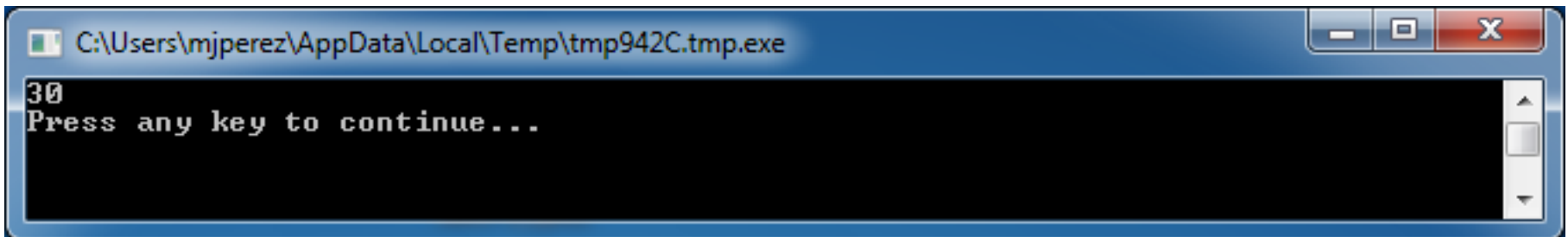


Sayılarla Oynamak

- ▶ Basit bir programla başlayalım:

```
1 a = 10
2 b = 20
3 c = a + b
4 TextWindow.WriteLine(c)
```

- ▶ Bu programı çalıştırdığınızda, aşağıdaki çıktıyı göreceksiniz:



The screenshot shows a Windows command prompt window with a blue title bar. The title bar text is "C:\Users\mjperrez\AppData\Local\Temp\tmp942C.tmp.exe". The window content is black with white text. The first line of text is "30". The second line of text is "Press any key to continue...". The window has standard Windows window controls (minimize, maximize, close) in the top right corner.



Sayılarla Oynamak



- ▶ Toplama işlemi için aşağıdaki programı yazalım.

```
1 a = 10
2 b = 20
3 c = a + b
4 TextWindow.WriteLine(c)
```



Dört İşlem Sembolleri:

+ → Toplama

- → Çıkarma

* → Çarpma

/ → Bölme




Programın Analizi

- ▶ Programın birinci satırında, **a** değişkenine 10 sayısını atıyoruz. İkinci satırda, **b** değişkenine 20 sayısını atıyoruz. Üçüncü satırda, **a** ve **b**'yi topluyor ve sonra çıkan sonucu **c**'ye atıyoruz. Böylece bu örnekte, **c**'nin değeri 30 olacaktır. TextWindow'da görüntülediğimiz değer de budur.

```
1 a = 10
2 b = 20
3 c = a + b
4 TextWindow.WriteLine(c)
```





Sayıların başında ve sonunda tırnak işareti olmadığına dikkat edin. Sayılar için, tırnak işaretine gerek yoktur. Tırnak işaretlerine yalnızca metin kullanırken ihtiyacınız vardır.

“”

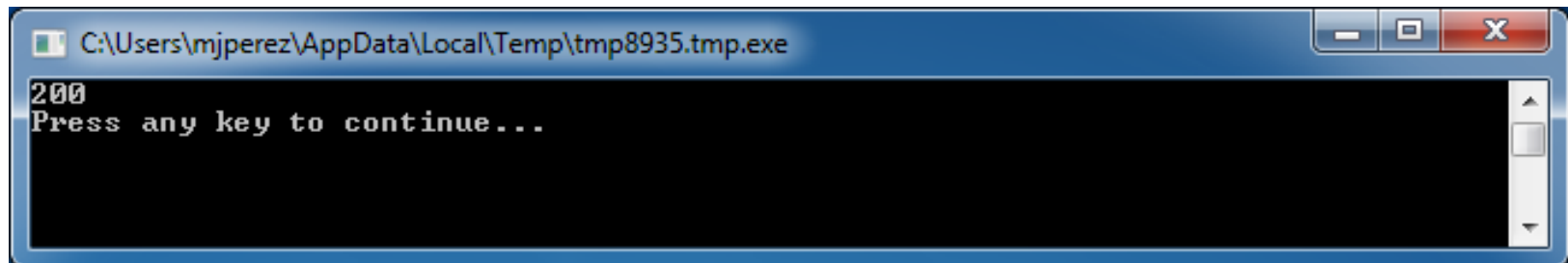


Sayılarla Oynamak

- ▶ Şimdi, programı biraz değiştirelim ve sonuçları görelim:

```
1 a = 10
2 b = 20
3 c = a * b
4 TextWindow.WriteLine(c)
```

- ▶ Yukarıdaki program, **a** ile **b**'yi çarpacak ve çıkan sonucu **c**'de saklayacaktır. Bu programın sonucunda şunu görebilirsiniz:



A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\mjperrez\AppData\Local\Temp\tmp8935.tmp.exe. The window contains the output of a program: 200. Below the output, it says "Press any key to continue...".

Sayılarla Oynamak

- ▶ Benzer şekilde, sayıları çıkarabilir ya da bölebilirsiniz. İşte bir çıkarma işlemi:

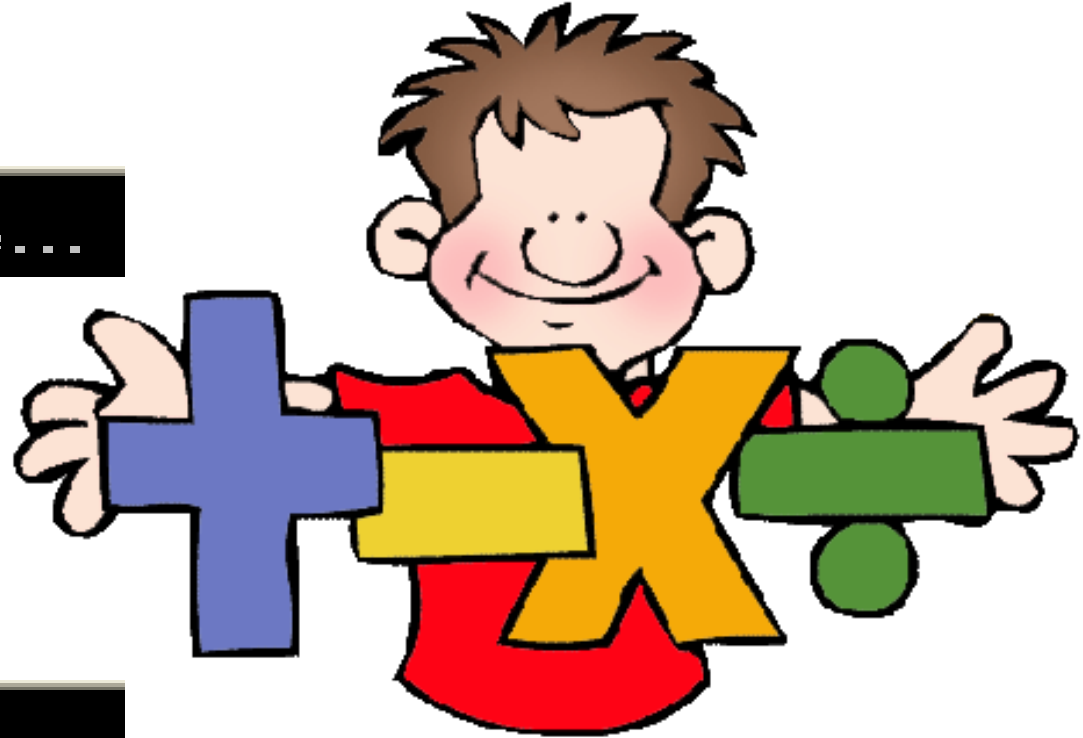
$$c = a - b$$

```
-10  
Press any key to continue...
```

- ▶ Ve bölme işlemi:

$$c = a / b$$

```
0,5  
Press any key to continue...
```



Örnek

- ▶ Klavyeden girilen iki sayının toplamını ekrana yazdıran programı yazalım.

```
1 TextWindow.Write(" 1.sayı: ")
2 a = TextWindow.Read()
3 TextWindow.Write(" 2.sayı: ")
4 b = TextWindow.Read()
5 c= a+b
6 TextWindow.WriteLine("iki sayının toplamı: " + c)
```

```
1.sayı: 5
2.sayı: 8
iki sayının toplamı: 13
Press any key to continue...
```

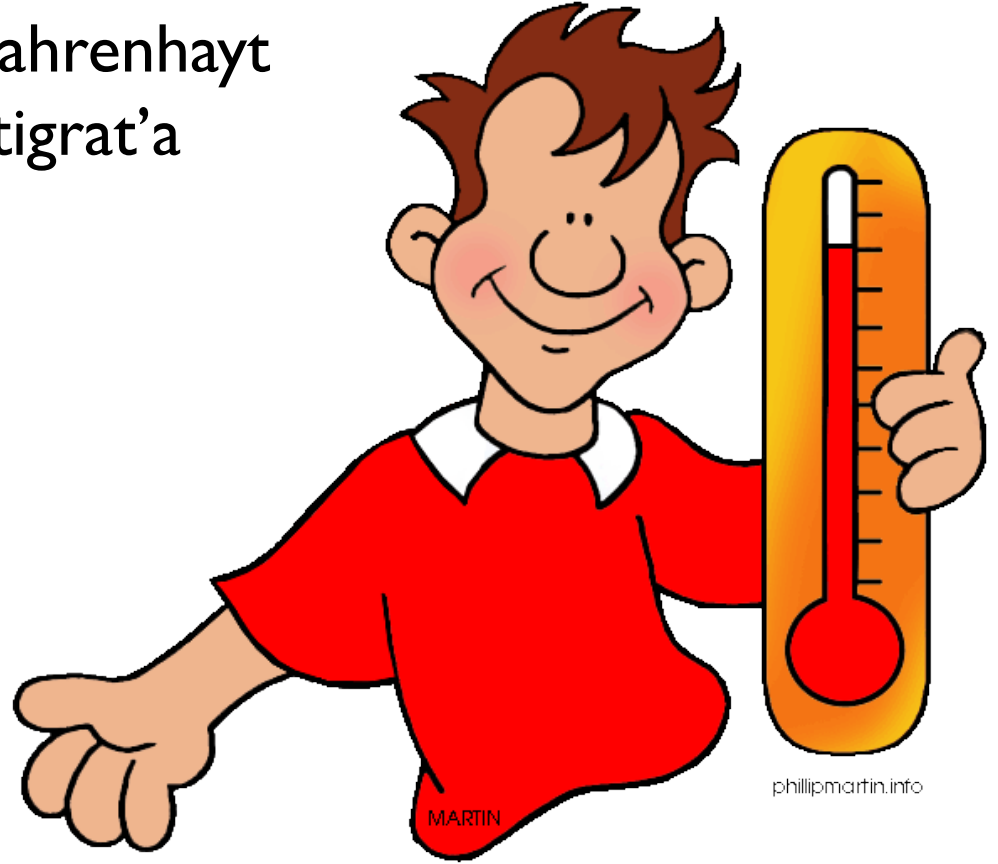


Basit Bir Sıcaklık Dönüştürücüsü

Bir sonraki programda, Fahrenheit cinsinden sıcaklıkları Santigrat'a çevirmek için,

$$^{\circ}\text{C} = \frac{5(^{\circ}\text{F} - 32)}{9}$$

formülünü kullanacağız.



Basit Bir Sıcaklık Dönüştürücüsü

- ▶ İlk olarak, kullanıcıdan sıcaklığı Fahrenheit cinsinden alacak ve bunu bir değişkende saklayacağız. Kullanıcıdan gelen sayıları okumamızı sağlayan özel bir işlem vardır ve bu da; **TextWindow.ReadNumber**.

```
1 TextWindow.Write("Sicakligi Fahrenheit cinsinden girin: ")
2 f = TextWindow.ReadNumber()
```

- ▶ Fahrenheit cinsinden sıcaklığı bir değişkende sakladıktan sonra, bunu şu şekilde Santigrat'a çevirebiliriz:

```
3 c = 5 * (f - 32) / 9
```

- ▶ Şimdi yapmamız gereken, sonucu kullanıcıya göstermektir.

```
4 TextWindow.WriteLine("Santigrat cinsinden sicaklik " + c)
```



Basit Bir Sıcaklık Dönüştürücüsü



```
1 TextWindow.Write("Sicakligi Fahrenhayt cinsinden girin: ")
2 f = TextWindow.ReadNumber()
3 c = 5 * (f - 32) / 9
4 TextWindow.WriteLine("Santigrat cinsinden sicaklik " + c)
```

- ▶ Ve bu programın sonucu şu olacaktır:

```
C:\Users\mjperrez\AppData\Local\Temp\tmp7376.tmp.exe
Sicakligi Fahrenhayt cinsinden girin: 77
Santigrat cinsinden sicaklik 25
Press any key to continue...
```

A cartoon illustration of a man with brown hair, wearing a red shirt, holding a large yellow thermometer. The thermometer has a red liquid column and a scale. The man is smiling and looking towards the viewer. The name 'MARTIN' is written at the bottom of the man's shirt, and 'phillipmccartin.info' is written at the bottom of the thermometer.

ReadNumber:

Metin penceresinden bir sayı okur ve Enter tuşuna basılana kadar bekler.



Not Ortalaması Hesaplayan Program

```
1 TextWindow.Write("1. not: ")
2 not1 = TextWindow.Read()
3 TextWindow.Write("2. not: ")
4 not2 = TextWindow.Read()
5 TextWindow.Write("3. not: ")
6 not3 = TextWindow.Read()
7 ort = (not1+not2+not3)/3
8 TextWindow.WriteLine("Not ortalamasi: " + ort)
```



```
1. not: 50
2. not: 60
3. not: 100
Not ortalamasi: 70
Press any key to continue...
```



Not Ortalaması Hesaplayan Program



```
1 TextWindow.Write("1. not: ")
2 not1 = TextWindow.Read()
3 TextWindow.Write("2. not: ")
4 not2 = TextWindow.Read()
5 TextWindow.Write("3. not: ")
6 not3 = TextWindow.Read()
7 ort = (not1+not2+not3)/3
8 TextWindow.WriteLine("Not ortalamasi: " + ort)
```





Microsoft®
Small Basic

Koşullar ve Dallanma

Koşullar

- ▶ Programımızı geliştirmeye devam ediyoruz. Önce isim sorulacak ve kullanıcının yazacağı isim, **ad** değişkenine atanacak. Ardından, saat 12:00'den önceyse “Günaydın”, değilse “İyi Akşamlar” ile birlikte girilen isim ekrana yazılacak.



```
1 TextWindow.Write ("İsminizi girin: ")
2 ad = TextWindow.Read()
3 If (Clock.Hour <12) Then
4 TextWindow.WriteLine ("Günaydın " + ad)
5 Else
6 TextWindow.WriteLine ("İyi akşamlar " + ad)
7 EndIf
```

Koşullar



- ▶ Örnek: Saat 12:00'den önceyse "Günaydın", değilse "İyi Akşamlar" ile birlikte klavyeden girilen ismi ekrana yazdıran program:

```
1 TextWindow.Write ("İsminizi girin: ")
2 ad = TextWindow.Read()
3 If (Clock.Hour <12) Then
4 TextWindow.WriteLine ("Günaydın " + ad)
5 Else
6 TextWindow.WriteLine ("İyi akşamlar " + ad)
7 EndIf
```


Koşullar

- ▶ Programı ne zaman çalıştırdığınıza bağlı olarak, aşağıdaki çıktılarından birisini göreceksiniz:
 - ▶ 12.00'dan önce ise:

```
İsminizi Girin: Burcu  
Günaydın Burcu  
Press any key to continue...
```

- ▶ 12.00'dan sonra ise:

```
İsminizi Girin: Burcu  
İyi Akşamlar Burcu  
Press any key to continue...
```



Programın Analizi

```
3 If (Clock.Hour <12) Then
4 TextWindow.WriteLine ("Günaydın " + ad)
5 Else
6 TextWindow.WriteLine ("İyi akşamlar " + ad)
7 EndIf
```

- ▶ Clock.Hour değerinin 12'den az olması durumunda, "Günaydın" yazısı ile birlikte klavye ile girilen isim yazdırılacaktır. **If**, **Then**, **Else** ve **EndIf** kelimeleri, program çalışırken bilgisayar tarafından anlaşılan özel kelimelerdir.



Programın Analizi

```
3 If (Clock.Hour <12) Then
4 TextWindow.WriteLine ("Günaydın " + ad)
5 Else
6 TextWindow.WriteLine ("İyi akşamlar " + ad)
7 EndIf
```

- ▶ **If** kelimesinin ardından daima bir koşul gelir, bu durumda bu koşul (**Clock.Hour<12**)'dir. Parantezler, bilgisayarın sizin niyetinizi anlaması için gereklidir. Koşulu **then** ve yürütülecek işlem izler. Koşulun sağlanmadığı durumda **Else** devreye girer ve sonrasındaki işlem yürütülür. **EndIf** ile koşulun uygulanması biter.
-



İkiden fazla koşulun gerektiği durumlarda **Else** kullanmak yerine her koşul için ayrı **If-Then-EndIf** kullanılabilir.

```
1 TextWindow.Write("İsminizi girin: ")
2 ad = TextWindow.Read()
3 If (Clock.Hour < 6) Then
4     TextWindow.WriteLine("İyi Geceler " + ad)
5 EndIf
6 If (Clock.Hour >= 6 And Clock.Hour < 12) Then
7     TextWindow.WriteLine("Günaydın " + ad)
8 EndIf
9 If (Clock.Hour >= 12 And Clock.Hour < 17) Then
10    TextWindow.WriteLine("İyi Günler " + ad)
11 EndIf
12 If (Clock.Hour >= 17 And Clock.Hour < 24) Then
13    TextWindow.WriteLine("İyi Akşamlar " + ad)
14 EndIf
```

Aşağıdaki örnekte, belli saat aralıklarında programın farklı şekillerde sonuç verdiğini göreceksiniz.

```
1 TextWindow.Write("İsminizi girin: ")
2 ad = TextWindow.Read()
3 If (Clock.Hour < 6) Then
4     TextWindow.WriteLine("İyi Geceler " + ad)
5 EndIf
6 If (Clock.Hour > = 6 And Clock.Hour < 12) Then
7     TextWindow.WriteLine("Günaydın " + ad)
8 EndIf
9 If (Clock.Hour > = 12 And Clock.Hour < 17) Then
10    TextWindow.WriteLine("İyi Günler " + ad)
11 EndIf
12 If (Clock.Hour > = 17 And Clock.Hour < 24) Then
13    TextWindow.WriteLine("İyi Akşamlar " + ad)
14 EndIf
```

Koşullar:

If: Farklı şeyleri yapabilmek için kararlarda bulunmanıza olanak verir.

Then: Koşulun sağlanması durumunda yürütülecek işlem yazılır.

Else: Koşulun sağlanmaması durumunda yürütülecek işlem yazılır.

Endif: Koşulun uygulanmasının bittiğini ifade eder.



Clock.Hour

- ▶ Small Basic'de, o andaki tarihe ve saate erişmek için, **Clock** nesnesini kullanabilirsiniz. Bu ayrıca size, o andaki günü, ayı, yılı, dakikayı, saniyeyi ayrı ayrı alabilmenizi sağlayan bir grup özellik sağlar.
- ▶ Az önceki programda saatin 12'den küçük olup olmadığını sorgulamak için **Hour** ifadesini kullandık.



Nesne

Clock

Sistem saatine erişmeyi sağlar.

- 🎨 Date
- 🎨 Day
- 🎨 ElapsedMilliseconds
- 🎨 Hour
- 🎨 Millisecond
- 🎨 Minute
- 🎨 Month
- 🎨 Second
- 🎨 Time
- 🎨 WeekDay
- 🎨 Year

Clock:

O andaki saate eriřmek için kullanılır.

Beraberinde isteęe göre Date, Day, Hour, Minute, Year gibi ifadeler kullanılır.



Örnek:



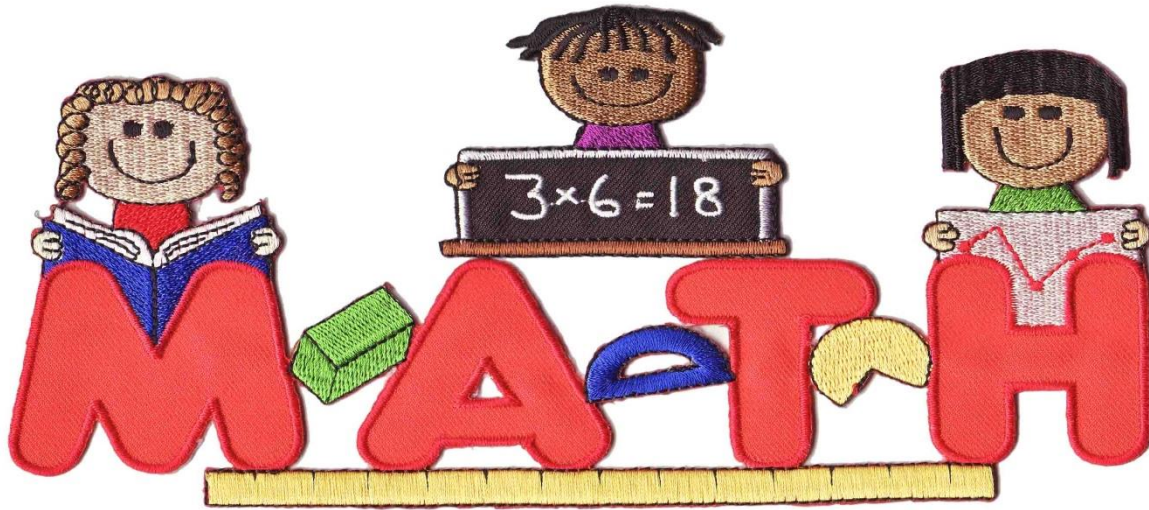
- ▶ Klavyeden girilen bir sayının tek mi, çift mi olduğunu söyleyecek program:

```
1 TextWindow.Write("Bir sayı girin: ")
2 a = TextWindow.ReadNumber()
3 r = Math.Remainder(a, 2)
4 If (r = 0) Then
5     TextWindow.WriteLine("Çift Sayı")
6 Else
7     TextWindow.WriteLine("Tek Sayı")
8 EndIf
```

```
Bir sayı girin: 35
Tek Sayı
Press any key to continue...
```

Math

- Bu programda, **Math** nesnesini kullandık. **Math**, matematik ile ilgili birçok yöntemin bulunduğu sınıf için kullanılır.



- ◉ Pi
- ◉ Abs
- ◉ ArcCos
- ◉ ArcSin
- ◉ ArcTan
- ◉ Ceiling
- ◉ Cos
- ◉ Floor
- ◉ GetDegrees
- ◉ GetRadians
- ◉ GetRandomNumber
- ◉ Log
- ◉ Max
- ◉ Min
- ◉ NaturalLog
- ◉ Power
- ◉ Remainder
- ◉ Round
- ◉ Sin
- ◉ SquareRoot

Math:

Matematik ile ilgili birçok yöntemin bulunduğu sınıf için kullanılır.



Remainder:

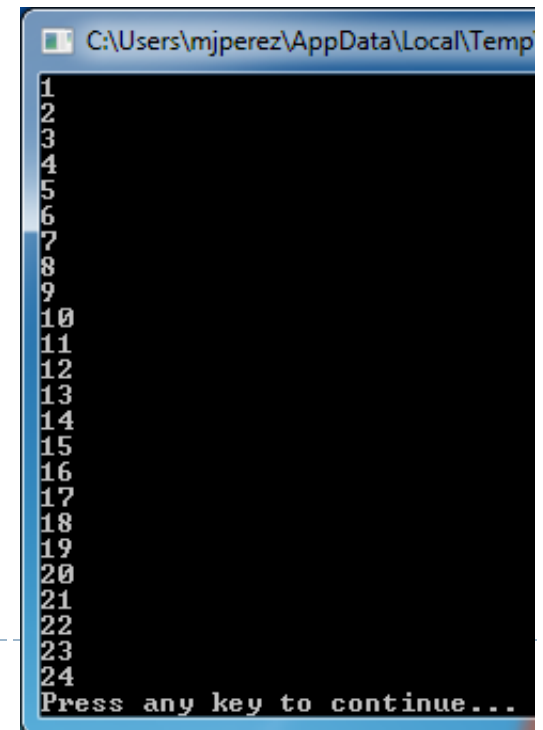
İlk sayıyı ikinci sayıya
böler ve kalanını verir.



Dallanma

- ▶ Hatırlarsanız, bilgisayarın bir programı yukarıdan aşağıya doğru her defasında bir ifadeyi işlemden geçirecek şekilde çalıştırdığını öğrenmiştiniz. Bununla birlikte, bilgisayarın sıranın dışına çıkarak bir başka ifadeye atlamasını sağlayan özel bir ifade vardır. Aşağıdaki programa bir göz atalım.

```
1 i = 1
2 listele:
3 TextWindow.WriteLine(i)
4 i = i + 1
5 If (i < 25) Then
6     Goto listele
7 EndIf
```



```
C:\Users\mjperrez\AppData\Local\Temp
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
Press any key to continue...
```

Örnek:

- ▶ 1'den 24'e kadar sayıları alt alta yazdıran program



```
1  i = 1
2  listele:
3  TextWindow.WriteLine(i)
4  i = i + 1
5  If (i < 25) Then
6      Goto listele
7  EndIf
```

Etiketler

- ▶ Bu programda, **i** değişkenine **1** değerini atadık. Ve sonra, iki nokta üst üste (**:**) ile biten yeni bir ifade ekledik.
 - ▶ **listele:**
- ▶ Bu, bir **etiket** olarak adlandırılır.

```
1  i = 1
2  listele:
3  TextWindow.WriteLine(i)
4  i = i + 1
5  If (i < 25) Then
6      Goto listele
7  EndIf
```

Etiketler:

Bilgisayarın anlayabildiđi yerimleridir. Örneđimizde “listele” bir etikettir.

Etiketleri istediđimiz gibi adlandırabiliriz ve her biri farklı adla olmak kořuluyla, istediđimiz kadar etiket ekleyebiliriz.



Değişkene Değer Atama

- ▶ Buradaki bir diğer ilginç ifade de şudur:
 - ▶ $i = i + 1$
- ▶ Bu yalnızca bilgisayara i değişkenine 1 eklemesini ve onu tekrar i değişkenine atmasını söyler. Yani, i 'nin değeri bu ifadeden önce 1 ise, ifade çalıştırıldıktan sonra 2 olacaktır.

```
1  i = 1
2  listele:
3  TextWindow.WriteLine(i)
4  i = i + 1
5  If (i < 25) Then
6    Goto listele
7  EndIf
```

Değişkene Değer Atama:

$$i = i + 1$$

Bu ifade, i değişkenine 1 eklenmesini ve tekrar i değişkenine atanmasını söyler. Eşittir işareti (=), değer atamak için kullanılır.



Koşullar ve Döngü

- ▶ Aşağıda belirtilmiş kısım, **i**'nin değeri 25'den küçükse, ifadeleri **listele** yer iminden başlayarak uygulanmasını söyleyen bölümdür.

```
1 i = 1
2 listele:
3 TextWindow.WriteLine(i)
4 i = i + 1
5 If (i < 25) Then
6     Goto listele
7 EndIf
```



Goto:

Program içerisinde
yeni bir konuma
gidebilmenize olanak
verir.



Sonsuz Uygulama

- ▶ **Goto** ifadesini kullanarak, bilgisayarın bir şeyi istediğiniz defa tekrarlamasını sağlayabilirsiniz. Örneğin; “Çift ya da Tek” programını alıp, aşağıdaki gibi değiştirdiğinizde, program sonsuza kadar çalışacaktır. Pencerenin üst sağ köşesindeki Kapatma (X) düğmesine basarak programı durdurabilirsiniz.

```
1  basla:
2  TextWindow.Write("Bir sayi girin: ")
3  a = TextWindow.ReadNumber()
4  r = Math.Remainder(a, 2)
5  If (r = 0) Then
6    TextWindow.WriteLine("Cift Sayi")
7  Else
8    TextWindow.WriteLine("Tek Sayi")
9  EndIf
10 Goto basla
```



Microsoft®
Small Basic

Döngüler

Döngüler

- ▶ Daha önceki bölümde yazdığımız bir programı ele alalım.

```
1 i = 1
2 listele:
3 TextWindow.WriteLine(i)
4 i = i + 1
5 If (i < 25) Then
6     Goto listele
7 EndIf
```

- ▶ Bu program, 1'den 24'e kadar sayıları sırayla yazdırıyor. Bu bir değişkeni artırma süreci programlamada oldukça yaygın olduğundan, programlama dilleri genellikle bunu yapmak için daha kolay bir yöntem sunarlar.
-



Döngüler

```
1 i = 1
2 listele:
3 TextWindow.WriteLine(i)
4 i = i + 1
5 If (i < 25) Then
6     Goto listele
7 EndIf
```

Yukarıdaki program, aşağıdaki programa eşdeğerdir:

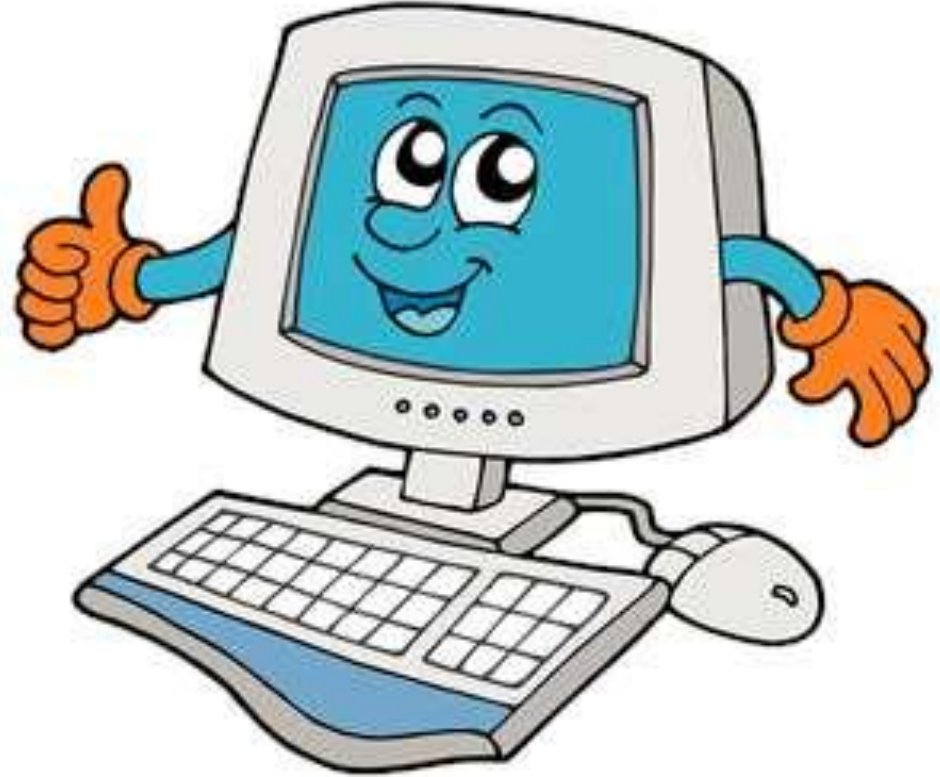
```
1 For i = 1 To 24
2     TextWindow.WriteLine(i)
3 EndFor
```



```
C:\Users\mjperrez\AppData\Local\Temp
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
Press any key to continue...
```


Döngüler

Gördüğünüz gibi, 7 satırlık bir programı 3 satırlık bir programa düşürdük ve yine de 7 satırlık programla aynı işi yapıyor! Daha önce, genellikle aynı şeyi yapmanın çeşitli yolları olduğunu söylediğimizi hatırlayın. İşte bu, harika bir örnek.



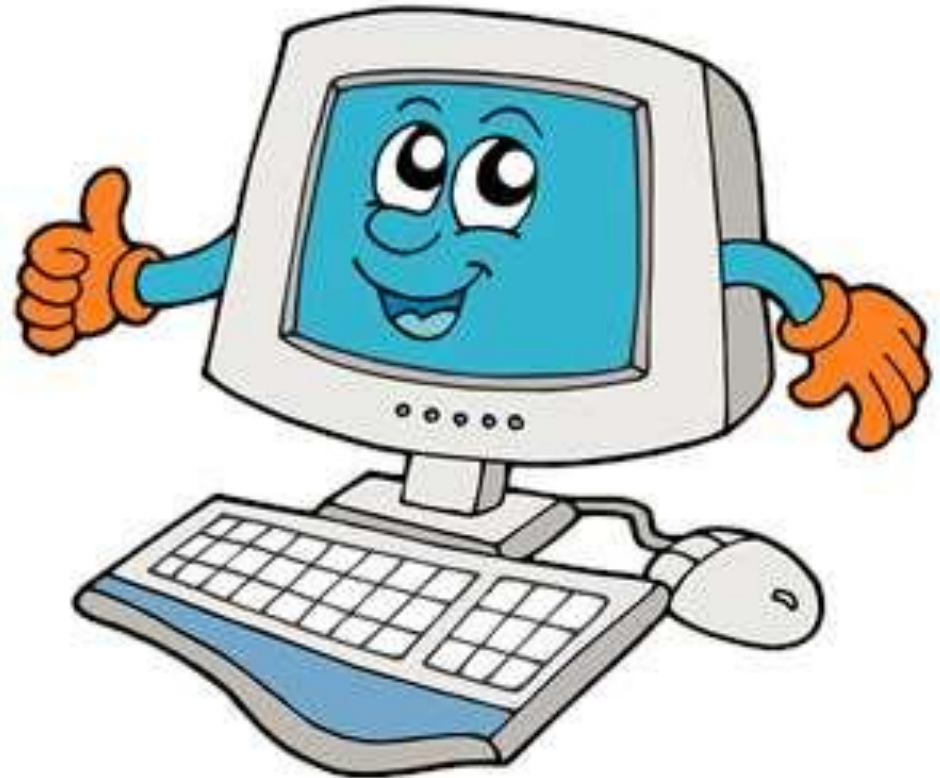
DÖNGÜLER

Kodlarımızı sürekli döndürmemizi sağlamada bize yardımcı olacak parçacıklara **döngü** denir.



For Döngüsü

- ▶ **For..EndFor**, programlama dilinde bir döngü olarak adlandırılır. Bu size, bir değişkeni alıp, ona bir başlangıç ve bitiş noktası vermenizi ve bilgisayarın değişkeni sizin için artırmasını sağlar. Bilgisayar değişkenin değerini her artırdığında, **For** ve **EndFor** arasındaki ifadeleri çalıştırır.



For Döngüsü:

For döngüsü, bir değişkeni alıp, ona bir başlangıç ve bitiş noktası vermenizi ve bilgisayarın değişkeni sizin için artırmasını sağlar.

Bilgisayar değişkenin değerini her artırdığında, **For** ve **EndFor** arasındaki ifadeler çalıştırılır.





-
- ▶ Örnek: 1'den 24'e kadar sayıları alt alta yazdıran program:

```
1 For i = 1 To 24
2   TextWindow.WriteLine(i)
3 EndFor
```



Step

- ▶ Eğer değişkenin birer birer yerine ikişer ikişer artmasını isteseydiniz, 1 ile 24 arasındaki tüm tek sayıları yazdırmak isteyecektiniz, döngüyü bunu yapmak için de kullanabilirsiniz.

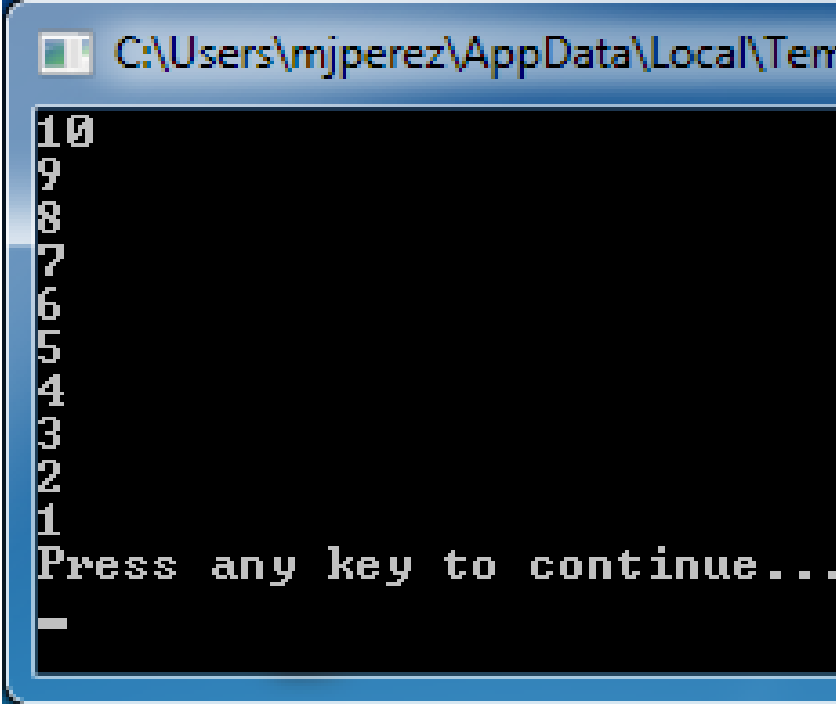
```
1 For i = 1 To 24 Step 2
2   TextWindow.WriteLine(i)
3 EndFor
```

- ▶ İfadenin **Step 2** bölümü, bilgisayara *i*'nin değerini 1 yerine 2 artırmasını söyler.
-



Step

Step'ı kullanarak, istediğiniz aralıklarla artırma yapabilirsiniz. Step için negatif bir değer de belirleyebilirsiniz ve bu durumda bilgisayar, bu örnekte olduğu gibi geriye doğru sayar.



```
C:\Users\mjperrez\AppData\Local\Temp
10
9
8
7
6
5
4
3
2
1
Press any key to continue...
_
```

```
1 For i = 10 To 1 Step -1
2   TextWindow.WriteLine(i)
3 EndFor
```

Step

For döngüsü içinde adım sayısını belirlemek için kullanılır.

```
1 For i = 5 To 30 Step 5
2   TextWindow.WriteLine(i)
3 EndFor
```



Yukarıdaki örnekte 5'ten 30'a kadar 5'er artarak sayılar alt alta yazılır.

While Döngüsü

- ▶ While döngüsü, döngü sayısı önceden bilinmediği zaman faydalıdır. Bir For döngüsü önceden tanımlandığı kadar çalışırken, While döngüsü verilen bir koşul doğru hale gelinceye kadar çalışır. Aşağıdaki örnekte, sonuç 1'den büyük olduğu sürece bir sayıyı ikiye bölüyoruz.

```
1 a = 100
2 While (a > 1)
3     TextWindow.WriteLine(a)
4     a = a / 2
5 EndWhile
```



While Döngüsü

- ▶ Aşağıdaki programda, a'ya 100 değerini atıyoruz ve sayı 1'den büyük olduğu sürece While döngüsünü çalıştırıyoruz. Döngünün içinde, sayıyı yazdırıyoruz ve sonra ikiye bölüp, yarısını buluyoruz. Böylece programın çıktısı, birbiri ardına yarıya inen sayılar oluyor.

```
1 a = 100
2 While (a > 1)
3     TextWindow.WriteLine(a)
4     a = a / 2
5 EndWhile
```

```
100
50
25
12,5
6,25
3,125
1,5625
```



While Döngüsü

- ▶ Bu programı For döngüsünü kullanarak yazmak gerçekten zor olurdu, çünkü döngünün kaç kez çalışması gerektiğini bilemezdik. Bir While döngüsü ile, bir koşulu kontrol etmek ve bilgisayara döngüyü sürdürmesini ya da bırakmasını söylemek kolaydır.

```
1 a = 100
2 While (a > 1)
3     TextWindow.WriteLine(a)
4     a = a / 2
5 EndWhile
```



While

Tüm While döngüleri, bir If..Then ifadesine dönüştürülebilir. Örneğin; yandaki iki program, aynı sonucu verir.

```
1 a = 100
2 While (a > 1)
3     TextWindow.WriteLine(a)
4     a = a / 2
5 Endwhile
```

```
1 a = 100
2 Basla:
3 TextWindow.WriteLine(a)
4 a = a / 2
5 If (a > 1) Then
6     Goto Basla
7 EndIf
```

While Döngüsü:

Verilen bir koşul doğru hale gelinceye kadar çalışır.

```
1 a = 100
2 While (a > 1)
3     TextWindow.WriteLine(a)
4     a = a / 2
5 Endwhile
```

Yukarıdaki örnekte, sonuç 1'den büyük olduğu sürece, önceden belirlenmiş sayı 2'ye bölünüyor.





Microsoft®
Small Basic

Grafiklere Giriş

Grafiklere Giriş

- Şimdiye kadar verdiğimiz tüm örneklerde, Small Basic dilinin temellerini açıklamak için, TextWindow'u kullandık. Bununla birlikte, Small Basic'de bu bölümde araştırmaya başlayacağımız güçlü bir Grafik özelliği seti de bulunur.



GraphicsWindow

- ▶ Tıpkı, metinlerle ve sayılarla çalışmamıza izin veren TextWindow gibi, Small Basic aynı zamanda bir şeyler çizmemizi sağlayan bir **GraphicsWindow** da sunar.

- ⚡ KeyDown
- ⚡ KeyUp
- ⚡ MouseDown
- ⚡ MouseMove
- ⚡ MouseUp
- ⚡ TextInput

- 🎨 BackgroundColor
- 🎨 BrushColor
- 🎨 CanResize
- 🎨 FontBold
- 🎨 FontItalic
- 🎨 FontName
- 🎨 FontSize
- 🎨 Height
- 🎨 LastKey
- 🎨 LastText
- 🎨 Left
- 🎨 MouseX
- 🎨 MouseY
- 🎨 PenColor
- 🎨 PenWidth
- 🎨 Title
- 🎨 Top
- 🎨 Width
- ⚙️ Clear
- ⚙️ DrawBoundText
- ⚙️ DrawEllipse
- ⚙️ DrawImage
- ⚙️ DrawLine
- ⚙️ DrawRectangle
- ⚙️ DrawResizedImage
- ⚙️ DrawText
- ⚙️ DrawTriangle
- ⚙️ FillEllipse
- ⚙️ FillRectangle
- ⚙️ FillTriangle
- ⚙️ GetColorFromPixel
- ⚙️ GetPixel
- ⚙️ GetRandomColor
- ⚙️ Hide
- ⚙️ SetPixel
- ⚙️ Show
- ⚙️ ShowMessage

GraphicsWindow

- ▶ GraphicsWindow'ü görüntüleyerek başlayalım.

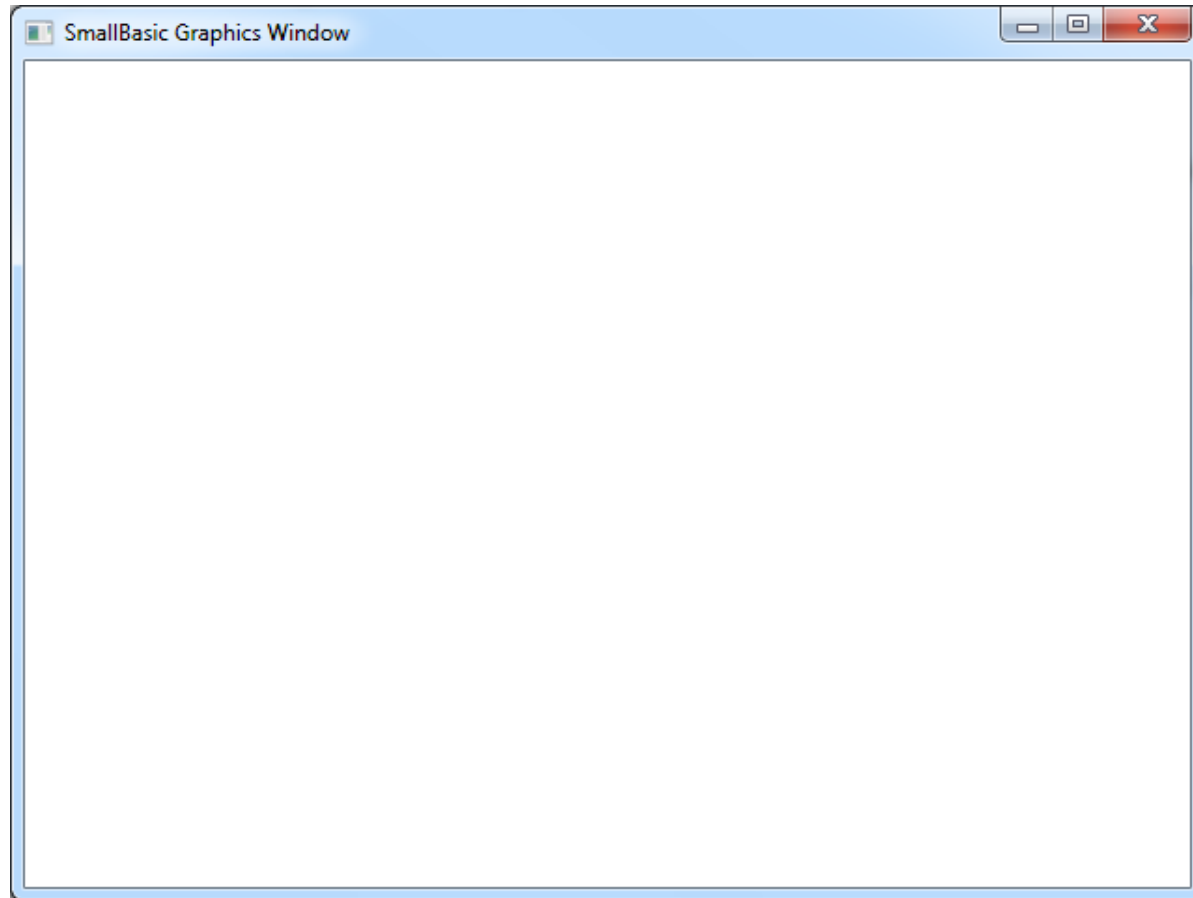
1 GraphicsWindow.Show()

- ▶ Bu programı çalıştırdığınızda, bildiğimiz siyah metin penceresi yerine, beyaz bir pencerenin açıldığını göreceksiniz. Henüz bu pencerede yapacak fazla bir şey yoktur. Ancak, bu bölümde üzerinde çalışacağımız zemin pencere bu olacak. Pencerenin üst sağ köşesindeki “X” düğmesine basarak bu pencereyi kapatabilirsiniz.



GraphicsWindow

```
1 GraphicsWindow.Show()
```



GraphicsWindow:

Grafiklerle çalışarak programı görsel olarak zenginleştirmek için GraphicsWindow'u kullanırız.

GraphicsWindow.Show()
komutu ile grafik penceresini görüntüleriz.



Grafik Penceresinin Kurulumu

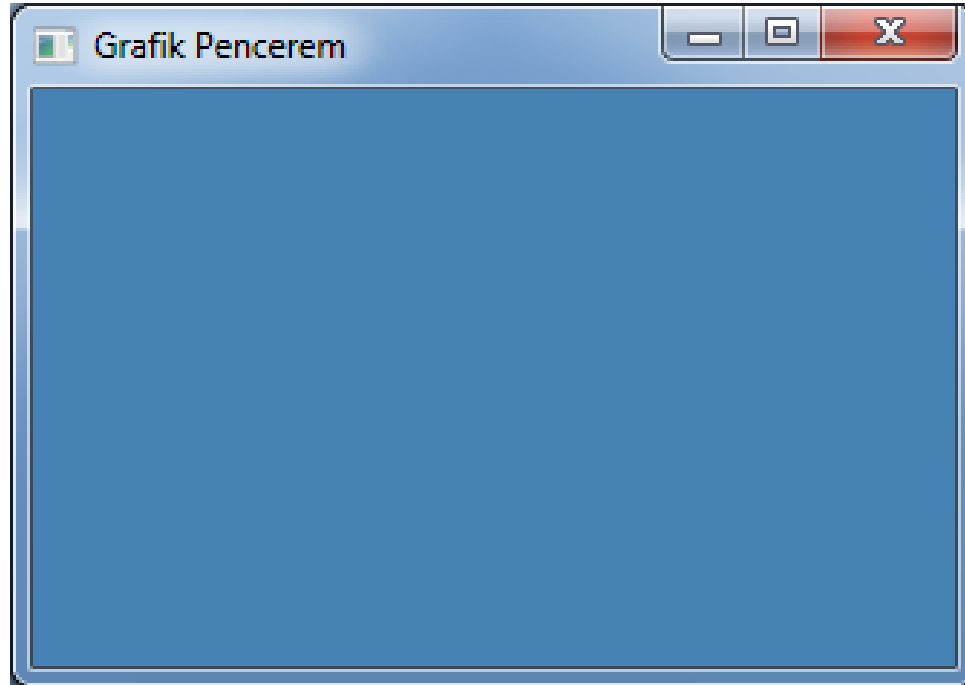
- ▶ Grafik penceresi, görünüşünü isteğiniz gibi ayarlamanıza izin verir. Bu pencerenin başlığını, arka planını ve boyutunu değiştirebilirsiniz. Şimdi devam edelim ve pencereyi daha iyi tanımak için, onu birazcık değiştirelim.

```
1 GraphicsWindow.BackgroundColor = "SteelBlue"  
2 GraphicsWindow.Title = "Grafik Pencerem"  
3 GraphicsWindow.Width = 320  
4 GraphicsWindow.Height = 200
```



Grafik Penceresinin Kurulumu

- ▶ Özelleştirilmiş bir pencere işte böyle görünür. Arka plan rengini, istediğiniz renkten birisiyle değiştirebilirsiniz. Pencerenin görünümünü nasıl değiştirebileceğinizi görmek için, bu özelliklerle oynayın.



Grafik Penceresinin Kurulumu

- ▶ Renkler için isimler kullanmak yerine, internet renk gösterimini de kullanabilirsiniz. Örneğin; #FF0000 Kırmızı renge karşılık gelir, #FFFF00 Sarı renge, vs.

```
1 GraphicsWindow.BackgroundColor = "#8B008B"  
2 GraphicsWindow.Title = "Grafik Pencere"  
3 GraphicsWindow.Width = 320  
4 GraphicsWindow.Height = 200
```

▶ Diğer renkler için [tıklayın](#)

Grafik Penceresini Değiştirme



- ▶ Grafik penceresinin başlığını, arka planını ve boyutunu değiştirmek için aşağıdaki programı yazalım:

```
1 GraphicsWindow.BackgroundColor = "SteelBlue"  
2 GraphicsWindow.Title = "Grafik Pencerem"  
3 GraphicsWindow.Width = 320  
4 GraphicsWindow.Height = 200
```



BackgroundColor:

Arka plan rengini
deęiřtirmek için
kullanılır.



Title:

Grafik penceresinin
başlığını belirtir.



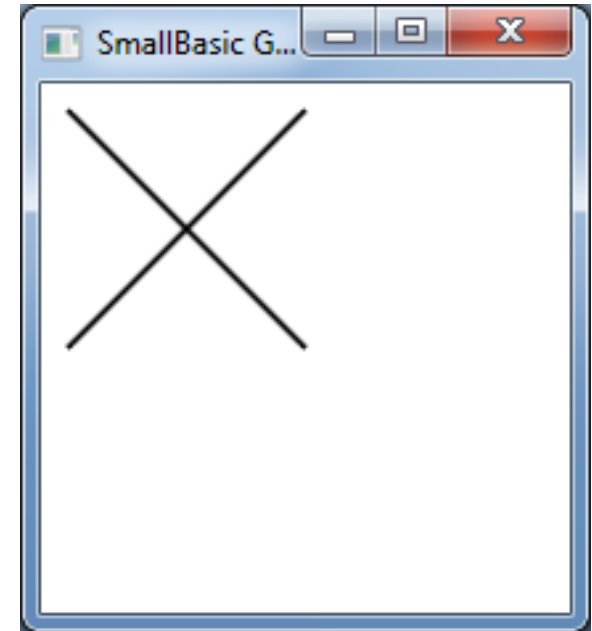
Width/Height :

Grafik penceresinin genişliđi için Width, yüksekliđi için Height kullanılır.



Çizgiler Çizmek

- ▶ GraphicsWindow'u açtıktan sonra, üzerine şekil, metin ve hatta resim çizebiliriz. Bazı basit şekiller çizmekle başlayalım. İşte, Grafik Penceresine bir çift çizgi çizen bir program.



```
1 GraphicsWindow.Width = 200
2 GraphicsWindow.Height = 200
3 GraphicsWindow.DrawLine(10, 10, 100, 100)
4 GraphicsWindow.DrawLine(10, 100, 100, 10)
```



Çizgiler Çizmek

- ▶ Programın ilk iki satırı, pencereyi ayarlar ve ondan sonraki iki satır da çapraz işaretinin çizgilerini çizer. DrawLine'dan sonra gelen ilk iki sayı, başlangıç x ve y koordinatlarını ve diğer ikisi de, bitiş x ve y koordinatlarını belirtir. Bilgisayar grafikleri ile ilgili ilginç olan şey, koordinatların (0, 0) pencerenin üst sol köşesinden başlamasıdır.

```
1 GraphicsWindow.Width = 200
2 GraphicsWindow.Height = 200
3 GraphicsWindow.DrawLine(10, 10, 100, 100)
4 GraphicsWindow.DrawLine(10, 100, 100, 10)
```



Çizgiler Çizmek



- ▶ Grafik penceresine bir çift çizgi çizmek için aşağıdaki programı yazalım:

```
1 GraphicsWindow.Width = 200
2 GraphicsWindow.Height = 200
3 GraphicsWindow.DrawLine(10, 10, 100, 100)
4 GraphicsWindow.DrawLine(10, 100, 100, 10)
```



DrawLine :

Bir noktadan diğesine bir çizgi çizer.

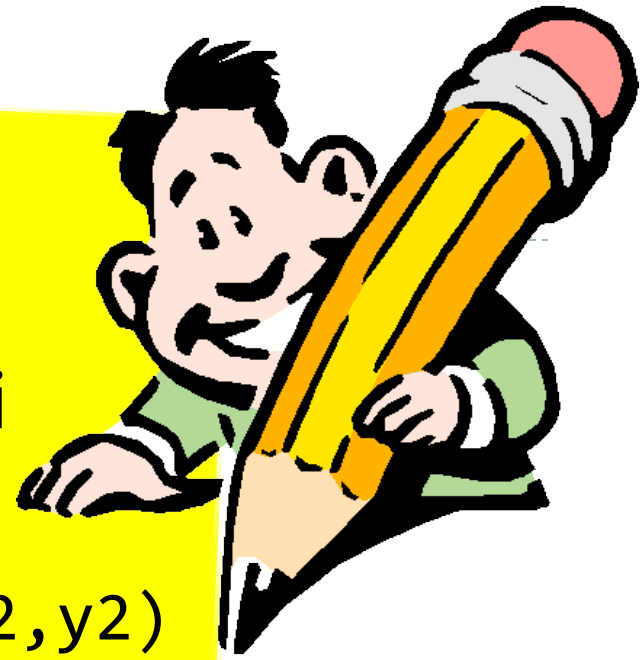
`GraphicWindow.DrawLine(x1,y1,x2,y2)`
şeklinde kullanılır.

x1: İlk noktanın x koordinatı

y1: İlk noktanın y koordinatı

x2: İkinci noktanın x koordinatı

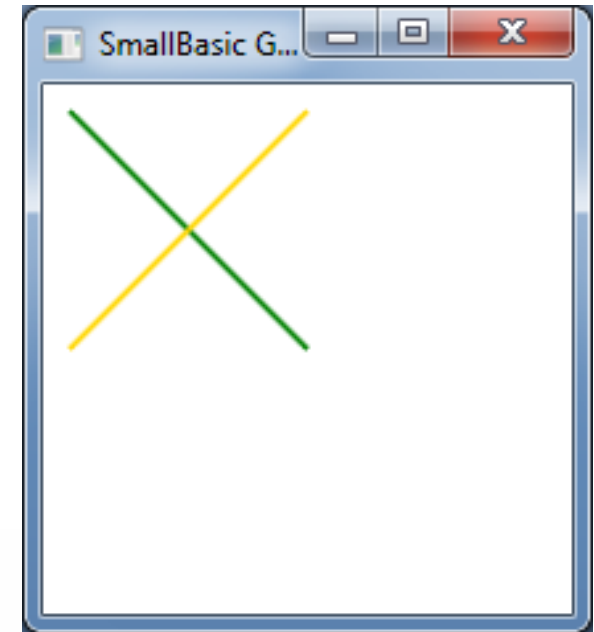
y2: İkinci noktanın y koordinatı



Çizgi Özelliklerini Değiştirmek

- ▶ Çizginin renk ve kalınlık gibi özelliklerini değiştirebiliriz. İlk önce, aşağıdaki programda gösterildiği şekilde, çizgilerin rengini değiştirelim.

```
1 GraphicsWindow.Width = 200
2 GraphicsWindow.Height = 200
3 GraphicsWindow.PenColor = "Green"
4 GraphicsWindow.DrawLine(10, 10, 100, 100)
5 GraphicsWindow.PenColor = "Gold"
6 GraphicsWindow.DrawLine(10, 100, 100, 10)
```



PenColor :

Grafik penceresinde şekil çizmek için kullanılan kalemin rengini belirtir.



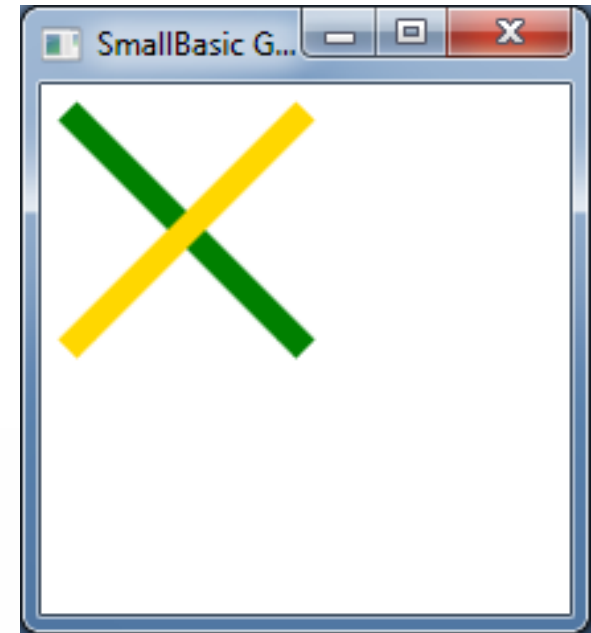
```
GraphicsWindow.PenColor = "Green"  
GraphicsWindow.DrawLine(10, 10, 100, 100)
```

Yukarıdaki örnekte, yeşil renkte çapraz yönde bir çizgi çizilir.

Çizgi Özelliklerini Değiştirmek

- ▶ Şimdi, boyutunu da değiştirelim. Aşağıdaki programda, çizgi kalınlığını varsayılan değer olan 1 yerine, 10 olarak değiştiriyoruz.

```
1 GraphicsWindow.Width = 200
2 GraphicsWindow.Height = 200
3 GraphicsWindow.PenWidth = 10
4 GraphicsWindow.PenColor = "Green"
5 GraphicsWindow.DrawLine(10, 10, 100, 100)
6 GraphicsWindow.PenColor = "Gold"
7 GraphicsWindow.DrawLine(10, 100, 100, 10)
```



PenWidth :

Grafik penceresinde Őekil çizmek için kullanılan kalemin kalınlıđını belirtir.

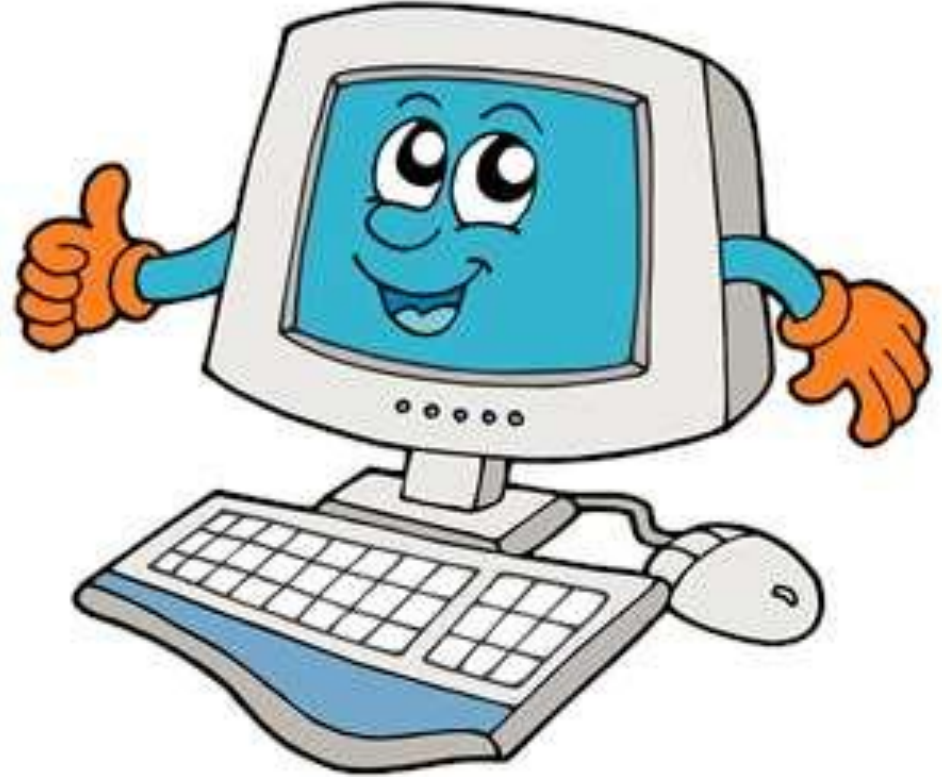


```
GraphicsWindow.PenWidth = 10
```

Yazdıđımız programa yukarıdaki satırı ekleyerek, çizgiyi 10 kat kalınlařtırırız.

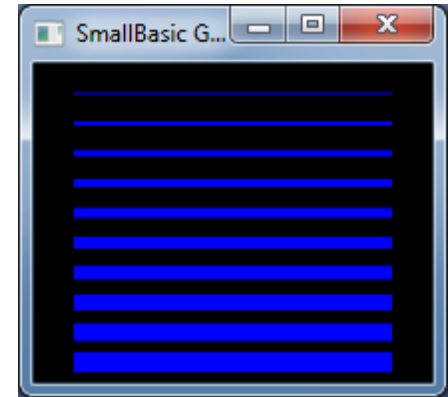
Çizgi Özelliklerini Deęiřtirmek

- ▶ *PenWidth* ve *PenColor*, bu çizgilerin çizildięi kalemi deęiřtirir. Bunlar yalnızca çizgileri deęil, özellikler güncellendikten sonra çizilen tüm şekilleri etkilerler.



Çizgi Özelliklerini Değiştirmek

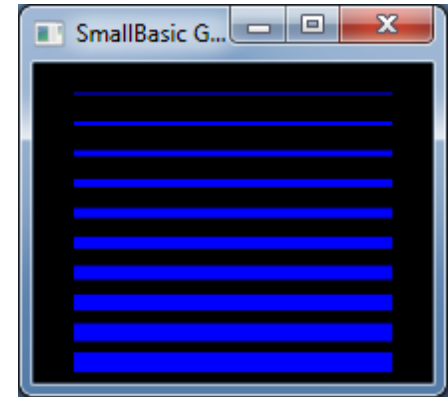
- ▶ Daha önceki bölümlerde öğrendiğimiz döngü yapan ifadeleri kullanarak, kalem kalınlığı gittikçe artan birden fazla çizgi çizebiliriz.



```
1 GraphicsWindow.BackgroundColor = "Black"
2 GraphicsWindow.Width = 200
3 GraphicsWindow.Height = 160
4 GraphicsWindow.PenColor = "Blue"
5
6 For a = 1 To 10
7     GraphicsWindow.PenWidth = a
8     GraphicsWindow.DrawLine(20, a * 15, 180, a * 15)
9 endfor
```

Çizgi Özelliklerini Değiştirmek

- ▶ Bu programın ilginç bölümü; döngü her çalıştığında PenWidth'in artması ve sonra eskisinin altına yeni bir çizgi çizmesidir.



```
1 GraphicsWindow.BackgroundColor = "Black"
2 GraphicsWindow.Width = 200
3 GraphicsWindow.Height = 160
4 GraphicsWindow.PenColor = "Blue"
5
6 For a = 1 To 10
7     GraphicsWindow.PenWidth = a
8     GraphicsWindow.DrawLine(20, a * 15, 180, a * 15)
9 endfor
```

Örnek:

- ▶ Kalem kalınlığı gittikçe artan birden fazla çizgi çizelim.

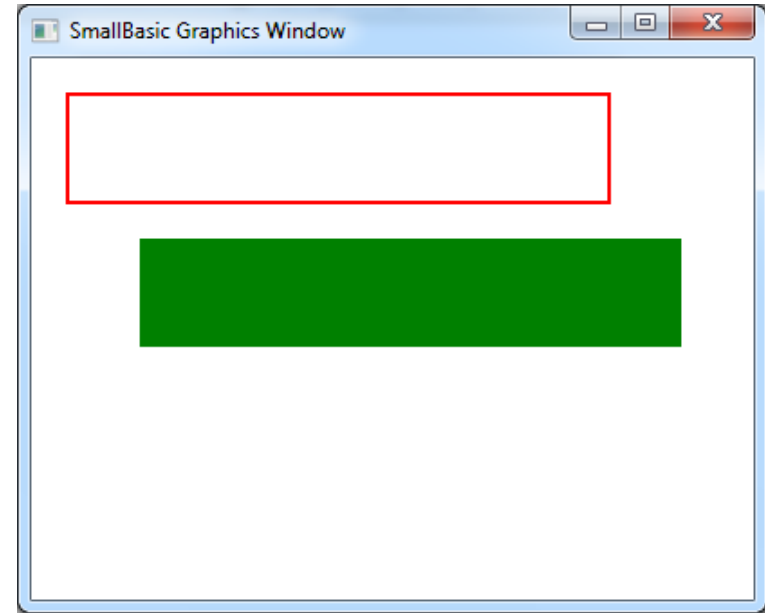


```
1 GraphicsWindow.BackgroundColor = "Black"
2 GraphicsWindow.Width = 200
3 GraphicsWindow.Height = 160
4 GraphicsWindow.PenColor = "Blue"
5
6 For a = 1 To 10
7     GraphicsWindow.PenWidth = a
8     GraphicsWindow.DrawLine(20, a * 15, 180, a * 15)
9 endfor
```



Şekiller Çizmek ve İçlerini Doldurmak

- ▶ İş şekiller çizmeye geldiğinde, her şekil için genellikle iki tip işlem vardır. Bunlar, *Çizme* ve *İçini Doldurma* işlemleridir. Çizme işlemleri, bir kalem kullanarak şeklin dış çerçevesini çizer ve İçini Doldurma işlemleri de, bir fırça kullanarak şekli boyar. Örneğin; yandaki programda, iki adet dikdörtgen var, bunlardan birisi kırmızı bir kalem kullanılarak çizilmiş ve diğeri de Yeşil Fırça kullanılarak içi doldurulmuş.



Şekiller Çizmek ve İçlerini Doldurmak

- ▶ Bir dikdörtgen çizmek veya içini doldurmak için, dört sayıya ihtiyacınız vardır. İlk iki sayı, dikdörtgenin üst sol köşesinin X ve Y koordinatlarını temsil eder. Üçüncü sayı, dikdörtgenin genişliğini, dördüncü ise yüksekliğini belirtir.

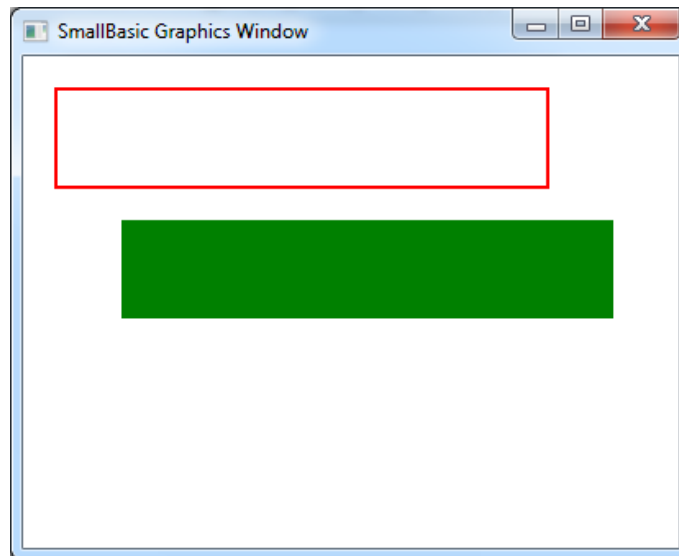
```
GraphicsWindow.PenColor = "Red"  
GraphicsWindow.DrawRectangle(20, 20, 300, 60)  
GraphicsWindow.BrushColor = "Green"  
GraphicsWindow.FillRectangle(60, 100, 300, 60)
```



Dikdörtgen Çizmek ve İçini Doldurmak



```
GraphicsWindow.PenColor = "Red"  
GraphicsWindow.DrawRectangle(20, 20, 300, 60)  
GraphicsWindow.BrushColor = "Green"  
GraphicsWindow.FillRectangle(60, 100, 300, 60)
```



BrushColor:

Şekilleri doldurmak için kullanılan fırça rengini belirtir.



DrawRectangle:

Seçilen kalem ile ekranda dikdörtgen çizer.

`DrawRectangle(x, y, width, height)` şeklinde kullanılır.

x ve y başlangıç noktalarını, width dikdörtgenin genişliğini, height yüksekliğini belirtir.



FillRectangle:

Seçilen fırça ile ekrandaki dikdörtgeni doldurur. Kullanım şekli DrawRectangle ile benzerdir.



Elips Çizmek ve İçini Doldurmak

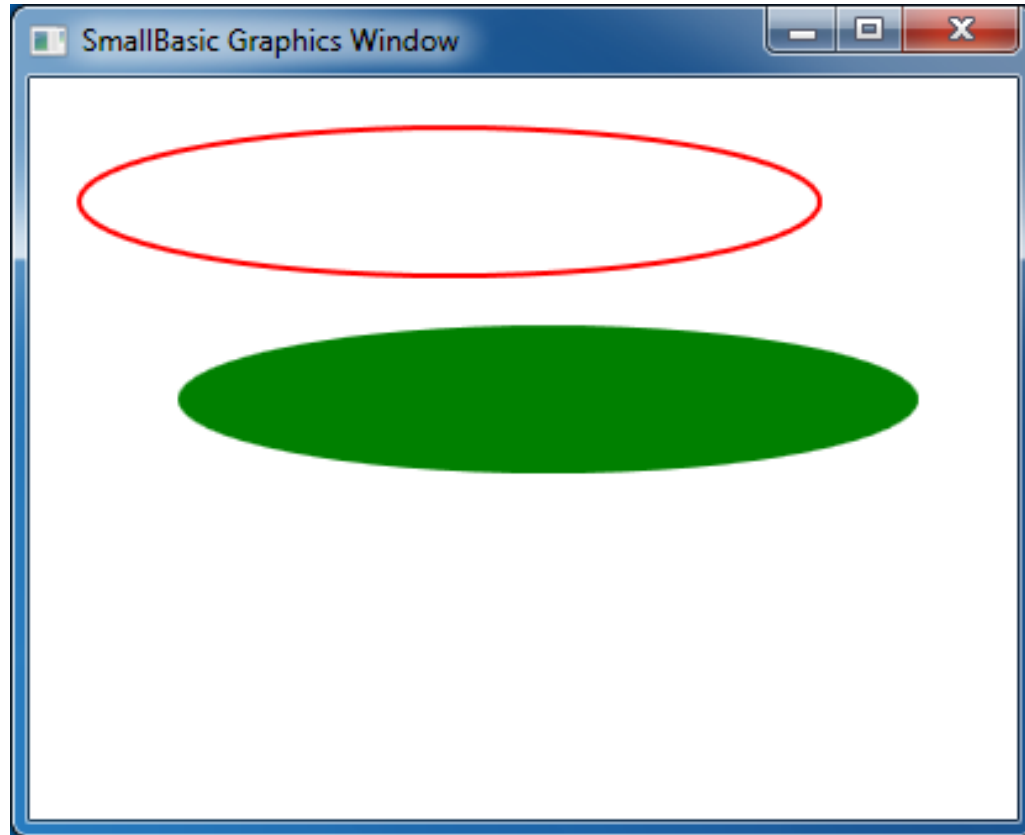
- ▶ Programdaki elipslerin çizilmesi ve içlerinin doldurulması da dikdörtgen ile benzerdir.

```
1 GraphicsWindow.Width = 400
2 GraphicsWindow.Height = 300
3 GraphicsWindow.PenColor = "Red"
4 GraphicsWindow.DrawEllipse(20, 20, 300, 60)
5 GraphicsWindow.BrushColor = "Green"
6 GraphicsWindow.FillEllipse (60, 100, 300, 60)
```



Elips Çizmek ve İçini Doldurmak

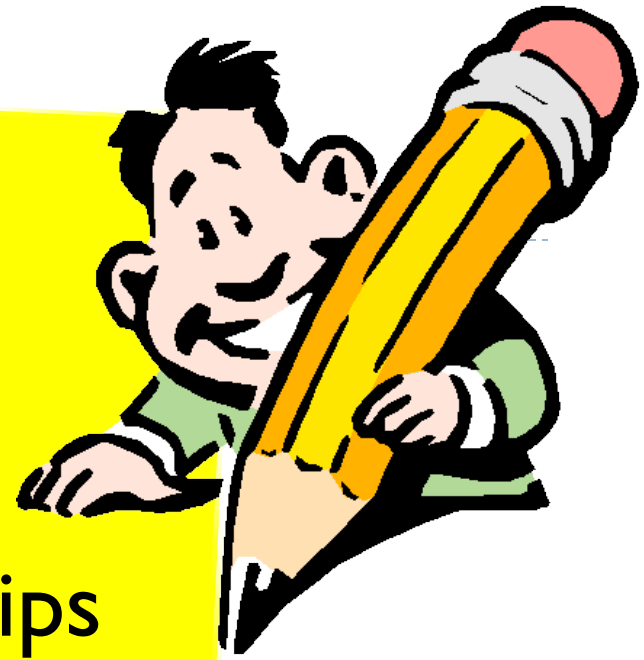
- ▶ Programdaki elipslerin çizilmesi ve içlerinin doldurulması da dikdörtgen ile benzerdir.



DrawEllipse:

Seçilen kalem ile ekranda elips çizer.

DrawEllipse(x, y, width, height) şeklinde kullanılır.



FillEllipse:

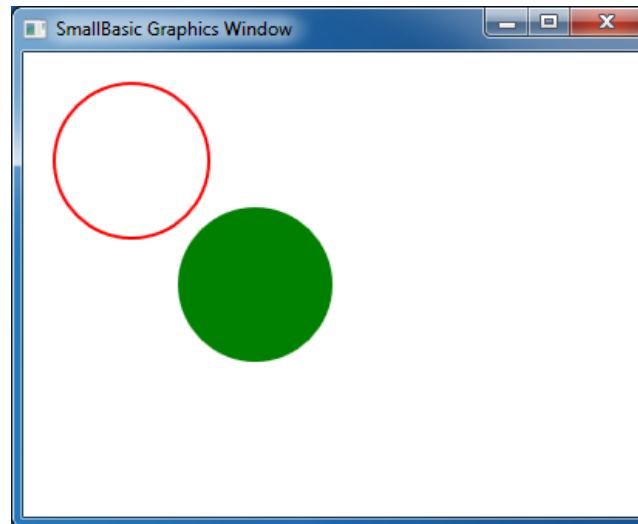
Seçilen fırça ile
ekrandaki elipsi
doldurur.



Elips Çizmek ve İçini Doldurmak

- ▶ Elipsler, yalnızca genel bir daire biçimidir. Daireler çizmek isterseniz, aynı genişliği ve yüksekliği belirtmeniz gerekir.

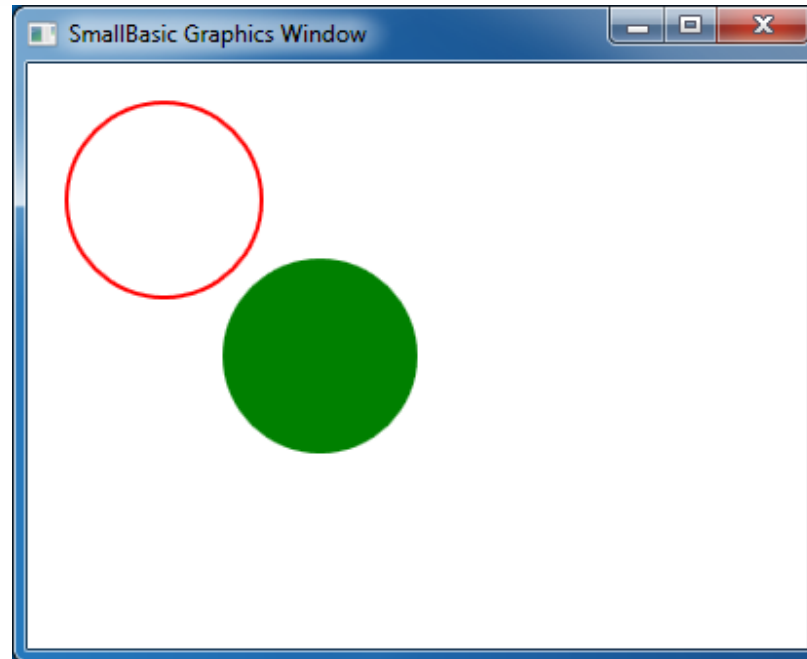
```
1 GraphicsWindow.PenColor = "Red"  
2 GraphicsWindow.DrawEllipse(20, 20, 100, 100)  
3 GraphicsWindow.BrushColor = "Green"  
4 GraphicsWindow.FillEllipse(100, 100, 100, 100)
```



Elips Çizmek ve İçini Doldurmak



```
1 GraphicsWindow.PenColor = "Red"  
2 GraphicsWindow.DrawEllipse(20, 20, 100, 100)  
3 GraphicsWindow.BrushColor = "Green"  
4 GraphicsWindow.FillEllipse(100, 100, 100, 100)
```



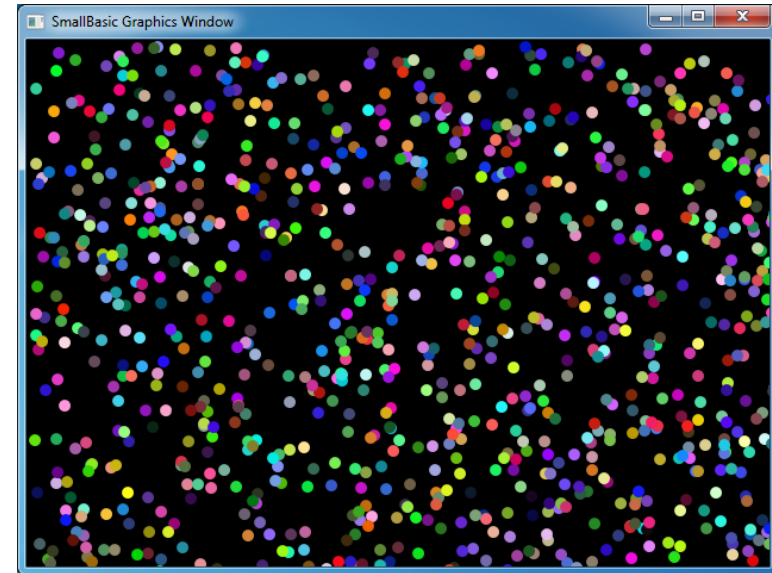
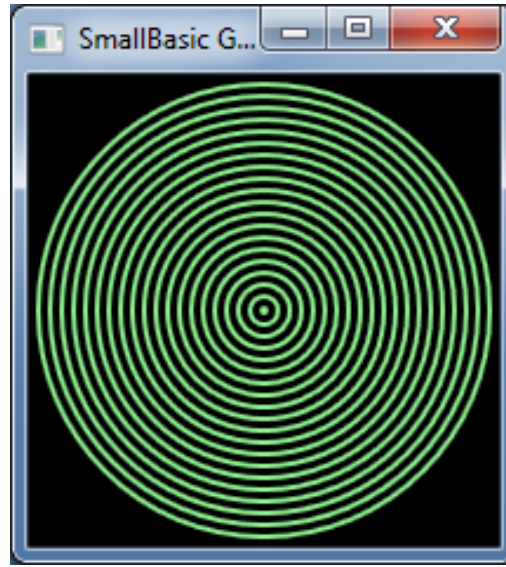
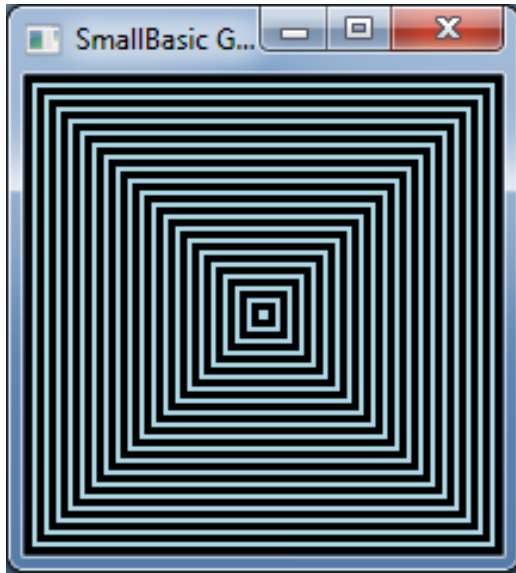


Microsoft®
Small Basic

Şekillerle Eğlence

Şekillerle Eğlence

- Bu bölümde, şu ana kadar öğrendiklerimizle biraz eğleneceğiz. Bu bölüm, bazı hoş görünümlü programlar yaratmak için, şu anda kadar öğrendiklerinizi birleştirmenin bazı ilginç yöntemlerini gösteren örnekler içerir.

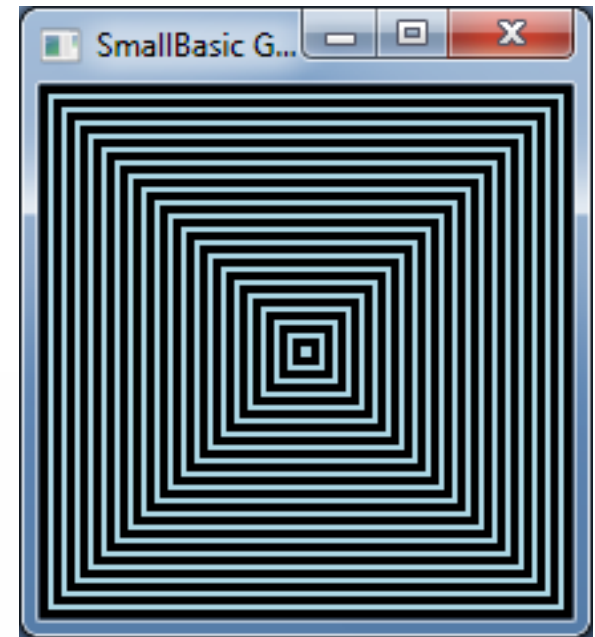


İç İçe Kareler



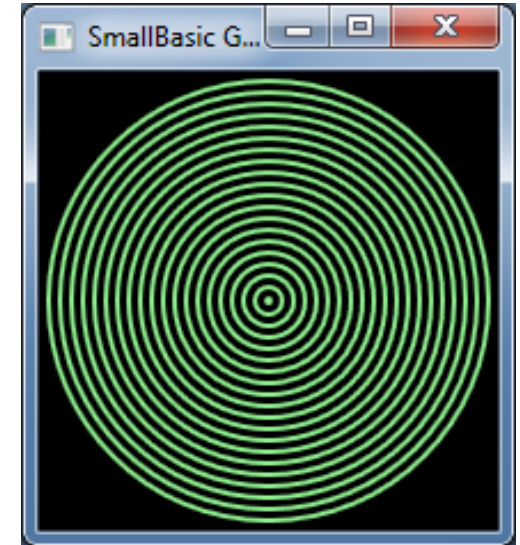
- Bir döngü içerisinde boyutları gittikçe artan kareler çizelim.

```
1 GraphicsWindow.BackgroundColor = "Black"  
2 GraphicsWindow.PenColor = "LightBlue"  
3 GraphicsWindow.Width = 200  
4 GraphicsWindow.Height = 200  
5 For i = 1 To 100 Step 5  
6   GraphicsWindow.DrawRectangle(100 - i, 100 - i, i * 2, i * 2)  
7 EndFor
```



İç İçe Daireler

- ▶ Bir önceki programa benzer olarak bu program, kareler yerine daireler çizer.



```
1 GraphicsWindow.BackgroundColor = "Black"
2 GraphicsWindow.PenColor = "LightGreen"
3 GraphicsWindow.Width = 200
4 GraphicsWindow.Height = 200
5 For i = 1 To 100 Step 5
6     GraphicsWindow.DrawEllipse(100 - i, 100 - i, i * 2, i * 2)
7 EndFor
```



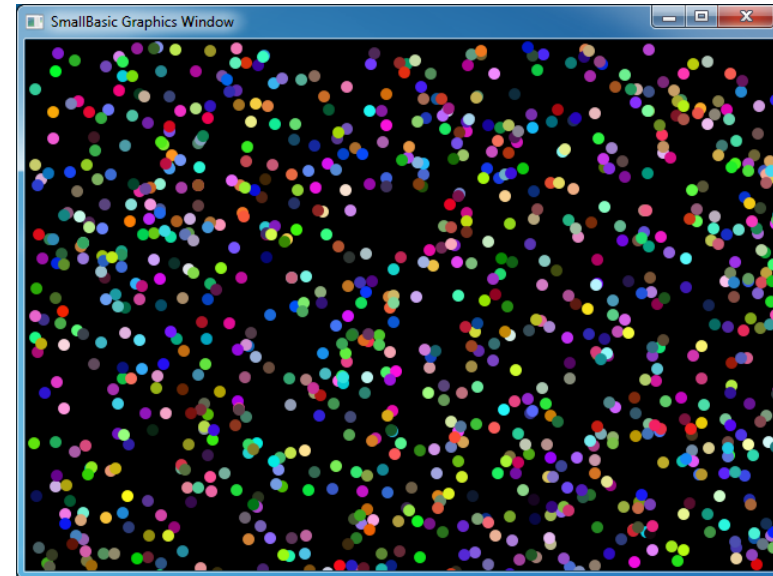
İç İçe Daireler

- ▶ Az önce yazdığımız programda “DrawRectangle” yerine “DrawEllipse” yazarsak, kare yerine daire çizmiş oluruz.



Rastgele Daireler

Bu program, fırçanın rengini rastgele seçmek için `GraphicsWindow.GetRandomColor` işlemini ve sonra da dairelerin x ve y koordinatlarını seçmek için `Math.GetRandomNumber` işlemini kullanır. Bu iki işlem, çalıştıkları her seferde farklı sonuçlar verir.



```
1 GraphicsWindow.BackgroundColor = "Black"
2 For i = 1 To 1000
3     GraphicsWindow.BrushColor = GraphicsWindow.GetRandomColor()
4     x = Math.GetRandomNumber(640)
5     y = Math.GetRandomNumber(480)
6     GraphicsWindow.FillEllipse(x, y, 10, 10)
7 EndFor
```


Rastgele Daireler



```
1 GraphicsWindow.BackgroundColor = "Black"  
2 For i = 1 To 1000  
3     GraphicsWindow.BrushColor = GraphicsWindow.GetRandomColor()  
4     x = Math.GetRandomNumber(640)  
5     y = Math.GetRandomNumber(480)  
6     GraphicsWindow.FillEllipse(x, y, 10, 10)  
7 EndFor
```





GetRandomColor:

Rastgele bir renk
seçer.



GetRandomNumber:

1 ile belirtilen sayı aralığında rastgele sayı üretir.



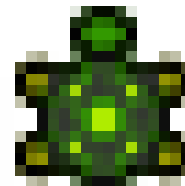
Microsoft®
Small Basic

Turtle Graphics

Turtle

- ▶ Small Basic'de, programların içerisindeki programlardan çağrılabilen pek çok komuta sahip olan bir **Turtle** (Kaplumbağa) nesnesi bulunur.
- ▶ Bu bölümde, kaplumbağayı ekranda grafikler çizmek için kullanacağız.
- ▶ Başlangıç olarak, kaplumbağanın ekranda görünür hale gelmesini sağlamamız gerekiyor. Bu tek satırlık basit bir programla yapılabilir.

```
1 Turtle.Show()
```




 Angle


 Speed

 X


 Y


 Hide

 Move


 MoveTo


 PenDown

 PenUp

 Show

 Turn

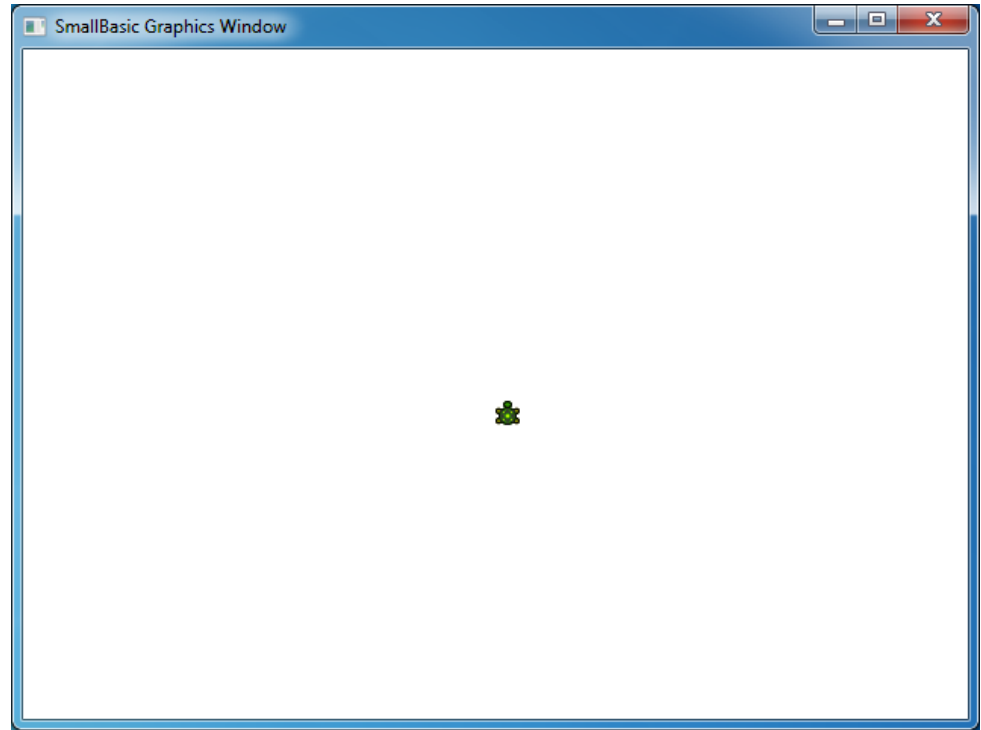
 TurnLeft

 TurnRight

Turtle

- ▶ Bu programı çalıştırdığınızda, merkezinde bir kaplumbağa bulunması dışında, tıpkı daha önceki bölümde gördüğümüz gibi beyaz bir pencerenin açıldığını göreceksiniz. Talimatlarımızı izleyecek ve çizmesini istediğimiz şeyi çizecek olan kaplumbağa budur.

```
1 Turtle.Show()
```



TURTLE:

Ekranda grafikler çizmek için kaplumbağa kullanacağız.

Kaplumbağanın ekranda görünmesi için aşağıdaki kodu yazalım:

```
1 Turtle.Show()
```



Hareket Ettirmek ve Şekiller Çizmek

- ▶ Kaplumbağanın anladığı talimatlardan birisi **Move**'dur. Bu işlem bir sayıyı girdi olarak alır. Bu sayı, kaplumbağaya ne kadar uzağa gitmesi gerektiğini söyler. Diyelim ki; aşağıdaki örnekte kaplumbağaya 100 piksel hareket etmesini söyleyeceğiz.



```
1 Turtle.Move(100)
```



Hareket Ettirmek ve Şekiller Çizmek

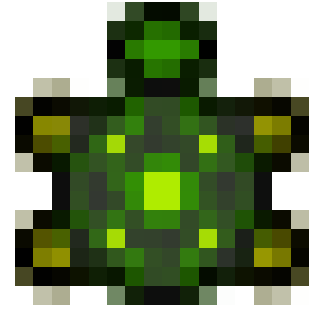
- ▶ Bu programı çalıştırdığınızda, kaplumbağanın yukarıya doğru yavaşça 100 piksel hareket ettiğini görebilirsiniz. Hareket ettikçe, arkasında bir çizgi çizdiğini de fark edeceksiniz. Kaplumbağa hareket etmeyi bitirdiğinde, sonuç yandaki şekildeki gibi olacaktır.



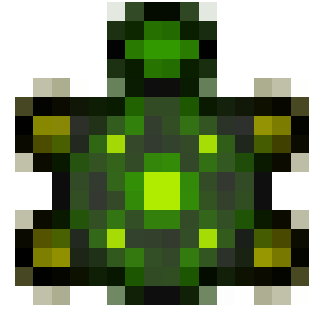
```
1 Turtle.Move(100)
```



Kaplumbağa üzerinde işlemler yaparken Show() komutunu kullanmaya gerek yoktur. Ne zaman bir kaplumbağa işlemi gerçekleştirilirse, kaplumbağa otomatik olarak görünür hale gelecektir.



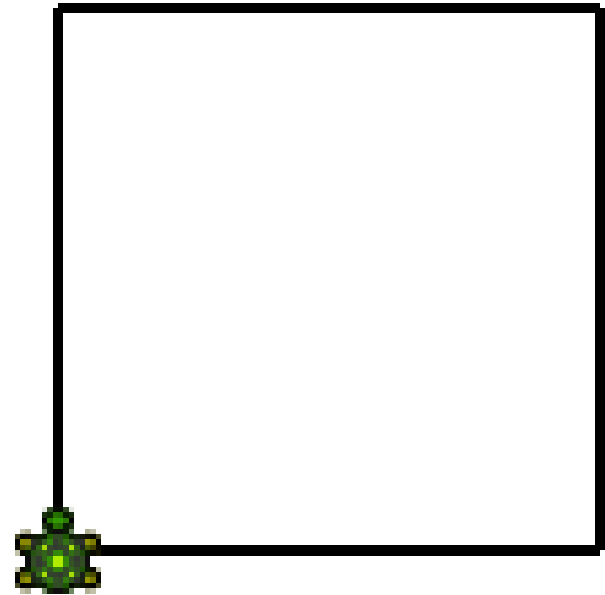
Kaplumbağa, daha önceki bölümde gördüğümüzle aynı GraphicsWindow'da çizim yapar. Bu da, önceki bölümde öğrendiğimiz tüm işlemlerin burada da geçerli olduğu anlamına gelir.



Bir Kare Çizmek

- ▶ Bir kare çizmek için, kaplumbağanın bir çizgi çizmesini, sağa dönmesini ve bir başka çizgi çizmesini ve dört kenar da tamamlanıncaya kadar bu işleme devam etmesini sağlamamız gerekir.

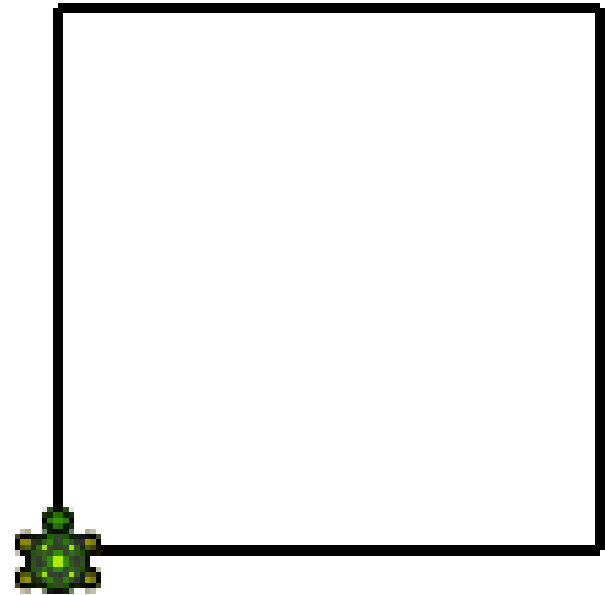
```
1 Turtle.Move(100)
2 Turtle.TurnRight()
3 Turtle.Move(100)
4 Turtle.TurnRight()
5 Turtle.Move(100)
6 Turtle.TurnRight()
7 Turtle.Move(100)
8 Turtle.TurnRight()
```



Bir Kare Çizmek

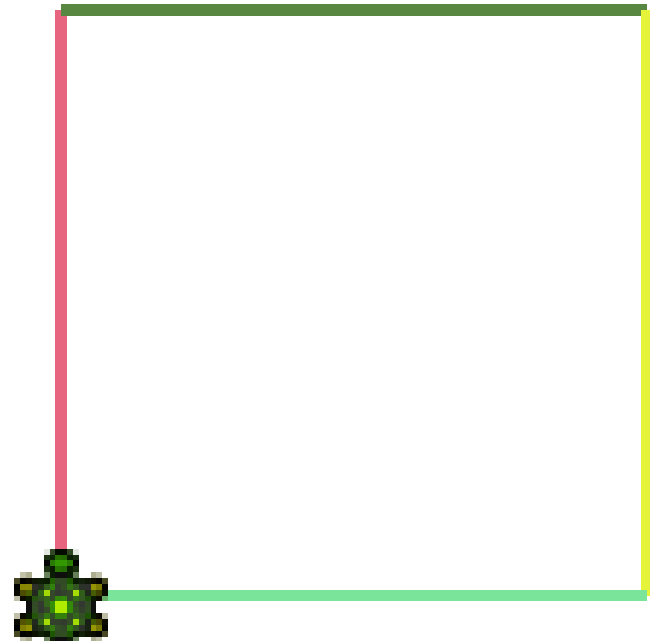
- ▶ Aynı talimatları tekrar tekrar yazmaktansa, tekrarlanan komutların döngü kullanılarak uygulanabileceğini öğrenmiştik. Yani, aynı programı **For..EndFor** döngüsünü kullanarak şekilde değiştirirsek, sonuçta ortaya çok daha basit bir program çıkacaktır.

```
1 For i = 1 To 4
2   Turtle.Move(100)
3   Turtle.TurnRight()
4 EndFor
```



Renkleri Değiştirmek

- ▶ Aynı programa aşağıdaki 2. satırı ekleyerek, karenin her bir kenarını farklı renkte çizmesini sağlayabiliriz.



```
1 For i = 1 To 4
2   GraphicsWindow.PenColor = GraphicsWindow.GetRandomColor()
3   Turtle.Move(100)
4   Turtle.TurnRight()
5 EndFor
```



Renkli Bir Kare Çizmek

- ▶ Kaplumbağaya, her kenarı farklı renkte bir kare çizdiren program:



```
1 For i = 1 To 4
2   GraphicsWindow.PenColor = GraphicsWindow.GetRandomColor()
3   Turtle.Move(100)
4   Turtle.TurnRight()
5 EndFor
```



Move:

Kaplumbağayı belirli bir uzaklığa götürür ve kalem kapatılmadığı sürece her harekette çizim yapılır.



TurnRight:

Kaplumbağa 90 derece sağına döner.

TurnLeft:

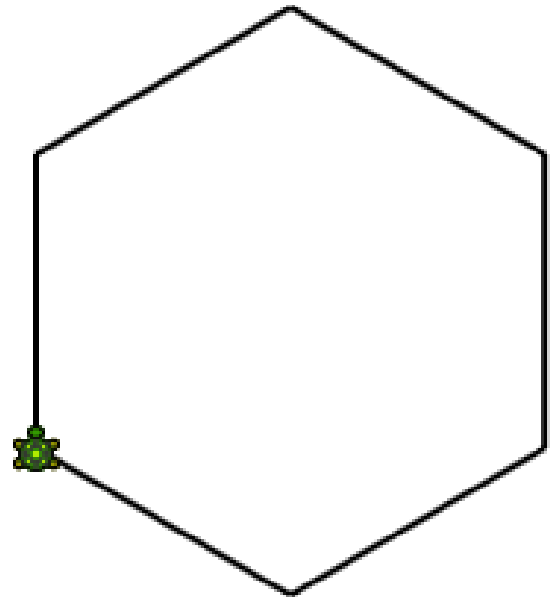
90 derece sola döner.



Altıgen Çizmek

- ▶ **TurnRight** ve **TurnLeft** işlemlerine ek olarak, kaplumbağanın bir de **Turn** işlemi vardır. Bu işlem, dönme açısını belirten bir girdi alır. Altıgenin kenarlarının arasındaki açı 60 derece olduğu için, **Turn(60)** komutunu kullandığımıza dikkat edin.

```
1 For i = 1 To 6
2   Turtle.Move(100)
3   Turtle.Turn(60)
4 EndFor
```



Altıgen Çizmek

- ▶ Kaplumbağaya altıgen çizdiren program:



```
1 For i = 1 To 6
2   Turtle.Move(100)
3   Turtle.Turn(60)
4 EndFor
```



Turn:

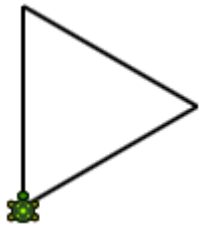
Kaplumbağayı belirli bir açıda döndürür. Açı derece cinsindedir ve pozitif ya da negatif olabilir. Eğer açı pozitifse kaplumbağa sağına, negatifse soluna döner.



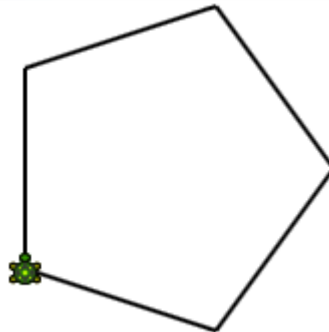
Çokgen Çizmek

- ▶ Tüm kenarları eşit olan çokgenler için, 360 sayısı kenar adedine bölünerek, kenarlar arasındaki açı kolayca elde edilebilir. Bu işlem kullanılarak, herhangi bir çokgeni çizmek mümkündür.

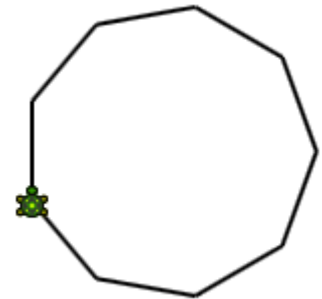
```
1 For i = 1 To 3
2   Turtle.Move(100)
3   Turtle.Turn(120)
4 EndFor
```



```
1 For i = 1 To 5
2   Turtle.Move(100)
3   Turtle.Turn(72)
4 EndFor
```



```
1 For i = 1 To 9
2   Turtle.Move(50)
3   Turtle.Turn(40)
4 EndFor
```



Çokgen Çizmek

Aynı işlemi, sadece kenar uzunluğunu değiştirerek yapabileceğimiz programı şu şekilde yazarız:

```
1 kenar = 5
2 uzunluk = 400 / kenar
3 aci = 360 / kenar
4 For i = 1 To kenar
5     Turtle.Move(uzunluk)
6     Turtle.Turn(acı)
7 EndFor
```

- ▶ Örneğin; **kenar** değişkenini 4 ile değiştirerek bir kare çizileceği gibi, 50 gibi yeterince büyük bir değer girmek, sonucun bir daireye çok benzemesine neden olacaktır.
-

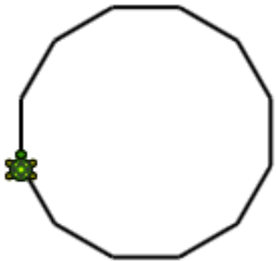


Çokgen Çizmek

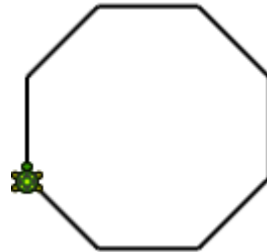
```
1 kenar = 5
2 uzunluk = 400 / kenar
3 aci = 360 / kenar
4 For i = 1 To kenar
5     Turtle.Move(uzunluk)
6     Turtle.Turn(aci)
7 EndFor
```

Sadece kenar uzunluğunu değiştirerek alınabilecek sonuçlara örnekleri görüyoruz.

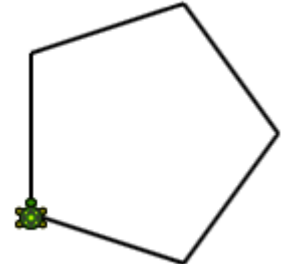
kenar = 12



kenar = 8



kenar = 5



Çokgen Çizmek

Kenar sayısını değiştirerek farklı çokgenler çizdirebileceğimiz program:



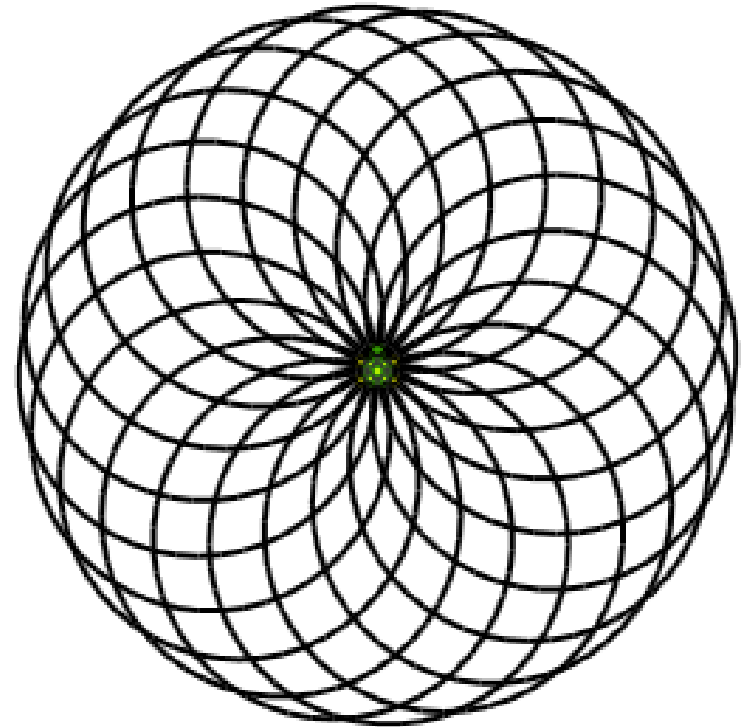
```
1 kenar = 5
2 uzunluk = 400 / kenar
3 aci = 360 / kenar
4 For i = 1 To kenar
5     Turtle.Move(uzunluk)
6     Turtle.Turn(acı)
7 EndFor
```



Karmaşık Şekiller Çizmek

- Biraz önce öğrendiğimiz tekniği kullanarak, ilginç bir sonuç elde edecek şekilde kaplumbağanın her seferinde biraz kayarak birden fazla daire çizmesini sağlayabiliriz.

```
1 kenar = 50
2 uzunluk = 400 / kenar
3 aci = 360 / kenar
4 Turtle.Speed = 9
5 For j = 1 To 20
6     For i = 1 To kenar
7         Turtle.Move(uzunluk)
8         Turtle.Turn(aci)
9     EndFor
10    Turtle.Turn(18)
11 EndFor
```



Karmaşık Şekiller Çizmek

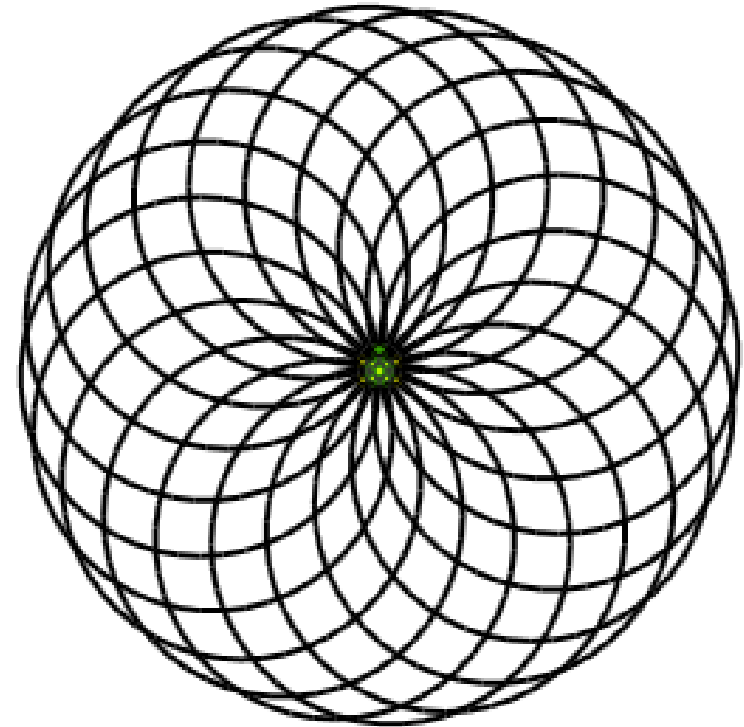
- ▶ Yandaki programda, birisi diğerinin içerisinde iki adet **For..EndFor** döngüsü vardır. İçteki döngü ($i = 1$ to kenar), çokgen programına benzer ve bir daire çizer. Dıştaki döngü, ($j = 1$ to 20) çizilen her bir daire için, kaplumbağayı biraz döndürür. Bu, kaplumbağaya 20 adet daire çizmesini söyler.

```
1 kenar = 50
2 uzunluk = 400 / kenar
3 aci = 360 / kenar
4 Turtle.Speed = 9
5 For j = 1 To 20
6     For i = 1 To kenar
7         Turtle.Move(uzunluk)
8         Turtle.Turn(acı)
9     EndFor
10    Turtle.Turn(18)
11 EndFor
```

Karmaşık Şekiller Çizmek

- ▶ Kaplumbağanın her seferinde biraz kayarak birden fazla daire çizdiren program:

```
1 kenar = 50
2 uzunluk = 400 / kenar
3 aci = 360 / kenar
4 Turtle.Speed = 9
5 For j = 1 To 20
6     For i = 1 To kenar
7         Turtle.Move(uzunluk)
8         Turtle.Turn(aci)
9     EndFor
10    Turtle.Turn(18)
11 EndFor
```



Speed:

Kaplumbağanın hareket hızını belirler. Hız değeri 1 ile 10 arasındadır.

Değer 10 olduğunda, kaplumbağa anında hareket eder ve döner.



Çevrede Dolaşmak

PenUp işlemi, kaplumbağayı çizim yapmadan ekranda herhangi bir yere hareket ettirebilmenizi sağlar. **PenDown** işlemi ile kaplumbağa tekrar çizmeye başlar. Bu, noktalı çizgiler gibi bazı ilginç efektler elde etmek için kullanılabilir. Aşağıdaki programı inceleyelim:



```
1 For i = 1 To 5
2     Turtle.Move(20)
3     Turtle.PenUp()
4     Turtle.Move(20)
5     Turtle.PenDown()
6 EndFor
```

|
|
|
|
|
|



Kesik Çizgiler Çizmek

```
1 For i = 1 To 5
2   Turtle.Move(20)
3   Turtle.PenUp()
4   Turtle.Move(20)
5   Turtle.PenDown()
6 EndFor
```



PenUp:

Kaplumbağa hareket ederken çizim yapmasını durdurur.

PenDown:

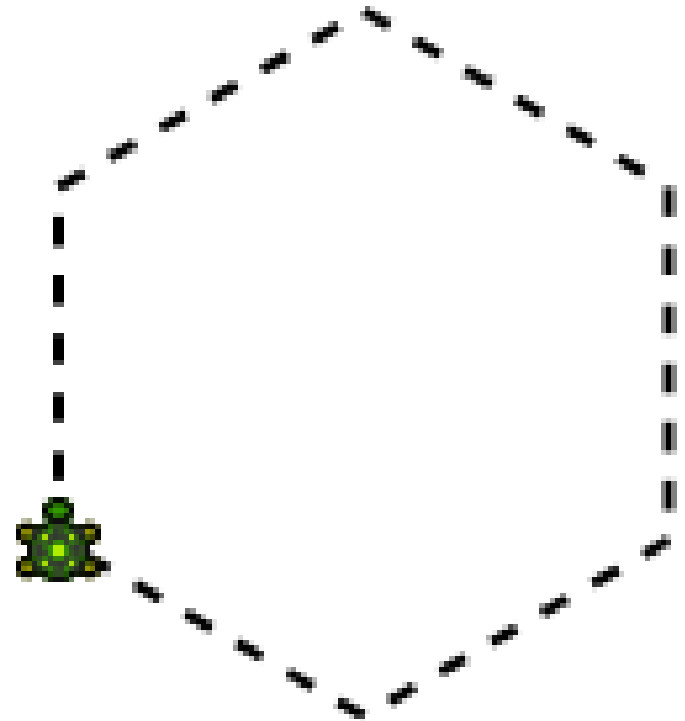
Kaplumbağa hareket ederken çizim yapması için kalemi aşağı indirir.

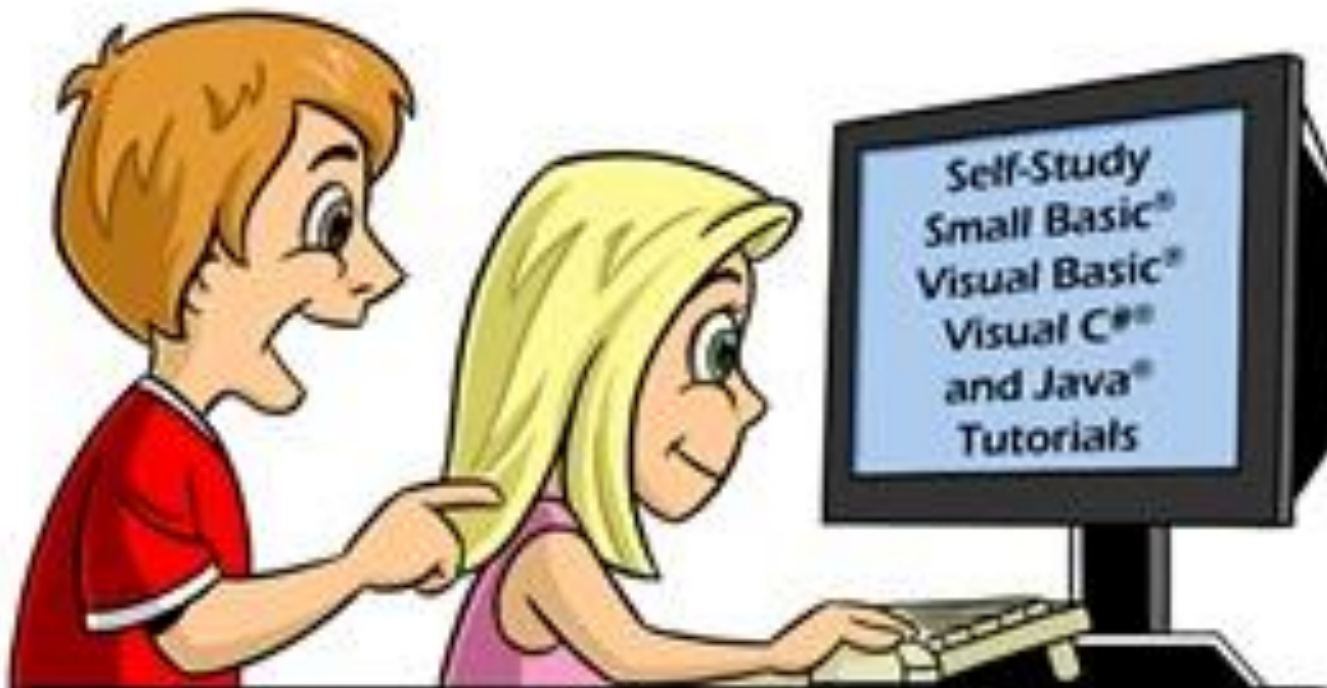


Kesik Çizgili Çokgen

- Bu programda da iki döngü var. İçteki döngü, tek bir noktalı çizgi çizer, dıştaki döngü kaç çizgi çizileceğini belirtir. Örneğimizde, **kenar** değişkeni için 6 sayısını kullanıyoruz ve kesik çizgili bir altıgen elde ediyoruz.

```
1 kenar = 6
2 uzunluk = 400 / kenar
3 aci = 360 / kenar
4 For i = 1 To kenar
5     For j = 1 To 6
6         Turtle.Move(uzunluk / 12)
7         Turtle.PenUp()
8         Turtle.Move(uzunluk / 12)
9         Turtle.PenDown()
10    EndFor
11    Turtle.Turn(acı)
12 EndFor
```



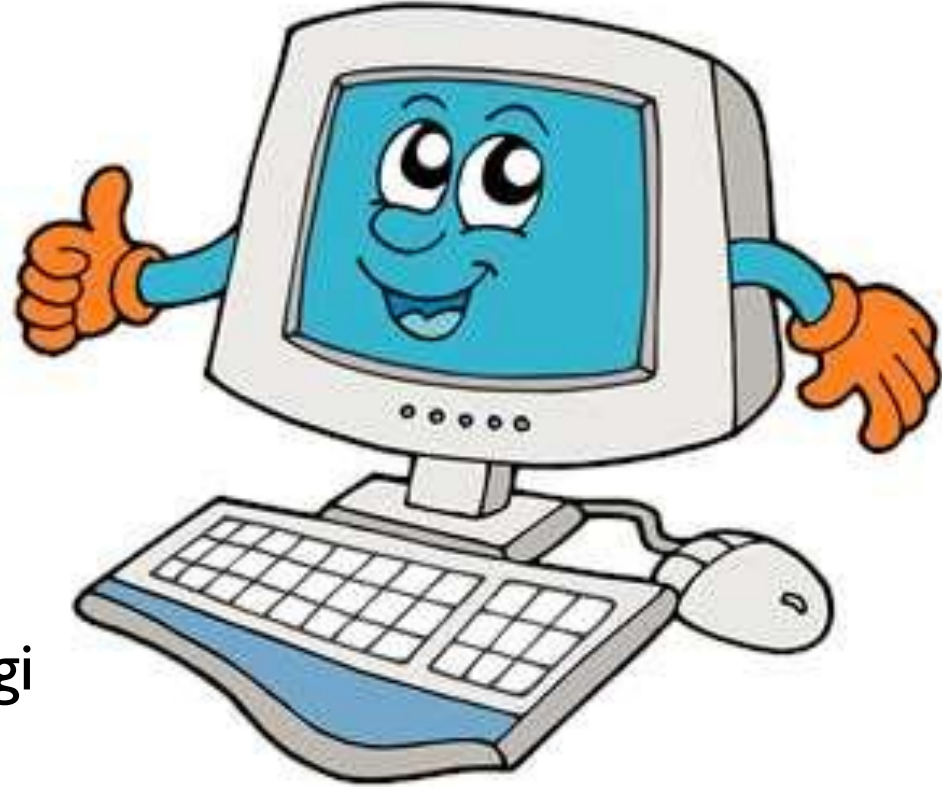


Microsoft®
Small Basic

Altyordamlar

Altyordamlar

- ▶ Program yazarken, sık sık aynı adım setini defalarca tekrar uygulamamız gerekecek durumlarda, aynı ifadeleri birden fazla kez yazmak yerine *Altyordamlar* faydalı olurlar.
- ▶ Bir altyordam, daha büyük bir program içerisindeki kodun, programdaki herhangi bir yerden çağırılabilen bir bölümdür.



Altyordamlar

- ▶ Altyordamlar, **Sub** anahtar kelimesinin ardından gelen bir isimle tanımlanırlar ve **EndSub** anahtar kelimesiyle sonlandırılırlar.
- ▶ Örneğin; aşağıdaki küçük parça ismi *saat* olan bir altyordamı temsil eder ve bu altyordam, o andaki zamanı `TextWindow`'a yazdırma işini yapar.

```
1 Sub saat
2     TextWindow.WriteLine(Clock.Time)
3 EndSub
```



ALTYORDAMLAR

Altyordam, daha büyük bir program içerisindeki kodun, programdaki herhangi bir yerden çağırılabilen bir bölümdür. **Sub** kelimesinin ardından gelen bir isimle tanımlanırlar ve **EndSub** ile sonlandırılırlar.



Altyordamlar

- ▶ Aşağıda, altyordam içeren ve onu çeşitli yerlerden çağıran bir program yer almaktadır.

```
1 Sub saat
2     TextWindow.WriteLine(Clock.Time)
3 EndSub
4 saat()
5 TextWindow.Write("İsminizi girin: ")
6 ad = TextWindow.Read()
7 TextWindow.Write(ad + ", şu anda saat: ")
8 saat()
```

```
12:10:17
İsminizi girin: Burcu
Burcu, şu anda saat: 12:10:20
Press any key to continue...
```

Örnek:

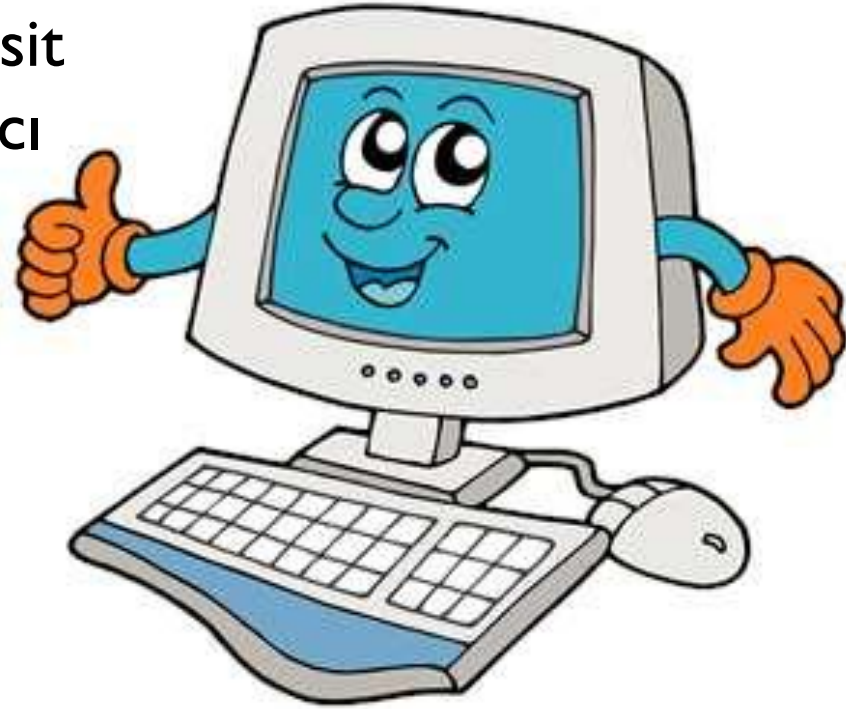


```
1 Sub saat
2     TextWindow.WriteLine(Clock.Time)
3 EndSub
4 saat()
5 TextWindow.Write("İsminizi girin: ")
6 ad = TextWindow.Read()
7 TextWindow.Write(ad + ", şu anda saat: ")
8 saat()
```



Altyordamları Kullanmanın Avantajları

- ▶ Altyordamlar, yazmanız gereken kod miktarını azaltırlar.
- ▶ Karmaşık problemlerin daha basit parçalara ayrılmasına da yardımcı olabilirler.
- ▶ Bir programın okunabilirliğinin iyileştirilmesine de yardımcı olabilirler. Programınızın ortak olarak çalışan bölümleri için iyi isimlendirilmiş altyordamlarınız varsa, programınızın okunması ve kavranması daha kolay hale gelir.



Değişkenlerin Kullanılması

- ▶ Bir programdaki herhangi bir değişkene, bir altıyordam içerisinde erişebilir ve onu kullanabilirsiniz. Aşağıdaki programda, *max* değişkeninin altıyordamın hem içinde, hem de dışında kullanıldığına dikkat edin.

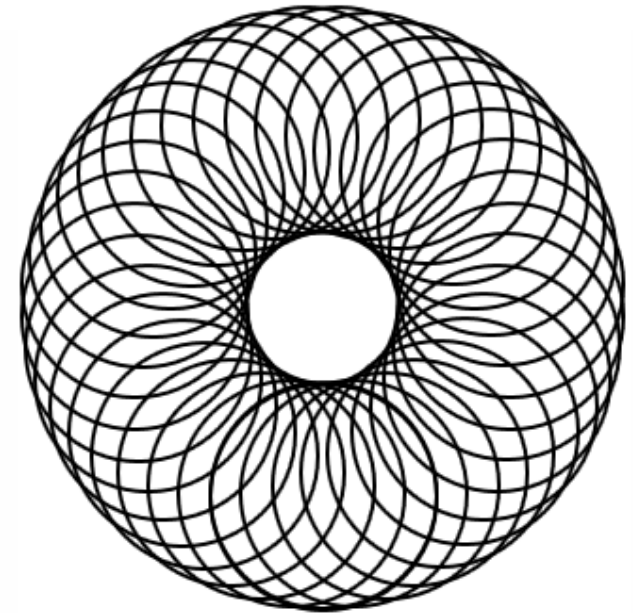
```
1 TextWindow.Write("1. sayi: ")
2 a = TextWindow.ReadNumber()
3 TextWindow.Write("2. sayi: ")
4 b = TextWindow.ReadNumber()
5 Buyuk()
6 TextWindow.WriteLine("Buyuk sayi: " + max)
7 Sub Buyuk
8   If (a > b) Then
9     max = a
10  Else
11    max = b
12  EndIf
13 EndSub
```

```
1. sayi: 8
2. sayi: 5
Buyuk sayi: 8
Press any key to continue...
```


Altyordamlar

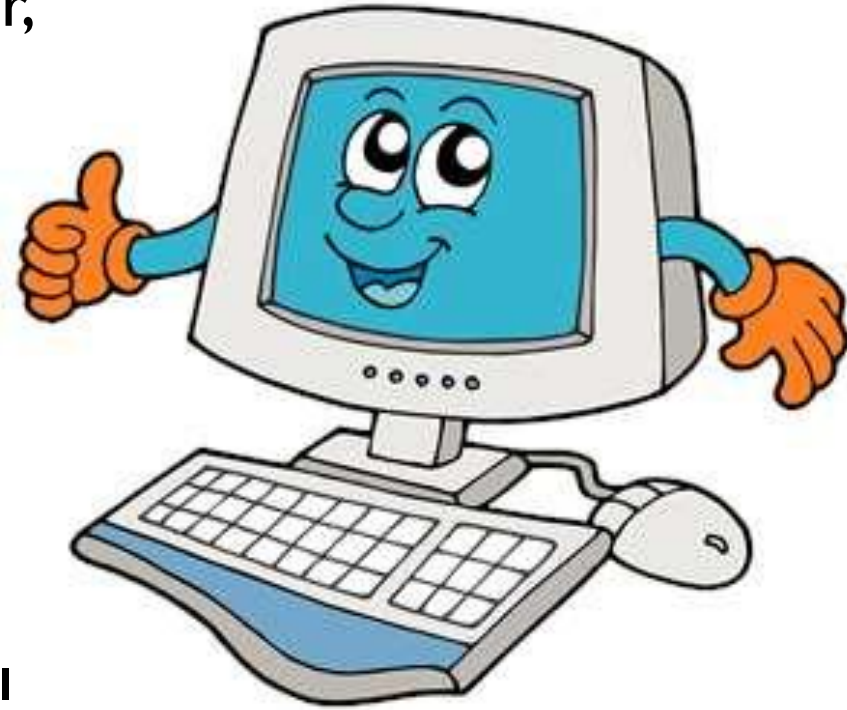
- Şimdi, x ve y değişkenlerinde saklayacak çeşitli noktaları hesaplayan bir grafik programı kullanacağız. Program, merkez olarak x ve y 'yi kullanarak bir daire çizmekten sorumlu olan **DaireCiz** isimli bir altyordam kullanacak.

```
1 For i = 0 To 6.4 Step 0.17
2   x = Math.Sin(i) * 100 + 200
3   y = Math.Cos(i) * 100 + 200
4   DaireCiz()
5 EndFor
6 Sub DaireCiz
7   baslaX = x - 40
8   baslaY = y - 40
9   GraphicsWindow.DrawEllipse(baslaX, baslaY, 120, 120)
10 EndSub
```



Döngüler İçerisinde Altyordamların Çağırılması

- ▶ Bazen, altyordamlar bir döngünün içinden çağırılırlar, bu sırada aynı ifade setini uygularlar, ancak bunu bir ya da daha fazla değışkende farklı değerlerle yaparlar. Örneğin; diyelim ki, *AsalKontrol* isimli bir altyordamınız var ve bu, bir sayının asal olup olmadığını belirliyor. Kullanıcıya bir değer girmesini söyleyen bir program yazabilir ve sonra bu altyordamı kullanarak, onun asal olup olmadığını söyleyebilirsiniz.



Döngüler İçerisinde Altyordamların Çağırılması

- ▶ Girilen sayının asal sayı olup olmadığını söyleyen program:

```
1 TextWindow.Write("Bir sayi girin: ")
2 i = TextWindow.ReadNumber()
3 Asal = "True"
4 AsalKontrol()
5 If (Asal = "True") Then
6     TextWindow.WriteLine(i + " asal sayidir")
7 Else
8     TextWindow.WriteLine(i + " asal sayi degildir")
9 EndIf
10 Sub AsalKontrol
11     For j = 2 To Math.SquareRoot(i)
12         If (Math.Remainder(i, j) = 0) Then
13             Asal = "False"
14             Goto Bitir
15         EndIf
16     Endfor
17 Bitir:
18 EndSub
```

Döngüler İçerisinde Altyordamların Çağırılması

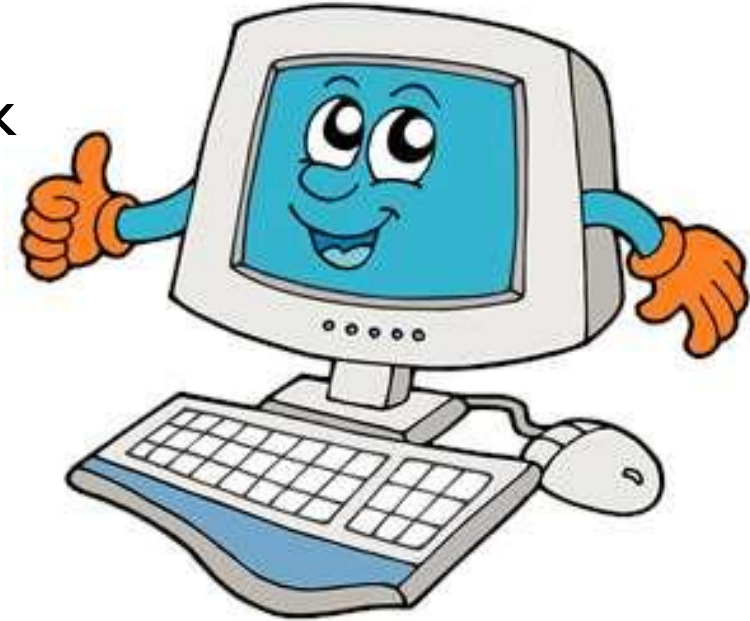
- ▶ `AsalKontrol` altyordamı, i değerini alır ve bunu daha küçük sayılara bölmeye çalışır. Eğer bir sayı i 'ye bölünür ve kalan olmazsa, bu durumda i bir asal sayı değildir. Bu noktada, altyordam `Asal`'ın değerini "False" olarak belirler ve çıkar. Sayı daha küçük sayılara bölünemezse, bu durumda `Asal`'ın değeri "True" olarak kalır.

```
Bir sayi girin: 15
15 asal sayi degildir
Press any key to continue...
```

```
Bir sayi girin: 29
29 asal sayidir
Press any key to continue...
```

Döngüler İçerisinde Altyordamların Çağırılması

- ▶ Artık bizim için Asal kontrolünü yapacak bir altyordamınız olduğuna göre, bunu diyelim ki, 100'ün altındaki tüm asal sayıları listelemek için kullanabilirsiniz. Önceki programı değiştirmek ve bir döngü içinden *AsalKontrol*'ü çağırmasını sağlamak gerçekten kolaydır. Bu, altyordama döngü her çalıştığında hesaplaması için farklı bir değer verir. Bir sonraki örnekle bunun nasıl yapıldığını görelim.



1'den 100'e kadar asal sayıları listeleyen program:

```
1 For i = 2 To 100
2   Asal = "True"
3   AsalKontrol()
4   If (Asal = "True") Then
5     TextWindow.WriteLine(i)
6   EndIf
7 EndFor
8 Sub AsalKontrol
9   For j = 2 To Math.SquareRoot(i)
10    If (Math.Remainder(i, j) = 0) Then
11      Asal = "False"
12      Goto Bitir
13    EndIf
14  Endfor
15 Bitir:
16 EndSub
```



Döngüler İçerisinde Altyordamların Çağırılması

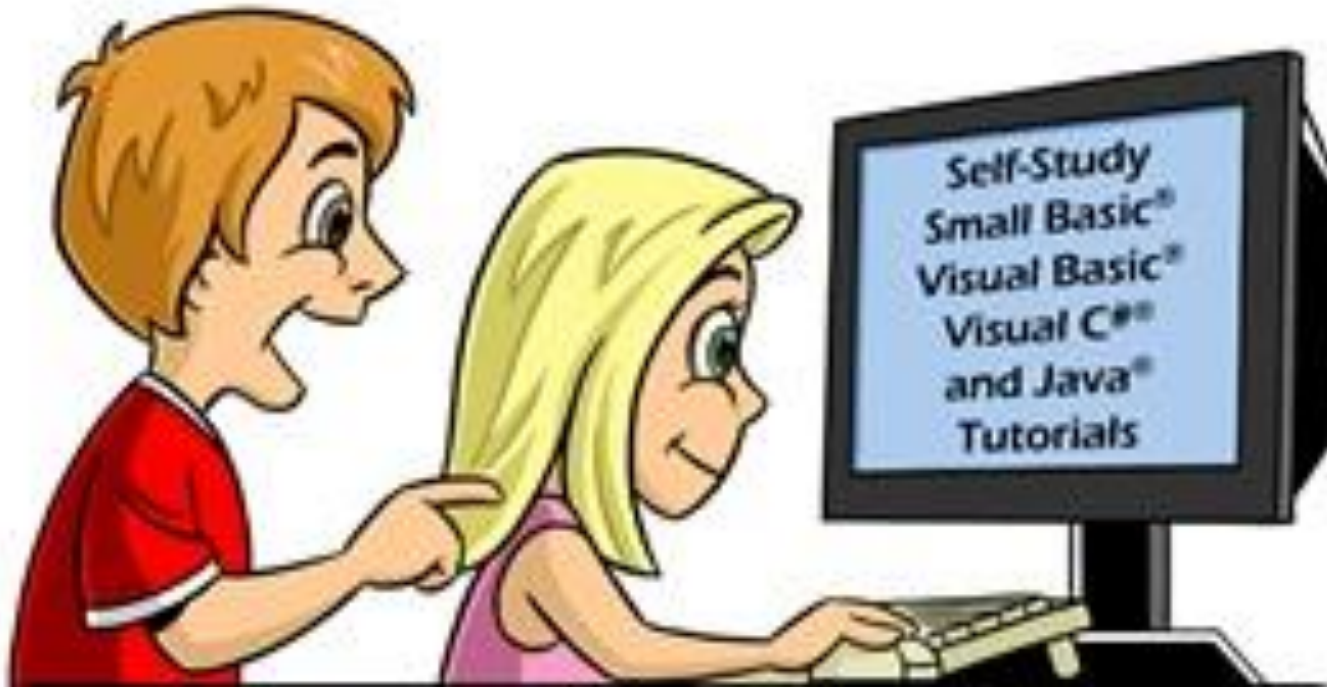
- ▶ Önceki programda, i değeri döngü her çalıştığında güncellenir. Döngünün içinde, *AsalKontrol* altyordamı için bir çağrı yapılır. Daha sonra, *AsalKontrol* altyordamı i değerini alır ve i 'nin bir asal sayı olup olmadığını hesaplar. Sonuç, daha sonra altyordamın dışındaki döngü tarafından erişilen *Asal* değişkeninde saklanır. Daha sonra, asal sayı olduğunun anlaşılması durumunda, i 'nin değeri yazdırılır. Ve döngü 2'den başlayıp 100'e kadar çıktığı için, 2 ile 100 arasındaki tüm asal sayıların bir listesini elde ederiz. Yanda programın sonucu yer almaktadır.

```
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
Press any k
```

SquareRoot:

Verilen sayının
karekökünü hesaplar.





Microsoft®
Small Basic

Diziler

Diziler

- ▶ Şu ana kadar, size değişkenlerin kullanımı konusunda detaylı bilgiler verdik – ne de olsa bu noktaya kadar geldiniz ve hala eğleniyorsunuz, öyle değil mi?
- ▶ Bir an için, değişkenlerle yazdığımız ilk programa geri dönelim:

```
1 TextWindow.Write("İsminizi Girin: ")
2 ad = TextWindow.Read()
3 TextWindow.WriteLine("Merhaba " + ad)
```

- ▶ Bu programda, kullanıcının ismini **name** isimli bir değişkene aldık ve orada sakladık. Daha sonra kullanıcıya “Merhaba” dedik.
-



Diziler

- ▶ Şimdi, diyelim ki; birden fazla kullanıcı var (mesela 5 kullanıcı). Hepsinin ismini nasıl saklayabiliriz? Bunu yapmanın bir yolu şudur:

```
1 TextWindow.Write("1. ismi girin: ")
2 ad1 = TextWindow.Read()
3 TextWindow.Write("2. ismi girin: ")
4 ad2 = TextWindow.Read()
5 TextWindow.Write("3. ismi girin: ")
6 ad3 = TextWindow.Read()
7 TextWindow.Write("4. ismi girin: ")
8 ad4 = TextWindow.Read()
9 TextWindow.Write("5. ismi girin: ")
10 ad5 = TextWindow.Read()
11 TextWindow.Write("Merhaba ")
12 TextWindow.Write(ad1 + ", ")
13 TextWindow.Write(ad2 + ", ")
14 TextWindow.Write(ad3 + ", ")
15 TextWindow.Write(ad4 + ", ")
16 TextWindow.WriteLine(ad5)
```

Diziler

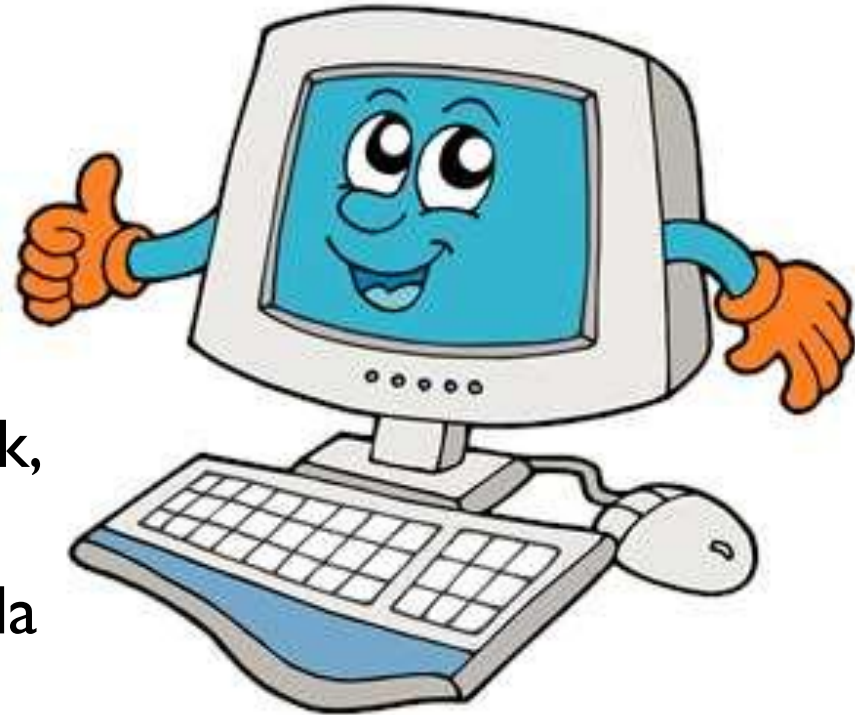
- ▶ Bu programı çalıştırdığınızda, aşağıdaki sonucu göreceksiniz:

```
1. ismi girin: Burcu
2. ismi girin: Baris
3. ismi girin: Burak
4. ismi girin: Arzu
5. ismi girin: Selcuk
Merhaba Burcu, Baris, Burak, Arzu, Selcuk
Press any key to continue...
```



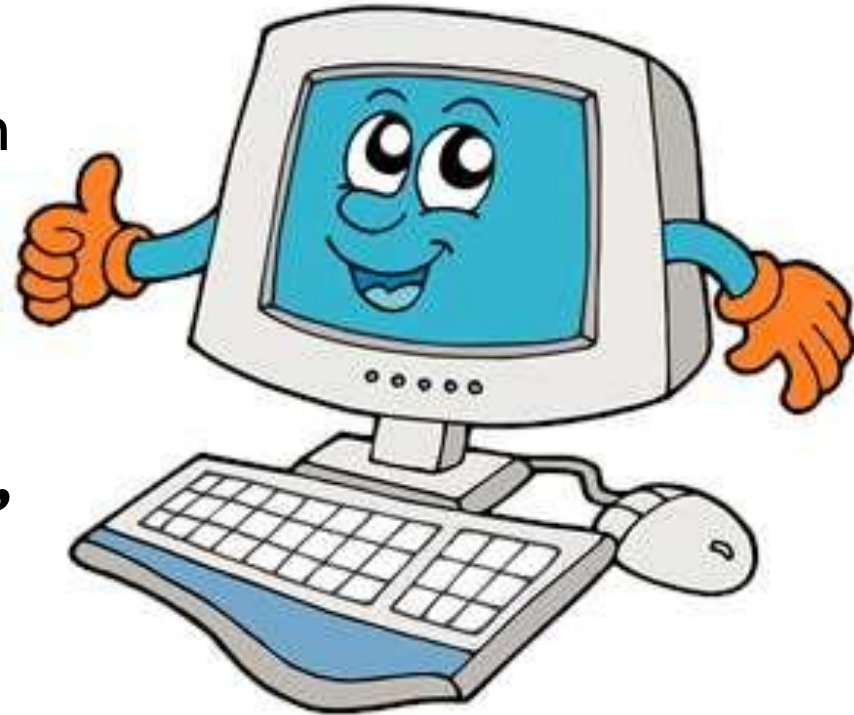
Diziler

- Bu kadar basit bir programı yazmanın daha iyi bir yolu olmalı, değil mi? Özellikle bilgisayarlar tekrarlanan görevleri yapmakta oldukça başarılı olduklarından, her bir yeni kullanıcı için aynı kodu tekrar tekrar yazmak yerine, aynı değişkeni kullanarak, birden fazla kullanıcının ismini saklayıp geri çağırabiliriz. Burada diziler imdadımıza yetişir.



Diziler

- ▶ Dizi, bir defada birden fazla deęeri tutabilen özel bir deęiřkendir.
- ▶ Beř kullanıcı adını saklamak için **ad1**, **ad2**, **ad3**, **ad4** ve **ad5** oluřturmak yerine, “indeks” kullanabiliriz.
- ▶ Örneęin; **ad[1]**, **ad[2]**, **ad[3]**, **ad[4]** ve **ad[5]**’in her biri bir deęer saklayabilir. 1, 2, 3, 4 ve 5 sayıları, dizi için *indeks* olarak adlandırılır.



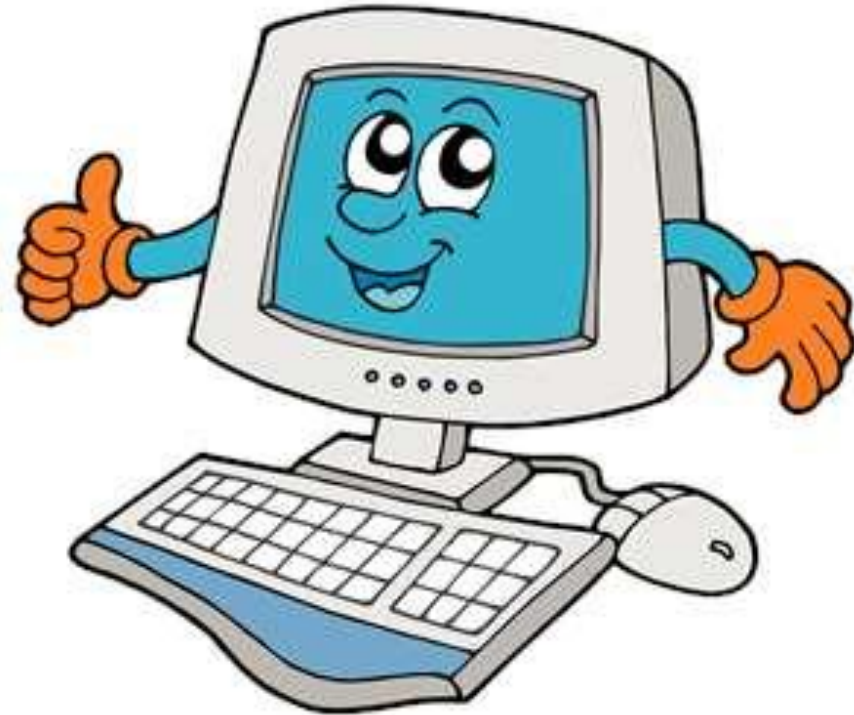
DİZİLER

Dizi, bir defada
birden fazla deęeri
tutabilen özel bir
deęiřkendir.



Diziler

- ▶ **ad[1], ad[2], ad[3], ad[4]** ve **ad[5]** farklı deęişkenler gibi görünse de, gerçekte bunların hepsi yalnızca tek bir deęişkendir.
- ▶ Deęerleri bir dizide saklamanın en iyi yanı, bir başka deęişkeni kullanarak indeksi belirtebilmenizdir. Bu da, döngülerin içerisindeki dizilere kolayca erişebilmemizi sağlar.



Diziler

- ▶ Şimdi, bir önceki programımızı dizilerle yazarak, yeni öğrendiğimiz bilgiyi nasıl kullanabileceğimize bir bakalım.

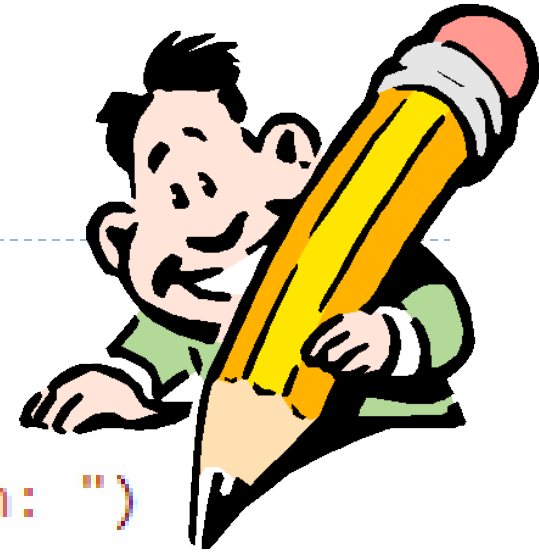
```
1 For i = 1 To 5
2   TextWindow.Write(i + ". ismi girin: ")
3   name[i] = TextWindow.Read()
4 EndFor
5 TextWindow.Write("Merhaba ")
6 For i = 1 To 5
7   TextWindow.Write(name[i])
8   If i < 5 Then
9     TextWindow.Write(", ")
10  EndIf
11 EndFor
12 TextWindow.WriteLine("")
```

- ▶ İşaretli satırlardan birincisi; bir değeri dizide saklar ve ikicisi de onu diziden okur.
-

Örnek:

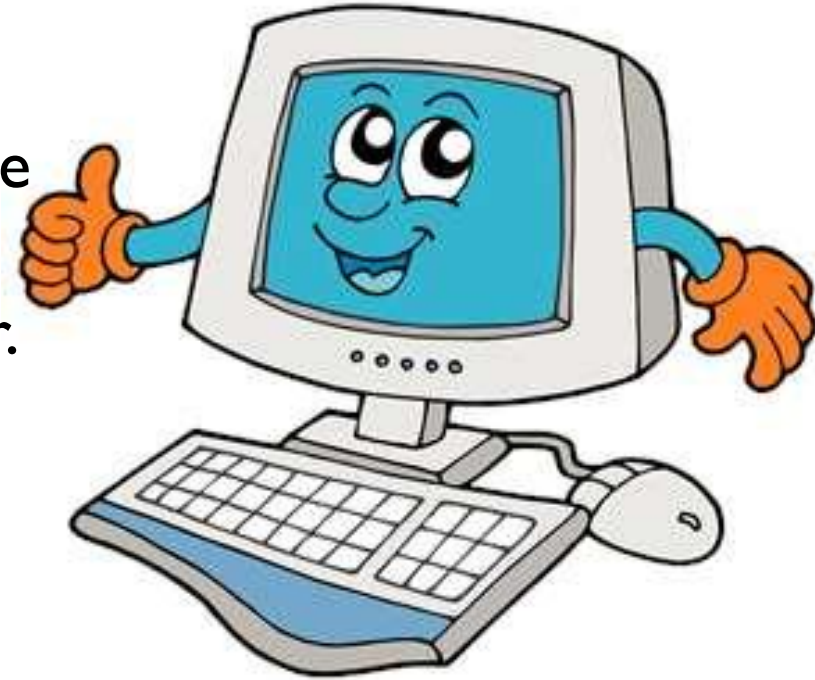
- Dizi kullanarak değişken tanımlama:

```
1 For i = 1 To 5
2   TextWindow.Write(i + ". ismi girin: ")
3   name[i] = TextWindow.Read()
4 EndFor
5 TextWindow.Write("Merhaba ")
6 For i = 1 To 5
7   TextWindow.Write(name[i])
8   If i < 5 Then
9     TextWindow.Write(", ")
10  EndIf
11 EndFor
12 TextWindow.WriteLine("")
```



Bir Dizinin İndekslenmesi

- ▶ Bir önceki programımızda, diziye değerleri saklayıp, sonra da geri çağırmak için indeks olarak sayıları kullandık. İndeksler yalnızca sayılarla sınırlı değildir ve uygulamada, metinsel indeksler kullanmak da oldukça avantajlıdır.
- ▶ Bir sonraki programda, bir kullanıcı hakkındaki çeşitli bilgi parçalarını sorup saklayacağız ve sonra da kullanıcının istediği bilgileri yazdıracağız.



Bir Dizinin İndekslenmesi

```
1 TextWindow.Write("İsminizi girin: ")
2 user["isim"] = TextWindow.Read()
3 TextWindow.Write("Yasinizi girin: ")
4 user["yas"] = TextWindow.Read()
5 TextWindow.Write("Sehrinizi girin: ")
6 user["sehir"] = TextWindow.Read()
7 TextWindow.Write("Posta kodunuzu girin: ")
8 user["posta kodu"] = TextWindow.Read()
9 TextWindow.Write("Hangi bilgiyi istiyorsunuz? ")
10 index = TextWindow.Read()
11 TextWindow.WriteLine(index + " = " + user[index])
```

```
İsminizi girin: Vijaye
Yasinizi girin: 30
Sehrinizi girin: Redmond
Posta kodunuzu girin: 98052
Hangi bilgiyi istiyorsunuz? posta kodu
posta kodu = 98052
Press any key to continue...
```

Birden Fazla Boyut

- ▶ Diyelim ki; tüm arkadaşlarınızın isimlerini ve telefon numaralarını saklamak ve daha sonra da ihtiyaç duyduğunuzda telefon numaralarına ulaşmak istiyorsunuz. Bu tip bir programı nasıl yazabiliriz?
- ▶ Bu durumda, ilgili iki indeks seti vardır. Her bir arkadaşınızı takma isimle tanımladığımızda, bu bizim dizideki ilk indeksimiz olur. İkincisi, **isim** ve **telefon numarası** o arkadaşın gerçek ismine ve telefon numarasına ulaşmamızı sağlayacaktır.



Birden Fazla Boyut

- ▶ Bu programı bir kez düzenledikten sonra, girdi olarak bir arkadaşımızın kod adını alabilir ve sonra o kişi hakkında sakladığımız bilgileri yazdırabiliriz. İşte bunu yapan program:

```
1 arkadas["Br"]["Ad"] = "Baris"
2 arkadas["Br"]["Telefon"] = "555-3456789"
3 arkadas["Ar"]["Ad"] = "Arzu"
4 arkadas["Ar"]["Telefon"] = "555-1234567"
5 arkadas["Sl"]["Ad"] = "Selcuk"
6 arkadas["Sl"]["Telefon"] = "555-2345678"
7 TextWindow.Write("Kod adi girin: ")
8 kod = TextWindow.Read()
9 TextWindow.WriteLine("İsim: " + arkadas[kod]["Ad"])
10 TextWindow.WriteLine("Telefon: " + arkadas[kod]["Telefon"])
```



Birden Fazla Boyut

```
1 arkadas["Br"]["Ad"] = "Baris"
2 arkadas["Br"]["Telefon"] = "555-3456789"
3 arkadas["Ar"]["Ad"] = "Arzu"
4 arkadas["Ar"]["Telefon"] = "555-1234567"
5 arkadas["Sl"]["Ad"] = "Selcuk"
6 arkadas["Sl"]["Telefon"] = "555-2345678"
7 TextWindow.Write("Kod adi girin: ")
8 kod = TextWindow.Read()
9 TextWindow.WriteLine("İsim: " + arkadas[kod]["Ad"])
10 TextWindow.WriteLine("Telefon: " + arkadas[kod]["Telefon"])
```

► Program şu şekilde çalışır:

```
Kod adi girin: br
İsim: Baris
Telefon: 555-3456789
Press any key to continue...
```

Tabloları Temsil Etmesi İçin Dizileri Kullanmak

- ▶ Çok boyutlu dizilerin oldukça yaygın bir kullanımı, tabloları temsil etmektir. Tabloların, iki boyutlu bir diziye çok iyi bir şekilde uyan sıraları ve kolonları vardır. Aşağıda, bir tabloya kutular yerleştiren bir program görüyoruz.

```
1 satir = 8
2 sutun = 8
3 boy = 40
4 For a = 1 To satir
5     For b = 1 To sutun
6         GraphicsWindow.BrushColor = GraphicsWindow.GetRandomColor()
7         kutu[a][b] = Shapes.AddRectangle(boy, boy)
8         Shapes.Move(kutu[a][b], b * boy, a * boy)
9     EndFor
10 EndFor
```

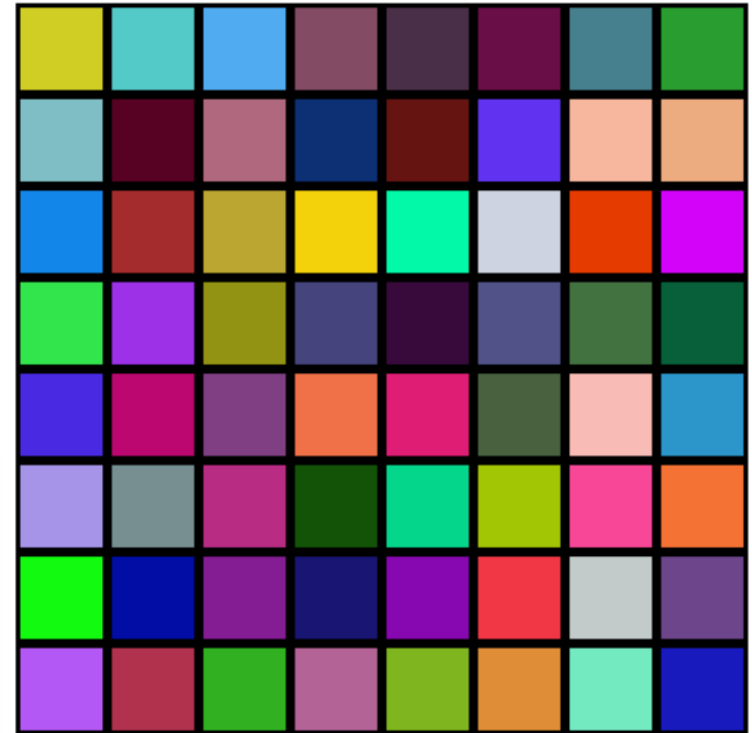

Tabloları Temsil Etmesi İçin Dizileri Kullanmak

- Bu program kareler ekler ve onları 8x8 bir tablo oluşturacak şekilde konumlandırır. Bu kutuları yerleştirmenin yanında, ayrıca onları bir dizi halinde de saklar. Bunu yapması, kutuları izlememizi ve gerektiğinde tekrar kullanmamızı kolaylaştırır.

```
1 satir = 8
2 sutun = 8
3 boy = 40
4 For a = 1 To satir
5     For b = 1 To sutun
6         GraphicsWindow.BrushColor = GraphicsWindow.GetRandomColor()
7         kutu[a][b] = Shapes.AddRectangle(boy, boy)
8         Shapes.Move(kutu[a][b], b * boy, a * boy)
9     EndFor
10 EndFor
```

Tabloları Temsil Etmesi İçin Dizileri Kullanmak

```
1 satir = 8
2 sutun = 8
3 boy = 40
4 For a = 1 To satir
5     For b = 1 To sutun
6         GraphicsWindow.BrushColor = GraphicsWindow.GetRandomColor()
7         kutu[a][b] = Shapes.AddRectangle(boy, boy)
8         Shapes.Move(kutu[a][b], b * boy, a * boy)
9     EndFor
10 EndFor
```



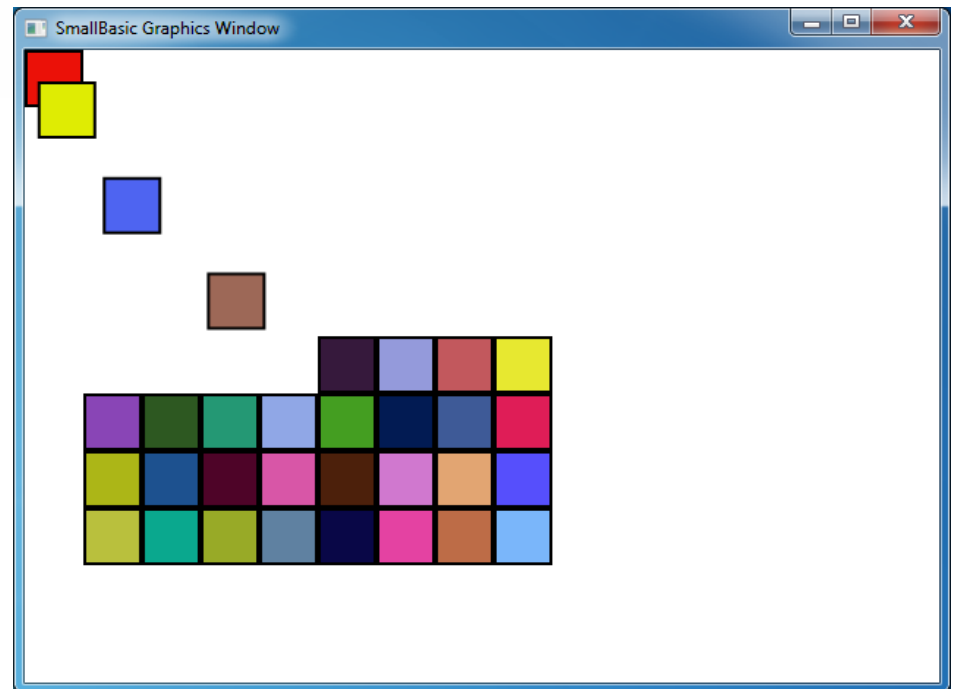
Tabloları Temsil Etmesi İçin Dizileri Kullanmak

- Bir önceki programın sonuna aşağıdaki kodun eklenmesi, bu kutuların sol üst köşeye gitmesine neden olacaktır.

```
1  satir = 8
2  sutun = 8
3  boy = 40
4  For a = 1 To satir
5      For b = 1 To sutun
6          GraphicsWindow.BrushColor = GraphicsWindow.GetRandomColor()
7          kutu[a][b] = Shapes.AddRectangle(boy, boy)
8          Shapes.Move(kutu[a][b], b * boy, a * boy)
9      EndFor
10 EndFor
11 For a = 1 To satir
12     For b = 1 To sutun
13         Shapes.Animate(kutu[a][b], 0, 0, 1000)
14         Program.Delay(300)
15     EndFor
16 EndFor
```

Tabloları Temsil Etmesi İçin Dizileri Kullanmak

```
11 For a = 1 To satir
12   For b = 1 To sutun
13     Shapes.Animate(kutu[a][b], 0, 0, 1000)
14     Program.Delay(300)
15   EndFor
16 EndFor
```



Örnek:

- Dizileri kullanarak kutular ekleyen ve kutuları sol üst köşede toplayan program:

```
1 satir = 8
2 sutun = 8
3 boy = 40
4 For a = 1 To satir
5     For b = 1 To sutun
6         GraphicsWindow.BrushColor = GraphicsWindow.GetRandomColor()
7         kutu[a][b] = Shapes.AddRectangle(boy, boy)
8         Shapes.Move(kutu[a][b], b * boy, a * boy)
9     EndFor
10 EndFor
11 For a = 1 To satir
12     For b = 1 To sutun
13         Shapes.Animate(kutu[a][b], 0, 0, 1000)
14         Program.Delay(300)
15     EndFor
16 EndFor
```



Shapes

- ▶ Grafik penceresinde şekil ekleme, şekli hareket ettirme, döndürme gibi işlemleri sağlar.

○ AddEllipse

○ AddImage

○ AddLine

○ AddRectangle

○ AddText

○ AddTriangle

○ Animate

○ GetLeft

○ GetOpacity

○ GetTop

○ HideShape

○ Move

○ Remove

○ Rotate

○ SetOpacity

○ SetText

○ ShowShape

○ Zoom

Shapes:

Grafik penceresinde
şekil ekleme, şekli
hareket ettirme ve
döndürme işlemlerini
sağlar.



AddRectangle:
Belirtilen genişlik
ve yükseklikte
dikdörtgen şekil
ekler.



Move:

Şekli başka konuma taşır.

Move(ad, x, y) şeklinde kullanılır.

ad: Çizdirilecek şeklin ismi

x ve y: Yeni pozisyonun x ve y koordinatları



Animate:

Şekli, belirtilen isimle yeni bir konuma çizdirir.

Animate(ad, x, y, zaman) şeklinde kullanılır.

zaman: Animasyonda kullanılacak milisaniye cinsinden süre



Program:

Programın alıřmasını
kontrol etmek iin
yardım saęlar.



Delay:

Programın alıřmasını
belirtilen sre
dođrultusunda geciktirir.

Delay(milisaniye)
řeklinde kullanılır.



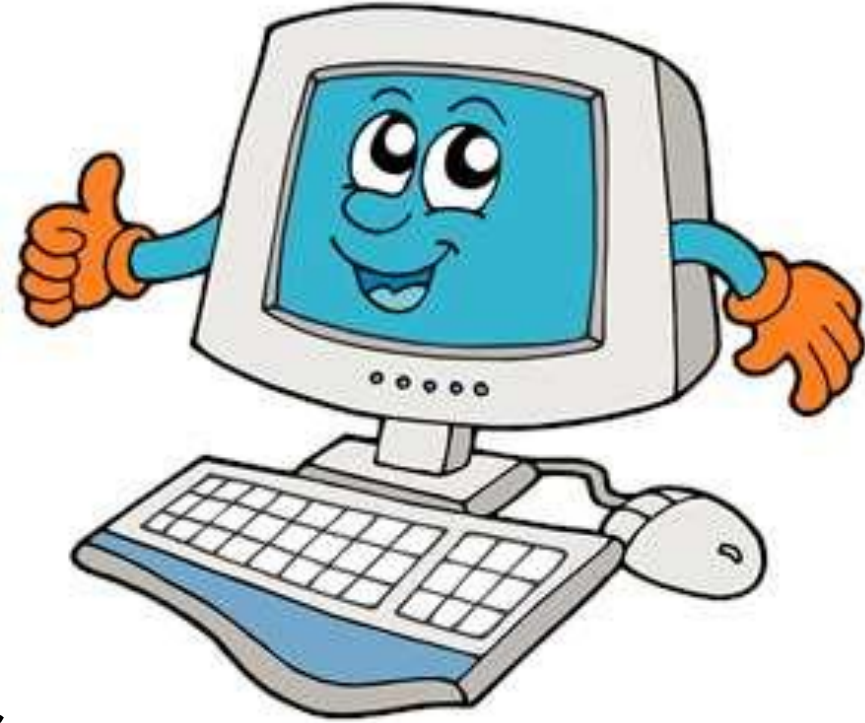


Microsoft®
Small Basic

Olaylar ve Etkileşim

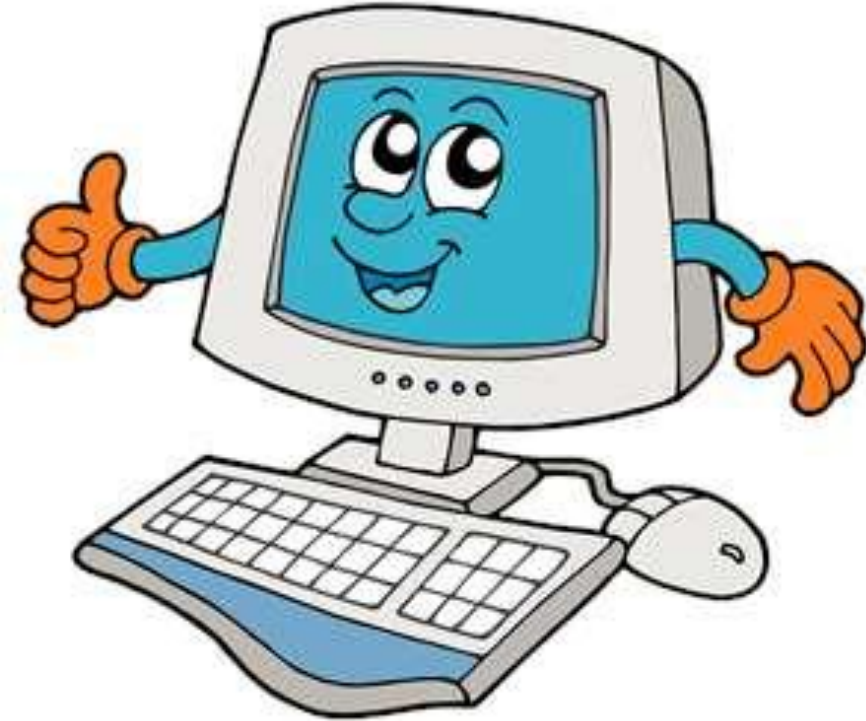
Olaylar ve Etkileşim

- ▶ Olaylar, örneğin; fareyi hareket ettirmek veya üzerine tıklamak gibi kullanıcı eylemlerine karşı verilen sinyaller gibidir.
- ▶ Bir anlamda, olaylar işlemlerin tam tersidir. İşlemler söz konusuysa, bir programcı olarak siz bilgisayara bir şeyler yaptırırsınız, olaylar söz konusuysa, bilgisayar ilginç bir şey olduğunda size haber verir.



Olaylar ve Etkileşim

- ▶ Bir kullanıcının programınızla etkileşime girmesini isterseniz, olayları kullanırsınız.
- ▶ Diyelim ki; bir Üç Taş oyunu yazıyorsunuz. Kullanıcının hamlesine karar vermesine izin vermek isteyeceksiniz. İşte burada olayları kullanarak programınız içerisinden girdi alırsınız.



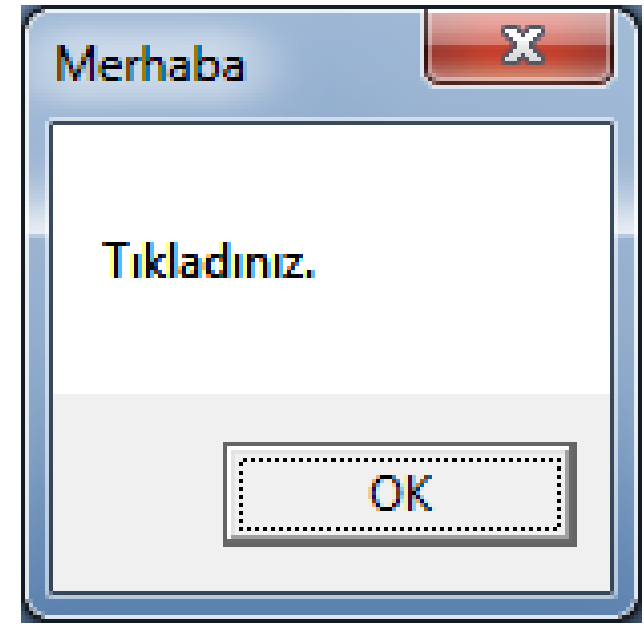
OLAYLAR

Fareyi hareket ettirmek veya üzerine tıklamak gibi kullanıcı eylemlerine karşı verilen sinyallerdir.



Olaylar ve Etkileşim

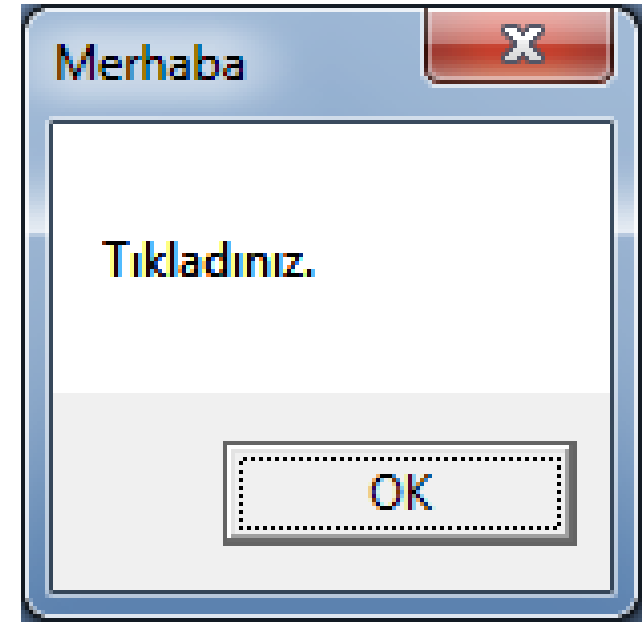
- ▶ Aşağıda, yalnızca tek bir ifadeden ve bir altyordamdan oluşan oldukça basit bir program yer alıyor. Altyordam, kullanıcıya bir mesaj kutusu göstermek için, GraphicsWindow nesnesinde *ShowMessage* işlemini kullanır.



```
1 GraphicsWindow.MouseDown = Tiklama
2 Sub Tiklama
3     GraphicsWindow.ShowMessage("Tıkladınız.", "Merhaba")
4 EndSub
```

Olaylar ve Etkileşim

- ▶ İlk satırda, GraphicsWindow nesnesinin **MouseDown** olayına *Tıklama* altyordamını atıyoruz. Olay gerçekleştiğinde, altyordam otomatik olarak çağırılır. Bu durumda, kullanıcı fareyi kullanarak her tıkladığında, yandaki mesaj kutusunu göreceksiniz.



```
1 GraphicsWindow.MouseDown = Tıklama
2 Sub Tıklama
3     GraphicsWindow.ShowMessage("Tıkladınız.", "Merhaba")
4 EndSub
```

Örnek:

- ▶ Fare ile tıklayınca mesaj kutusu açılan program:



```
1 GraphicsWindow.MouseDown = Tiklama
2 Sub Tiklama
3     GraphicsWindow.ShowMessage("Tıkladınız.", "Merhaba")
4 EndSub
```



MouseDown:

Fare tuşuna
basıldığında bir olayı
aktifleştirir.



ShowMessage:

Ekranında girilen mesajı gösterir.

ShowMessage(mesaj, başlık)
şeklinde kullanılır.



Olaylar ve Etkileşim

- ▶ *Tıklama* altyordamını, bir mesaj kutusu açmak yerine başka şeyler yapmak üzere değiştirebilirsiniz. Örneğin; aşağıdaki programda olduğu gibi, kullanıcının fareyi tıkladığı yerlere büyük mavi noktalar çizebilirsiniz.

```
1 GraphicsWindow.BrushColor = "Blue"  
2 GraphicsWindow.MouseDown = Tiklama  
3 Sub Tiklama  
4     x = GraphicsWindow.MouseX - 10  
5     y = GraphicsWindow.MouseY - 10  
6     GraphicsWindow.FillEllipse(x, y, 20, 20)  
7 EndSub
```



Olaylar ve Etkileşim

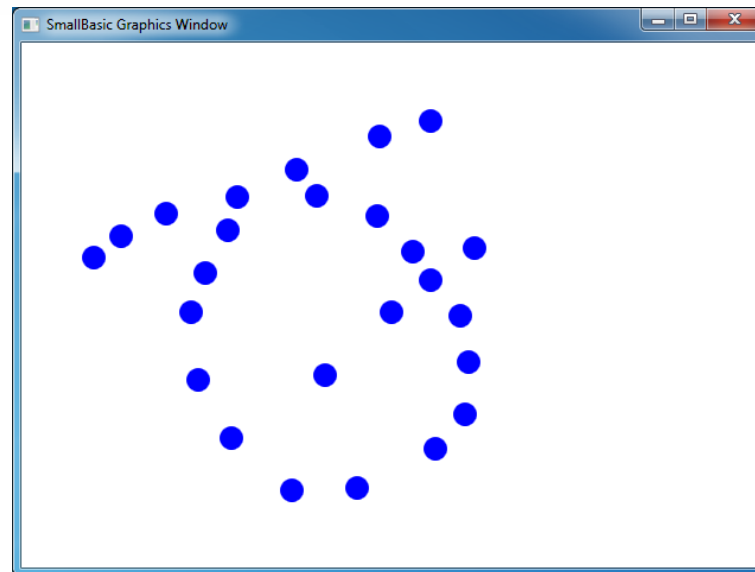
- ▶ Bu programda, farenin koordinatlarını elde etmek için *MouseX* ve *MouseY*'yi kullandığımıza dikkat edin. Bunları, farenin koordinatları dairenin merkezi olacak şekilde, bir daire çizmek için kullanıyoruz.

```
1 GraphicsWindow.BrushColor = "Blue"  
2 GraphicsWindow.MouseDown = Tiklama  
3 Sub Tiklama  
4     x = GraphicsWindow.MouseX - 10  
5     y = GraphicsWindow.MouseY - 10  
6     GraphicsWindow.FillEllipse(x, y, 20, 20)  
7 EndSub
```



Mavi Daireler!

```
1 GraphicsWindow.BrushColor = "Blue"  
2 GraphicsWindow.MouseDown = Tiklama  
3 Sub Tiklama  
4     x = GraphicsWindow.MouseX - 10  
5     y = GraphicsWindow.MouseY - 10  
6     GraphicsWindow.FillEllipse(x, y, 20, 20)  
7 EndSub
```



Birden Fazla Olayın Kullanılması

- ▶ Ne kadar olayı kullanmak istediğinizle ilgili bir sınır yoktur. Ancak, bir olayı yalnızca bir kez kullanabilirsiniz. Aynı olaya iki altıyordam atamaya çalışırsanız, ikincisi aktif olur.
- ▶ Şimdi bir önceki örneğe, tuşlara basmayı kullanan bir altıyordam ekleyelim. Ayrıca, bu altıyordama fırçanın rengini deęiştirtelim, böylece fareye tıkladığımızda farklı bir renkte nokta elde edelim.



Renkli Noktalar

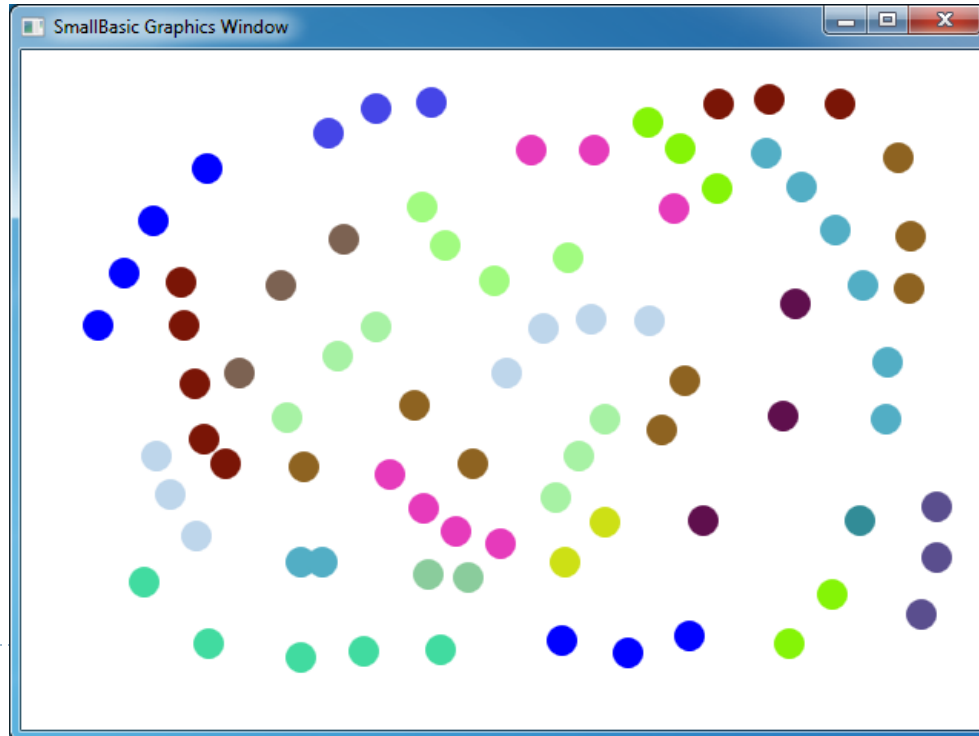


```
1 GraphicsWindow.BrushColor = "Blue"
2 GraphicsWindow.MouseDown = Tiklama
3 GraphicsWindow.KeyDown = Birtusabasma
4 Sub Birtusabasma
5     GraphicsWindow.BrushColor = GraphicsWindow.GetRandomColor()
6 EndSub
7 Sub Tiklama
8     x = GraphicsWindow.MouseX - 10
9     y = GraphicsWindow.MouseY - 10
10    GraphicsWindow.FillEllipse(x, y, 20, 20)
11 EndSub
```



Renkli Noktalar

- Bu programı çalıştırır ve pencereye tıklarsanız, mavi bir nokta çıkacaktır. Herhangi bir tuşa bir kez basar ve sonra yine tıklarsanız, farklı renkte bir nokta çıkacaktır. Bir tuşa bastığınızda, fırçanın rengini rastgele bir renge değiştiren *Birtusabasma* altyordamı uygulanır.



MouseX:

Grafik penceresine göre farenin x konumunu belirtir.

MouseY:

Farenin y konumunu belirtir.

MouseDown:

Fare tuşuna basıldığında bir olayı aktifleştirir.



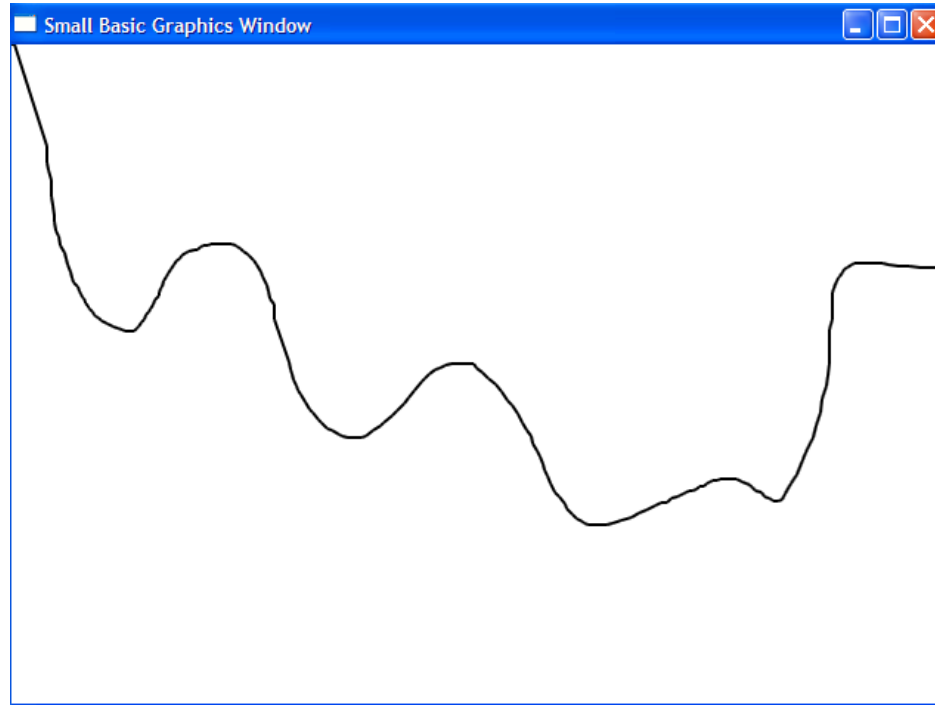
Bir Çizim Programı

- ▶ Olayları ve altyordamları öğrendikten sonra, artık kullanıcıların pencerede çizim yapmasına izin veren bir program yazabiliriz.
- ▶ İlk adım olarak, kullanıcıların fareyi grafik penceresinde herhangi bir yere hareket ettirmesine izin verecek ve fare hareket ettirildiğinde bir iz bırakacak bir program yazalım.

```
1 GraphicsWindow.MouseMove = FareHareket
2 Sub FareHareket
3     x = GraphicsWindow.MouseX
4     y = GraphicsWindow.MouseY
5     GraphicsWindow.DrawLine(pX, pY, x, y)
6     pX = x
7     pY = y
8 EndSub
```

Bir Çizim Programı

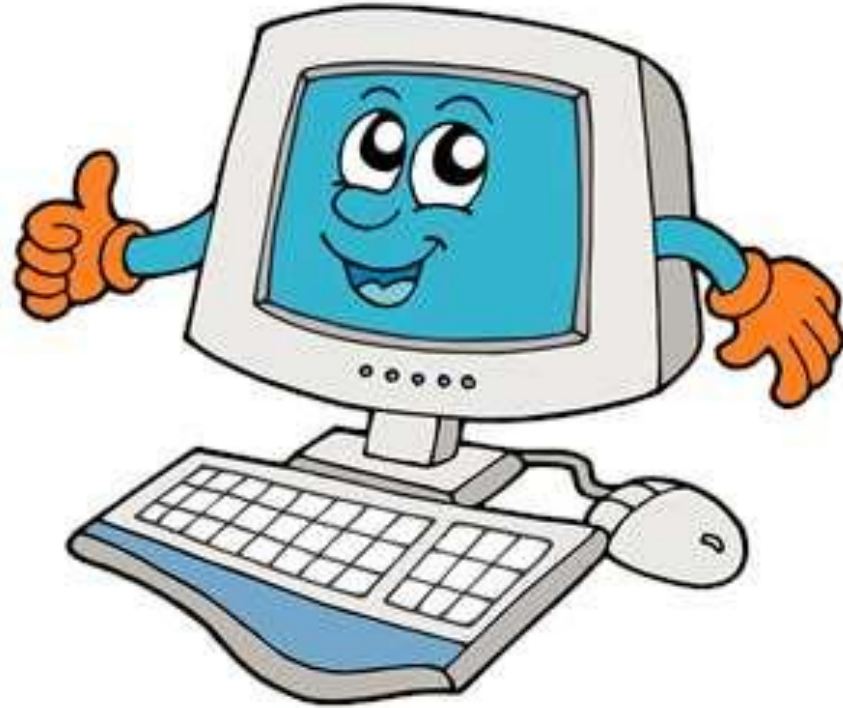
- ▶ Ancak, bu programı çalıştırdığınızda, ilk satır daima pencerenin sol üst kenarından (0, 0) başlar.



- ▶ *MouseDown* olayını kullanarak ve o olay geldiğinde pX ve pY değerlerini alarak bu sorunu giderebiliriz.

Bir Çizim Programı

- ▶ İze yalnızca kullanıcı fare düğmesini basılı tuttuğunda ihtiyacımız vardır. Diğer zamanlarda, çizgi çizmememiz gerekir. Bu davranışı elde etmek için, **Mouse** nesnesinde *IsLeftButtonDown* özelliğini kullanacağız. Bu özellik, Sol düğmesine basılı tutulup tutulmadığını söyler. Bu değer doğruysa, çizgiyi çizeriz, değilse çizmeyiz.



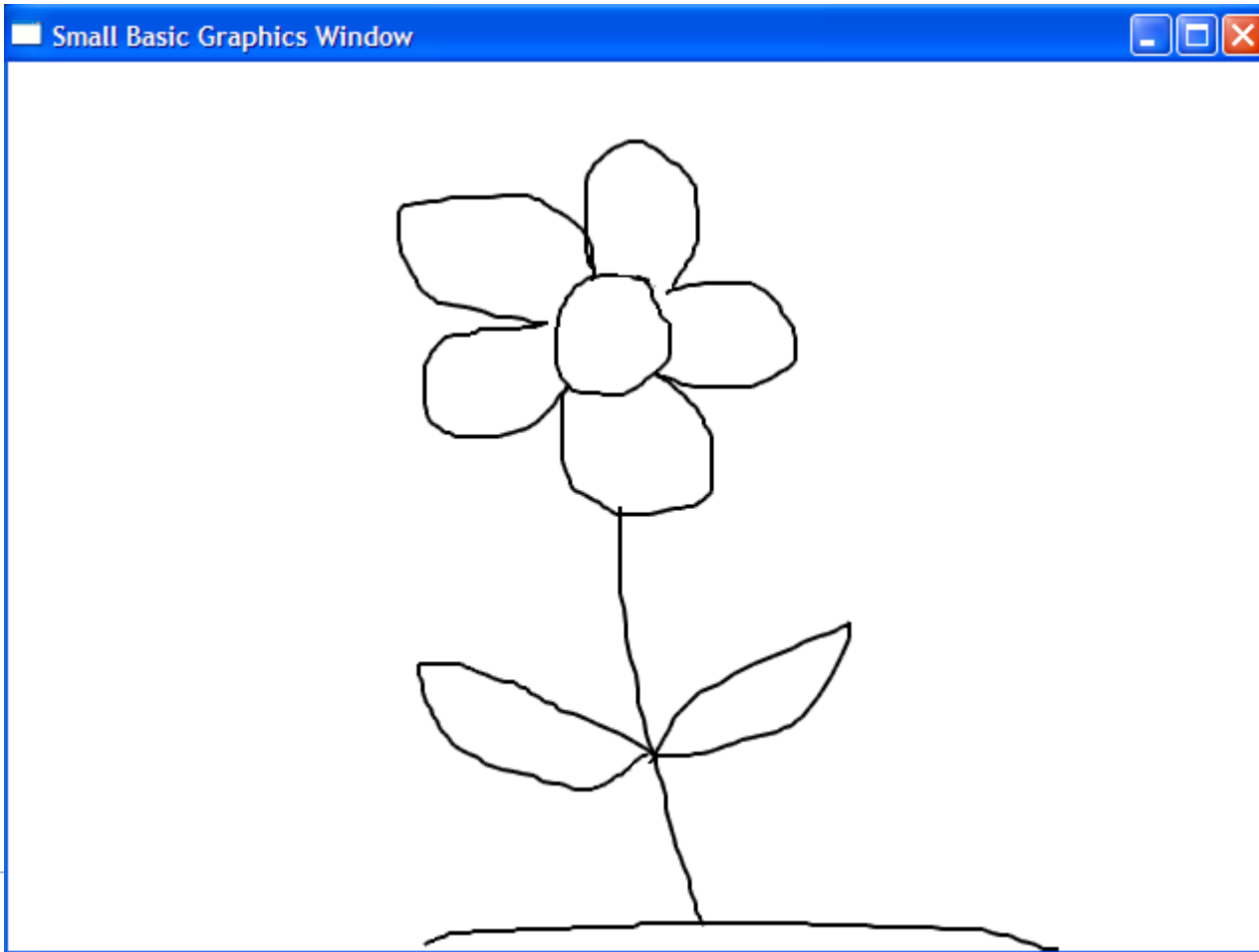
Bir Çizim Programı

```
1 GraphicsWindow.MouseMove = FareHareket
2 GraphicsWindow.MouseDown = Tiklatinca
3 Sub Tiklatinca
4     pX = GraphicsWindow.MouseX
5     pY = GraphicsWindow.MouseY
6 EndSub
7 Sub FareHareket
8     x = GraphicsWindow.MouseX
9     y = GraphicsWindow.MouseY
10    If (Mouse.IsLeftButtonDown) Then
11        GraphicsWindow.DrawLine(pX, pY, x, y)
12    EndIf
13    pX = x
14    pY = y
15 EndSub
```



Bir Çizim Programı

- Programı çalıştırdığımızda artık serbestçe istediğimizi çizebiliriz.



MouseMove:

Fare hareket ettirildiğinde bir olayı gerçekleştirir.



Mouse:

İmleç konumu,
işaretçi gibi, fare ile
ilgili özellikleri almak
veya ayarlamak için
için kullanılır.



IsLeftButtonDown:

Sol tuşa basılıp
basılmadığını belirtir.

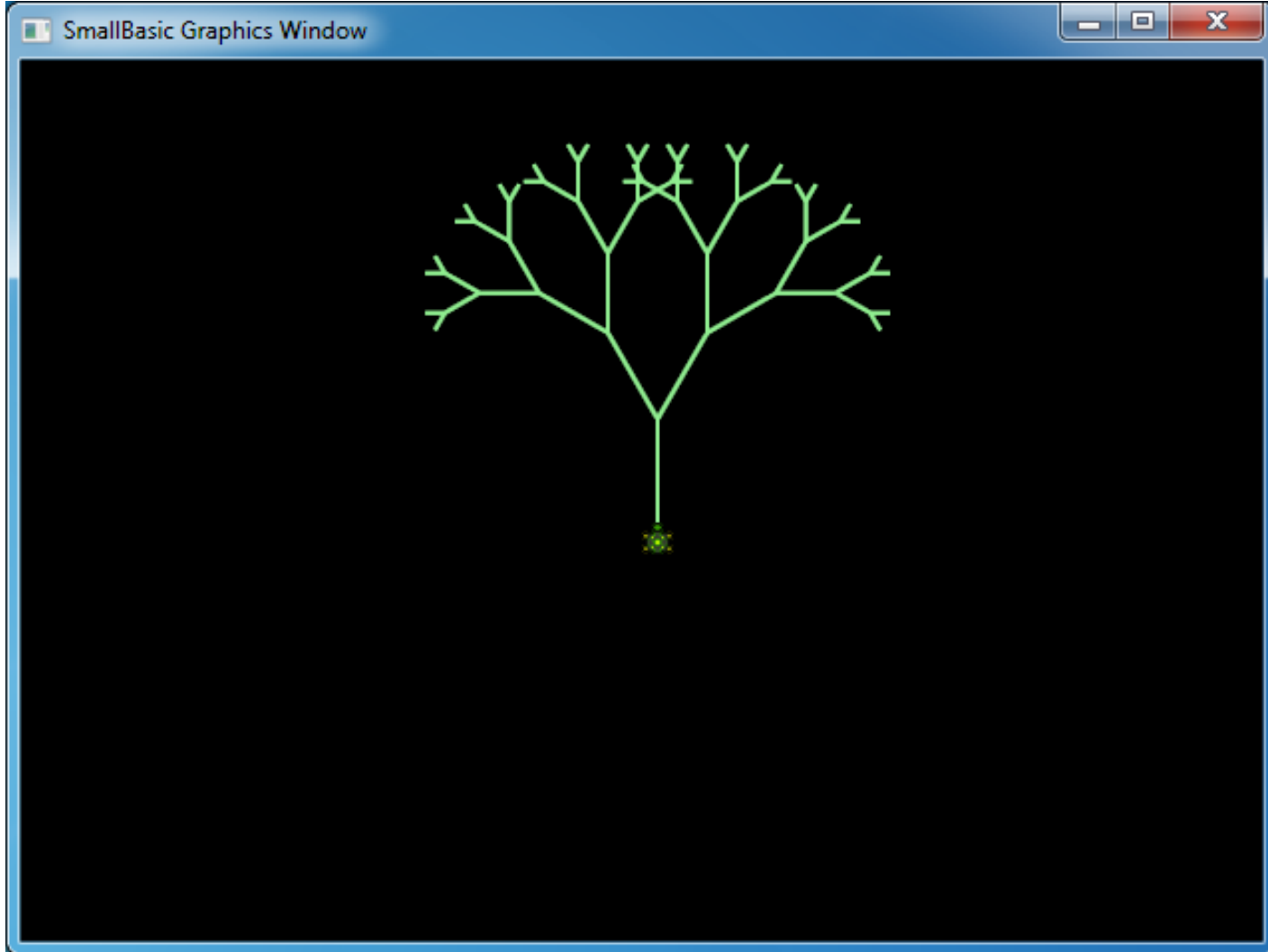




Microsoft®
Small Basic

Eğlenceli Örnekler

Benzer Şekilleri Kullanarak Ağaç Çizen Kaplumbağa



Benzer Şekilleri Kullanarak Ağaç Çizen Kaplumbağa



```
1 aci = 30
2 delta = 10
3 mesafe = 60
4 Turtle.Speed = 9
5 GraphicsWindow.BackgroundColor = "Black"
6 GraphicsWindow.PenColor = "LightGreen"
7 AgacCiz()
8 Sub AgacCiz
9   If (mesafe > 0) Then
10     Turtle.Move(mesafe)
11     Turtle.Turn(aci)
12     Stack.PushValue("mesafe", mesafe)
13     mesafe = mesafe - delta
14     AgacCiz()
15     Turtle.Turn(-aci * 2)
16     AgacCiz()
17     Turtle.Turn(aci)
18     mesafe = Stack.PopValue("mesafe")
19     Turtle.Move(-mesafe)
20   EndIf
21 EndSub
```

Stack:

Değerleri depolamanın yolunu sağlar. Yığının en tepesine bir değer gönderebilir ve oradan alabilirsiniz.



PushValue:

Belirtilen yığına bir deęer atar.

PushValue(yığın adı, atanan deęer) şeklinde kullanılır.



Pop Value:

Belirtilen yığından
değer alır.



Flickr'dan Fotoğraf Almak



```
1 GraphicsWindow.BackgroundColor = "Black"  
2 GraphicsWindow.MouseDown = Tiklatinca  
3 Sub Tiklatinca  
4     foto = Flickr.GetRandomPicture("mountains, river")  
5     GraphicsWindow.DrawResizedImage(foto, 0, 0, 640, 480)  
6 EndSub
```



Flickr:

Flickr fotoğraf servisine erişimi sağlar.



GetRandomPicture:

Belirtilen bir etiket
doğrultusunda rastgele bir
fotoğrafın URL'sini belirtir.



DrawResizedImage:

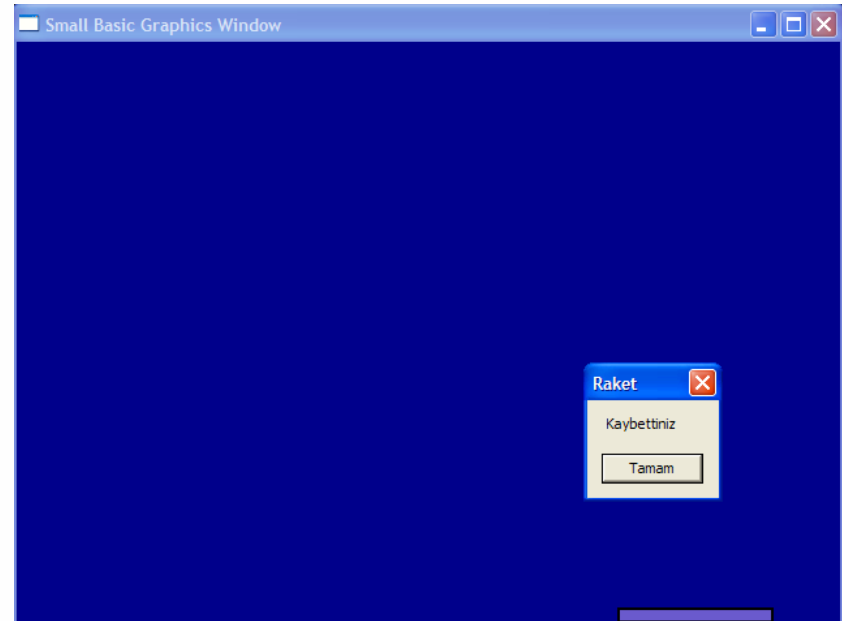
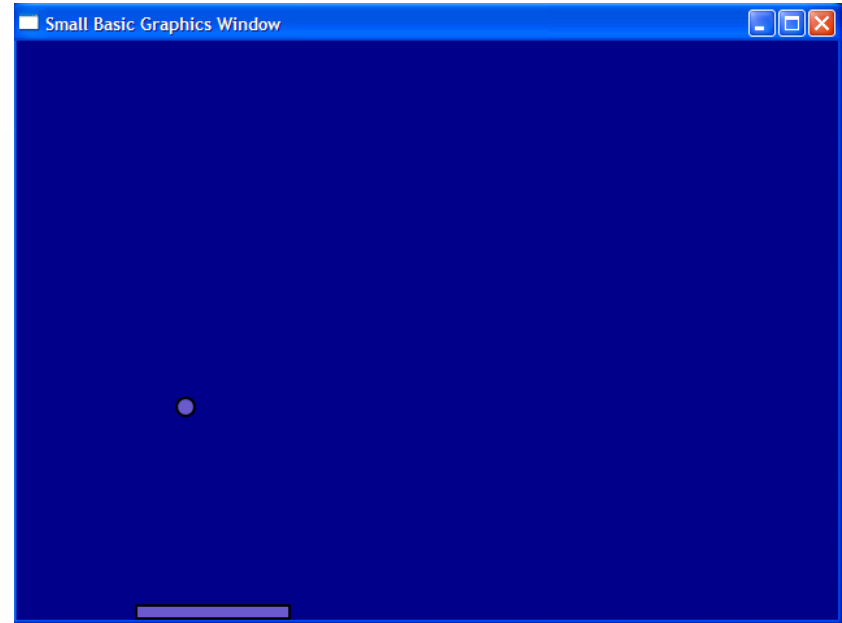
Hafızadaki belirtilen bir resmi belirtilen boyutta ekrana çizdirir.

DrawResizedImage(ad, x, y, genişlik, yükseklik) şeklinde kullanılır.



Raket Oyunu

```
1 GraphicsWindow.BackgroundColor = "DarkBlue"
2 raket = Shapes.AddRectangle(120, 12)
3 top = Shapes.AddEllipse(16, 16)
4 GraphicsWindow.MouseMove = FareHareket
5 x = 0
6 y = 0
7 deltaX = 1
8 deltaY = 1
9 Calistir:
10  x = x + deltaX
11  y = y + deltaY
12  gw = GraphicsWindow.Width
13  gh = GraphicsWindow.Height
14  If (x >= gw - 16 or x <= 0) Then
15    deltaX = -deltaX
16  EndIf
17  If (y <= 0) Then
18    deltaY = -deltaY
19  EndIf
20  rX = Shapes.GetLeft (raket)
21  If (y = gh - 28 and x >= rX and x <= rX + 120) Then
22    deltaY = -deltaY
23  EndIf
24  Shapes.Move(top, x, y)
25  Program.Delay(5)
26  If (y < gh) Then
27    Goto Calistir
28  EndIf
29 GraphicsWindow.ShowMessage("Kaybettiniz", "Raket")
30 Sub FareHareket
31  raketX = GraphicsWindow.MouseX
32  Shapes.Move(raket, raketX - 60, GraphicsWindow.Height - 12)
33 EndSub
```





Microsoft®
Small Basic

Renkler

Kırmızı

IndianRed	#CD5C5C
LightCoral	#F08080
Salmon	#FA8072
DarkSalmon	#E9967A
LightSalmon	#FFA07A
Crimson	#DC143C
Red	#FF0000
FireBrick	#B22222
DarkRed	#8B0000



Pembe

Pink	#FFC0CB
LightPink	#FFB6C1
HotPink	#FF69B4
DeepPink	#FF1493
MediumVioletRed	#C71585
PaleVioletRed	#DB7093



Turuncu

LightSalmon	#FFA07A
Coral	#FF7F50
Tomato	#FF6347
OrangeRed	#FF4500
DarkOrange	#FF8C00
Orange	#FFA500



Sari

Gold	#FFD700
Yellow	#FFFF00
LightYellow	#FFFFE0
LemonChiffon	#FFFACD
LightGoldenrodYellow	#FAFAD2
PapayaWhip	#FFEFD5
Moccasin	#FFE4B5
PeachPuff	#FFDAB9
PaleGoldenrod	#EEE8AA
Khaki	#F0E68C



Mor

Lavender	#E6E6FA
Thistle	#D8BFD8
Plum	#DDA0DD
Violet	#EE82EE
Orchid	#DA70D6
Fuchsia	#FF00FF
Magenta	#FF00FF
MediumOrchid	#BA55D3
MediumPurple	#9370DB

BlueViolet	#8A2BE2
DarkViolet	#9400D3
DarkOrchid	#9932CC
DarkMagenta	#8B008B
Purple	#800080
Indigo	#4B0082
SlateBlue	#6A5ACD
DarkSlateBlue	#483D8B
MediumSlateBlue	#7B68EE



Yeşil

GreenYellow	#ADFF2F
Chartreuse	#7FFF00
LawnGreen	#7CFC00
Lime	#00FF00
LimeGreen	#32CD32
PaleGreen	#98FB98
LightGreen	#90EE90
MediumSpringGreen	#00FA9A
SpringGreen	#00FF7F
MediumSeaGreen	#3CB371
SeaGreen	#2E8B57
ForestGreen	#228B22

Green	#008000
DarkGreen	#006400
YellowGreen	#9ACD32
OliveDrab	#6B8E23
Olive	#808000
DarkOliveGreen	#556B2F
MediumAquamarine	#66CDAA
DarkSeaGreen	#8FBC8F
LightSeaGreen	#20B2AA
DarkCyan	#008B8B
Teal	#008080



Mavi

Aqua	#00FFFF
Cyan	#00FFFF
LightCyan	#E0FFFF
PaleTurquoise	#AFEEEE
Aquamarine	#7FFFD4
Turquoise	#40E0D0
MediumTurquoise	#48D1CC
DarkTurquoise	#00CED1
CadetBlue	#5F9EA0
SteelBlue	#4682B4
LightSteelBlue	#B0C4DE
PowderBlue	#B0E0E6
LightBlue	#ADD8E6

SkyBlue	#87CEEB
LightSkyBlue	#87CEFA
DeepSkyBlue	#00BFFF
DodgerBlue	#1E90FF
CornflowerBlue	#6495ED
MediumSlateBlue	#7B68EE
RoyalBlue	#4169E1
Blue	#0000FF
MediumBlue	#0000CD
DarkBlue	#00008B
Navy	#000080
MidnightBlue	#191970



Kahverengi

Cornsilk	#FFF8DC
BlanchedAlmond	#FFEBCD
Bisque	#FFE4C4
NavajoWhite	#FFDEAD
Wheat	#F5DEB3
BurlyWood	#DEB887
Tan	#D2B48C
RosyBrown	#BC8F8F
SandyBrown	#F4A460

Goldenrod	#DAA520
DarkGoldenrod	#B8860B
Peru	#CD853F
Chocolate	#D2691E
SaddleBrown	#8B4513
Sienna	#A0522D
Brown	#A52A2A
Maroon	#800000



Beyaz

White	#FFFFFF	Beige	#F5F5DC
Snow	#FFFAFA	OldLace	#FDF5E6
Honeydew	#F0FFF0	FloralWhite	#FFFAF0
MintCream	#F5FFFA	Ivory	#FFFFF0
Azure	#F0FFFF	AntiqueWhite	#FAEBD7
AliceBlue	#F0F8FF	Linen	#FAF0E6
GhostWhite	#F8F8FF	LavenderBlush	#FFF0F5
WhiteSmoke	#F5F5F5	MistyRose	#FFE4E1
Seashell	#FFF5EE		



Gri

Gainsboro	#DCDCDC
LightGray	#D3D3D3
Silver	#C0C0C0
DarkGray	#A9A9A9
Gray	#808080
DimGray	#696969
LightSlateGray	#778899
SlateGray	#708090
DarkSlateGray	#2F4F4F
Black	#000000



KAYNAKLAR

- ▶ Microsoft resmi sitesinde yayınlanan Introducing Small Basic dosyasından,
- ▶ Microsoft Small Basic programı içerisindeki tanıtımlardan yararlanılmıştır.

Hazırlayan: durmuşali cilas
Bilişim teknolojileri öğretmeni

