

# De naslaggids voor ontwikkelaars voor Small Basic: Hoofdstuk 4: TextWindow-object

[Small Basic](#) > [Kleine basisboeken](#) > [De naslaggids voor ontwikkelaars voor small basic](#) > 4. Object TextWindow

## Voorbeeld

In dit hoofdstuk kijken we naar het object Small Basic **TextWindow**. Het wordt gebruikt om te werken met tekstinput en -output.

## TextWindow Object

De **TextWindow** is een object waar we tekstinput kunnen ontvangen en tekstoutput kunnen schrijven. In Small Basic worden tekstvensters meestal gebruikt voor eenvoudige programma's zonder grafische elementen. We hebben het tekstvenster gebruikt in de voorbeelden in hoofdstuk 3.

## TextWindow-eigenschappen:

### BackgroundColor

Hiermee wordt de achtergrondkleur van de tekst opgehaald of ingesteld die in het tekstvenster moet worden uitgevoerd.

### CursorLeft

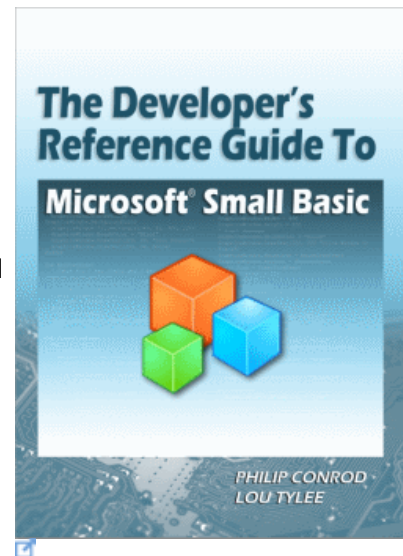
Hiermee wordt de kolompositie van de cursor in het tekstvenster opgehaald of ingesteld.

### CursorTop

Hiermee wordt de rijpositie van de cursor in het tekstvenster opgehaald of ingesteld.

### ForegroundColor

Hiermee wordt de voorgrondkleur van de tekst opgehaald of ingesteld om in het tekstvenster te worden uitgevoerd.



Dit hoofdstuk is een bewerking van het boek *The Developer's Reference Guide To Microsoft Small Basic* van Philip Conrod en Lou Tylee.

Om dit boek in zijn geheel te kopen, zie de [Computer Science For Kids website](#).

Links

Hiermee wordt de linkerpositie van het tekstvenster op het computerscherm opgehaald of ingesteld.

Titel

Krijgt of stelt de titel voor het tekstvenster in.

Top

Hiermee wordt de bovenste positie van het tekstvenster op het computerscherm opgehaald of ingesteld.

### **TextWindowMethods:**

Clear()

Hiermee wist u het tekstvenster.

Hide()

Hiermee verbergt u het tekstvenster.

Pause()

Wacht op gebruikersinvoer voordat u terugkeert.

PauseIfVisible()

Wacht alleen op gebruikersinvoer wanneer het tekstvenster is geopend.

PauseWithoutMessage()

Wacht op gebruikersinvoer voordat u terugkeert (maar er is geen bericht 'Druk op een toets').

Read()

Leest een regel tekst uit het tekstvenster. Deze methode wordt pas geretourneerd als de gebruiker op **Enter** drukt. Retourneert ingevoerde tekst.

ReadKey()

Leest een toetsaanslag in het tekstvenster. Hiermee geeft u als resultaat de ingedrukte toets.

ReadNumber()

Leest een getal uit het tekstvenster. Deze methode wordt pas geretourneerd als de gebruiker op **Enter** drukt. Retourneert ingevoerd nummer.

Show()

Toont het tekstvenster.

Write(data)

Hiermee schrijft u tekst of getal (**data**) naar het tekstvenster. In tegenstelling tot **WriteLine** voegt dit geen

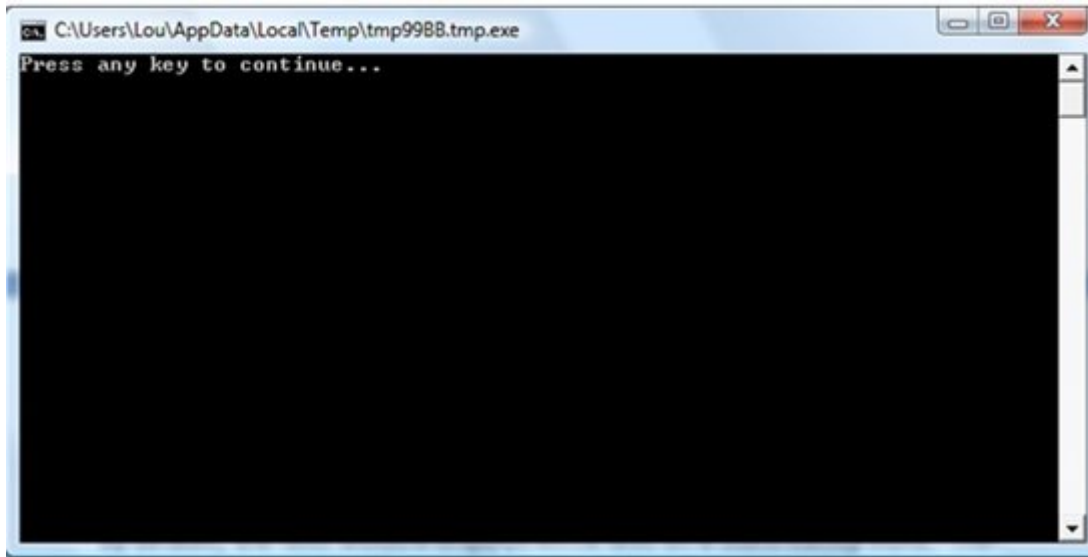
nieuw regelteken toe, wat betekent dat alles wat na deze aanroep naar het tekstvenster is geschreven, op dezelfde regel staat.

WriteLine(data)

Schrijft tekst of getal (**data**) naar het tekstvenster. Er wordt een nieuw regelteken aan de uitvoer toegevoegd, zodat de volgende keer dat iets naar het tekstvenster wordt geschreven, het in een nieuwe regel wordt geplaatst.

Functies van TextWindow

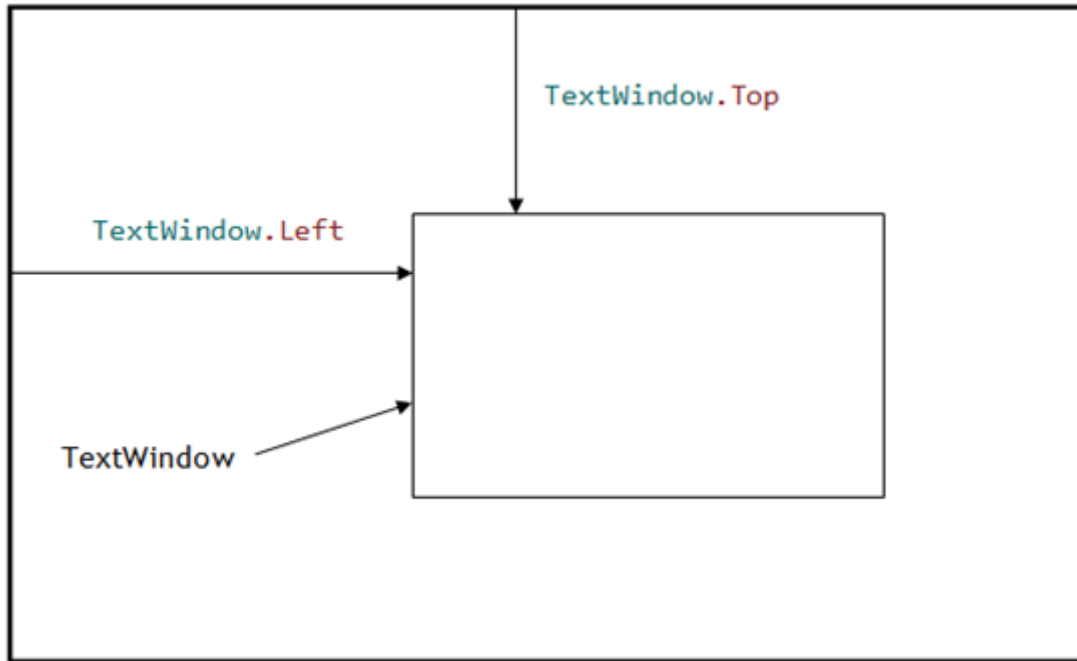
Standaard geeft het tekstvenster witte tekst weer op een zwarte achtergrond:



The default window has 25 rows (**CursorTop** values from 0 to 24) and 80 columns (**CursorLeft** values from 0 to 79)

To change the background color for the entire window, set the **TextWindow.BackgroundColor** property followed by a **Clear** method. The **ForegroundColor** property sets the text color.

The text window is located on your computer screen as follows:



#### Example 4-1. TextWindow Properties

Write a program that writes blue text on a yellow background. Write a line of text near the middle of the window.

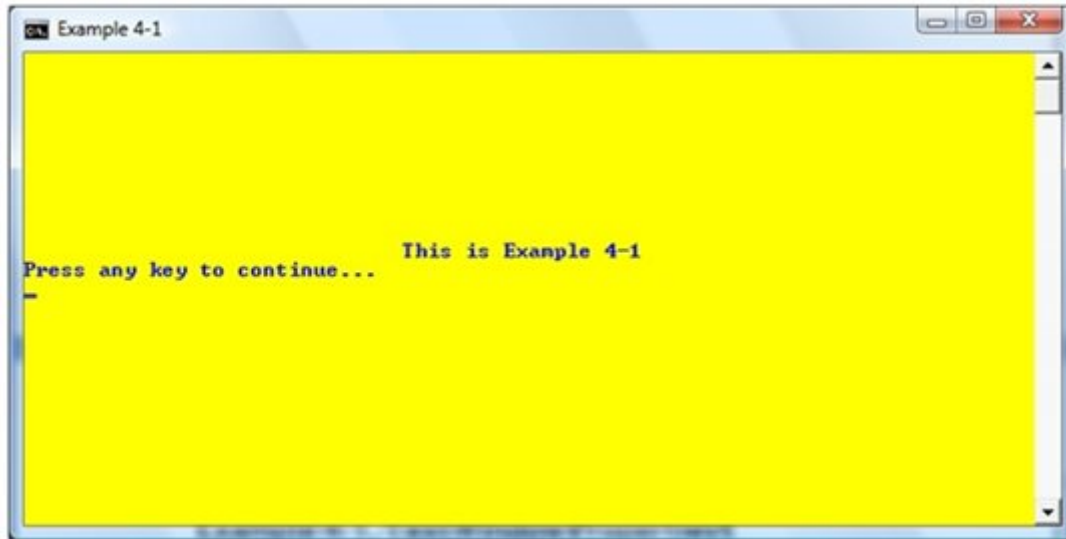
##### Small Basic Code:

```
1. ' Guide to Small Basic, Example 4-1
2. TextWindow.Title = "Example 4-1"
3. TextWindow.BackgroundColor = "Yellow"
4. TextWindow.ForegroundColor = "Blue"
5. TextWindow.Clear()
6. TextWindow.Left = 200
7. TextWindow.Top = 100
8. TextWindow.CursorLeft = 30
9. TextWindow.CursorTop = 10
10. TextWindow.WriteLine("This is Example 4-1")
```

Saved as **Example 4-1** in **Guide to Small Basic\Programs\Chapter 4** folder.

This program illustrates use of all the properties of the **TextWindow** object. The **Title** property is set as are **BackgroundColor** and **ForegroundColor**. We use the **WriteLine** method (preceded by **CursorLeft** and **CursorTop**) to position the text in the window. **Left** and **Top** position the window on your screen.

**Save** and **Run** the program to see the results:



## TextWindow Input

There are three methods used to get user input in the text window. The **Read** method is used to obtain text information. The **<Enter>** key is pressed to complete the input. Similarly, the **ReadNumber** method is used to obtain numeric information. With **ReadNumber**, non-numeric keystrokes are ignored. Lastly, the **ReadKey** method obtains individual keystrokes with no need to press **<Enter>**.

## Example 4-2. TextWindow Input

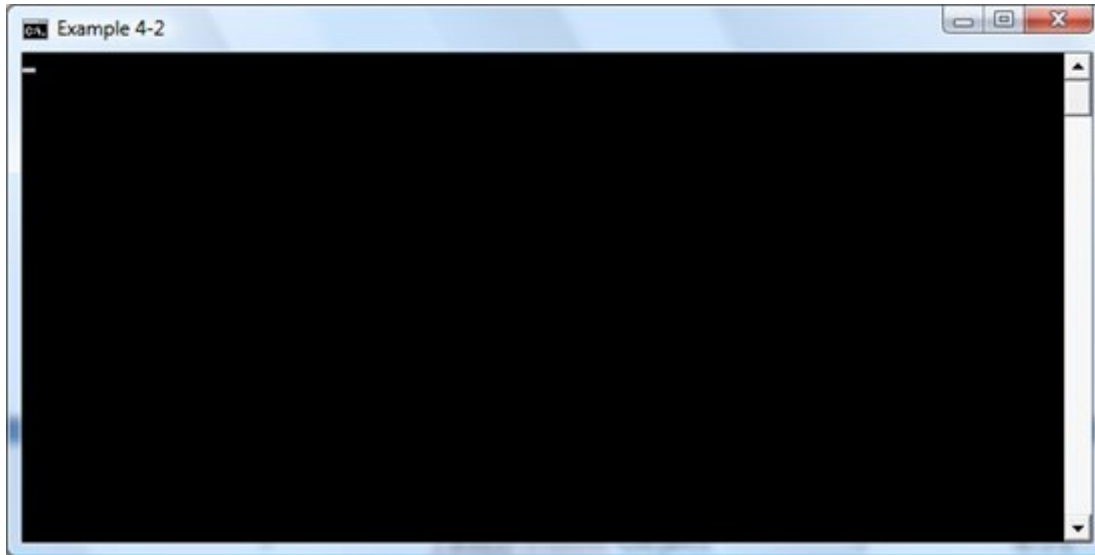
Write a program where the user first inputs a line of text then a numeric value. Lastly, intercept keystrokes typed by the user.

### Small Basic Code:

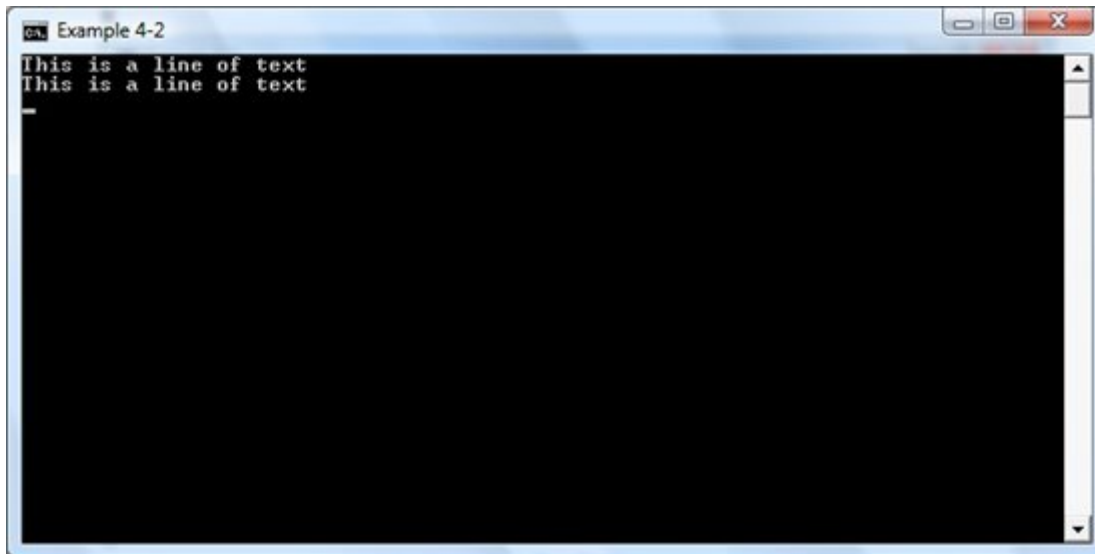
1. ' Guide to Small Basic, Example 4-2
2. TextWindow.Title = "Example 4-2"
3. A = TextWindow.Read()
4. TextWindow.WriteLine(A)
5. B = TextWindow.ReadNumber()
6. TextWindow.WriteLine(B)
7. GetKeystroke:
8. C = TextWindow.ReadKey()
9. TextWindow.WriteLine(C)
10. Goto GetKeystroke

Saved as **Example 4-2** in **Guide to Small Basic\Programs\Chapter 4** folder.

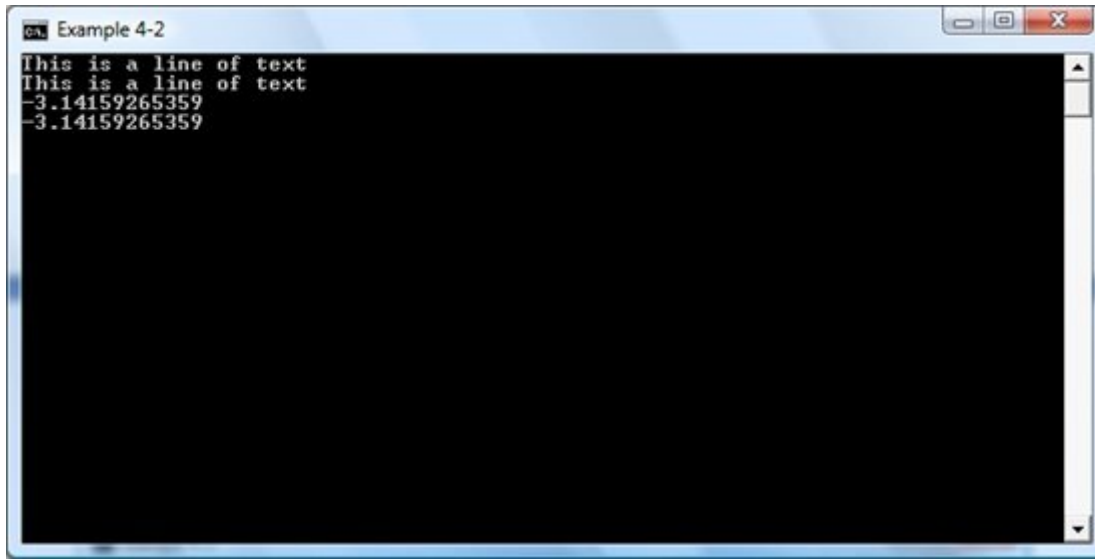
**Save** and **Run** the program. First, you see a flashing cursor waiting for you to type a line of text, then press **<Enter>**:



Type something in and press **<Enter>**. Here's what I see now:

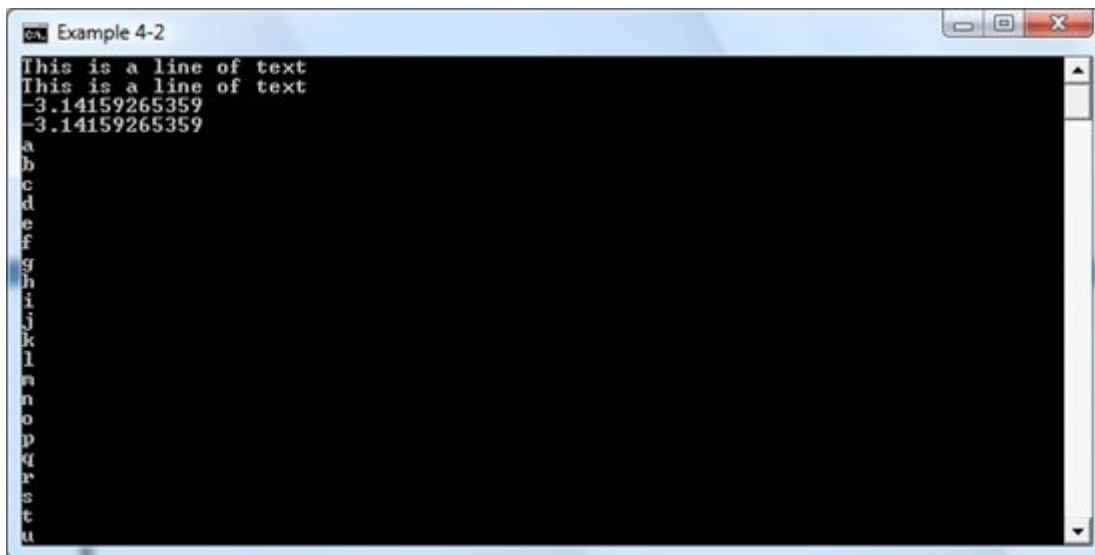


You see the line I input and the line written back out with **WriteLine**. The program is now waiting for a numeric input. Type such an input and press <**Enter**>. You will only be able to type numeric characters (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, single decimal point, single negative sign to start input). Try non-numeric characters – they won't appear. Here's what I have now (my input and it's written value):



```
Example 4-2
This is a line of text
This is a line of text
-3.14159265359
-3.14159265359
```

Now, the program is waiting for individual keystrokes. Type some. Notice there is no need to press <**Enter**> after the keystrokes. With each keystroke, the program will write out what you typed:



```
Example 4-2
This is a line of text
This is a line of text
-3.14159265359
-3.14159265359
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
```

To stop the program, click the **X** in the upper right corner of the window.

## TextWindow Output

There are two methods used to write information in the text window. We have used the **WriteLine** method. It writes the information to the window and appends a new line character. Any subsequent output methods start on a new line. **WriteLine** can also be used to write a blank line using:

```
TextWindow.WriteLine("")
```

An alternate to **WriteLine** is the **Write** method. It writes information to the window, but does not append a new line character. Subsequent output methods start at the end of the information written by the **Write** method. The **Write** method is useful for providing prompts prior to a **Read**, **ReadNumber** or **ReadKey** method.

Output using both **WriteLine** and **Write** can be positioned anywhere in the text window by setting the **CursorLeft** and **CursorTop** properties prior to using the output method. By default, each output method uses a **CursorLeft** value of 0. **CursorTop** is incremented by one with each **WriteLine** method.

### Example 4-3. Dice Rolling (TextWindow Output)

Write a program that rolls a dice until an input number of sixes is rolled (we showed this code when discussing **While** loops in Chapter 2). Once the rolls are complete, write out how many rolls it took to achieve the input value.

#### Small Basic Code:

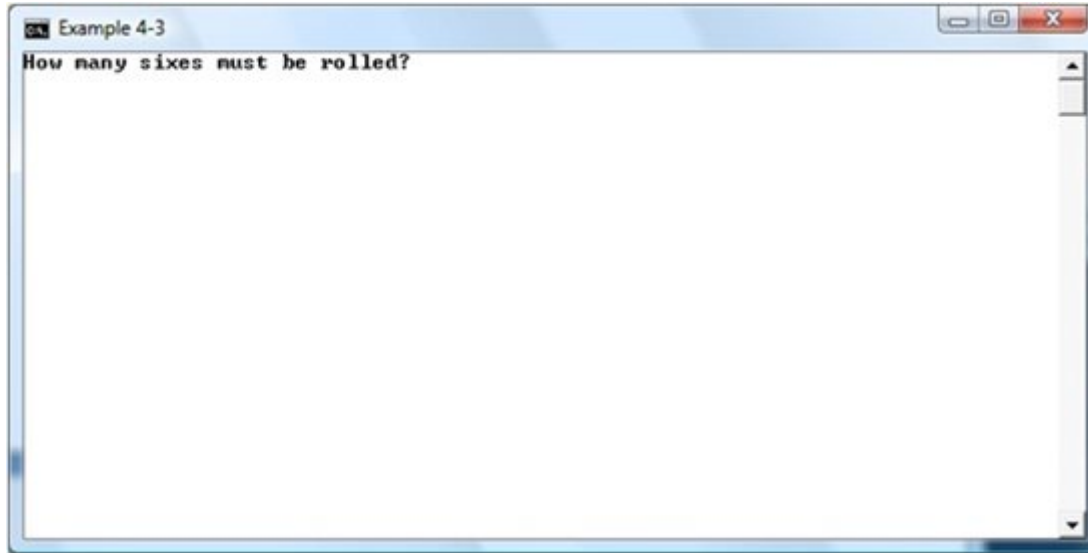
```
1. ' Guide to Small Basic, Example 4-3
2. TextWindow.Title = "Example 4-3"
3. TextWindow.BackgroundColor = "White"
4. TextWindow.ForegroundColor = "Black"
5. TextWindow.Clear()
6. GetNumberSixes:
7. TextWindow.Write("How many sixes must be rolled? ")
8. Number = TextWindow.ReadNumber()
9. Rolls = 0
10. Counter = 0
11. While (Counter < Number)
12. ' Roll a simulated die
13. Rolls = Rolls + 1
14. If (Math.GetRandomNumber(6) = 6) Then
15. Counter = Counter + 1
16. EndIf
17. EndWhile
18. TextWindow.WriteLine("It took " + Rolls + " rolls to get " + Number + " sixes.")
```



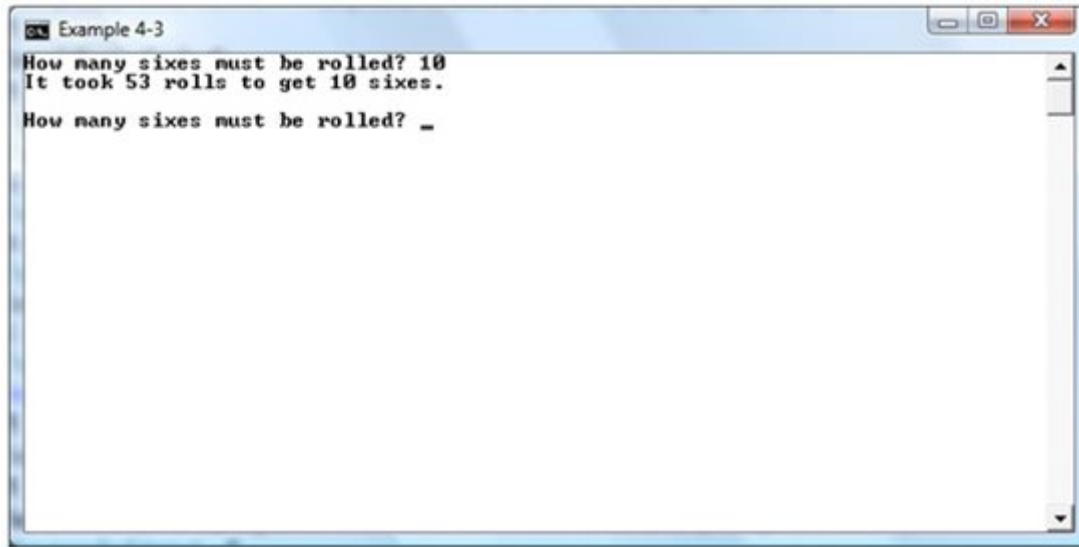
```
19. TextWindow.WriteLine("")
20. Goto GetNumberSixes
```

Saved as **Example 4-3** in **Guide to Small Basic\Programs\Chapter 4** folder.

**Save** and **Run** the program. The **Write** statement is used to prompt the user for the needed input (notice no new line is started – the flashing cursor is to the right of the question mark):



I picked 10 and see that it took 53 rolls to achieve 10 sixes:



```
Example 4-3
How many sixes must be rolled? 10
It took 53 rolls to get 10 sixes.
How many sixes must be rolled? _
```

Theoretically, it should take 60 rolls (since each roll has a 1 in 6 chance of rolling a six). At this point, I can type in another value. Try several values. Stop the program when you are done.

## Chapter Review

After completing this chapter, you should understand:

- Gebruik van het object **TextWindow**.
- **Tekstvenster-eigenschappen** instellen.
- Verschil tussen **Read**, **ReadNumber** en **ReadKey** methoden.
- Verschil tussen **WriteLine**- en **Write**-methoden.

Het object **TextWindow** kan alleen tekstinformatie accepteren en uitvoeren. Vervolgens kijken we naar het **GraphicsWindow-object** dat grafische elementen toestaat.

[Volgende hoofdstuk > >](#)