

De naslaggids voor ontwikkelaars voor Small Basic: Hoofdstuk 5: GraphicsWindow-object

[Small Basic](#) > [Kleine Basic E-Books](#) > [The Developer's Reference Guide to Small Basic](#) > 5. Grafischwindow-object

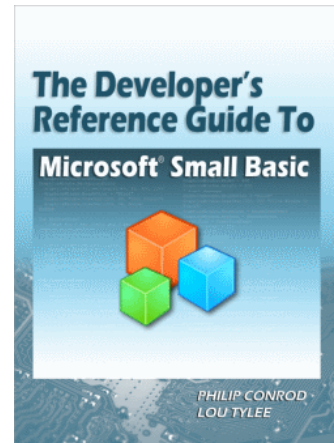
Voorbeeld

We zagen dat het tekstvenster nuttig is voor invoer en uitvoer van op tekst gebaseerde informatie. In dit hoofdstuk kijken we naar het object dat wordt gebruikt om de meeste Small Basic-programma's te hosten - de **GraphicsWindow**. Met dit object kunnen we zeer krachtige toepassingen bouwen.

Grafischwindow-object

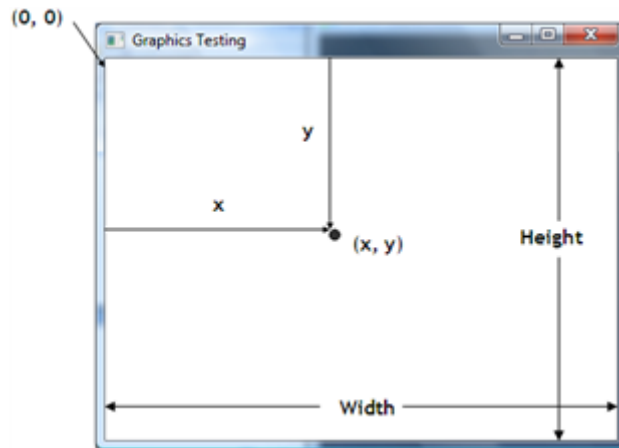
GraphicsWindow Coördinaten:

Het object **GraphicsWindow** is een hoeksteen van small basic programmeren. In het grafische venster kunnen we lijnen, vormen en tekst in vele kleuren tekenen. We kunnen besturingselementen (knoppen en tekstvakken) hosten. We kunnen muis- en toetsenbordinput van een gebruiker ontvangen. Het coördinatensysteem dat door het grafische venster wordt gebruikt, is:



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](#).



Het venster is **Breedte** pixels breed en **Hoogte** pixels hoog. We gebruiken twee waarden (coördinaten) om een enkele pixel in het venster te identificeren. De **x** (horizontale) coördinaat neemt toe van links naar rechts, beginnend bij **0**. De **y** (verticale) coördinaat neemt toe van boven naar beneden, ook beginnend bij **0**. Punten in het gebied worden aangeduid met de twee coördinaten tussen haakjes, of **(x, y)**.

GraphicsWindowEigenschappen:

BackgroundColor

Hiermee wordt de achtergrondkleur van het grafische venster opgehaald of ingesteld.

BrushColor

Hiermee wordt de penseelkleur opgehaald of ingesteld die moet worden gebruikt om vormen te vullen die in het afbeeldingsvenster zijn getekend.

CanResize

Hiermee geeft u op of het formaat van het grafische venster door de gebruiker kan worden gewijzigd. Kan "waar" of "onwaar" zijn.

FontBold

Krijgt of stelt in of het lettertype dat moet worden gebruikt bij het tekenen van tekst op het afbeeldingsvenster vet is.

FontItalic

Krijgt of stelt in of het lettertype dat moet worden gebruikt bij het tekenen van tekst op het afbeeldingsvenster, cursief is.

FontName

Hiermee wordt de lettertypenaam opgehaald of ingesteld wanneer u tekst tekent in het afbeeldingsvenster.

FontSize

Hiermee wordt de tekeningrootte opgehaald of ingesteld die moet worden gebruikt bij het tekenen van tekst in het afbeeldingsvenster.

Hoogte

Hiermee wordt de hoogte van het grafische venster opgehaald of ingesteld.

LastKey

Hiermee krijgt u de laatste toets die is ingedrukt of vrijgegeven.

LastText

Hiermee wordt de laatste tekst opgehaald die in het grafische venster is ingevoerd.

Links

Hiermee wordt de linkerpositie van het grafische venster opgehaald of ingesteld.

MouseX

hiermee wordt de x-positie van de muis ten opzichte van het grafische venster opgehaald.

MuisY

Hiermee wordt de y-positie van de muis ten opzichte van het grafische venster opgehaald.

PenColor

Hiermee wordt de kleur van de pen opgehaald of ingesteld die wordt gebruikt om vormen in het afbeeldingsvenster te tekenen.

PenWidth

Hiermee wordt de breedte van de pen opgehaald of ingesteld die wordt gebruikt om vormen in het afbeeldingsvenster te tekenen.

Titel

Hiermee wordt de titel voor het grafische venster opgehaald of ingesteld.

Top

Hiermee wordt de bovenste positie van het grafische venster opgehaald of ingesteld.

Breedte

Hiermee wordt de breedte van het afbeeldingsvenster opgehaald of ingesteld.

GraphicsWindowMethods:

Clear()

Clears the window.

DrawBoundText(x, y, text, w, h)

Draws a line of **text** on the screen at the specified location (**x**, **y**) within a region bounded by width **w** and height **h**. Helps define when text should wrap. Uses current brush and font properties.

DrawEllipse(x, y, w, h)

Draws an ellipse (width **w**, height **h**) at (**x**, **y**) on the screen using the current pen.

DrawImage(image, x, y)

Draws the specified **image** from memory on to the screen at (**x**, **y**).

DrawLine(x1, y1, x2, y2)

Draws a line from one point (**x1**, **y1**) to another (**x2**, **y2**). Uses current pen.

DrawRectangle(x, y, w, h)

Draws a rectangle (width **w**, height **h**) on the screen at (**x**, **y**) using the current pen.

DrawResizedImage(image, x, y, w, h)

Draws the specified **image** from memory on to the screen at (**x**, **y**), in the specified size (width **w**, height **h**).

DrawText(x, y, text)

Draws a line of **text** on the screen at the specified location (**x**, **y**). Uses current brush and font properties.

DrawTriangle(x1, y1, x2, y2, x3, y3)

Draws a triangle connecting the three input points on the screen using the current pen.

FillEllipse(x, y, w, h)

Fills an ellipse (width **w**, height **h**) on the screen at (**x**, **y**) using the current brush.

FillRectangle(x, y, w, h)

Fills a rectangle (width **w**, height **h**) on the screen at (**x**, **y**) using the current brush.

FillTriangle(x1, y1, x2, y2, x3, y3)

Fills a triangle connecting the three input points on the screen using the current brush.

GetColorFromRGB(red, green, blue)

Constructs a color give the **red**, **green**, **blue** values (0-255). Returns the color.

GetRandomColor()

Gets a valid random color. Returns the color.

Hide()

Hides the graphics window.

SetPixel(x, y, c)

Draws the pixel specified by (**x**, **y**) in the color **c**.

Show()

Shows the graphics window to enable interactions with it.

ShowMessage(text, title)

Displays a message box (with message **text** and **title**) to the user.

GraphicsWindowEvents:

KeyDown

Raises an event when a key is pressed down on the keyboard.

KeyUp

Raises an event when a key is released on the keyboard.

MouseDown

Raises an event when the mouse button is clicked down.

MouseMove

Raises an event when the mouse is moved around.

MouseUp

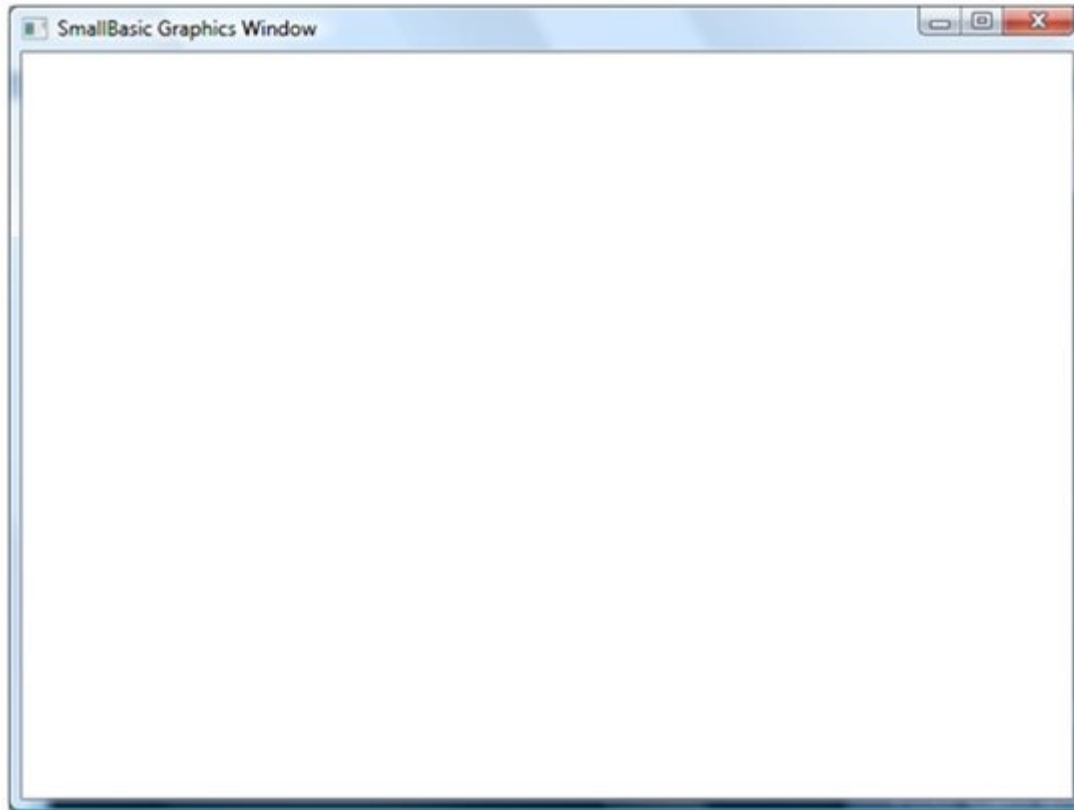
Raises an event when the mouse button is released.

TextInput

Raises an event when text is entered on the graphics window.

GraphicsWindow Features

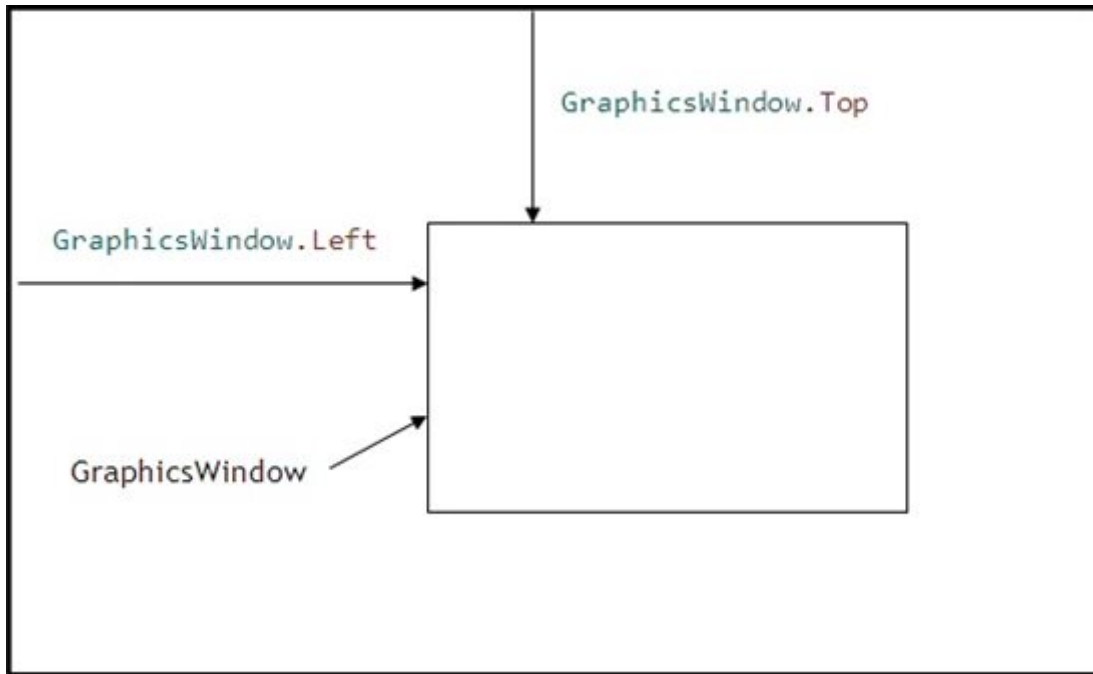
By default, the graphics window has a white background:



The default window is 624 pixels wide (**Width**) by 444 pixels high (**Height**).

To change the background color for the entire window, set the **GraphicsWindow.BackgroundColor**.

Similar to the text window, the graphics window is located on your computer screen as follows:



To center the graphics window on your computer screen, use these relations:

1. `GraphicsWindow.Left = 0.5 * (Desktop.Width - GraphicsWindow.Width)`
2. `GraphicsWindow.Top = 0.5 * (Desktop.Height - GraphicsWindow.Height)`

GraphicsWindow Colors

The graphics window (and also the text window) uses colors for various program elements. Colors are specified by color names. The color names used by Small Basic are listed in Appendix I to this guide. There are two other ways to get colors in Small Basic. The **GetRandomColor** method:

```
GraphicsWindow.GetRandomColor()
```

will return a random color. It is fun to use for display and games. The **GetColorFromRGB** method:

```
GraphicsWindow.GetColorFromRGB(Red, Green, Blue)
```

builds a color based on three specified components: **Red, Green, Blue**, each of which range from 0 to 255.

The graphics window background color is set by:

```
GraphicsWindow.BackgroundColor
```

And, as mentioned, the default value for this color is "White".

Lines and shapes in Small Basic are drawn using a "pen." The color and width of the pen is specified by:

1. GraphicsWindow.PenColor
2. GraphicsWindow.PenWidth

The default value for **PenColor** is "Black" and the default **PenWidth** is 2.

Shapes and text (yes, text) are filled (painted) using a "brush." The color of the brush is specified by:

GraphicsWindow.BrushColor

The default value for **BrushColor** is "SlateBlue".

GraphicsWindow Font

The font used to draw text in the graphics window is specified by four different properties

1. GraphicsWindow.FontName
2. GraphicsWindow.FontSize
3. GraphicsWindow.FontBold
4. GraphicsWindow.FontItalic

The **FontName** property is the name of the font. The default value is "Tahoma". Other font names can be found by opening a word processor and selecting the change font option.

The **FontSize** property sets the size of the current font. The default value is 12. **FontBold** can have one of two values. If "true", the font will be bold. If "false", it will not be bold. The default value is "true". Similarly, **FontItalic** indicates if a font is italicized. The default value is "false" – no italics.

Example 5-1. GraphicsWindow Properties

Write a program that writes "Graphics Window" in a large, bold, italic font in the middle of a yellow graphics window. Set window size to 400 by 150 pixels.

Small Basic Code:

1. 'Guide to Small Basic, Example 5-1
2. GraphicsWindow.Show()
3. GraphicsWindow.Title = "Example 5-1"
4. GraphicsWindow.Width = 400
5. GraphicsWindow.Height = 150
6. GraphicsWindow.BackgroundColor = "Yellow"
7. GraphicsWindow.FontSize = 36


```
8. GraphicsWindow.FontBold = "true"  
9. GraphicsWindow.FontItalic = "true"  
10. GraphicsWindow.BrushColor = "Black"  
11. GraphicsWindow.DrawText(20, 40, "Graphics Window")
```

Saved as **Example 5-1** in **Guide to Small Basic\Programs\Chapter 5** folder.

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



Example 5-2. SetPixel Method

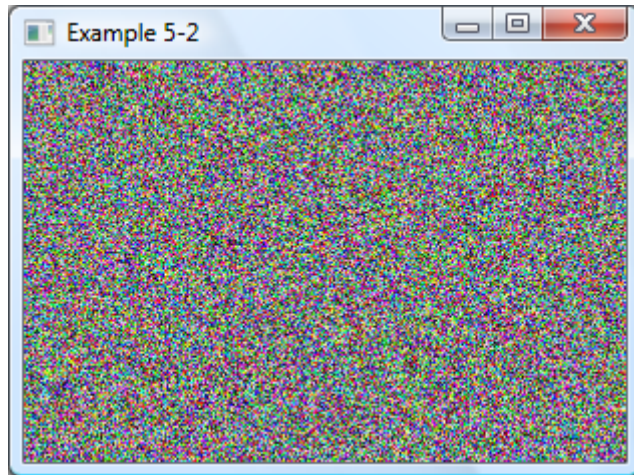
Write a program that fills a graphics window with randomly colored pixels.

Small Basic Code:

```
1. 'Guide to Small Basic, Example 5-2  
2. GraphicsWindow.Show()  
3. GraphicsWindow.Title = "Example 5-2"  
4. GraphicsWindow.Width = 300  
5. GraphicsWindow.Height = 200  
6. For X = 0 To 299  
7.   For Y = 0 To 199  
8.     GraphicsWindow.SetPixel(X, Y, GraphicsWindow.GetRandomColor())  
9.   EndFor  
10. EndFor
```

Saved as **Example 5-2** in **Guide to Small Basic\Programs\Chapter 5** folder. This code just "marches" through all the pixels and assigns a random color to each.

Save and **Run** the program to see the results (it takes a while for the window to fill):



Example 5-3. DrawLine Method

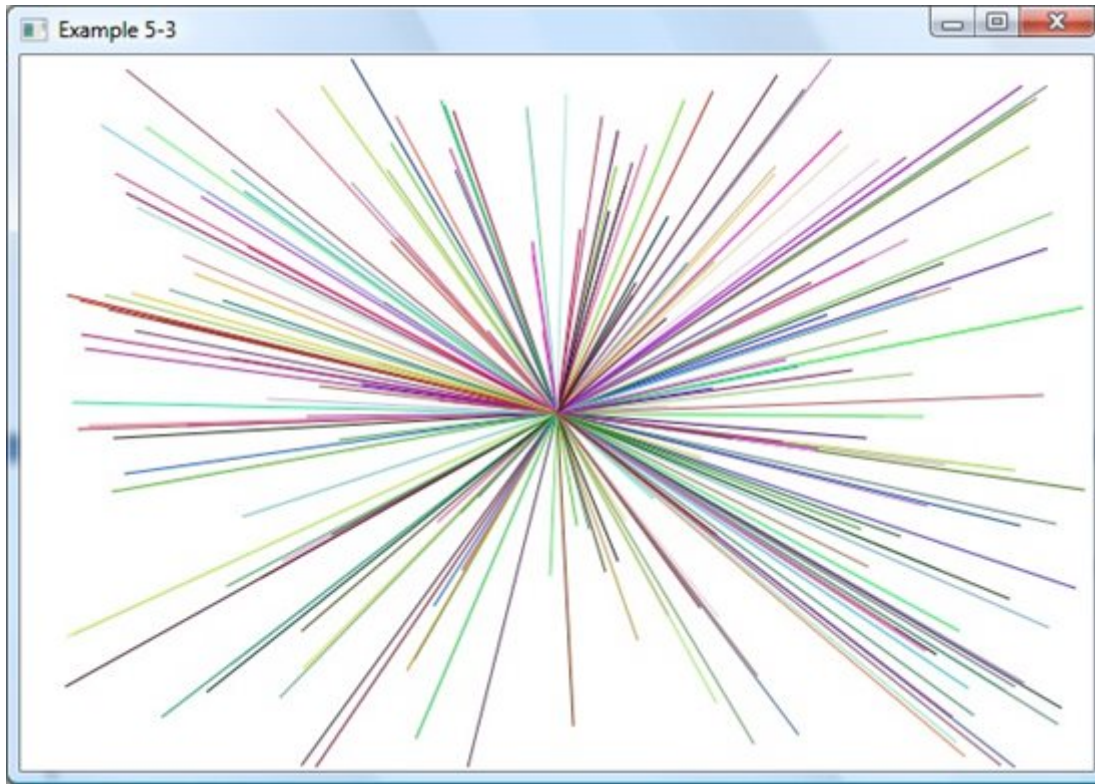
Write a program that draws randomly colored lines from the center of a graphics window out to some random point in the window.

Small Basic Code:

```
1. 'Guide to Small Basic, Example 5-3
2. GraphicsWindow.Show()
3. GraphicsWindow.Title = "Example 5-3"
4. GraphicsWindow.Width = 600
5. GraphicsWindow.Height = 400
6. GraphicsWindow.PenWidth = 1
7. For I = 1 To 200
8.   GraphicsWindow.PenColor = GraphicsWindow.GetRandomColor()
9.   X = Math.GetRandomNumber(600) - 1
10.  Y = Math.GetRandomNumber(400) - 1
11.  GraphicsWindow.DrawLine(300, 200, X, Y)
12. EndFor
```

Saved as **Example 5-3** in **Guide to Small Basic\Programs\Chapter 5** folder. Each line has the same starting point (the center of the window). Then, lines are drawn to the random point (X, Y).

Save and **Run** the program to see the results. Look at all the pretty lines:



Example 5-4. Drawing Rectangles

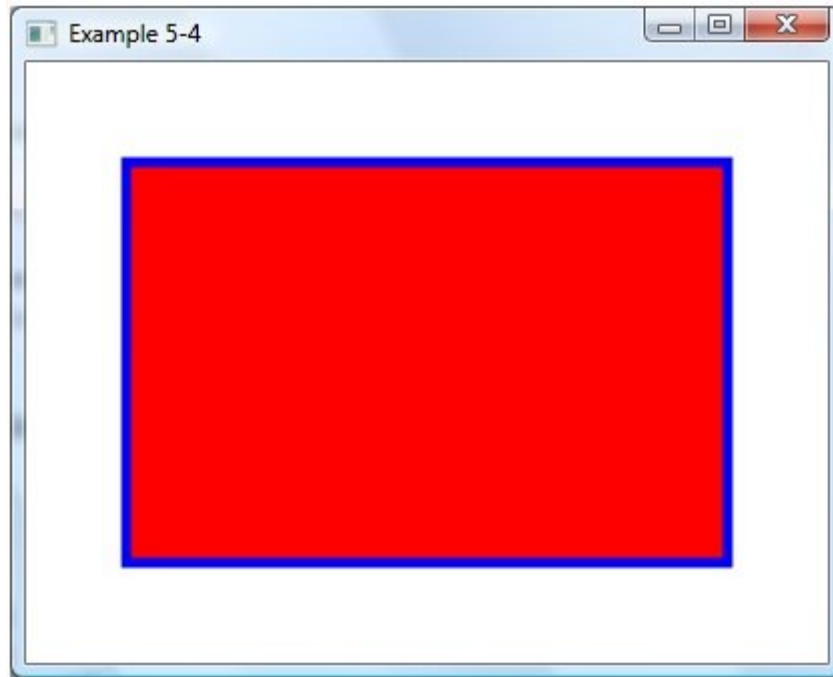
Write a program that draws a red rectangle, surround by a blue border.

Small Basic Code:

1. Guide to Small Basic, Example 5-4
2. GraphicsWindow.Show()
3. GraphicsWindow.Title = "Example 5-4"
4. GraphicsWindow.Width = 400
5. GraphicsWindow.Height = 300
6. GraphicsWindow.BrushColor = "Red"
7. GraphicsWindow.FillRectangle(50, 50, 300, 200)
8. GraphicsWindow.PenColor = "Blue"
9. GraphicsWindow.PenWidth = 5
10. GraphicsWindow.DrawRectangle(50, 50, 300, 200)

Saved as **Example 5-4** in **Guide to Small Basic\Programs\Chapter 5** folder.

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



We fill the rectangle first, then draw. This insures the border is not erased by the fill operation.

Example 5-5. Drawing Ellipses

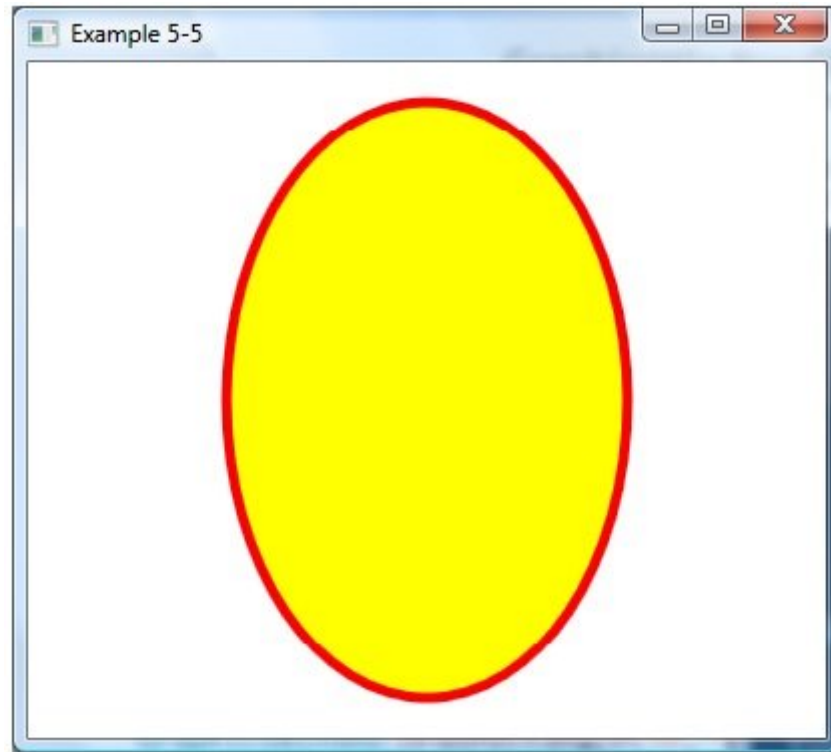
Write a program that draws a yellow ellipse, surround by a red border.

Small Basic Code:

```
1. 'Guide to Small Basic, Example 5-5
2. GraphicsWindow.Show()
3. GraphicsWindow.Title = "Example 5-5"
4. GraphicsWindow.Width = 400
5. GraphicsWindow.Height = 340
6. GraphicsWindow.BrushColor = "Yellow"
7. GraphicsWindow.FillEllipse(100, 20, 200, 300)
8. GraphicsWindow.PenColor = "Red"
9. GraphicsWindow.PenWidth = 5
10. GraphicsWindow.DrawEllipse(100, 20, 200, 300)
```

Saved as **Example 5-5** in **Guide to Small Basic\Programs\Chapter 5** folder.

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



Again, fill then draw to see the full border.

Example 5-6. Drawing Triangles

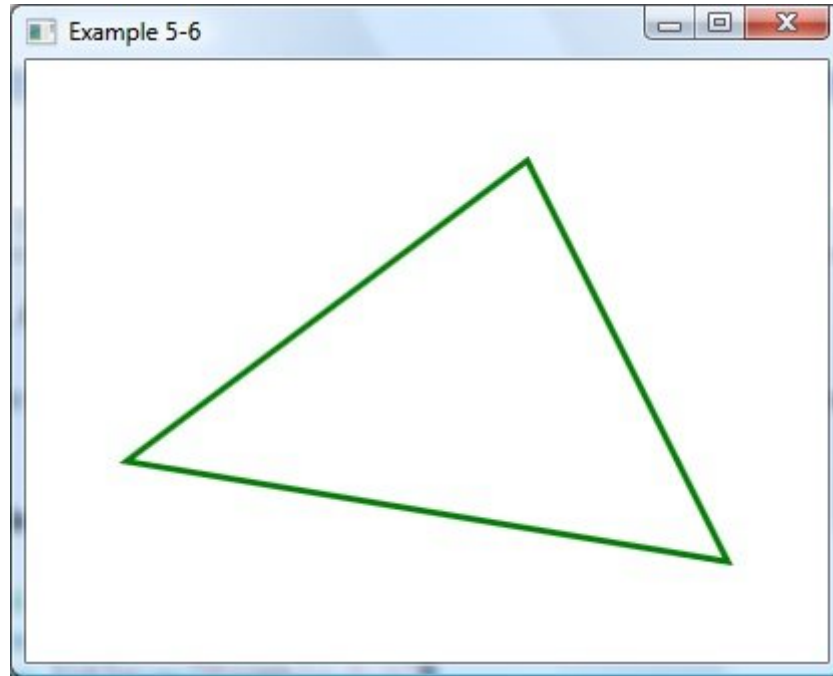
Write a program that draws a green-bordered triangle that connects these three points (250, 50), (50, 200), (350, 250).

Small Basic Code:

1. 'Guide to Small Basic, Example 5-6
2. GraphicsWindow.Show()
3. GraphicsWindow.Title = "Example 5-6"
4. GraphicsWindow.Width = 400
5. GraphicsWindow.Height = 300
6. GraphicsWindow.PenWidth = 3
7. GraphicsWindow.PenColor = "Green"
8. GraphicsWindow.DrawTriangle(250, 50, 50, 200, 350, 250)

Saved as **Example 5-6** in **Guide to Small Basic\Programs\Chapter 5** folder.

Save and **Run** the program to see the triangle (identify the three points):



Hoofdstuk Review

Na het voltooien van dit hoofdstuk moet u begrijpen:

1. Gebruik van het **GraphicsWindow**.
2. Hoe kleuren en lettertypen worden gebruikt in de grafische vensters.
3. Teken met **DrawText**, **DrawPixel**, **DrawLine** methoden.
4. Teken en vullen van rechthoeken, ellipsen, driehoeken.

We zullen veel meer doen met het grafische venster naarmate we meer te weten komen over Small Basic. In het bijzonder zullen we leren over het gebruik van de grafische venstergebeurtenissen om toetsaanslagen en muisklikken te herkennen.

Vervolgens leren we over enkele van de bedieningselementen die zijn gekoppeld aan Small Basic en het idee van gebeurtenisgestuurd programmeren.

