

C# tutorial-Drawing

Drawing text on the form

In C#, text and other shapes can be drawn on a form or on other controls. To start drawing objects on the control, you need to create graphic object for the control. The graphic object is created by using `CreateGraphics()` method of the control. To draw string on the control, you can use the `drawString()` method of the graphic object.

Example: drawing text on the form

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace WindowsFormsApplication2
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void Form2_Paint(object sender, PaintEventArgs e)
        {
            //place the C# code about drawing here
        }
    }
}
```

```
//Text to draw

string textToDraw = "Drawing text in C#";

//Create graphic object for the current form

Graphics gs = this.CreateGraphics();

//Create brush objec

Brush br = new SolidBrush(Color.BlueViolet);

//Create font object

Font f = new Font("Courier New", 20);

//Create point object

PointF pf = new PointF(30, 100);

PointF pf1 = new PointF(30, 140);

//Then you are ready to draw the text on your form

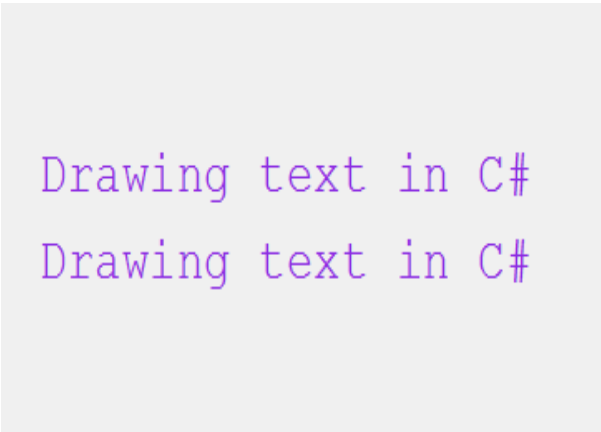
gs.DrawString(textToDraw, f, br, pf);

gs.DrawString(textToDraw, f, br, pf1);

}

}

}
```



Drawing text in C#
Drawing text in C#

Drawing lines on the form

After the graphic object is created by using `CreateGraphics()` method of the control. To draw a line on the control, you can use the `drawLine()` method of the graphic object. The `drawLine()` method requires three parameters. The first parameter is a pen object. You can specify the color of line to draw with the pen object. The other two parameters are point objects--start and end points of the line.

Example: drawing two different color lines on the form

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace WindowsFormsApplication2
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void Form2_Paint(object sender, PaintEventArgs e)
        {
```

```
//Create graphic object for the current form

Graphics gs = this.CreateGraphics();

//Create brush object

Brush br1 = new SolidBrush(Color.Red );

Brush br2 = new SolidBrush(Color.CadetBlue);

//Create pen objects

Pen p1 = new Pen(br1);

Pen p2 = new Pen(br2);

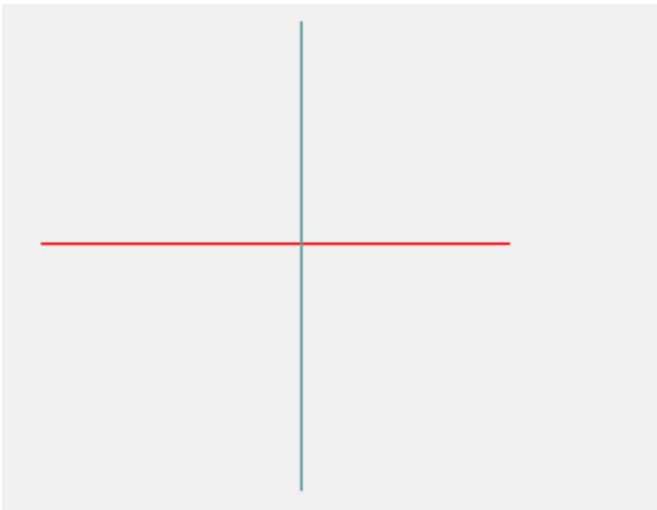
//Draw lines

gs.DrawLine(p1, new Point(20, 100), new Point(200, 100));

gs.DrawLine(p2, new Point(120, 10), new Point(120, 200));

}

}
```



Drawing rectangles on the form

To draw a rectangle shape on the control, you can use the `drawRectangle()` method of the graphic object. The `drawRectangle()` method requires two parameters. The first parameter is a pen object. The second is the rectangle object. If you want to draw many rectangles, the `drawRectangles()` method can be used instead. By using the latter method, you can pass an array of rectangle objects to draw.

Example: drawing three rectangles on the form using two different methods

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.ComponentModel;
```

```
using System.Data;
```

```
using System.Drawing;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Windows.Forms;
```

```
namespace WindowsFormsApplication2
```

```
{
```

```
    public partial class Form2 : Form
```

```
    {
```

```
        public Form2()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
        private void Form2_Paint(object sender, PaintEventArgs e)
```

```
        {
```

```
            //Create graphic object for the current form
```

```
            Graphics gs = this.CreateGraphics();
```

```
            //Create rectangle objects
```

```
            Rectangle rt1 = new Rectangle(30,90,100,200);
```

```
            Rectangle[] Crect = { new Rectangle(140, 90, 200, 200), new Rectangle(350, 90, 400, 200)
```

```
        };
```

```
//Create pen objects

Pen p = new Pen(new SolidBrush(Color.Green));

Pen p1 = new Pen(new SolidBrush(Color.BlueViolet));

//Draw rectangles

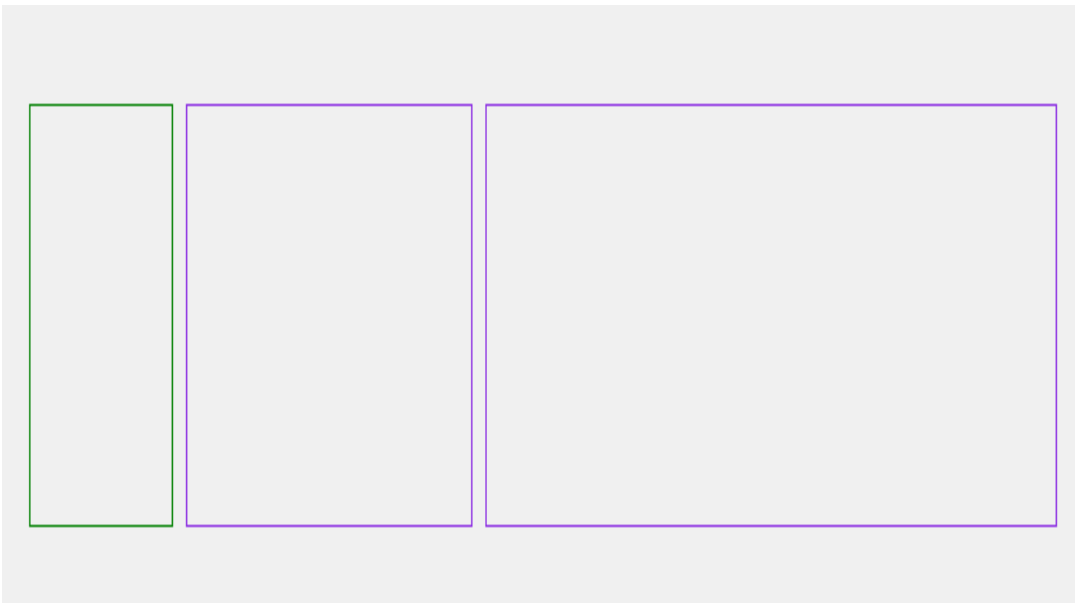
gs.DrawRectangle(p, rt1); //draw one rectangle

gs.DrawRectangles(p1, Crect); //draw more two rectangles

}

}

}
```



Drawing ellipses on the form

To draw an ellipse shape on the control, you can use the drawEllipse() method of the graphic object. The drawEllipse() method requires two parameters. The first parameter is a pen object. The second is the rectangle object.

Example: drawing two ellipses on the form

```
using System;

using System.Collections.Generic;

using System.ComponentModel;
```

```
using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace WindowsFormsApplication2
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void Form2_Paint(object sender, PaintEventArgs e)
        {

            //Create graphic object for the current form

            Graphics gs = this.CreateGraphics();

            //Create pen objects with thick width

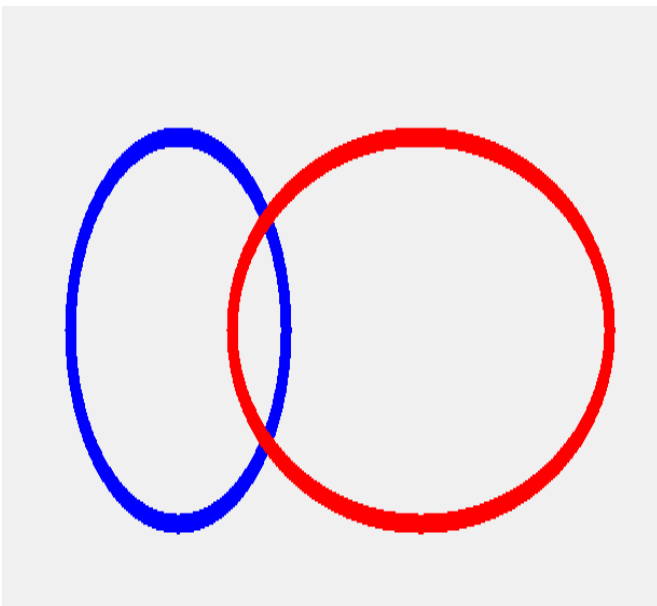
            Pen p = new Pen(new SolidBrush(Color.Blue), 10);

            Pen p1 = new Pen(new SolidBrush(Color.Red), 10);

            //Create rectangle objects

            Rectangle rt = new Rectangle(100, 80, 200, 200);
```

```
Rectangle rt1 = new Rectangle(250, 80, 350, 200);  
  
//Draw ellipses  
  
gs.DrawEllipse(p, rt);  
  
gs.DrawEllipse(p1, rt1);  
  
}  
  
}  
  
}
```



Drawing arcs on the form

To draw a curved shape using graphic object, you need to use its drawArc() method. This method accepts four parameters. The first parameter is the pen object. Second parameter is the rectangle object that determines the boundaries of arc shapes. The third and fourth parameters are start angle and end angle(respectively).

Example: drawing five arcs on the form

```
using System;  
  
using System.Collections.Generic;  
  
using System.ComponentModel;  
  
using System.Data;  
  
using System.Drawing;
```



```
using System.Linq;
```

```
using System.Text;
```

```
using System.Windows.Forms;
```

```
namespace WindowsFormsApplication2
```

```
{
```

```
    public partial class Form2 : Form
```

```
    {
```

```
        public Form2()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
        private void Form2_Paint(object sender, PaintEventArgs e)
```

```
        {
```

```
            //Create graphic object for the current form
```

```
            Graphics gs = this.CreateGraphics();
```

```
            //Create pen objects with thick width
```

```
            Pen p = new Pen(new SolidBrush(Color.Blue), 10);
```

```
            //Create a rectangle
```

```
            Rectangle rt = new Rectangle(100, 50, 200, 200);
```

```
            Rectangle rt1 = new Rectangle(120, 60, 160, 210);
```

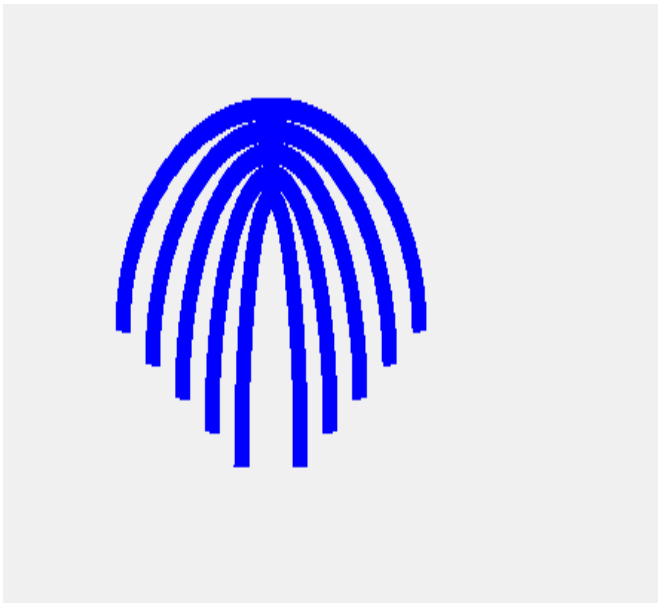
```
            Rectangle rt2 = new Rectangle(140, 70, 120, 220);
```

```
            Rectangle rt3 = new Rectangle(160, 80, 80, 230);
```

```
            Rectangle rt4 = new Rectangle(180, 90, 40, 240);
```

```
//Draw arcs
```

```
gs.DrawArc(p, rt, 0, -180);  
gs.DrawArc(p, rt1, 0, -180);  
gs.DrawArc(p, rt2, 0, -180);  
gs.DrawArc(p, rt3, 0, -180);  
gs.DrawArc(p, rt4, 0, -180);  
  
}  
}  
}
```



Filling rectangle shapes with one color

Rectangles that are drawn on the control can be filled with a color or a mix of colors by using `FillRectangle()` method of the graphic object.

Example: drawing fill the rectangle with red color

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.ComponentModel;
```

```
using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace WindowsFormsApplication2

{

    public partial class Form2 : Form

    {

        public Form2()

        {

            InitializeComponent();

        }

        private void Form2_Paint(object sender, PaintEventArgs e)

        {

            //Create graphic object for the current form

            Graphics gs = this.CreateGraphics();

            //Create a rectangle object

            Rectangle rt = new Rectangle(100, 100, 200, 200);

            //Fill the rectangle with red color

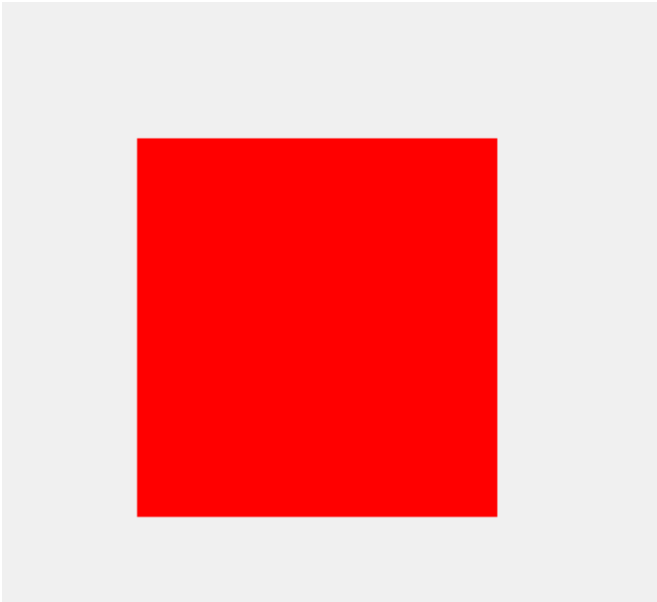
            gs.FillRectangle(new SolidBrush(Color.Red), rt);

        }

    }

}
```

```
    }  
  }  
}
```



Filling shapes with a mix of colors

Closed shapes(rectangles, ellipses,...) that are drawn on the control can be filled with a mix of colors by using `LinearGradientBrush` object. You can create a `LinearGradientBrush` object from the `LinearGradientBrush` class of `System.Drawing.Drawing2D` library.

Example: filling the rectangle and ellipse with a mix of colors

```
using System;  
  
using System.Collections.Generic;  
  
using System.ComponentModel;  
  
using System.Data;  
  
using System.Drawing;  
  
using System.Drawing.Drawing2D;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Windows.Forms;
```

```
namespace WindowsFormsApplication2
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void Form2_Load(object sender, EventArgs e)
        {
            this.WindowState = FormWindowState.Maximized;
        }

        private void Form2_Paint(object sender, PaintEventArgs e)
        {
            //Create graphic object for the current form
            Graphics gs = this.CreateGraphics();

            //Create a rectangle object
            Rectangle rt = new Rectangle(20, 100, 200, 200);
            Rectangle rt1 = new Rectangle(230, 100, 200, 200);

            //Create brush object
            LinearGradientBrush lb = new LinearGradientBrush(
                rt,
```

```
Color.FromArgb(255,0,255,255),
```

```
Color.FromArgb(255, 0, 0, 0),
```

```
LinearGradientMode.Horizontal);
```

```
LinearGradientBrush lb1 = new LinearGradientBrush(
```

```
rt1,
```

```
Color.FromArgb(255, 0, 255, 0),
```

```
Color.FromArgb(255, 0, 0, 255),
```

```
LinearGradientMode.Horizontal);
```

```
//fill the shapes
```

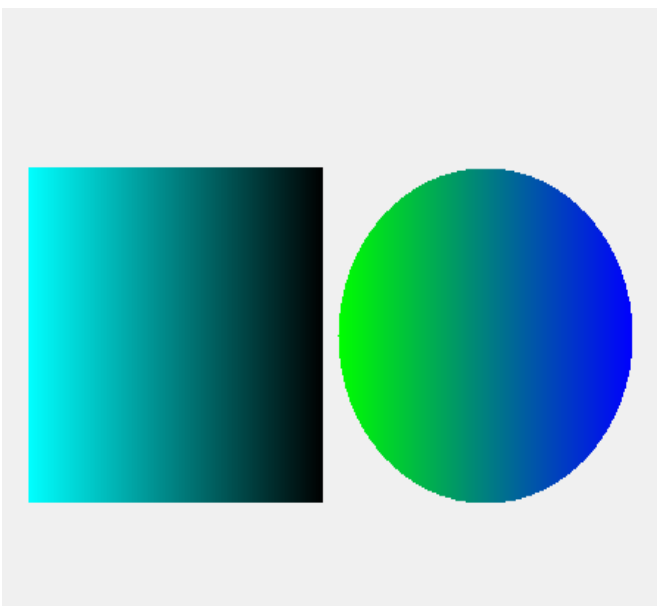
```
gs.FillRectangle(lb, rt);
```

```
gs.FillEllipse(lb1, rt1);
```

```
}
```

```
}
```

```
}
```



Filling shapes with a pattern

In this page you will learn to fill the closed shapes using different patterns. You can create a HatchBrush object from the HatchBrush class of System.Drawing.Drawing2D library and use it to fill your shapes.

Example: filling the rectangle and ellipse with different patterns

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace WindowsFormsApplication2
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void Form2_Load(object sender, EventArgs e)
        {
            this.WindowState = FormWindowState.Maximized;
        }
    }
}
```

```
}
```

```
private void Form2_Paint(object sender, PaintEventArgs e)
```

```
{
```

```
//Create graphic object for the current form
```

```
Graphics gs = this.CreateGraphics();
```

```
//Create rectangle objects
```

```
Rectangle rt = new Rectangle(20, 100, 200, 200);
```

```
Rectangle rt1 = new Rectangle(230, 100, 200, 200);
```

```
//Create brush objects
```

```
HatchBrush hb = new HatchBrush(HatchStyle.BackwardDiagonal, Color.FromArgb(255, 255, 100, 10));
```

```
HatchBrush hb1 = new HatchBrush(HatchStyle.LightDownwardDiagonal, Color.FromArgb(0, 10, 200, 255));
```

```
//fill the shapes
```

```
gs.FillRectangle(hb, rt);
```

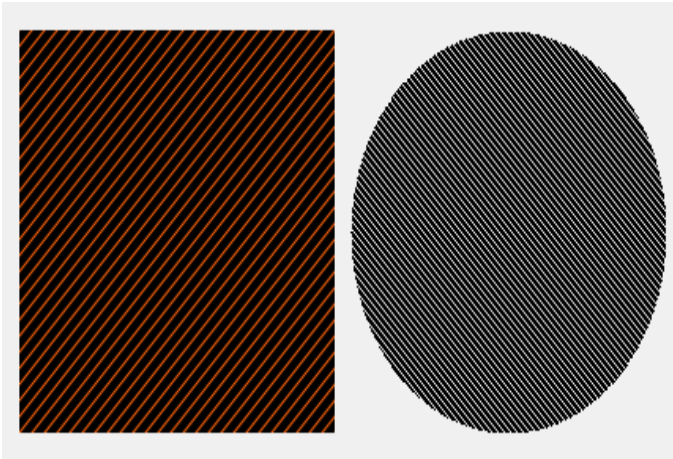
```
gs.FillEllipse(hb1, rt1);
```



```
}
```

```
}
```

```
}
```



Filling shapes with an image

In this page you will learn to fill the closed shapes using different images. To fill the shapes with images, You need to create a TextureBrush object from the TextureBrush class of System.Drawing library and use it to fill your shapes.

Example: filling the rectangle and ellipse with images

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.ComponentModel;
```

```
using System.Data;
```

```
using System.Drawing;
```

```
using System.Drawing.Drawing2D;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Windows.Forms;
```

```
namespace WindowsFormsApplication2
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void Form2_Load(object sender, EventArgs e)
        {
            this.WindowState = FormWindowState.Maximized;
        }

        private void Form2_Paint(object sender, PaintEventArgs e)
        {
            //Create graphic object for the current form
            Graphics gs = this.CreateGraphics();

            //Create a rectangle object
            Rectangle rt = new Rectangle(20, 100, 200, 200);
            Rectangle rt1 = new Rectangle(230, 100, 200, 200);

            //Create bitmap object to contain image in current pro. folder
            Image img = Image.FromFile("image1.bmp");
```

```
Bitmap bimage = new Bitmap(img);
```

```
//Create brush object
```

```
TextureBrush tb = new TextureBrush(bimage);
```

```
TextureBrush tb1 = new TextureBrush(bimage);
```

```
//fill the shapes
```

```
gs.FillRectangle(tb, rt);
```

```
gs.FillEllipse(tb1, rt1);
```

```
}
```

```
}
```

```
}
```

