Drawing rectangles - immediate drawing mode

INTRODUCTION

In the previous sections, we learned how to draw filled or wireframe rectangles.

As soon as the `ctx.strokeRect(x, y, width, height)` or the `ctx.fillRect(x, y, width, height)` method is called, a rectangle is indeed drawn *immediately* in the canvas.

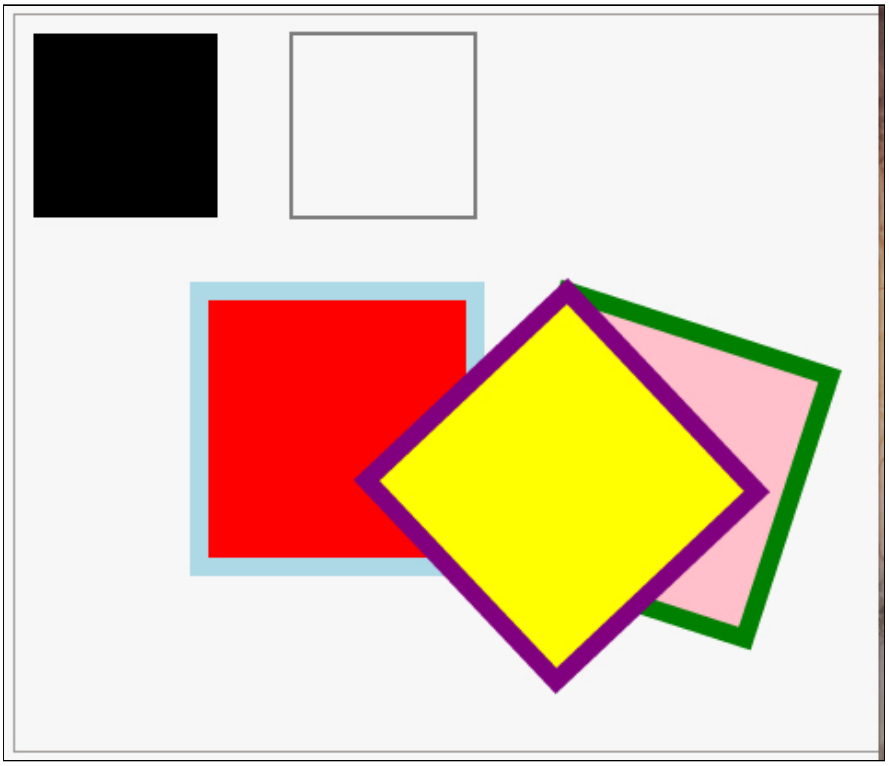
While drawing rectangles with `strokeRect` or `fillRect`, drawing text or drawing images, all these shapes will be drawn in *immediate mode*.

Another mode called "path mode" or "buffered mode" will be seen later in this course, which will be useful for drawing lines, curves, arcs, and also rectangles. Rectangles are the only shapes that have methods for drawing them *immediately* and also other methods for drawing them in "*path/buffered mode*".

EXAMPLE THAT DRAWS RECTANGLES IN IMMEDIATE MODE AND ALSO SHOWS GOOD PRACTICE

We'll just give an example here that draws several rectangles, filled or wireframe, with different colors and line widths.

Online example on JS Bin



Source code:

```
<!DOCTYPE HTML>
<html>
 <head>
 <title>Some rectangles drawn in immediate mode</title>
 <style>
   #myCanvas {
      border: 1px solid #9C9898;
   }
 </style>
 <script>
   var canvas, ctx;

   window.onload = function () {
     canvas = document.getElementById('myCanvas');
     ctx = canvas.getContext('2d');

     // black rectangle, default color (black)
     ctx.fillRect(10, 10, 100, 100);
     // outlined rectangle, default color
     ctx.strokeRect(150, 10, 100, 100);
```

```
    // outlined rectangle filled in red, outline blue
    ctx.fillStyle = 'red';
    ctx.strokeStyle = 'lightBlue';
    ctx.lineWidth = 10;
    ctx.fillRect(100, 150, 150, 150);
    ctx.strokeRect(100, 150, 150, 150);

    // A function that automatizes previous drawing
    var angle = Math.PI / 10;
    drawFilledRectangle(300, 150, 150, 150, 'pink', 'green', 10, angle);
    drawFilledRectangle(300, 150, 150, 150, 'yellow', 'purple', 10, angle + 0.5);
};

function drawFilledRectangle(x, y, w, h, fillColor, strokeColor, lw, angle) {
  // GOOD PRACTICE : save if the function change the context
  // or coordinate
  // system
  ctx.save();

  // position coordinate system
  ctx.translate(x, y);
  ctx.rotate(angle);

  // set colors, line width...
  ctx.lineWidth = lw;
  ctx.fillStyle = fillColor;
  ctx.strokeStyle = strokeColor;

  // draw at 0, 0 as we translated the coordinate
  // system already
 ctx.fillRect(0, 0, w, h);
 ctx.strokeRect(0, 0, w, h);

 // GOOD PRACTICE : a restore for a save!
 ctx.restore();
}
</script>
</head>
<body>
   <canvas id="myCanvas" width="578" height="400">
   </canvas>
</body>
```

```
</html>
```

KNOWLEDGE CHECK 3.3.1 (NOT GRADED)

What does immediate drawing mode mean?

- The HTML5 canvas' graphic context provides a few drawing methods that work in "immediate drawing mode". When calling these methods, the drawings appear on the canvas as soon as the instructions are executed.
- The HTML5 canvas works all the time in a special mode called "immediate drawing mode". It's easier to program as all graphic instructions produce results as soon as the instruction are executed. All methods that produce drawings work like that.