HTML Canvas Basics

HTML canvas is a drawing surface that allows you to dynamically render graphics using JavaScript. It provides a rectangular area where you can draw graphics, create animations, and generate interactive images. The canvas is initially blank and is only a container for graphics, which can be drawn using JavaScript.

To use the canvas, you first need to create it in your HTML code by using the canvas element:

```
<canvas id="myCanvas"></canvas>
```

Once you have created the canvas element, you can use JavaScript to obtain the canvas object and its context, which provides the necessary methods to draw graphics on the canvas:

```
const canvas = document.getElementById("myCanvas");

const ctx = canvas.getContext("2d");
```

The above code obtains the canvas object using its ID and then gets its context object by specifying "2d" as the argument. The context object provides several methods for drawing shapes, images, and text on the canvas.

Example :

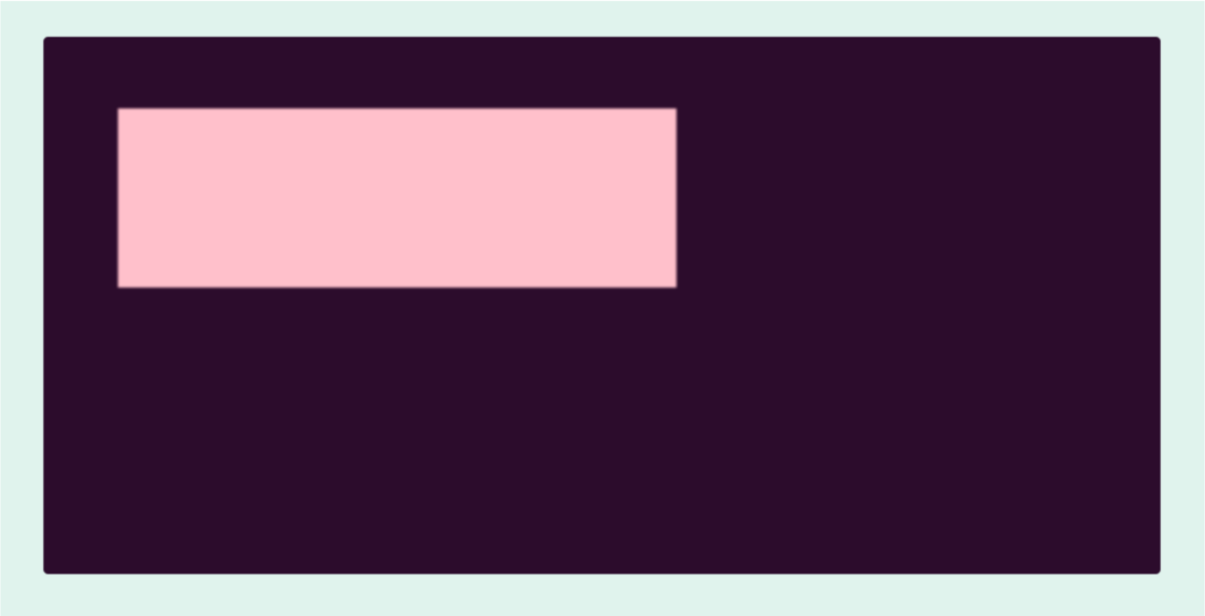
HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<style>
    *{
        margin: 0;
        padding: 0;
        box-sizing: border-box;
    }
    body{
        background-color: rgb(2, 28, 20);
        display: flex;
        align-items: center;
        color: black;
        justify-content: center;
        height: 100vh;
    }
    canvas{
        border-radius: 5px;
        background-color: rgb(255, 255, 255);
        width: 80%;
        height: 80%;
    }
</style>
<body>
    <canvas id="canvas"></canvas>
</body>
<script>
    const canvas=document.getElementById("canvas")
    const ctx=canvas.getContext('2d')
    const width=window.innerWidth;

    //rectangle
   /*ctx.fillStyle="pink"
    ctx.fillRect(20,20,150,50)*/

//stroke rectangle
/*ctx.linewidth=5
ctx.strokeStyle="white"
ctx.strokeRect(20,20,80,30)

ctx.linewidth=2
ctx.strokeStyle="yellow"
ctx.strokeRect(120,120,80,30)*/
```

```
//fill text
/*ctx.font = "30px April";
      ctx.fillStyle = "aqua";
      ctx.fillText("Hey there", 13, 50);*/


      //stroke text
/*ctx.strokeStyle="purple"
ctx.strokeText("H e l l o W o r l d",15,50)*/

//lines

/*ctx.beginPath()
ctx.moveTo(50, 50);
ctx.lineTo(100, 50);
ctx.lineTo(100, 100);
ctx.lineTo(50, 100);
ctx.closePath()

ctx.lineWidth=5
ctx.strokestyle="darkred"
ctx.stroke()
ctx.fillStyle="palegreen"
ctx.fill()*/


//circle
ctx.beginPath()
ctx.arc(60,50,40,0,Math.PI*2)
ctx.strokeStyle="darkred"
ctx.stroke()
ctx.fillStyle="palegreen"
ctx.fill()


</script>
</html>
```

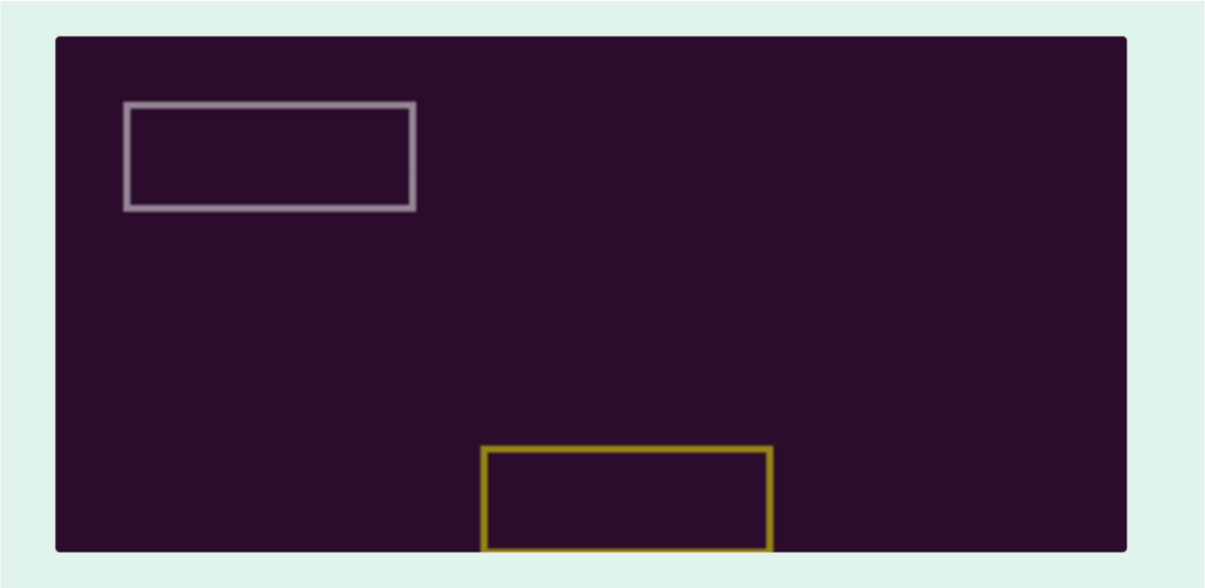
Output :

1. Rectangle :



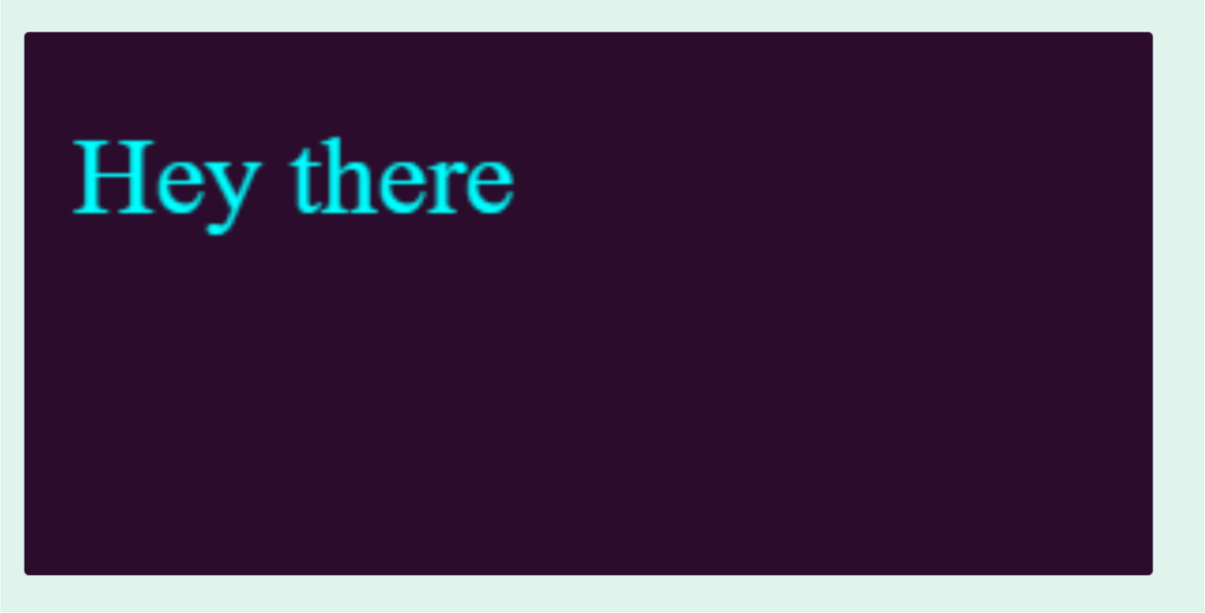
Rectangle

2. Stroke Rectangle :



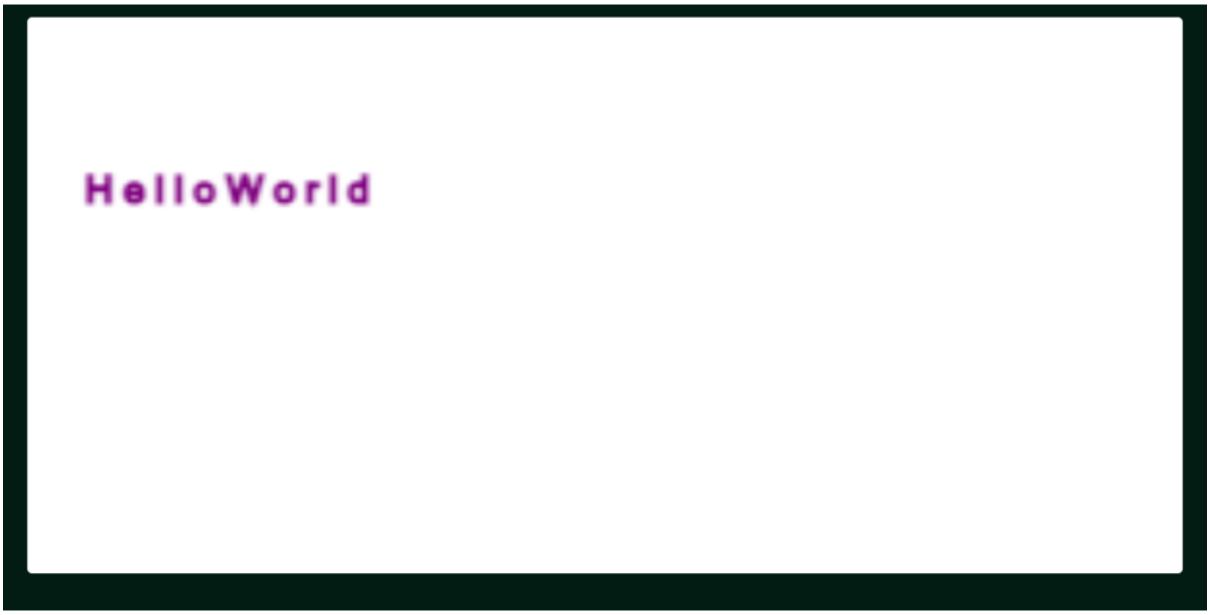
Stroke Rectangle

3. Fill Text :



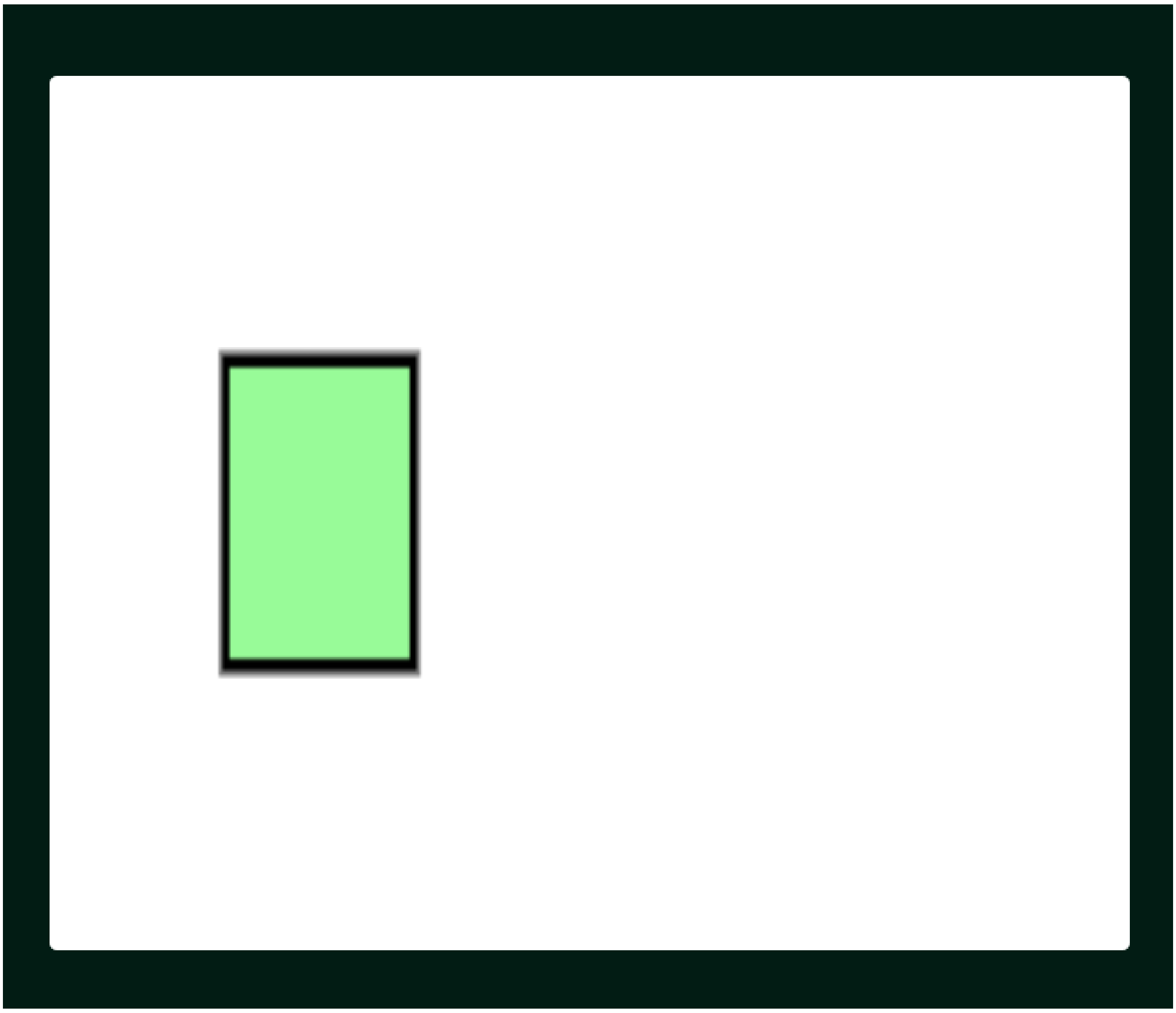
Fill Text

4. Stroke Text :



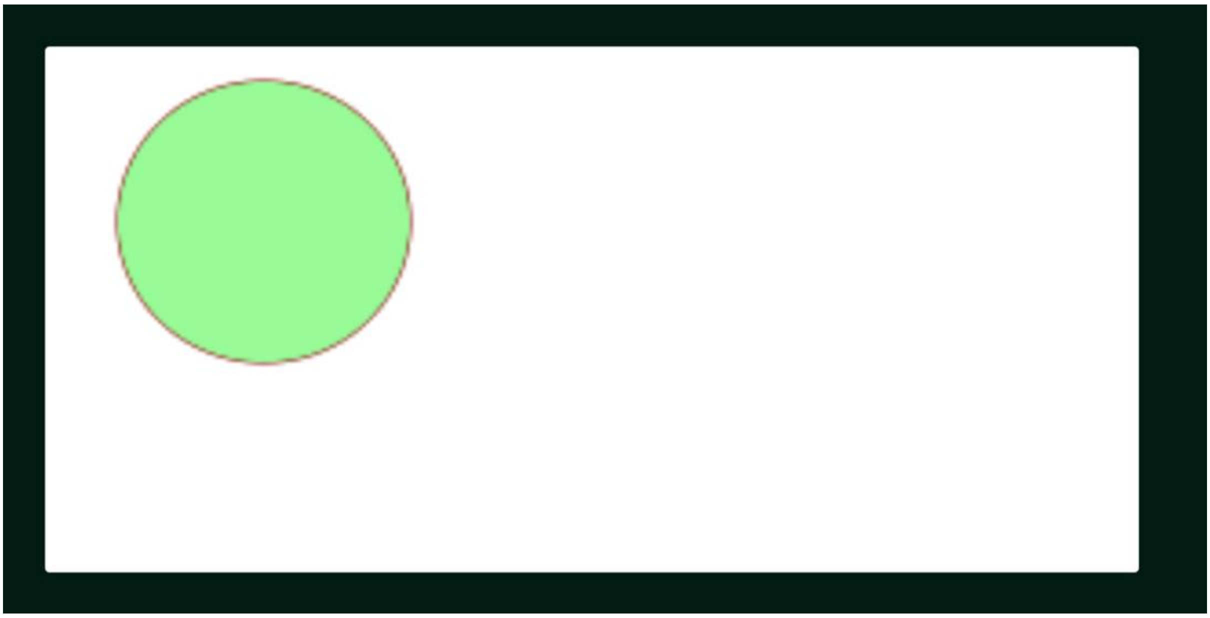
Stroke Text

5. Lines :



Lines

6. Circle :



Circle

Explanation :

This HTML document contains a *<canvas>* element, which allows for dynamic, scriptable rendering of graphics. The document includes a *<style>* section for defining the style of the page, including a background color, a centered layout with a border-radius and a white background color for the canvas element.

In the *<body>* section, there is a *<canvas>* element with the ID of "canvas". In the *<script>* section, the canvas element is accessed using the *getElementById()* method, and a 2D rendering context is obtained using the *getContext()* method.

In this document, various 2D graphics are drawn on the canvas element.

1. First, a rectangle is drawn using the *fillRect()* method, which takes in four parameters: the x-coordinate, the y-coordinate, the width, and the height. The *fillStyle* property is set to "pink" to fill the rectangle with a pink color.
2. Second, a stroke rectangle is drawn using the *strokeRect()* method. The *lineWidth* and *strokeStyle* properties are set to define the stroke style.
3. Third, a text is added to the canvas using the *fillText()* and *strokeText()* methods, which take in the text to be added, as well as the x- and y-coordinates of where the text should be positioned on the canvas.
4. Fourth, lines are added to the canvas using the *moveTo()* and *lineTo()* methods. The *beginPath()* method is used to start a new path, and the *closePath()* method is used to close the path. The *stroke()* method is used to stroke the path, and the *fill()* method is used to fill the path with the *fillStyle* property.
5. Fifth, a circle is drawn on the canvas using the *arc()* method. This method takes in six parameters: the x-coordinate, the y-coordinate, the radius, the start angle, the end angle, and whether the circle should be drawn in a clockwise or

counterclockwise direction. The *stroke()* and *fill()* methods are used to stroke and fill the circle with the *strokeStyle* and *fillStyle* properties.