Creating A Smiley And Bouncing Animation Using HTML Canvas

1. Creating A Smiley

Code :

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);

    }
</style>
```

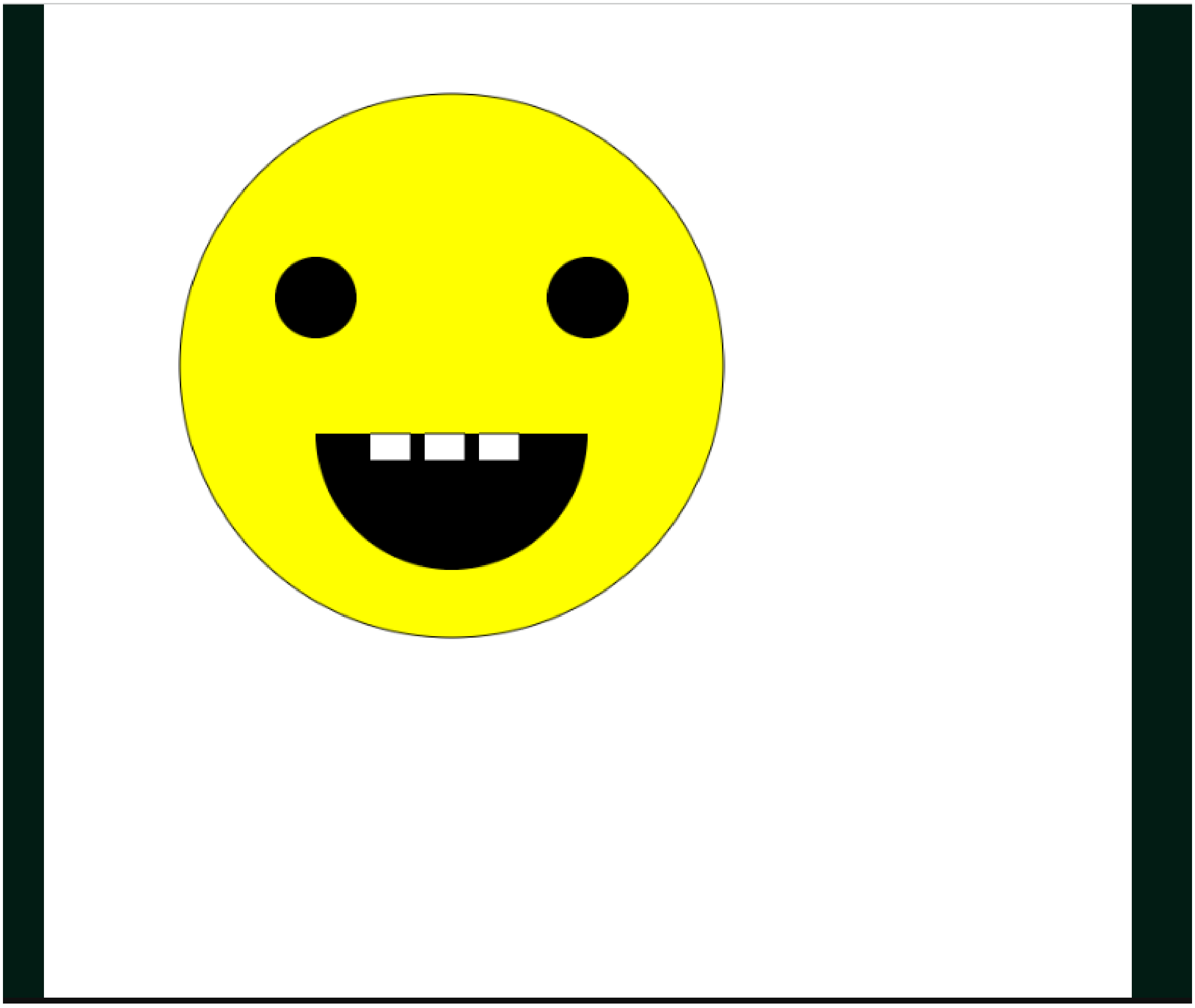
```
<body>
    <canvas id="canvas" width="800" height="800"></canvas>
</body>
<script>
    const canvas=document.getElementById("canvas")
    const ctx=canvas.getContext('2d')
    const width=window.innerWidth-100;
    ctx.beginPath()
    ctx.arc(300,300,200,0,Math.PI*2)
    ctx.fillStyle="yellow"
    ctx.fill()
    ctx.strokeStyle="black"
    ctx.stroke()
//eyes
ctx.beginPath()
ctx.arc(400,250,30,0,Math.PI*2)
ctx.fillStyle="black"
ctx.fill()
ctx.beginPath()
ctx.arc(200,250,30,0,Math.PI*2)
ctx.fillStyle="black"
ctx.fill()
//mouth
ctx.beginPath()
ctx.arc(300,350,100,0,Math.PI)
ctx.fillStyle="black"
ctx.fill()

//teeth
ctx.beginPath()
ctx.moveTo(240,350)
ctx.lineTo(270,350)
ctx.lineTo(270,370)
ctx.lineTo(240,370)
ctx.moveTo(240,350)
ctx.fillStyle="white"
ctx.fill()
ctx.stroke()

ctx.beginPath()
ctx.moveTo(280,350)
ctx.lineTo(310,350)
ctx.lineTo(310,370)
ctx.lineTo(280,370)
ctx.moveTo(280,350)
ctx.fillStyle="white"
ctx.fill()
ctx.stroke()

ctx.beginPath()
ctx.moveTo(320,350)
ctx.lineTo(350,350)
ctx.lineTo(350,370)
ctx.lineTo(320,370)
ctx.moveTo(320,350)
ctx.fillStyle="white"
ctx.fill()
ctx.stroke()
</script>
</html>
```

Output :



Smiley

Explanation :

The code is an HTML document that creates a canvas element with an id of "canvas" and a width and height of 800 pixels. The canvas is styled with a white background and a border radius of 5 pixels. The rest of the page is styled with a dark green background, centered content, and no margins or padding.Inside the script tag, the canvas context is retrieved and stored in a variable called "ctx". A yellow circle is drawn on the canvas using the "arc" method and a black stroke. Two smaller black circles are added for the eyes, and a black semicircle is added for the mouth. The teeth are drawn using a combination of lines and rectangles filled with white.

2. Creating Bouncing Ball

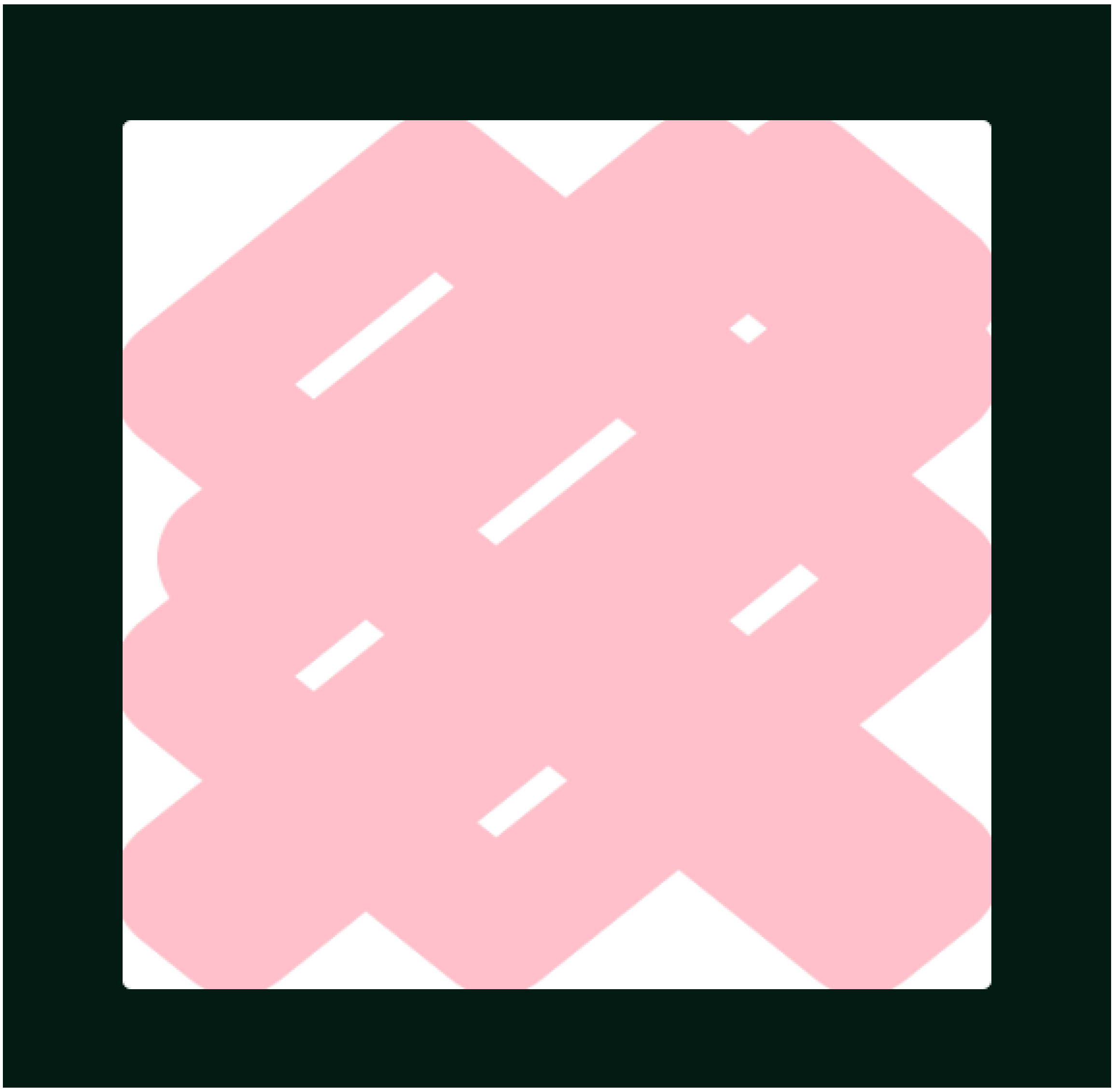
Code :

HTML

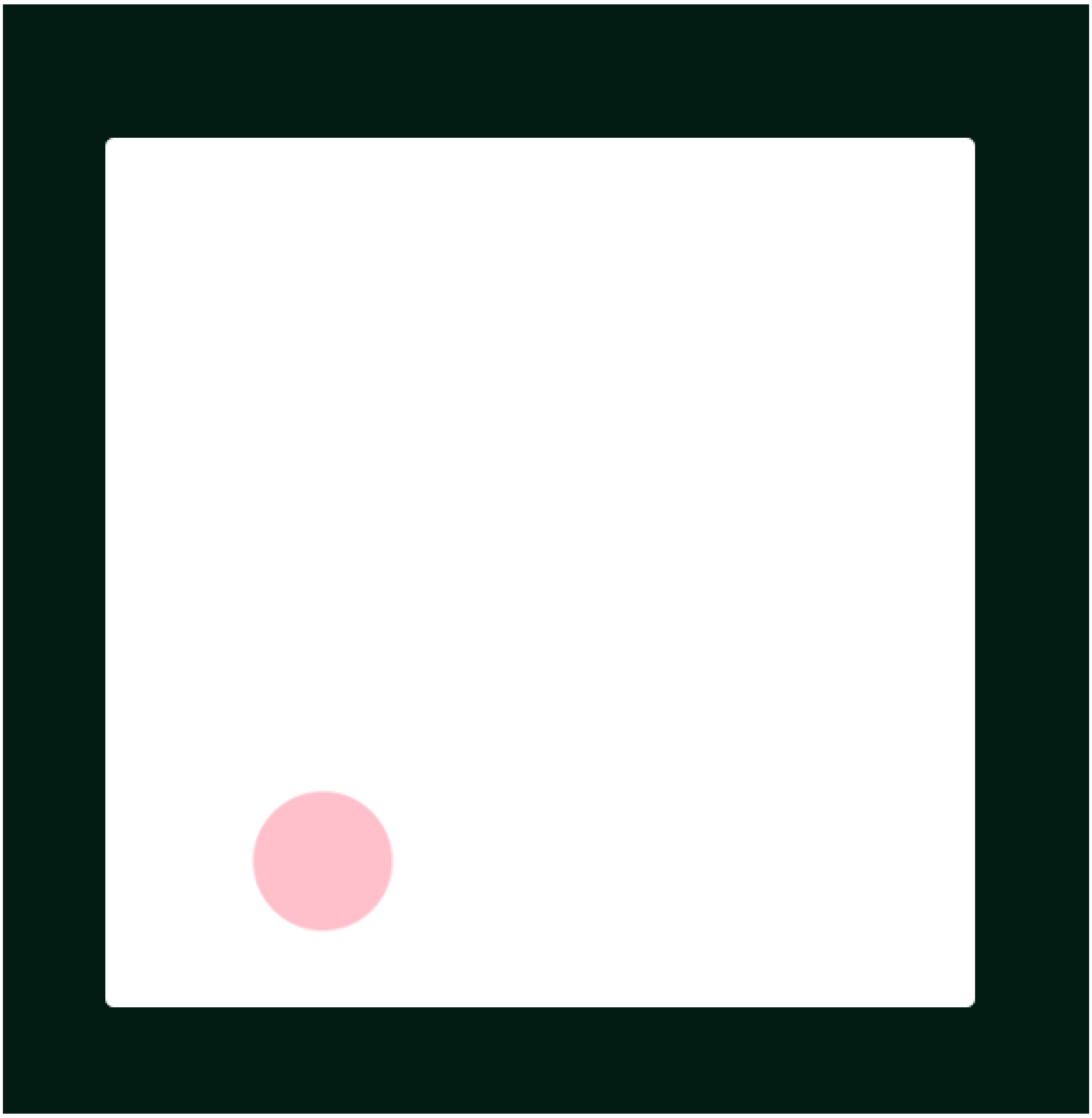
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<style>
    *{
        margin: 0;
        padding: 0;
        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);

    }
</style>
<body>
    <canvas id="canvas" width="500" height="500"></canvas>
</body>
<script>
    const canvas=document.getElementById("canvas")
    const ctx=canvas.getContext('2d')
    const circle={
        x:300,
        y:300,
        radius:40,
        dx:5,
        dy:4
    }
    function drawcircle(){
        ctx.beginPath()
        ctx.arc(circle.x,circle.y,circle.radius,0,Math.PI*2)
        ctx.fillStyle="pink"
        ctx.fill()
    }
    drawcircle()
    function update(){
        //ctx.clearRect(0,0,canvas.width,canvas.height)
        drawcircle()
        circle.x+=circle.dx
        circle.y+=circle.dy
        if (circle.x+circle.radius>canvas.width||circle.x-circle.radius<0){
            circle.dx*=-1
        }
        if (circle.y+circle.radius>canvas.height||circle.y-circle.radius<0){
            circle.dy*=-1
        }
        requestAnimationFrame(update)
    }
    update()
</script>
</html>
```

Output :



Bouncing Ball Without clearRect()



Bouncing Ball With clearRect()

Explanation :

This is an HTML code that displays a canvas element, which is a rectangular area on the screen where graphics can be drawn using JavaScript. The canvas has a width and height of 500 pixels.In the CSS section, the code sets some styles for the canvas, including a border-radius of 5 pixels, and a background color of white.In the JavaScript section, the code gets a reference to the canvas element and its context, which is a 2D rendering context.The code defines an object named *circle* that represents a circle that will be drawn on the canvas. The circle has an initial position of (300,300), a radius of 40 pixels, and a velocity represented by the *dx* and *dy* properties.The code defines a function named *drawcircle()* that uses the context to draw the circle on the canvas with a pink color.The code defines a function named *update()* that updates the position of the circle and redraws it on the canvas using *requestAnimationFrame()* method.

The *requestAnimationFrame()* method provides a smooth animation by taking the most optimal amount of time to redraw the canvas.The update function also checks if the circle hits the edge of the canvas and reverses the direction of the velocity if it does.