**LAB NO.1**

**Advanced HTML & CSS(CANVAS)**

## **What is HTML Canvas?**

* The HTML <canvas> element is used to draw graphics, on the fly, via scripting (usually JavaScript).
* The <canvas> element is only a container for graphics. You must use a script to actually draw the graphics.
* Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

## **Canvas Example**

In HTML, a <canvas> element looks like this:

<canvas id="myCanvas" width="200" height="100"></canvas>

* The <canvas> element must have an id attribute so it can be referred to by JavaScript.
* The width and height attribute is necessary to define the size of the canvas.

To add a border, use a style attribute:

### **Example**

<html>

<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">

</canvas>

</body>

</html>

It will produce the Rectangle with respect to given size.

# **HTML Canvas Drawing**

## **Draw on the Canvas With JavaScript**

All drawing on the HTML canvas must be done with JavaScript:

**Line Drawing**

<html>

<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">

Your browser does not support the HTML5 canvas tag.</canvas>

<script>

var c = document.getElementById("myCanvas"); Find the <canvas> element.

var ctx = c.getContext("2d"); The HTML Canvas Element.getContext() method returns a drawing context on the canvas.

ctx.moveTo(0,0);

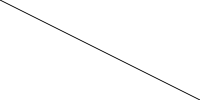
ctx.lineTo(200,100);

ctx.stroke(); The stroke() method actually draws the path you have defined with all those moveTo() and lineTo() methods. The default color is black.

</script>

</body>

</html>



### **Draw a Text**

<html>

<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">

Your browser does not support the HTML5 canvas tag.</canvas>

<script>

var c = document.getElementById("myCanvas");

var ctx = c.getContext("2d");

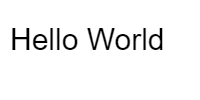
ctx.font = "30px Arial";

ctx.fillText("Hello World",10,50);

</script>

</body>

</html>



### **Example**

<html>

<body>

<canvas id="myCanvas" width="200" height="100"

style="border:1px solid #c3c3c3;">

Your browser does not support the canvas element.

</canvas>

<script>

var canvas = document.getElementById("myCanvas");

var ctx = canvas.getContext("2d");

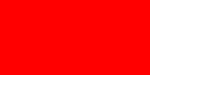
ctx.fillStyle = "#FF0000";

ctx.fillRect(0,0,150,75);

</script>

</body>

</html>



## **Step 1: Find the Canvas Element**

First of all, you must find the <canvas> element.

This is done by using the HTML DOM method getElementById():

var canvas = document.getElementById("myCanvas");

## **Step 2: Create a Drawing Object**

Secondly, you need a drawing object for the canvas.

The getContext() is a built-in HTML object, with properties and methods for drawing:

var ctx = canvas.getContext("2d");

## **Step 3: Draw on the Canvas**

Finally, you can draw on the canvas.

Set the fill style of the drawing object to the color red:

ctx.fillStyle = "#FF0000";

The fillStyle property can be a CSS color, gradient, or a pattern. The default fillStyle is black.

The fillRect(*x,y,width,height*) method draws a rectangle, filled with the fill style, on the canvas:

ctx.fillRect(0, 0, 150, 75);

**A Simple Program to draw a triangle ,also apply any style and colors to the shape**.

<html>

<head>

<title>Triangle Canvas Example</title>

</head>

<body>

<canvas id="myCanvas" width="500" height="500"></canvas>

<script>

var canvasElement = document.getElementById("myCanvas");

var context = canvasElement.getContext("2d");

The first line gets a pointer the canvas element in our HTML. The second line gets you access to the canvas element context object that allows you to actually draw things into the canvas.

// the triangle

context.beginPath();The beginPath method signals to the canvas that you intend to draw a path.

context.moveTo(100, 100);That is handled by the moveTo function which takes an x and y co-ordinate value.

context.lineTo(100, 300);

context.lineTo(300, 300);The first lineTo function draws a line from our starting point of (100, 100) to (100, 300):

context.closePath();TheclosePath() method tells your pen to draw a line back to the starting point.

// the outline

context.lineWidth = 10;

context.strokeStyle = '#666666';

context.stroke();The lineWidth and strokeStyle properties specify the thickness and color of the line we want to draw. The actual drawing of the line is handled by calling the stroke method. At this point, you will see a triangle whose outlines are actually visible:

// the fill color

context.fillStyle = "#FFCC00";

context.fill();The fillStyle property allows you to define the color. The fill method tells your canvas to go ahead and fill up the insides of our closed path with that color.

</script>

</body>

</html>

**LAB TASK**

1. Write a Program to draw a triangle and rectangle both in one code ,also apply any style and colors to both shapes .