5) formtarget → <input formtarget="_blank" >

6) formnovalidate → <input formnovalidate="formnovalidate" value="Submit without validation">

18/5/21

Day 4 – Graphics (Canvas)

- The <canvas> element is used to display graphics on a web page, via JS.

→ It is only a container for Graphics.

→ No borders, no content by default

Syntax:

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

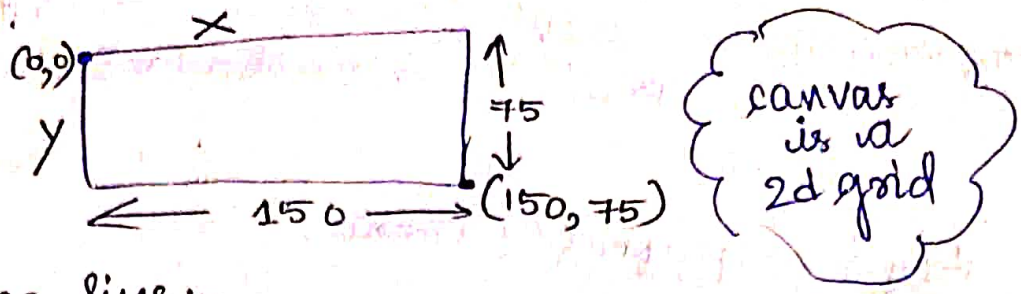
id → important while using in JS

width, height → size of canvas.

Drawing on canvas with JS :

```
<!DOCTYPE HTML>
<html>
<body>
<canvas id="myCanvas" width="200" height="100"
  style="border:1px solid #c3c3c3;">
Your browser doesn't support canvas → alt text
</canvas>

<script>
→ finding canvas element
var canvas = document.getElementById("myCanvas");
var ctx = canvas.getContext("2d"); //creating a drawing object for canvas

ctx.fillStyle = "#FF0000"; //to draw on canvas, we need a color/gradient or pattern
→ default → black.

ctx.fillRect(0,0,150,75);
             x y w h → height
                 ↓ width

</script>
</body>
</html>
```

* The co-ordinates start at (0,0) at top-left corner.



Drawing line :-

ctx.moveTo(0,0); → define a starting point.
ctx.lineTo(200,100); → define ending point.
ctx.stroke(); → stroke() method to strike from start to end.

Drawing circle

ctx.beginPath(); — begins a path
ctx.arc(95/x, 50/y, 40/r, 0/start angle, 2*Math.PI/endangle);
ctx.stroke();

(x,y) → centre of circle coordinates

Draw a Text

ctx.font = "30px Arial";
ctx.fillText("HelloWorld", 10, 50);

Hello World

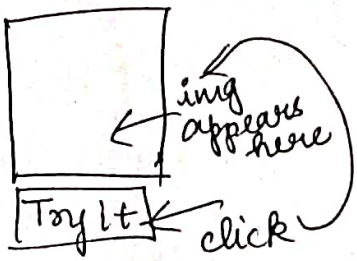
Stroke Text:

ctx.font = "30px Arial";
ctx.strokeText("HelloWorld", 10, 50);

HELLO WORLD

* Drawing images:

```
<img id="scream" src=" " >
<p> <button onclick = "myCanvas()" > Try it </button></p>
<script>
    function myCanvas() {
        var c = document.getElementById("myCanvas");
        var ctx = c.getContext("2d");
        var img = document.getElementById("scream");
        ctx.drawImage(img, 10,10);
    }
</script>
```



* SVG (<svg>)

→ Scalable Vector Graphics
→ container for svg graphics

SVG circle

```
<svg width="100" height="100">
    <circle cx="50" cy="50" r="40" stroke="green"
    stroke-width="4" fill="yellow" />
</svg>
```

Rectangle

```
<rect width=" " height=" "
style="fill: red; stroke-width: 10; stroke: black;" />
```

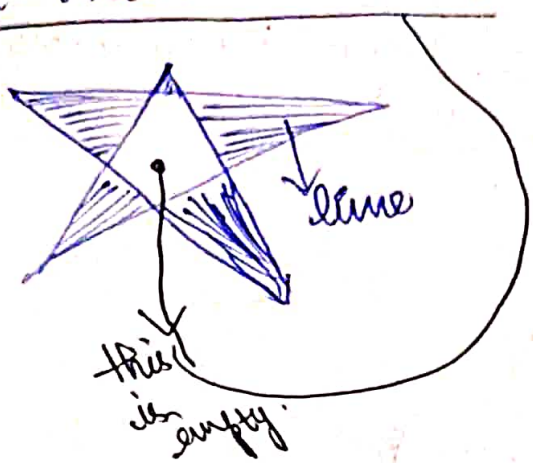
Rounded Rectangle

```
<rect x="50" y="20"
    style=" " >
```

rx = "20" ry="20" width height

SVG star

```
<svg width="300" height="200">
  <polygon points="100,10 40,198 190,
    78 10, 78 160, 198"
  style="fill: lime; stroke: purple;
  stroke-width: 5;
  fill-rule: evenodd;" />
</svg>
```



Canvas vs SVG

Canvas	SVG (→ based on graphics for XML)
→ Resolution dependent.	→ Resolution independent
→ No support for event handlers.	→ supported.
→ poor text rendering capabilities.	→ large rendering areas (Google Maps)
→ .png or .jpg (resulting image)	→ Not suited for game applic's
→ Suited for graphic-intensive games.	

* Multimedia

⇒ Only mp3, WAV and Ogg audio are supported by HTML format

→ Only mp4, WebM, Ogg video are supported

1) <video>

<video width="320" height="240" controls>
<source src="movie.mp4" type="video/mp4">
<source src="movie.ogg" type="video/ogg">
Browser doesn't support this!
</video>

controls → Adds controls like play, pause, volume

anyone will be played, which is available & supported

only shown in unsupported browsers.

⇒ <video width="" height="" autoplay>

when page is loaded, it is played automatically.

⇒ <video width="" height="" autoplay muted>

starts playing but muted

⇒ <track> → for subtitles

2) Audio <audio>

Adds controls

Ex: <audio controls autoplay muted>
<source src=".mp3" type="audio/mpeg">
</audio>