HTML Graphics

HTML Canvas Graphics

The HTML <canvas> element is used to draw graphics on a web page.

The graphic to the left is created with <canvas>. It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.

What is HTML Canvas?

The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript.

The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

Canvas is supported by all major browsers.

Canvas Examples

A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

The markup looks like this:

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

Example

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

Here are some examples:

Draw a Line

Example

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0, 0);
ctx.lineTo(200, 100);
ctx.stroke();
</script>
```

Draw a Circle

Example

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95, 50, 40, 0, 2 * Math.PI);
ctx.stroke();
</script>
```

Draw a Text

Example

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.font = "30px Arial";
ctx.fillText("Hello World", 10, 50);
</script>
```

Stroke Text

Example

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.font = "30px Arial";
ctx.strokeText("Hello World", 10, 50);
</script>
```

Draw Linear Gradient

Example

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");

// Create gradient
var grd = ctx.createLinearGradient(0, 0, 200, 0);
grd.addColorStop(0, "red");
grd.addColorStop(1, "white");

// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(10, 10, 150, 80);
</script>
```

Draw Circular Gradient

Example

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");

// Create gradient
```

```
var grd = ctx.createRadialGradient(75, 50, 5, 90, 60, 100);
grd.addColorStop(0, "red");
grd.addColorStop(1, "white");

// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(10, 10, 150, 80);
</script>
```

HTML SVG Graphics

SVG (Scalable Vector Graphics)

SVG defines vector-based graphics in XML, which can be directly embedded in HTML pages.

SVG graphics are scalable, and do not lose any quality if they are zoomed or resized:

SVG is supported by all major browsers.

What is SVG?

- SVG stands for Scalable Vector Graphics
- SVG is used to define vector-based graphics for the Web
- SVG defines graphics in XML format
- Each element and attribute in SVG files can be animated
- SVG is a W3C recommendation
- SVG integrates with other standards, such as CSS, DOM, XSL and JavaScript

---

The <svg> Element

The HTML <svg> element is a container for SVG graphics.

SVG has several methods for drawing paths, rectangles, circles, polygons, text, and much more.

SVG Circle

Example

```
<!DOCTYPE html>
<html>
<body>

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

</body>
</html>
```

SVG Rectangle

Example

```
<svg width="400" height="120">
  <rect x="10" y="10" width="200" height="100" stroke="red" stroke-
width="6" fill="blue" />
</svg>
```

SVG Rectangle with Opacity and Rounded Corners

Example

```
<svg width="400" height="180">
  <rect x="50" y="20" rx="20" ry="20" width="150" height="150"
```

```
  style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />
</svg>
```

SVG Star

Example

```
<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>
```

SVG Gradient Ellipse and Text

SVG

Example

```
<svg height="130" width="500">
  <defs>
    <linearGradient id="grad1">
      <stop offset="0%" stop-color="yellow" />
      <stop offset="100%" stop-color="red" />
    </linearGradient>
  </defs>
  <ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)" />
  <text fill="#ffffff" font-size="45" font-
family="Verdana" x="50" y="86">SVG</text>
  Sorry, your browser does not support inline SVG.
</svg>
```

Differences Between SVG and Canvas

Feature	SVG	Canvas
Resolution	Independent (sharp zoom)	Dependent (blurs on zoom)
Event Handlers	Supported (easy clicks)	Not supported (need JS code)
Text Rendering	Good (clear, editable)	Poor (pixelated, not editable)
Complex Rendering	Slow	Fast
Save Image	Not direct (XML)	Yes (PNG/JPG)
Games	Not suited	Well suited (graphic-heavy)