Alumna: Analia Noemi Dressler Lang

CANVAS

¿Qué es el Canva?

Es un elemento HTML que se utiliza para crear gráficos en una pagina web, a través de JavaScript

En una página de HTML trabaja en un lienzo de área rectangular, sin bordes ni contenido.

Se debe de ver de esta forma:

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

Etiquetas

Rectángulo:

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

Línea:

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

Circulo:

<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>

Dibujar un texto:

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

Texto de trazo

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

Degradado Lineal:

<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>

Degradado Circular:

<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>

Imagen

<script>  
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
var img = document.getElementById("scream");  
ctx.drawImage(img, 10, 10);  
</script>

SVG

¿Qué es SVG?

* SVG son las siglas de Scalable Vector Graphics
* SVG se utiliza para definir gráficos para la Web
* SVG es una recomendación del W3C

El elemento HTML <svg>

El <svg>elemento HTML es un contenedor para gráficos SVG.

SVG tiene varios métodos para dibujar trazados, cuadros, círculos, texto e imágenes gráficas.

Etiquetas

Circulo SVG:

<!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>

Rectangulo SVG

<svg width="400" height="100">  
  <rect width="400" height="100" style="fill:rgb(0,0,255);stroke-width:10;stroke:rgb(0,0,0)" />  
</svg>

Rectangulo redondeado SVG

<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>

Estrella SVG

<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>

Logotipo de SVG

<svg height="130" width="500">  
  <defs>  
    <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">  
      <stop offset="0%" style="stop-color:rgb(255,255,0);stop-opacity:1" />  
      <stop offset="100%" style="stop-color:rgb(255,0,0);stop-opacity:1" />  
    </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>



Diferencias entre SVG y Canvas

SVG es un lenguaje para describir gráficos 2D en XML.

Canvas dibuja gráficos 2D sobre la marcha (con JavaScript).

SVG está basado en XML, lo que significa que todos los elementos están disponibles dentro del DOM de SVG. Puede adjuntar controladores de eventos de JavaScript para un elemento.

En SVG, cada forma dibujada se recuerda como un objeto. Si se cambian los atributos de un objeto SVG, el navegador puede volver a renderizar automáticamente la forma.

El lienzo se representa píxel a píxel. En el lienzo, una vez que se dibuja el gráfico, el navegador lo olvida. Si se debe cambiar su posición, es necesario volver a dibujar toda la escena, incluidos los objetos que podrían haber sido cubiertos por el gráfico.

<https://www.w3schools.com/html/html5_svg.asp>