

Maquetación y animaciones Web mediante HTML5 y CSS3







HTML5

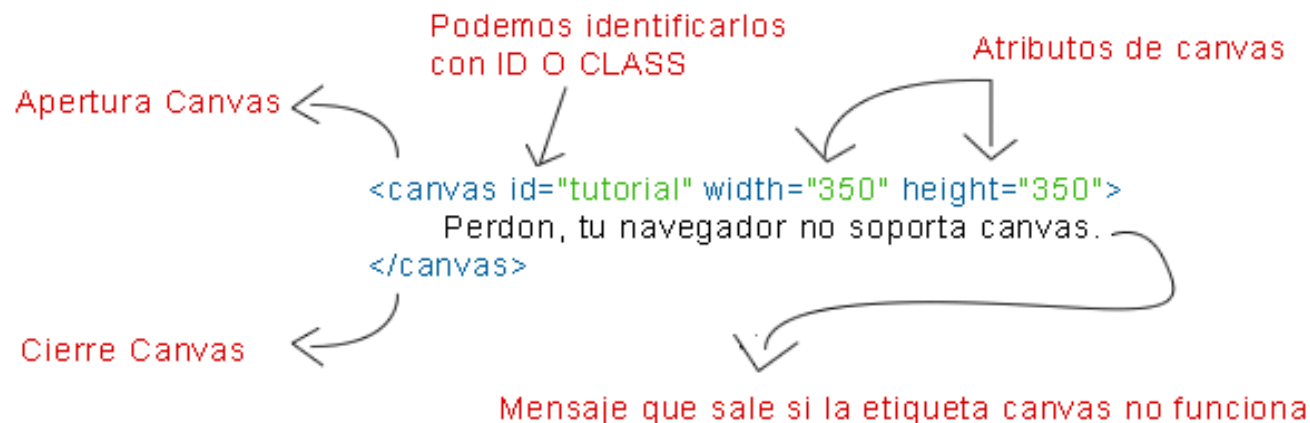
Módulo: Desarrollo de Interfaces Web

CFGS Desarrollo de Aplicaciones Web

Nuevo elemento <canvas> en HTML5

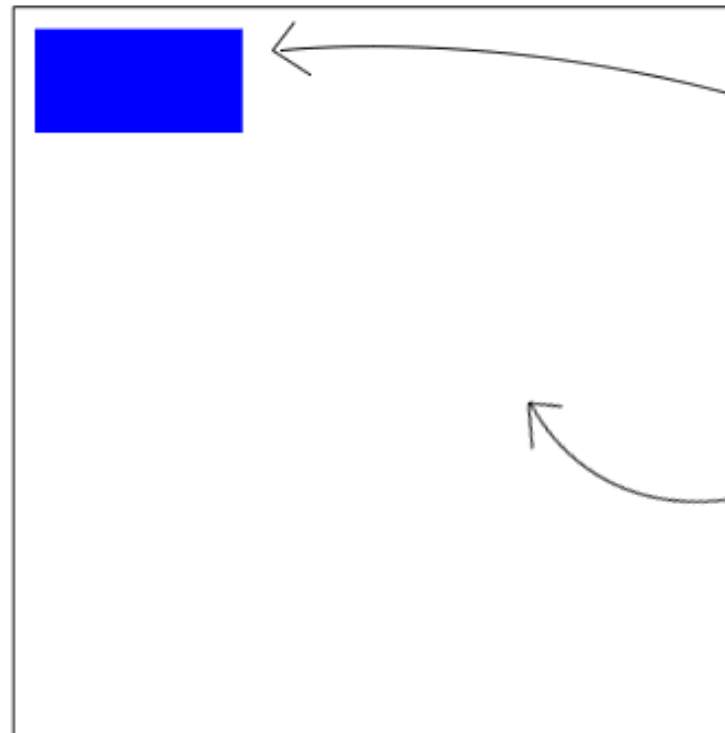
- El nuevo elemento <canvas> nos permite dibujar gráficos en un documento web utilizando JS. Entre las ventajas que podemos encontrar de usar este nuevo elemento es que vamos a dejar de necesitar, y por lo tanto cargar, muchas imágenes ya que podremos dibujarlas con canvas.
- Podremos crear formas, líneas, degradados, textos, arcos, etc.
- El soporte de canvas en los diferentes navegadores sería:

Soporte de navegadores de la etiqueta <canvas>					
 IE9+	 Safari 2.0+	 Chrome 3.0+	 Firefox 3.0+	 Opera 10.0+	 Android 1.0+



Nuevo elemento <canvas> en HTML5

```
<html>
</head>
<body>
<canvas id="tutorial" width="350" height="350" style="border: 1px solid #000000;">
  Perdon, tu navegador no soporta canvas.
</canvas>
<script type="text/javascript">
  var c=document.getElementById("tutorial");
  var cxt=c.getContext("2d");
  cxt.fillStyle="blue";
  cxt.fillRect(10,10,100,50);
</script>
</body>
</html>
```

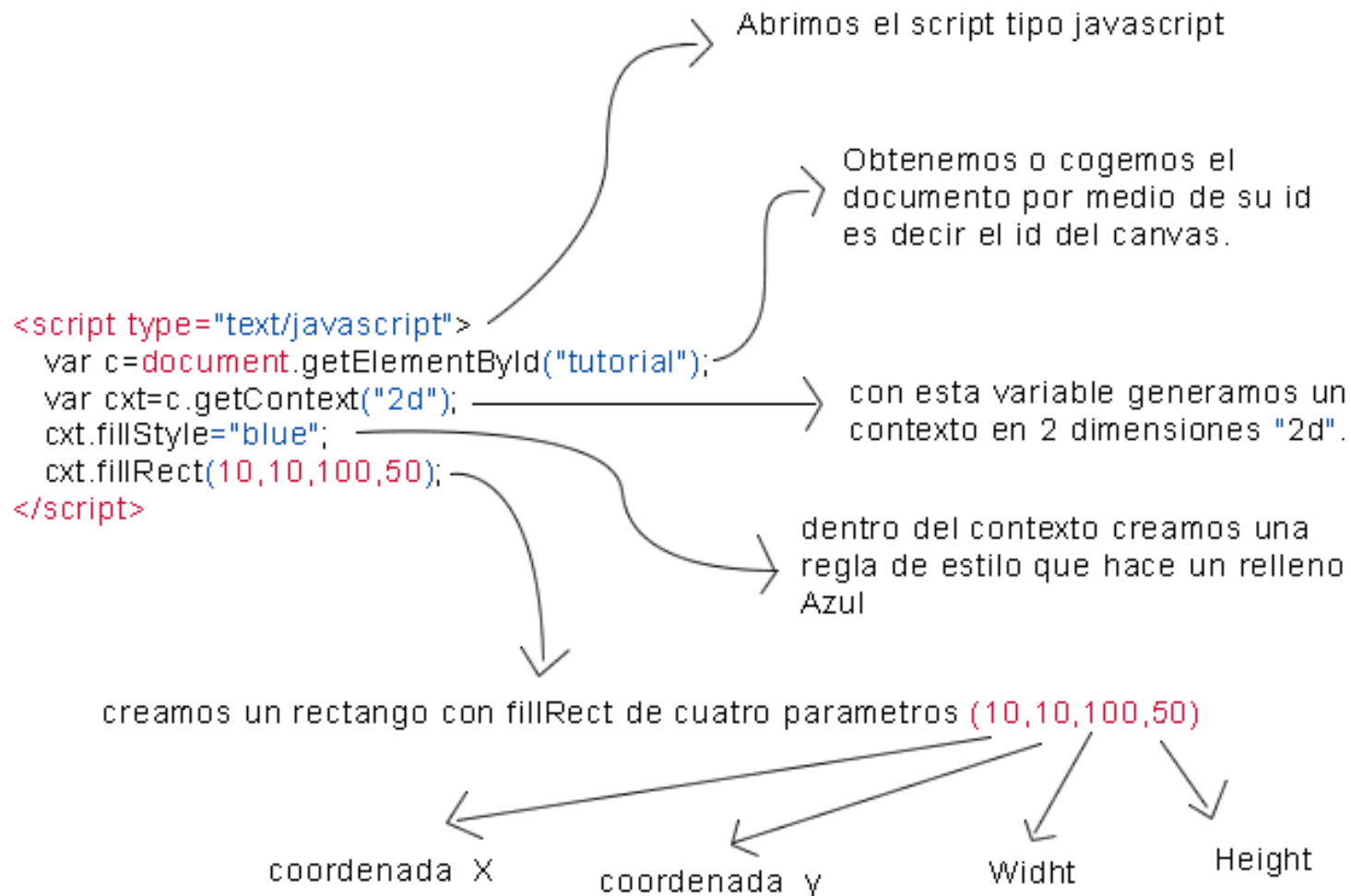


Elemento que creamos con el Script

Lienzo que creamos con canvas, con un alto de 350px y 350px de ancho y un borde de 1px negro solido.

Nuevo elemento <canvas> en HTML5

Explicando el script



Propiedades de <canvas> en HTML5

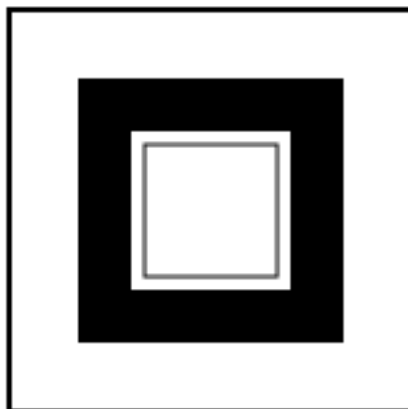
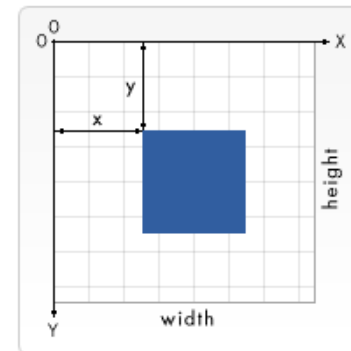
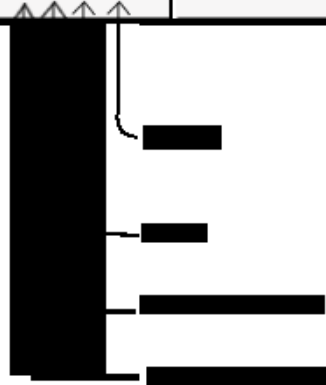
- `fillStyle = color|style` ➔ El color|estilo de relleno del dibujo.
- `strokeStyle = color|style` ➔ El color|estilo del contorno del dibujo.
- `lineCap = butt|round|square` ➔ El estilo del acabado de línea.
- `lineJoin = bevel|round|miter` ➔ El estilo de las esquinas de las líneas.
- `lineWidth = number` ➔ El grosor de las líneas.
- `miterLimit = number` ➔ El límite de esquinas en una línea.
- `shadowColor = color` ➔ El color de la sombra.
- `shadowOffsetX = number` ➔ La distancia horizontal de la sombra.
- `shadowOffsetY = number` ➔ La distancia vertical de la sombra.
- `shadowBlur = number` ➔ El tamaño del efecto de desenfoque.

Métodos de <canvas> en HTML5

- `fillRect(x, y, width, height)` → Dibuja un rectángulo relleno usando el color/estilo del atributo `fillStyle`. Las coordenadas `x` e `y` empiezan en la esquina superior izquierda.
- `strokeRect(x, y, w, h)` → Dibuja las líneas de un rectángulo usando el color/estilo del atributo `strokeStyle`.
- `clearRect(x, y, w, h)` → Limpia un área rectangular.
- `rect(x, y, w, h)` → Crea un rectángulo.
- `moveTo(x, y)` → Mueve el camino al punto especificado, sin crear una línea.
- `lineTo(x, y)` → Crea una línea desde el último punto en el camino hasta el punto dado.
- `arc(x, y, r, sAngle, eAngle, aClockwise)` → Crea un círculo, o parte de un círculo.
- `arcTo(x1, y1, x2, y2, radius)` → Crea un arco entre dos puntos.

Dibujar rectángulos con <canvas>

Nombre	Función	Returns
<code>clearRect(x, y, w, h)</code>	Borra el rectángulo, será transparente	void
<code>fillRect(x, y, w, h)</code>	dibuja un rectángulo relleno	void
<code>strokeRect(x, y, w, h)</code>	dibuja un contorno rectangular	void



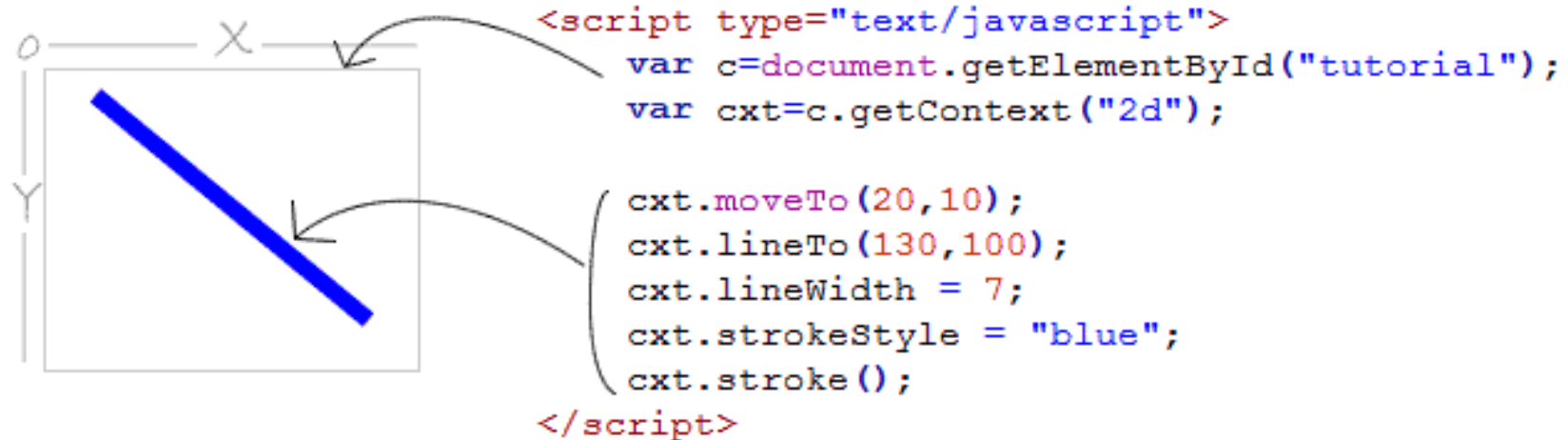
```
<script type="text/javascript">
  var c=document.getElementById('tutorial');
  var ctx=c.getContext('2d');
```

```
  // Formas rectangulares
  ctx.fillRect(25,25,100,100);
  ctx.clearRect(45,45,60,60);
  ctx.strokeRect(50,50,50,50);
```

```
</script>
```

- 1º – se crea un cuadro negro de 100*100px con la función FillRect
- 2º – se elimina un cuadro de 60*60px con la función ClearRect
- 3º – se crea un contorno rectangular de 50*50px con StrokeRect

Trazar una línea con <canvas>



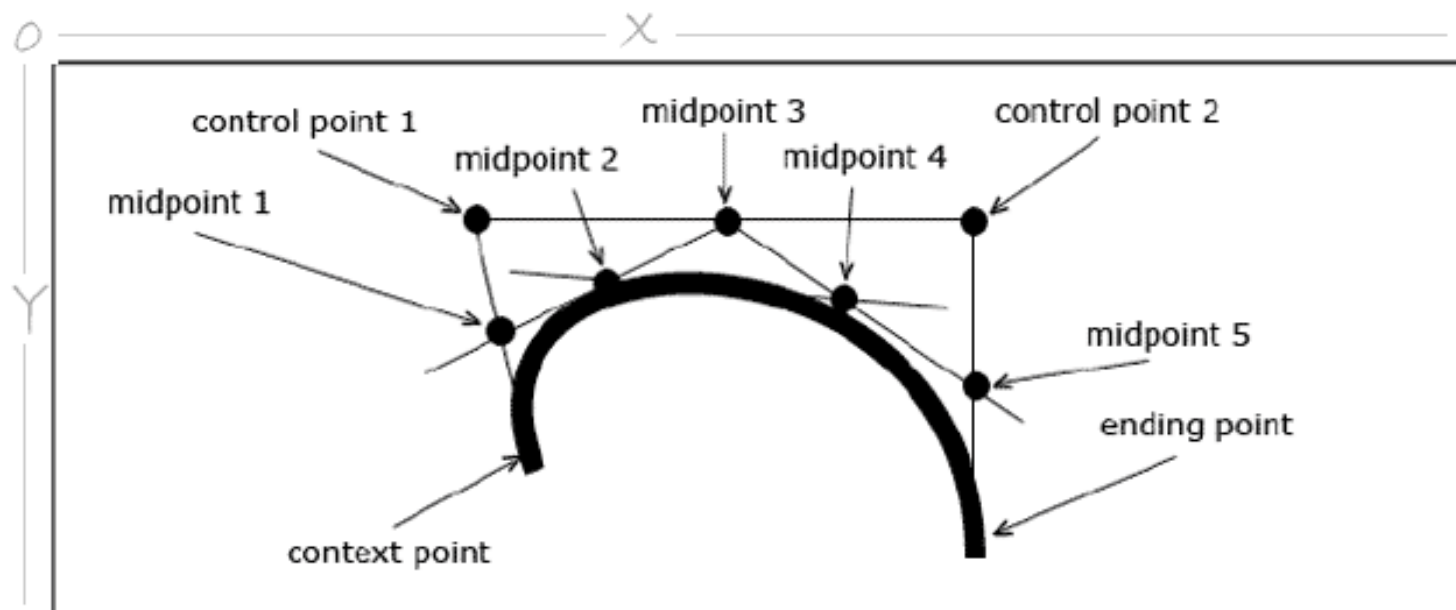
- **cxt.moveTo(20,10);** = Creamos un punto de partida para la línea en el punto (x,y) de acuerdo a las coordenadas partiendo del eje 0.
- **cxt.lineTo(130,100);** = Creamos un punto de llegada para la línea en el punto (x,y) de acuerdo a las coordenadas partiendo del eje 0.
- **cxt.lineWidth = 7;** = Establecemos un ancho para la línea de 7px.
- **cxt.strokeStyle = "blue";** = Asignamos un color a la línea, en este caso será azul.
- **cxt.stroke();** = hacemos que la línea sea visible con el color del trazo

Trazar una curva Béizier con <canvas>

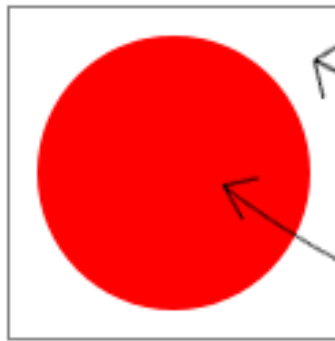
```
<script type="text/javascript">  
  var c=document.getElementById("myCanvas");  
  var cxt=c.getContext("2d");  
  cxt.lineWidth = 10;  
  cxt.strokeStyle = "black";  
  cxt.moveTo(180, 130);  
  cxt.bezierCurveTo(150, 10, 420, 10, 420, 180);  
  cxt.stroke();  
</script>
```

```
cxt.bezierCurveTo(150, 10, 420, 10, 420, 180);
```

```
(controlPointX1, controlPointY1, controlPointX2, controlPointY2, endPointX, endPointY);
```



Dibujar un círculo con <canvas>



```
<script type="text/javascript">
  var c=document.getElementById("tutorial");
  var cxt=c.getContext("2d");

  cxt.fillStyle ="red";
  cxt.beginPath();
  cxt.arc(60,60,50,0,Math.PI*2,true);
  cxt.closePath();
  cxt.fill();
</script>
```

cxt.fillStyle ="red"; = Color del relleno.

cxt.beginPath(); = Sirve para empezar un recorrido, no se le añadió ningún parámetro.

cxt.arc(60,60,50,0,Math.PI*2,true); = Esta función es la más importante ya que nos crea un arco y nos sirve para hacer el círculo, y los parámetros son: 60 en su centro en X, 60 en su centro en Y, 50px radio en desde el centro, que es el punto que creamos con las coordenadas X, Y, luego empezamos un círculo desde 0 hasta el final del arco expresado en radianes por que no soporta grados, (1 vuelta completa de círculo es igual a 2 PI Radianes y por último true para decir que queremos ver la parte positiva del arco es decir se hace el recorrido de acuerdo a como giran las manecillas del reloj.

cxt.closePath(); = Cerramos el recorrido.

cxt.fill(); = Rellenamos el recorrido.

Dibujar gradientes con <canvas>

```
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);
```



```
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);
```



Dibujar textos con <canvas>

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



Hello World

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



Hello World

```
var canvas = document.getElementById("myCanvas");  
var ctx = canvas.getContext("2d");  
ctx.font = "30px Comic Sans MS";  
ctx.fillStyle = "red";  
ctx.textAlign = "center";  
ctx.fillText("Hello World", canvas.width/2,  
canvas.height/2);
```



Hello World

Dibujar imágenes con <canvas>

```
window.onload = function() {  
    var canvas = document.getElementById("myCanvas");  
    var ctx = canvas.getContext("2d");  
    var img = document.getElementById("scream");  
    ctx.drawImage(img, 10, 10);  
};
```



Ejemplo completo de <canvas>



HTML5_Canvas_Ejemplo.html

```

<!DOCTYPE html>
<html>
<head>
<title>HTML5 Logo con Canvas</title>
<meta charset="UTF-8">
<script language="JavaScript">
var ctx;
var factorvalue = 1;
var fontfamily = "65px 'Gill Sans Ultra Bold', sans-serif";
function init() {
  ctx = document.getElementById('canvas').getContext('2d');
  ctx.font = fontfamily;
  dologo();
}
function dologo() {
  var offsety = 80;
  ctx.fillText("HTML", 31,60);
  ctx.translate(0,offsety);
}

// Escudo de fondo fondo naranja
ctx.fillStyle = "#E34C26";
ctx.beginPath();
ctx.moveTo(39, 250);
ctx.lineTo(17, 0);
ctx.lineTo(262, 0);
ctx.lineTo(239, 250);
ctx.lineTo(139, 278);
ctx.closePath();
ctx.fill();

// naranja iluminado de la parte de la derecha
ctx.fillStyle = "#F86529";
ctx.beginPath();
ctx.moveTo(139, 257);
ctx.lineTo(220, 234);
ctx.lineTo(239, 20);
ctx.lineTo(139, 20);
ctx.closePath();
ctx.fill();

// Gris claro del 5 de su parte izquierda
ctx.fillStyle = "#EBE8EB";
ctx.beginPath();
ctx.moveTo(139, 113);
ctx.lineTo(98, 113);
ctx.lineTo(98, 82);
ctx.lineTo(139, 82);
ctx.lineTo(139, 51);
ctx.lineTo(62, 51);
ctx.lineTo(70, 144);
ctx.lineTo(139, 144);
ctx.closePath();
ctx.fill();

ctx.beginPath();
ctx.moveTo(139, 193);
ctx.lineTo(105, 184);
ctx.lineTo(103, 159);
ctx.lineTo(72, 159);
ctx.lineTo(76, 207);
ctx.lineTo(139, 225);
ctx.closePath();
ctx.fill();

// el 5 en lado de la derecha
ctx.fillStyle = "#FFFFFF";
ctx.beginPath();
ctx.moveTo(139, 113);
ctx.lineTo(139, 144);
ctx.lineTo(177, 144);
ctx.lineTo(173, 184);
ctx.lineTo(139, 193);
ctx.lineTo(139, 225);
ctx.lineTo(202, 207);
ctx.lineTo(210, 113);
ctx.closePath();
ctx.fill();

ctx.beginPath();
ctx.moveTo(139, 51);
ctx.lineTo(139, 82);
ctx.lineTo(213, 82);
ctx.lineTo(216, 51);
ctx.closePath();
ctx.fill();
}
</script>
</head>

<body onload="init();">
  <canvas id="canvas" width="600" height="400">
    Tu navegador no soporta canvas
  </canvas>
</body>
</html>

```

Generamos el contexto del canvas, la fuente que vamos a utilizar en el lienzo

Iniciamos la función dologo que es donde dibujaremos el logo, ajustamos las coordenadas, dibujamos las letras HTML.

HTML
Pintamos este escudo que formamos por medio de línea con lineTo y moveTo, y lo rellenos con ese color utilizando fillStyle

HTML
Con este bloque de código generamos el reflejo del escudo de la derecha

HTML
Con este bloque de código generamos la primera parte del 5, que tiene como color #EBE8EB que es un gris claro.

HTML
Generamos la otra parte del 5 en su parte gris claro

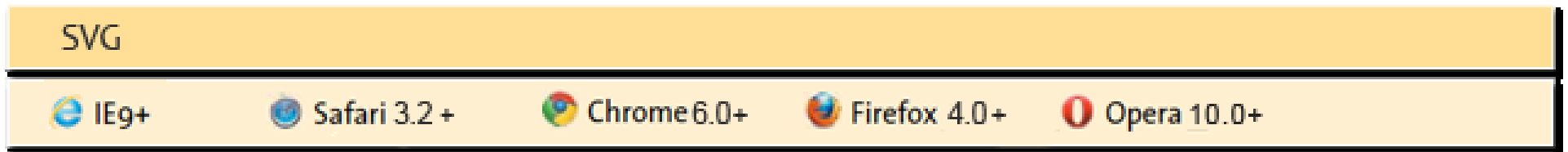
HTML
Generamos la parte del 5 de la derecha, que tiene como relleno #FFFFFF que es un blanco limpio

HTML
Terminamos el 5 con este bloque de código que nos genera el cuadro de la derecha

cuerpo del HTML con una propiedad donde llamamos la función que abrimos al comienzo de todo el script, y por supuesto la etiqueta Canvas con su altura y anchura.

Nuevo elemento <svg> en HTML5

HTML5 nos permite incorporar archivos o imágenes vectoriales directamente mediante SVG (Scalable Vector Graphics). Las imágenes vectoriales siempre conservan su calidad, nitidez y aspecto y se trata de ficheros XML.



Formas:

- Rectángulo <rect>
- Círculo <circle>
- Elipse <ellipse>
- Línea <line>
- Polilínea <polyline>
- Polígono <polygon>
- Ruta <path>

Nuevo elemento <svg> en HTML5

```
1 <html>
2 <head>
3 </head>
4 <body>
5   <svg height=100 width=100>
6     <circle cx=50 cy=50 r=50 fill=red />
7   </svg>
8 </body>
9 </html>
```

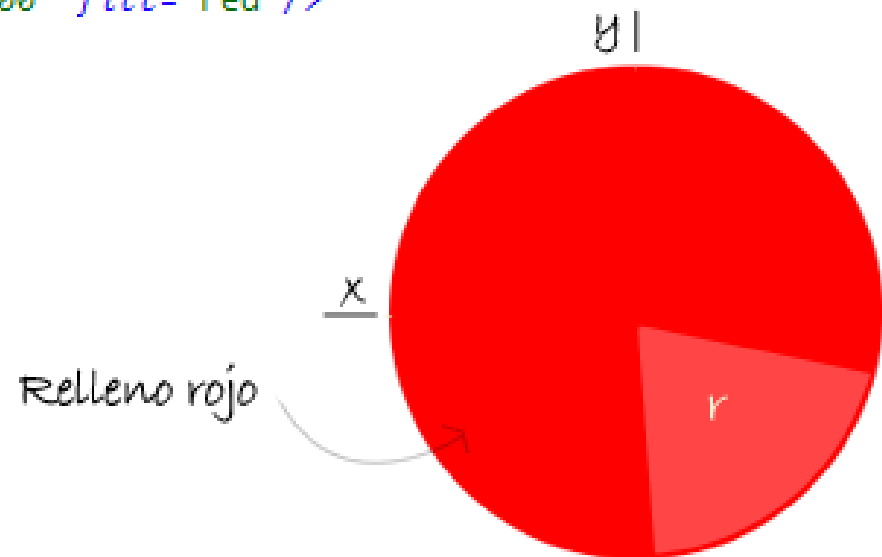
Creamos el elemento <svg> y lo cerramos </svg>, también le damos una altura y anchura, Luego dentro del elemento svg dibujamos un círculo con cx=50, es decir, su centro x es 50 y es igual para el centro y, r=50, es decir, su radio es de 50 y fill o relleno será rojo. Lo que nos devolvería un círculo rojo.

Una forma efectiva y de más posibilidades es definir los gráficos o contenido por medio del formato XML. La declaración sería la siguiente:

```
<svg xmlns="http://www.w3.org/2000/svg" id="circulo" height="200" width="200">
  <circle cx="100" cy="100" r="100" fill="red" />
</svg>
```

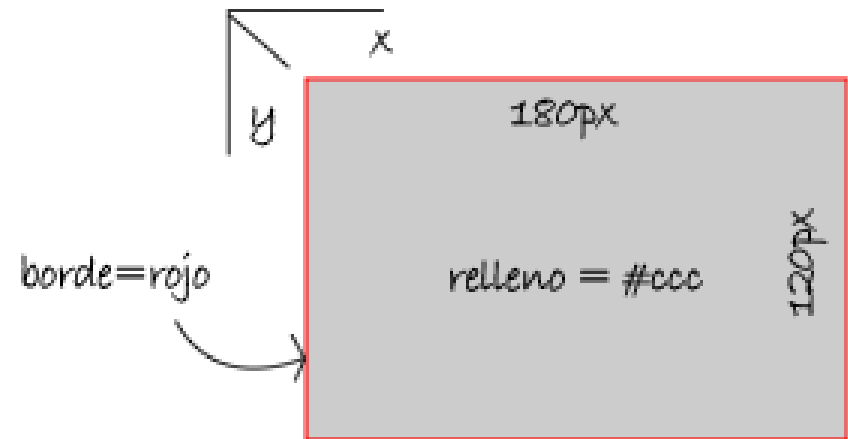

Dibujando un círculo con <svg>

```
1 <html>
2 <head>
3   <title></title>
4 </head>
5 <body>
6   <svg id="circulo" height="200" width="200" xmlns="http://www.w3.org/2000/svg">
7     <circle cx="100" cy="100" r="100" fill="red"/>
8   </svg>
9 </body>
10 </html>
```



Dibujando un cuadrado con <svg>

```
1  <html>
2  <head>
3      <title></title>
4  </head>
5  <body>
6      <svg height="200" width="200" xmlns="http://www.w3.org/2000/svg">
7          <rect x="10" y="20" width="180" height="120" stroke="red" fill="#ccc" />
8      </svg>
9  </body>
10 </html>
```



Añadiendo estilos a <svg>

```
1 <html>
2 <head>
3   <title></title>
4 </head>
5 <body>
6   <svg xmlns="http://www.w3.org/2000/svg" version="1.1">
7     <rect x="50" y="20" rx="20" ry="20" width="150" height="100"
8     style="fill:red; stroke:black; stroke-width:5; opacity:0.5" />
9   </svg>
10 </body>
11 </html>
```

Con *rx*, *ry* Damos el radio o curvatura de los
angulos del rectangulo

style =
fill= relleno rojo
stroke=borde negro
stroke-width=grosor del borde
opacity= opacidad del elemento



Dibujando elipses con <svg>

```
<svg height="150" width="500">  
  <ellipse cx="240" cy="100" rx="220" ry="30" style="fill:purple" />  
  <ellipse cx="220" cy="70" rx="190" ry="20" style="fill:lime" />  
  <ellipse cx="210" cy="45" rx="170" ry="15" style="fill:yellow" />  
</svg>
```



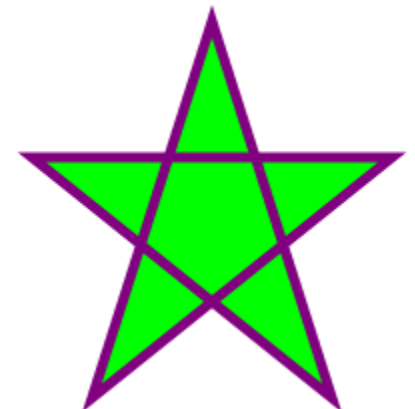
Dibujando líneas con <svg>

```
<svg height="210" width="500">  
  <line x1="0" y1="0" x2="200" y2="200" style="stroke:rgb(255,0,0);stroke-width:2" />  
</svg>
```



Dibujando polígonos con <svg>

```
<svg height="210" width="500">  
  <polygon points="100,10 40,198 190,78 10,78 160,198"  
    style="fill:lime;stroke:purple;stroke-width:5;fill-rule:nonzero;" />  
</svg>
```



Textos con <svg>

```
<svg height="60" width="200">
  <text x="0" y="15" fill="red" transform="rotate(30 20,40)">I love SVG</text>
</svg>
```

I love SVG

Gradientes con <svg>

```
<svg height="150" width="400">
  <defs>
    <linearGradient id="grad3" 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="200" cy="70" rx="85" ry="55" fill="url(#grad3)" />
  <text fill="ffffff" font-size="45" font-family="Verdana" x="150" y="86">
    SVG</text>
</svg>
```



- El atributo id de <linearGradient> define de manera unívoca el nombre del gradiente
- Los atributos x1, x2, y1,y2 definen donde comienza y termina el gradiente
- El rango de colores puede componerse por uno o más colores. Cada color está especificado con una etiqueta <stop> y el atributo offset define donde empieza y termina el gradiente del color.
- El atributo fill del tag <ellipse> enlaza la elipse con el gradiente.

Mirar...

- Canvas: <http://www.w3schools.com/canvas/default.asp>
- Crear un reloj con canvas: http://www.w3schools.com/canvas/canvas_clock.asp
- SVG: <http://www.w3schools.com/svg/default.asp>
- Ejemplos de SVG: http://www.w3schools.com/svg/svg_examples.asp
- En MSDN: [Cómo elegir entre SVG y Canvas](#)