



Elemento Canvas en HTML5

-
- C.F.G.S. DAW
 - 6 horas semanales
 - Mes aprox. de impartición: Ene - Feb
 - iPasen - cjaedia071@g.educaand.es

Carmelo José Jaén Díaz

Índice



Objetivo

Glosario

Interacción persona - ordenador

Objetivos

Características. Usable.

Características. Visual.

Características. Educativo y actualizado.

OBJETIVO



- Identificar los formatos de audio y vídeo que se van a utilizar.
- Realizar animaciones a partir de imágenes fijas.
- Aplicar la guía de estilo sobre los elementos multimedia de audio, vídeo y animaciones.
- Reconocer y analizar las tecnologías relacionadas con la inclusión de contenido multimedia e interactivo.
- Agregar elementos multimedia a los documentos web.
- Añadir interactividad a los elementos de un documento web.
- Verificar el funcionamiento de los elementos multimedia en diferentes navegadores.

GLOSARIO



Formato. De audio o de vídeo, tipo de fichero o archivo contenedor de alguno de dichos elementos. En función de este, el audio o vídeo será indicado para un determinado fin; incluso determina en qué navegador podrá ser reproducido.

Fotograma. Cada una de las imágenes que componen un vídeo, en concreto, el vídeo final se produce por la secuencia de fotogramas visualizada de acuerdo con una determinada frecuencia de imágenes por segundo.

Frecuencia. Número de vibraciones por segundo que dan origen al sonido analógico. El espectro de un sonido se caracteriza por su rango de frecuencias, medido en hercios (Hz).

Relación de aspecto. Relación que existe entre el ancho y el alto de un fotograma, y, por tanto, del vídeo, puesto que normalmente se mantiene el mismo tipo de formato a lo largo de toda la grabación. Las más conocidas son 4:3 y 16:9.

Sonido. Sensación producida en el órgano del oído por el movimiento vibratorio de los cuerpos, transmitido por un medio elástico como el aire.

GLOSARIO



Velocidad de transmisión (*bit rate*). Cantidad de espacio físico (en bits) que ocupa un segundo de duración de un audio. El audio tendrá más calidad cuanto mayor sea su *bit rate*, y el archivo que lo contiene tendrá mayor peso.

INTRODUCCIÓN



En unidades anteriores, se estudió cómo hacer transformaciones y transiciones sobre los elementos de nuestra página web. De esta forma conseguíamos proporcionar a los elementos un efecto de animación sobre algunas propiedades básicas de CSS. Además, también hemos estudiado cómo utilizar la propiedad animation de CSS para crear animaciones concatenadas.



INTRODUCCIÓN



En este apartado vamos a seguir viendo cómo crear animaciones, pero en esta ocasión estudiaremos el elemento `<canvas>`. Este elemento nos permite pintar gráficos, crear animaciones y desarrollar videojuegos mediante una API (*Application Programming Interface*, Interfaz de Programación de Aplicaciones) de JavaScript.

ELEMENTO CANVAS EN HTML5

Pasos para crear un Canvas



Lo primero que tenemos que hacer para trabajar con Canvas es crear un lienzo y, a partir de ese lienzo, incluir elementos y modificarlos. Veamos los pasos a seguir.

1) Crearemos un lienzo (espacio rectangular vacío) en el que serán mostrados todos los elementos:

```
<canvas id="lienzo" width="500" height="300">
```

El navegador no soporta canvas

```
</canvas>
```


ELEMENTO CANVAS EN HTML5

Pasos para crear un Canvas



2) Accedemos al elemento `<canvas>` desde Javascript:

Primero, accedemos al objeto DOM del elemento, normalmente, a través del método `getElementById()`, y posteriormente, utilizamos el método `getContext()` para obtener el contexto de dibujo, sobre el cual, se puede dibujar.

```
function iniciar()  
{  
    let canvas=document.getElementById( 'lienzo' );  
    let ctx=canvas.getContext( '2d' );  
}  
window.addEventListener( "load", iniciar, false );
```

En este caso, guardamos una referencia al elemento `<canvas>` en la variable `canvas` y el contexto de dibujo lo creamos utilizando `getContext('2d')`. Con esto obtenemos un contexto para dibujar en 2 dimensiones.

ELEMENTO CANVAS EN HTML5

Pasos para crear un Canvas



3) Dibujar formas en el lienzo:

A continuación, podemos empezar a crear formas y realizar manipulaciones según las coordenadas concretas del contexto. Veamos las formas básicas que podemos crear en nuestro lienzo.

FORMAS BÁSICAS EN CANVAS

Rectángulos



La función `fillRect(x, y, width, height)` nos permite dibujar un rectángulo 'relleno' y la función `strokeRect(x, y, width, height)` un rectángulo con sólo su contorno. Mediante `lineWidth` especificamos el ancho de su contorno.

Se puede establecer el color mediante las propiedades `fillStyle` y `strokeStyle`. Es posible usar rgb, rgba, un código hexadecimal o la palabra clave de un color.

Por ejemplo:

```
ctx.fillStyle = "rgba(0, 0, 0, 0.5)"; //Color
ctx.fillRect(100, 10, 60, 50); //Rectángulo relleno
ctx.lineWidth = 5 //ancho de contorno
ctx.strokeStyle = "red"; //Color
ctx.strokeRect(200, 70, 60, 50); //Rectángulo de solo contorno
```

FORMAS BÁSICAS EN CANVAS

Ejemplo - Rectángulos



Código HTML

```
<canvas id="miCanvas" width="550"
height="300">
    Tu navegador no soporta
    canvas.
</canvas>
```

Código JS

```
function iniciar(){
    let miCanvas =
    document.getElementById("miCanvas");
    let ctx = miCanvas.getContext("2d");
    //Color
    ctx.fillStyle = "rgba(0, 0, 0, 0.5)";
    //Rectángulo relleno
    ctx.fillRect(100, 10, 60, 50);
    ctx.lineWidth = 5 //ancho de contorno
    ctx.strokeStyle = "red"; //Color
    //Rectángulo de solo contorno
    ctx.strokeRect(200, 70, 60, 50);
}
window.addEventListener("load", iniciar,
false);
```

FORMAS BÁSICAS EN CANVAS

Ejemplo - Rectángulos



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/mybGaRJ>



FORMAS BÁSICAS EN CANVAS

Rectángulos



Mediante la función `clearRect(x, y, width, height)` podemos borrar cualquier elemento que haya sido dibujado en el Canvas dentro de un área.

Por ejemplo:

```
ctx.clearRect(210, 20, 25, 25);
```

FORMAS BÁSICAS EN CANVAS

Líneas



Para indicar que queremos dibujar una línea lo primero que debemos hacer es llamar a la función `beginPath()`.

A continuación llamaremos a `moveTo(x, y)` para movernos a la posición donde nuestra línea comenzará. Y a `lineTo(x, y)` para *dibujar* la línea. Para hacerla visible utilizaremos la función `stroke()`.

Podemos especificar el ancho y el color de la línea con las propiedades `lineWidth` y `strokeStyle`.

FORMAS BÁSICAS EN CANVAS

Líneas



Por ejemplo:

```
ctx.beginPath();  
ctx.moveTo(5, 5);  
ctx.lineTo(5, 295);  
ctx.moveTo(295, 5);  
ctx.lineTo(295, 295);  
ctx.lineWidth = 1;  
ctx.strokeStyle = "#CC0000";  
ctx.stroke();
```


FORMAS BÁSICAS EN CANVAS

Ejemplo - Líneas



Código HTML

```
<canvas id="miCanvas" width="550"
height="300">
    Tu navegador no soporta
    canvas.
</canvas>
```

Código JS

```
function iniciar() {
    let miCanvas =
document.getElementById("miCanvas");
    let ctx = miCanvas.getContext("2d");
    ctx.beginPath();
    ctx.moveTo(5, 5);
    ctx.lineTo(5, 295);
    ctx.moveTo(295, 5);
    ctx.lineTo(295, 295);
    ctx.lineWidth = 1;
    ctx.strokeStyle = "#CC0000";
    ctx.stroke();
}
window.addEventListener("load", iniciar,
false);
```

FORMAS BÁSICAS EN CANVAS

Ejemplo - Líneas



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/MYqqZOX>



FORMAS BÁSICAS EN CANVAS

Arcos



Para indicar que vamos a dibujar un arco llamaremos inicialmente a la función `beginPath()`. Dibujaremos un arco o un círculo con la función `arc(x, y, radio, angIni, angFin, sentidoAntihorario)`. Donde `x` e `y` indican el centro del círculo, y los ángulos han de indicarse en radianes. El arco no será visible hasta llamar a `fill()` o a `stroke()`.

Por ejemplo:

```
ctx.beginPath();  
    // let radianes = (Math.PI / 180) * grados;  
    ctx.arc(150, 150, 50, 0, 2*Math.PI); // 2*Math.PI es la manera corta de  
expresar 360 grados en radianes  
ctx.lineWidth = 10;  
ctx.strokeStyle = "yellow";  
ctx.stroke();
```

FORMAS BÁSICAS EN CANVAS

Ejemplo - Arcos



Código HTML

```
<canvas id="miCanvas" width="550"
height="300">
    Tu navegador no soporta
    canvas.
</canvas>
```

Código JS

```
function iniciar() {
    let miCanvas =
document.getElementById("miCanvas");
    let ctx = miCanvas.getContext("2d");
    ctx.beginPath();
    ctx.arc(150, 150, 50, 0, 2 *
Math.PI);
    ctx.lineWidth = 10;
    ctx.strokeStyle = "brown";
    ctx.stroke();
}
window.addEventListener("load",
iniciar, false);
```

FORMAS BÁSICAS EN CANVAS

Ejemplo - Arcos



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/xbKamWN>



FORMAS BÁSICAS EN CANVAS

Texto



A través de la función `fillText(texto, x, y)` podemos dibujar texto dentro del Canvas. Podemos dar formato al texto estableciendo previamente las propiedades `font`, `textAlign` y `fillStyle`.

Por ejemplo:

```
ctx.font = 'Bold 20px Tahoma';  
ctx.textAlign = 'center';  
ctx.fillStyle = 'black';  
ctx.fillText('ENIUN', 50, 50);
```

FORMAS BÁSICAS EN CANVAS

Ejemplo - Texto



Código HTML

```
<canvas id="miCanvas" width="550"
height="300">
    Tu navegador no soporta
    canvas.
</canvas>
```

Código JS

```
function iniciar(){
    let miCanvas =
document.getElementById("miCanvas");
    let ctx = miCanvas.getContext("2d");
ctx.font = 'Bold 20px Tahoma';
ctx.textAlign = 'center';
ctx.fillStyle = 'black';
ctx.fillText('DIW', 50, 50);
}
window.addEventListener("load",
iniciar, false);
```

FORMAS BÁSICAS EN CANVAS

Ejemplo - Texto



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/VYZGqGv>

DIW

FORMAS BÁSICAS EN CANVAS

Imágenes



Para añadir una imagen lo primero que tenemos que hacer es crear un objeto `Image` y establecer su propiedad `src`.

Con la función `drawImage(objImg, x, y)` añadiremos la imagen al Canvas.

Por ejemplo:

```
let img = new Image();
img.src =
'https://upload.wikimedia.org/wikipedia/commons/f/f2/Emblema_de_la_Junta_de
_Andaluc%C3%AD_a_2020.svg';
img.onload = function(){
    ctx.drawImage(img, 10, 10);
};
```

FORMAS BÁSICAS EN CANVAS

Ejemplo - Imágenes



Código HTML

```
<canvas id="miCanvas" width="550"  
height="300">
```

Tu navegador no soporta

canvas.

```
</canvas>
```

Código JS

```
function iniciar(){  
    let miCanvas =  
document.getElementById("miCanvas");  
    let ctx = miCanvas.getContext("2d");  
let img = new Image();  
img.src =  
'https://upload.wikimedia.org/wikipedia/c  
ommons/f/f2/Emblema_de_la_Junta_de_Andalu  
c%C3%ADa_2020.svg';  
img.onload = function(){  
    ctx.drawImage(img, 10, 10);  
};  
}  
window.addEventListener("load", iniciar,  
false);
```

FORMAS BÁSICAS EN CANVAS

Imágenes



Si ejecutamos el método `drawImage()` sin llamar al evento `onload` puede que no se ejecute correctamente si todavía no se ha cargado la imagen. Por este motivo se utiliza el evento `onload`.

Adicionalmente, hay que tener en cuenta las restricciones de CORS (*Cross-Origin Resource Sharing*). Estas suceden si el servidor que aloja la imagen no permite que sea utilizada en contextos de terceros como un `<canvas>`.

Puedes encontrar más información en:

- [HTML5 Canvas reference](#)
- [HTML5 Canvas Cheat Sheet](#)

FORMAS BÁSICAS EN CANVAS

Ejemplo - Texto



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/xb>

Kammv



FORMAS BÁSICAS EN CANVAS

Ejemplo con todas las formas



Código HTML

```
<canvas id="miCanvas" width="550"
height="300">
    Tu navegador no soporta
    canvas.
</canvas>
```

Código JS

```
function iniciar(){
    let miCanvas =
document.getElementById("miCanvas");
    let context =
miCanvas.getContext("2d");
    //Cuadrados
    context.fillStyle = "rgb(186,85,211)";
    context.fillRect(5, 5, 100, 100);
    context.fillStyle = "rgb(0, 162,
232)";
    context.fillRect(40, 55, 100, 100);
    context.fillStyle = "rgba(255, 0, 0,
0.2)";
    context.fillRect(75, 105, 100, 100);
```

FORMAS BÁSICAS EN CANVAS

Ejemplo con todas las formas



Código JS

```
//Círculo
context.beginPath();
context.fillStyle = "rgba(85,107,47,
0.5)";
context.arc(200, 40, 40, 0, 2 *
Math.PI, true);
context.fill();
context.beginPath();
context.fillStyle = "rgba(0,139,139,
0.5)";
context.arc(200, 70, 40, 0, 2 *
Math.PI, true);
context.fill();
```

```
//Contorno Círculo
context.beginPath();
context.strokeStyle = "rgba(255, 0, 0,
0.5)";
context.arc(200, 90, 40, 0, 2 *
Math.PI, true);
context.lineWidth = 3;
context.stroke();
//Línea
context.beginPath();
context.moveTo(550, 0);
context.lineTo(0, 550);
context.lineWidth = 2;
context.strokeStyle = "#008080";
context.stroke();
```

FORMAS BÁSICAS EN CANVAS

Ejemplo con todas las formas



Código JS

```
//Imagen
let img = new Image();
img.src =
'https://upload.wikimedia.org/wikipedia/c
ommons/f/f2/Emblema_de_la_Junta_de_Andalu
c%C3%ADa_2020.svg';
img.onload = function(){
context.drawImage(img, 110, 10);
context.drawImage(img, 150, 40);
context.drawImage(img, 180, 100);
context.drawImage(img, 300, 10);
context.drawImage(img, 350, 80);
context.drawImage(img, 210, 300);
context.drawImage(img, 210, 10);    };
```

```
//Arcos
context.beginPath();

context.arc(450,150,50,Math.PI,Math.PI*0.
5,false);

context.arc(450,150,20,Math.PI*0.5,Math.P
I*1,false);
context.closePath()
context.fill();
context.beginPath();
```

FORMAS BÁSICAS EN CANVAS

Ejemplo con todas las formas



Código JS

```
//Rectángulo
context.fillStyle = 'rgba(0, 0, 0,
0.5)';
context.fillRect(300, 10, 60, 50);
context.lineWidth = 5;
context.strokeStyle = 'blue';
context.strokeRect(400, 70, 60, 50);
```

```
//Gradiente lineal
let grad =
context.createLinearGradient(0,0,300,0);
grad.addColorStop(0.2, 'red');
grad.addColorStop(0.8, 'blue');
grad.addColorStop(1, 'pink');
context.fillStyle = grad;
context.fillRect(250, 200, 180, 50);
//Texto
context.font = 'Bold 28px Tahoma';
context.textAlign = 'center';
context.fillStyle = 'grey';
context.fillText('Canvas', 360, 235);
```


FORMAS BÁSICAS EN CANVAS

Ejemplo con todas las formas



Código JS

```
//Sombras y arcos  
context.beginPath();  
context.shadowBlur=50;  
context.shadowColor="blue";  
context.fillStyle="black";  
context.arc(180,150,100,0,0.5*Math.PI)  
;  
context.fill();  
}  
window.addEventListener("load", iniciar,  
false);
```

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/xbKammv>



[Ver más ejemplos de creación de formas](#)