

Canvas

Módulo 01

Implementación

¿Qué rol cumple?

Canvas

El elemento canvas es un elemento **de HTML pero cobra sentido si se lo trabaja desde javascript.**

Esto significa que no tiene ninguna funcionalidad sólo **implementado de la siguiente forma,**

```
<canvas id="canvas"></canvas>
```



¿Qué rol cumple?

Canvas

Para poder comenzar a **trabajar con canvas**, es importante generar el siguiente método para **comenzar a trabajar**,

```
var canvas = document.getElementById("canvas");  
var ctx = canvas.getContext("2d");
```

De esta forma podemos empezar a dibujar. La realidad que a diferencia de **SVG**, el trabajo con **CANVAS** es imposible sin Javascript.

Por otro lado su uso en **videojuegos generados en HTML** forma parte de su principal uso.



¿Qué rol cumple?

Canvas

Como [vimos anteriormente](#) comenzamos a trabajar con **CANVAS** pero las opciones su múltiples.

Por ejemplo si simplemente queremos dibujar una forma,

```
ctx.fillStyle = "green";  
ctx.fillRect(0,0,200,100);
```



¿Qué rol cumple?

Canvas

En el [slide anterior](#), podemos entender que los dos primeros valores fijan la posición dentro del elemento situado en **HTML** llamado canvas.

Los últimos valores se refieren al ancho y alto del propio **dibujo que estamos realizando**.

Vamos a obtener el siguiente resultado,



¿Qué rol cumple?

Canvas

Sin embargo de forma más interesante, **podemos trazar líneas,**

```
ctx.moveTo(20,20);  
ctx.lineTo(150,100);  
ctx.stroke();
```

El primer método `moveTo` nos permite decir desde donde en cuanto al eje **X primero e Y después comienza esa línea.**

Luego `lineTo`, **implica en cuanto al eje X e Y el final de la misma.**



¿Qué rol cumple?

Canvas

En el [slide anterior](#), genera el siguiente resultado,



¿Qué rol cumple?

Canvas

Para generar un círculo debemos trabajar con el siguiente código o similar,

```
ctx.beginPath();  
ctx.arc(100,10,80,0,2*Math.PI);  
ctx.stroke();
```

Los dos primeros valores implican el inicio del elemento dentro de la superficie del **canvas tanto en eje X e Y luego.**

El **número 80** en este caso representa el **radio del círculo.**

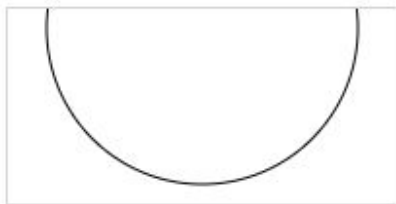
Si copiamos tal cual el resto del código generamos un círculo. **Si alteramos el 0 por otro número, se generará un arco no un círculo.**



¿Qué rol cumple?

Canvas

En el [slide anterior](#), genera el siguiente resultado, ya que hemos corrido el inicio del **círculo y por tanto no entra en su totalidad en la superficie del canvas.**



¿Qué rol cumple?

Canvas

Luego en cualquier superficie, **sea cuadrado, rectángulo o elipse o círculo podemos generar un fondo en gradiente.**

Esto podemos observar en el ejemplo debajo,

```
var gradiente = ctx.createLinearGradient(0, 0, 100, 0);  
gradiente.addColorStop(0, "green");  
gradiente.addColorStop(1, "yellow");  
ctx.fillStyle = gradiente;  
ctx.fillRect(0,0,20,80);
```



¿Qué rol cumple?

Canvas

Podemos **utilizar imágenes**, de la siguiente forma,

```
var imagen = document.getElementById("cuadrado");  
ctx.drawImage(imagen, 110, 110);
```

De esta manera luego podemos utilizarlo para el trabajo con Javascript e incluirla dentro de **un juego por ejemplo que necesitemos realizar utilizando imágenes en él.**



¿Qué rol cumple?

Canvas

Lo interesante del canvas, es qué podemos luego intervenir con cualquier elemento de Javascript ya sea funciones, arreglos, operadores, etc.

Todos sus métodos, elementos y así también características son plausibles de ser trabajadas con Javascript por esa razón constituye una base importante para su uso posterior en juegos como por ejemplo podemos observar [en esta página.](#)

Lo interesante de esta forma de trabajo es el bajo peso de los elementos así como la facilidad de manipularlos con **Javascript**.

En internet podemos encontrar muchísimos juegos **utilizando canvas**.



Revisión

- Repasar los conceptos vistos de **CANVAS**.
- Trabajar con el **bonus track** debajo de todo para practicar los **elementos individualmente**.
- Ver todos los videos y materiales necesarios antes de continuar



¡Muchas gracias!

¡Sigamos trabajando!