

JS-Canvas

- **Canvas:**
 - Elemento HTML que nos permite dibujar gráficos

```
<canvas id="myCanvas" width="500" height="500">fallback content</canvas>
```

- Por defecto es un rectángulo blanco
- fallback content: Contenido recomendable añadir para que se muestre en navegadores antiguos que no soporten la etiqueta (pej. IE < 9)
- Necesita la etiqueta de cierre
- Para trabajar con canvas con JS necesitamos el contexto 2d:

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

- **Plantilla:**

```
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="utf-8">  
<title>Canvas</title>  
</head>  
<body>  
<canvas width="500" height="500" id="myCanvas"></canvas>  
<script type="text/javascript">  
let canvas = document.getElementById("myCanvas");  
let ctx= canvas.getContext("2d");  
</script>  
</body>  
</html>
```

- **Formas:**
 - Dibujar una línea:

```
ctx.moveTo(0, 0); //Arriba a la izquierda  
ctx.lineTo(200, 100); //Coordenadas en px
```

```
ctx.stroke();
```

- Dibujar una figura:

```
ctx.moveTo(100, 200);
```

```
ctx.lineTo(200, 200);
```

```
ctx.lineTo(150, 100);
```

```
ctx.lineTo(100, 200);
```

```
ctx.stroke();
```

```
//ctx.fill(); //rellena
```

- Dibujar un rectángulo:

```
//strokeRect(x, y, width, height)
```

```
ctx.strokeRect(50, 50, 100, 50);
```

- Dibujar un rectángulo lleno:

```
//fillRect(x, y, width, height)
```

```
ctx.fillRect(50, 50, 100, 50);
```

- Borrar un rectángulo:

```
//clearRect(x, y, width, height)
```

```
ctx.fillRect(50, 50, 100, 50);
```

```
ctx.clearRect(60, 60, 80, 30);
```

- Dibujar un círculo:

```
//arc(x, y, radius, startAngle, endAngle, counterclockwise)
```

```
ctx.beginPath();
```

```
ctx.arc(95, 50, 40, 0, 2 * Math.PI);
```

```
ctx.stroke();
```

- Dibujar un círculo lleno:

```
ctx.beginPath();
```

```
ctx.arc(95, 50, 40, 0, 2 * Math.PI);
```

```
ctx.fill();
```

- **Texto:**

- Añadir texto:

```
ctx.font = "30px Arial";
```

```
ctx.fillText("Hello World", 10, 50);
```

- Añadir StrokeText:



```
ctx.font = '48px serif';
```

```
ctx.strokeText('Hello world', 10, 50);
```

- Definir color:

```
//fillStyle = color
```

```
//strokeStyle = color
```

```
ctx.fillStyle = 'orange';
```

```
ctx.strokeStyle = 'blue';
```

```
ctx.beginPath();
```

```
ctx.arc(95, 50, 40, 0, 2 * Math.PI);
```

```
ctx.stroke();
```

```
ctx.fill();
```

- Imágenes

```
//drawImage(image, x, y)
```

```
//drawImage(image, x, y, width, height)
```

```
let img = new Image();
```

```
img.onload = function() {
```

```
  ctx.drawImage(img, 0, 0);
```

```
};
```

```
img.src = 'image.png';
```

- Más:
 - funcionalidades: josh.org/books/canvasdeepdive/toc.html
- **Animaciones:**
 - Para dibujar un frame necesitamos:
 - Limpiar el canvas
 - Dibujar los elementos que queramos
 - Para controlar las animaciones:
 - Para invocar una función cada delay milisegundos:
 - [setInterval\(\) global function - Web APIs | MDN \(mozilla.org\)](https://developer.mozilla.org/en-US/docs/Web/API/setInterval).

setInterval(function[, delay]);

- Invoca una función una sola vez dentro de delay milisegundos
- [setTimeout\(\)_global function - Web APIs | MDN \(mozilla.org\)](#).

setTimeout(function[, delay]);

- Para controlar las animaciones:
 - **requestAnimationFrame(callback);**
 - Ejecuta una animación antes del siguiente repaint del navegador
 - Se ejecuta una sola vez
 - La tasa de refresco es unas 60 veces por segundo
 - La tasa de refresco puede variar según el display
 - La tasa de refresco puede ser más lenta si el navegador está sobrecargado

```
let canvas = document.getElementById("myCanvas");
let ctx = canvas.getContext("2d");
let x = 0, y = 0, speed = 4;
const width = 20, height = 20;
window.addEventListener("load", (event) => {
  init();
});
function init(){
  window.requestAnimationFrame(draw);
}
function draw(timestamp){
  console.log(timestamp);
  if(x < 0 || x > canvas.width - width){
    speed *= -1;
  }
  x += speed;
  ctx.clearRect(0, 0, canvas.width, canvas.height); //Limpiar el canvas
  ctx.fillRect(x, y, width, height);
  window.requestAnimationFrame(draw);
}
```

- Ejemplos:
 - Sistema solar:

- https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Basic_animations#an_animated_solar_system
- Reloj:
 - https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Basic_animations#an_animated_clock
- Snake:
 - https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Basic_animations#snake_game
- Más información:
 - https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API