



**Materia:** Web Security.  
**Tema:** Introducción a **canvas**.

## PARTE II.

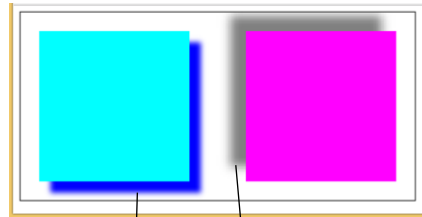
### 1. Sombras

- El atributo `shadowBlur` especifica el desenfoque y el color o una sombra. De manera predeterminada, el desenfoque es 0 (sin desenfoque). Entre mayor sea el valor, más desenfocados aparecerán los bordes de la sombra.
- Un atributo `shadowOffsetX` positivo mueve la sombra a la derecha del rectángulo.
- Un atributo `shadowOffsetY` positivo mueve la sombra hacia abajo del rectángulo.
- El atributo `shadowColor` especifica el color de la sombra.
- Si usamos un valor negativo de `shadowOffsetX` la sombra se mueve a la izquierda del rectángulo.
- Si usamos un valor negativo de `shadowOffsetY` la sombra se mueve hacia arriba del rectángulo.
- El valor predeterminado para `shadowOffsetX` y `shadowOffsetY` es 0 (sin sombra).

### EJEMPLO 1.

#### Archivo `sombras.html`:

```
<!--Creación de sombras en un canvas. -->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Sombras</title>
  </head>
  <body>
    <canvas id = "sombra" width = "525" height = "250"
      style = "border: 1px solid black;">
    </canvas>
    <script>
      // efecto de sombra con desplazamientos positivos
      var canvas = document.getElementById("sombra");
      var contexto = canvas.getContext("2d")
      contexto.shadowBlur = 10;
      contexto.shadowOffsetX = 15;
      contexto.shadowOffsetY = 15;
      contexto.shadowColor = "blue";
      contexto.fillStyle = "cyan";
      contexto.fillRect(25, 25, 200, 200);
      // efecto de sombra con desplazamientos negativos
      contexto.shadowBlur = 20;
      contexto.shadowOffsetX = -20;
      contexto.shadowOffsetY = -20;
      contexto.shadowColor = "gray";
      contexto.fillStyle = "magenta";
      contexto.fillRect(300, 25, 200, 200);
    </script>
  </body>
</html>
```

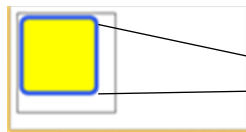


Azul con sombra nítida abajo derecha

Magenta con sombra gris difuminada arriba izquierda

## 2. Curvas cuadráticas

- Las curvas cuadráticas tienen un punto inicial, un punto final y un solo punto de inflexión.
- El método `quadraticCurveTo` usa cuatro argumentos. Los primeros dos (`cpx` y `cpy`) son las coordenadas del punto de control: el punto de inflexión de la curva. Los argumentos tercero y cuarto (`x` y `y`) son las coordenadas del punto final. El punto inicial es el destino de la última subruta y se especifica mediante el uso de los métodos `moveTo` o `lineTo`.



Cada esquina con un radio de 10

## 3. Curvas de Bezier

- Las curvas de Bezier tienen un punto inicial, un punto final y dos puntos de control a través de los cuales pasa la curva. Estos pueden usarse para dibujar curvas con uno o dos puntos de inflexión, dependiendo de las coordenadas de los cuatro puntos.
- El método `bezierCurveTo` usa seis argumentos. Los primeros dos argumentos (`cp1x` y `cp1y`) son las coordenadas del primer punto de control. Los argumentos tercero y cuarto (`cp2x` y `cp2y`) son las coordenadas del segundo punto de control. Por último, los argumentos quinto y sexto (`x` y `y`) son las coordenadas del punto final. El punto inicial es el destino de la última subruta, la cual se especifica mediante el método `moveTo` o `lineTo`.

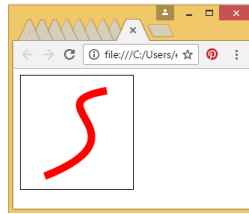
### EJEMPLO 5.

#### Archivo `curvabezier.html`:

```
<!-- Dibujo de una curva de Bezier sobre un canvas. -->
<!DOCTYPE html>
<html>
  <head>
    <meta charset = "utf-8">
    <title>Curvas de Bezier</title>
  </head>
  <body>
    <canvas id = "dibujarBezier" width = "150" height = "150"
      style = "border: 1px solid black;">
    </canvas>
    <script>
      var canvas = document.getElementById("dibujarBezier");
      var contexto = canvas.getContext("2d");
      contexto.beginPath();
      contexto.moveTo(115, 20);
      contexto.bezierCurveTo(12, 37, 176, 77, 32, 133);
      contexto.lineWidth = 10;
      contexto.strokeStyle = "red";
      contexto.stroke();
    </script>
  </body>
```



</html>



**Curva de Bezier.**

#### 4. Gradientes lineales

- El método `createLinearGradient` tiene cuatro argumentos que representan `x0`, `y0`, `x1`, `y1`, en donde los primeros dos argumentos son las coordenadas `x` y `y` del inicio del gradiente y los últimos dos son las coordenadas `x` y `y` del final.
- El inicio y el final tienen las mismas coordenadas `x` pero distintas coordenadas `y`, por lo que el inicio del gradiente es un punto en la parte superior del `canvas` que está justo encima del punto al final del gradiente en la parte inferior. Esto crea un gradiente de línea vertical que comienza en la parte superior y cambia a medida que el gradiente se mueve hacia la parte inferior del `canvas`.
- El método `addColorStop` agrega topes de color al gradiente. Cabe mencionar que cada tope de color tiene un valor positivo entre 0 (el inicio del gradiente) y 1 (el final del gradiente). Para cada tope de color especificamos un color.
- El método `fillStyle` especifica un gradiente y el método `fillRect` dibuja el gradiente en el `canvas`.
- Para dibujar un gradiente horizontal, use el método `createLinearGradient` en donde el inicio y el final tienen distintas coordenadas `x` pero las mismas coordenadas `y`.

#### EJEMPLO 6.

##### Archivo `gradientelineal.html`:

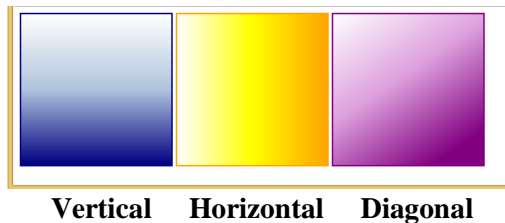
```
<!-- Dibujo de gradientes lineales en un canvas. -->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Gradientes lineales</title>
  </head>
  <body>
    <!-- gradiente lineal vertical -->
    <canvas id = "gradienteLineal" width = "200" height = "200"
      style = "border: 2px solid navy;">
    </canvas>
    <script>
      var canvas = document.getElementById("gradienteLineal");
      var contexto = canvas.getContext("2d");
      var gradiente = contexto.createLinearGradient(0, 0, 0, 200);
      gradiente.addColorStop(0, "white");
      gradiente.addColorStop(0.5, "lightsteelblue");
      gradiente.addColorStop(1, "navy");
      contexto.fillStyle = gradiente;
      contexto.fillRect(0, 0, 200, 200);
    </script>
    <!-- gradiente lineal horizontal -->
    <canvas id = "gradienteLineal2" width = "200" height = "200"
      style = "border: 2px solid orange;">
    </canvas>
    <script>
```



```

var canvas = document.getElementById("gradienteLineal2");
var contexto = canvas.getContext("2d");
var gradiente = contexto.createLinearGradient(0, 0, 200, 0);
gradiente.addColorStop(0, "white");
gradiente.addColorStop(0.5, "yellow" );
gradiente.addColorStop(1, "orange");
contexto.fillStyle = gradiente;
contexto.fillRect(0, 0, 200, 200);
</script>
<!-- gradiente lineal diagonal -->
<canvas id = "gradienteLineal3" width = "200" height = "200"
  style = "border: 2px solid purple;">
</canvas>
<script>
  var canvas = document.getElementById("gradienteLineal3");
  var contexto = canvas.getContext("2d");
  var gradiente = contexto.createLinearGradient(0, 0, 135, 200);
  gradiente.addColorStop(0, "white");
  gradiente.addColorStop(0.5, "plum");
  gradiente.addColorStop(1, "purple");
  contexto.fillStyle = gradiente;
  contexto.fillRect( 0, 0, 200, 200);
</script>
</body>
</html>

```



**Vertical      Horizontal      Diagonal**

## 5. Gradientes radiales

- Un gradiente radial está compuesto de dos círculos: un círculo interno en donde comienza el gradiente y un círculo externo en donde termina el gradiente.
- El método `createRadialGradient` tiene seis argumentos que representan `x0`, `y0`, `r0`, `x1`, `y1`, `r1`, en donde los primeros tres argumentos son las coordenadas `x` y `y` junto con el radio del círculo inicial del gradiente, y los últimos tres argumentos son las coordenadas `x` y `y` junto con el radio del círculo final.
- Al dibujar círculos concéntricos con los mismos coordenados `x` y `y` pero distintos radios se crea un gradiente radial que comienza en un centro común y cambia a medida que se desplaza hacia el círculo final.
- Si los círculos inicial y final no son concéntricos, se altera el efecto.

### EJEMPLO 7.

#### Archivo `gradienteradial.html`:

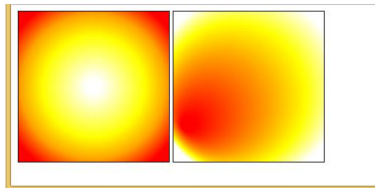
```

<!-- Dibujo de gradientes radiales en un canvas. -->
<!DOCTYPE html>
<html>
  <head>
    <meta charset = "utf-8">
    <title>Gradientes radiales</title>
  </head>
  <body>
    <!-- gradiente radial con círculos concéntricos -->
    <canvas id = "gradienteRadial" width = "200" height = "200"

```



```
        style = "border: 1px solid black;";
    </canvas>
    <script>
        var canvas = document.getElementById("gradienteRadial");
        var contexto = canvas.getContext("2d");
        var gradiente = contexto.createRadialGradient(
            100, 100, 10, 100, 100, 125);
        gradiente.addColorStop(0, "white");
        gradiente.addColorStop(0.5, "yellow");
        gradiente.addColorStop(0.75, "orange");
        gradiente.addColorStop(1, "red");
        contexto.fillStyle = gradiente;
        contexto.fillRect(0, 0, 200, 200);
    </script>
    <!-- no gradiente radial con círculos no concéntricos -->
    <canvas id = "gradienteRadial2" width = "200" height = "200"
        style = "border: 1px solid black;";
    </canvas>
    <script>
        var canvas = document.getElementById("gradienteRadial2");
        var contexto = canvas.getContext("2d");
        var gradiente = contexto.createRadialGradient(
            20, 150, 10, 100, 100, 125);
        gradiente.addColorStop(0, "red");
        gradiente.addColorStop(0.5, "orange");
        gradiente.addColorStop(0.75, "yellow");
        gradiente.addColorStop(1, "white");
        contexto.fillStyle = gradiente;
        contexto.fillRect(0, 0, 200, 200);
    </script>
    </body>
</html>
```



**Cambio de gradientes desde el centro hacia afuera.**