

El elemento HTML <canvas> se utiliza para dibujar gráficos, sobre la marcha, a través de secuencias de comandos (normalmente JavaScript). El elemento <canvas> es sólo un contenedor de gráficos.

Debe utilizar un script para dibujar los gráficos.

El elemento <canvas> tiene varios métodos para dibujar rutas, cajas, círculos, texto y añadir imágenes.

## Elemento Canvas

### ATRIBUTOS

Nombre	Tipo	Por defecto
width	<i>unsigned long</i>	300
height	<i>unsigned long</i>	150

### MÉTODOS

Devuelve	Nombre
<i>string</i>	toDataURL( [Optional] <i>string</i> type, [Variadic] <i>any</i> args )
<i>Object</i>	getContext( <i>string</i> ctxId )

## Transformación

### MÉTODOS

Devuelve	Nombre
<i>void</i>	scale( <i>float</i> x, <i>float</i> y )
<i>void</i>	rotate( <i>float</i> angle )
<i>void</i>	translate( <i>float</i> x, <i>float</i> y )
<i>void</i>	transform( <i>float</i> m11, <i>float</i> m12, <i>float</i> m21, <i>float</i> m22, <i>float</i> dx, <i>float</i> dy )
<i>void</i>	setTransform( <i>float</i> m11, <i>float</i> m12, <i>float</i> m21, <i>float</i> m22, <i>float</i> dx, <i>float</i> dy )

## Contexto 2D

### ATRIBUTOS

Nombre	Tipo
canvas	<i>HTMLCanvasObject</i> [readonly]

### MÉTODOS

Devuelve	Nombre
<i>void</i>	save()
<i>void</i>	restore()

## Dibujo de imagen

### MÉTODOS

Devuelve	Nombre
<i>void</i>	drawImage( <i>Object</i> image, <i>float</i> dx, <i>float</i> dy, [Optional] <i>float</i> dw, <i>float</i> dh )
El argumento "image" puede ser de tipo <i>HTMLImageElement</i> , <i>HTMLCanvasElement</i> or <i>HTMLVideoElement</i>	
<i>void</i>	drawImage( <i>Object</i> image, <i>float</i> sx, <i>float</i> sy, <i>float</i> sw, <i>float</i> sh, <i>float</i> dx, <i>float</i> dy, <i>float</i> dw, <i>float</i> dh )

## Colores, estilos y sombras

### ATRIBUTOS

Nombre	Tipo	Por defecto
strokeStyle	<i>any</i>	black
fillStyle	<i>any</i>	black
shadowOffsetX	<i>float</i>	0.0
shadowOffsetY	<i>float</i>	0.0
shadowBlur	<i>float</i>	0.0
shadowColor	<i>string</i>	transparent black

### INTERFAZ CanvasGradient

<i>void</i>	addColorStop( <i>float</i> offset, <i>string</i> color )
-------------	--

### INTERFAZ CanvasPattern

Sin atributos ni métodos

### MÉTODOS

Devuelve	Nombre
<i>CanvasGradient</i>	createLinearGradient( <i>float</i> x0, <i>float</i> y0, <i>float</i> x1, <i>float</i> y1 )
<i>CanvasGradient</i>	createRadialGradient( <i>float</i> x0, <i>float</i> y0, <i>float</i> r0, <i>float</i> x1, <i>float</i> y1, <i>float</i> r1 )
<i>CanvasPattern</i>	createPattern( <i>Object</i> image, <i>string</i> repetition )
El argumento "image" puede ser del tipo <i>HTMLImageElement</i> , <i>HTMLCanvasElement</i> o <i>HTMLVideoElement</i>	
"repetition" admite cualquiera de los siguientes valores [repeat (por defecto), repeat-x, repeat-y, no-repeat]	

## Manipulación de píxeles

### MÉTODOS

Devuelve	Nombre
<i>ImageData</i>	createImageData( <i>float</i> sw, <i>float</i> sh )
<i>ImageData</i>	createImageData( <i>ImageData</i> imageData )
<i>ImageData</i>	getImageData( <i>float</i> sx, <i>float</i> sy, <i>float</i> sw, <i>float</i> sh )
<i>void</i>	putImageData( <i>ImageData</i> imageData, <i>float</i> dx, <i>float</i> dy )

### INTERFAZ ImageData

width	<i>unsigned long</i>	[readonly]
height	<i>unsigned long</i>	[readonly]
height	<i>unsigned long</i>	[readonly]

### INTERFAZ CanvasPixelArray

length	<i>unsigned long</i>	[readonly]
--------	----------------------	------------

## Rutas

### MÉTODOS

Devuelve	Nombre
<i>void</i>	<code>beginPath()</code>
<i>void</i>	<code>closePath()</code>
<i>void</i>	<code>fill()</code>
<i>void</i>	<code>stroke()</code>
<i>void</i>	<code>clip()</code>
<i>void</i>	<code>moveTo( float x, float y )</code>
<i>void</i>	<code>lineTo( float x, float y )</code>
<i>void</i>	<code>quadraticCurveTo( float cpx, float cpy, float x, float y )</code>
<i>void</i>	<code>bezierCurveTo( float cp1x, float cp1y, float cp2x, float cp2y, float x, float y )</code>
<i>void</i>	<code>arcTo( float x1, float y1, float x2, float y2, float radius )</code>
<i>void</i>	<code>arc( float x, float y, float radius, float startAngle, float endAngle, boolean anticlockwise )</code>
<i>void</i>	<code>rect( float x, float y, float w, float h )</code>
<i>boolean</i>	<code>isPointInPath( float x, float y )</code>

## Estilos de líneas

### ATRIBUTOS

Nombre	Tipo	Por defecto
<code>lineWidth</code>	<i>float</i>	1.0
<code>lineCap</code>	<i>string</i>	<code>butt</code>

Admite cualquiera de los siguientes valores

`butt`

`rounded`

`square`



`lineJoin`



`String`



`miter`

Admite cualquiera de los siguientes valores

`rounded`

`bevel`

`miter`



`miterLimit`



`Float`



10

## Rectángulos

### MÉTODOS

Devuelve	Nombre
<i>void</i>	<code>clearRect( float x, float y, float w, float h )</code>
<i>void</i>	<code>fillRect( float x, float y, float w, float h )</code>
<i>void</i>	<code>strokeRect( float x, float y, float w, float h )</code>

## Composición

### ATRIBUTOS

Nombre	Tipo	Por defecto
<code>globalAlpha</code>	<i>float</i>	1.0
<code>globalCompositeOperation</code>	<i>string</i>	<code>source-over</code>

Admite cualquiera de los siguientes valores



`source-over`



`source-in`



`source-atop`



`source-out`



`destination-atop`



`destination-in`



`destination-out`



`destination-atop`



`lighter`



`XOR`



`copy`



`darker`

## Texto

### ATRIBUTOS

Nombre	Tipo	Por defecto
<code>font</code>	<i>string</i>	10px sans-serif
<code>textAlign</code>	<i>string</i>	<code>start</code>

Admite cualquiera de los siguientes valores

[ `start`, `end`, `left`, `right`, `center` ]

`textBaseline` *string* alphabetic

Admite cualquiera de los siguientes valores

[ `top`, `hanging`, `middle`, `alphabetic`, `ideographic`, `bottom` ]

### MÉTODOS

Devuelve	Nombre
<i>void</i>	<code>fillText( string text, float x, float y, [Optional] float maxWidth )</code>
<i>void</i>	<code>strokeText( string text, float x, float y, [Optional] float maxWidth )</code>
<i>TextMetrics</i>	<code>measureText( string text )</code>

### INTERFAZ TextMetrics

<code>width</code>	<i>string</i>	<code>width</code>
--------------------	---------------	--------------------