

HTML5 CANVAS

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			Contexto 2D		
ATRIBUTOS			ATRIBUTOS		
Nombre	Tipo	Por defecto	Nombre	Tipo	
width	<i>unsigned long</i>	300	canvas	<i>HTMLCanvasObject</i> [readonly]	
height	<i>unsigned long</i>	150	MÉTODOS		
MÉTODOS			Devuelve	Nombre	
Devuelve	Nombre		<i>void</i>	save()	
<i>string</i>	toDataURL(<i>void</i>	restore()	
	[Optional] <i>string</i> type,		Dibujo de imagen		
	[Variadic] <i>any</i> args)				
<i>Object</i>	getContext(<i>string</i> ctxId)		MÉTODOS		
Transformación			Devuelve	Nombre	
MÉTODOS			<i>void</i>	drawImage(
Devuelve	Nombre			<i>Object</i> image,	
<i>void</i>	scale(<i>float</i> x, <i>float</i> y)			<i>float</i> dx, <i>float</i> dy,	
<i>void</i>	rotate(<i>float</i> angle)			[Optional] <i>float</i> dw, <i>float</i> dh)	
<i>void</i>	translate(<i>float</i> x, <i>float</i> y)			El argumento “image” puede ser de tipo	
<i>void</i>	transform(HTMLImageElement, HTMLCanvasElement or	
	<i>float</i> m11, <i>float</i> m12,			HTMLVideoElement	
	<i>float</i> m21, <i>float</i> m22,		<i>void</i>	drawImage(
	<i>float</i> dx, <i>float</i> dy)			<i>Object</i> image,	
<i>void</i>	setTransform(<i>float</i> sx, <i>float</i> sy,	
	<i>float</i> m11, <i>float</i> m12,			<i>float</i> sw, <i>float</i> sh,	
	<i>float</i> m21, <i>float</i> m22,			<i>float</i> dx, <i>float</i> dy,	
	<i>float</i> dx, <i>float</i> dy)			<i>float</i> dw, <i>float</i> dh)	
Colores, estilos y sombras					
ATRIBUTOS			MÉTODOS		
Nombre	Tipo	Por defecto	Devuelve	Nombre	
strokeStyle	<i>any</i>	black	<i>CanvasGradient</i>	createLinearGradient(
fillStyle	<i>any</i>	black		<i>float</i> x0, <i>float</i> y0,	
shadowOffsetX	<i>float</i>	0.0		<i>float</i> x1, <i>float</i> y1)	
shadowOffsetY	<i>float</i>	0.0	<i>CanvasGradient</i>	createRadialGradient(
shadowBlur	<i>float</i>	0.0		<i>float</i> x0, <i>float</i> y0, <i>float</i> r0,	
shadowColor	<i>string</i>	transparent black		<i>float</i> x1, <i>float</i> y1, <i>float</i> r1)	
INTERFAZ CanvasGradient			<i>CanvasPattern</i>	createPattern(
<i>void</i>	addColorStop(<i>float</i> offset, <i>string</i> color)			<i>Object</i> image,	
INTERFAZ CanvasPattern				<i>string</i> repetition)	
Sin atributos ni métodos				El argumento "image" puede ser del tipo	
				HTMLImageElement, HTMLCanvasElement o	
				HTMLVideoElement	
				"repetiton" admite cualquiera de los siguientes valores	
				[repeat (por defecto), repeat-x, repeat-y, no-repeat]	
Manipulación de píxeles					
MÉTODOS			INTERFAZ ImageData		
Devuelve	Nombre		width	<i>unsigned long</i>	[readonly]
<i>ImageData</i>	createImageData(<i>float</i> sw, <i>float</i> sh)		height	<i>unsigned long</i>	[readonly]
<i>ImageData</i>	createImageData(<i>ImageData</i> imageData)		height	<i>unsigned long</i>	[readonly]
<i>ImageData</i>	getImageData(<i>float</i> sx, <i>float</i> sy, <i>float</i> sw, <i>float</i> sh)		INTERFAZ CanvasPixelArray		
<i>void</i>	putImageData(<i>ImageData</i> imageData, <i>float</i> dx, <i>float</i> dy)		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`



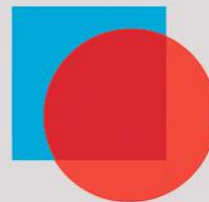
`lighter`



`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>
--------------------	---------------	--------------------