

# CANVAS como herramienta para el desarrollo web

## CANVAS as a tool for web development.

Valentina Varela Alzate

Risaralda, Universidad Tecnológica de Pereira, Perera, Colombia  
valentina.varela@utp.edu.co

**Resumen**—Canvas es un elemento HTML el cual puede ser usado para dibujar gráficos usando scripts (normalmente JavaScript). También puede ser usado para dibujar gráficos, realizar composición de fotos o simples y no tan simples animaciones.

El elemento o herramienta canvas no es difícil de implementar pero se necesitan conocimientos básicos de HTML y JavaScript para entender su funcionamiento. Es un elemento que no está soportado en navegadores antiguos, sin embargo está soportado en la mayoría de las versiones más recientes de los navegadores actuales.

**Palabras clave**—Desarrollo web, HTML, Javascript, programación, código, navegadores, diseño.

**Abstract**—Canvas is an HTML element which can be used to draw graphics using scripts (usually JavaScript). It can also be used to draw graphics, make composition of photos or simple and not so simple animations.

The element or canvas tool is not difficult to implement but basic knowledge of HTML and JavaScript is needed to understand its operation. It is an element that is not supported in old browsers, however it is supported in most of the most recent versions of current browsers.

**Key Word** —Web development, HTML, Javascript, programming, code, browsers, design.

## I. INTRODUCCIÓN

Canvas es un elemento HTML que permite la creación de gráficos y animaciones de forma dinámica por medio de scripts. Fue creado por Apple, pero se ha liberado la propiedad intelectual para englobarlo dentro de los estándares de HTML, no requiere ningún plugin adicional, sólo una un navegador que soporte HTML5.

El elemento canvas permite especificar un área de la página donde se puede, a través de scripts, dibujar y renderizar imágenes, lo que amplía notablemente las posibilidades de las páginas dinámicas y permite hacer cosas que hasta ahora estaban reservadas a los desarrolladores en Flash. Canvas se desarrolló inicialmente por Apple para su navegador Safari y

luego fue utilizado y estandarizado por la organización WHATWG para incorporarlo a HTML 5. Posteriormente también ha sido adoptado por navegadores como Firefox y Opera. Por lo que respecta a Chorme, es un navegador que utiliza el mismo motor de renderizado que Safari, por lo que también soporta el elemento Canvas.

Canvas permite dibujar en la página y actualizar dinámicamente estos dibujos, por medio de scripts y atendiendo a las acciones del usuario. Todo esto da unas posibilidades de uso tan grandes como las que disponemos con el plugin de Flash, en lo que respecta a renderización de contenidos dinámicos. Las aplicaciones pueden ser grandes como podamos imaginar, desde juegos, efectos dinámicos en interfaces de usuario, editores de código, editores gráficos, aplicaciones, efectos 3D, etc.

Actualmente algunas de las aplicaciones más novedosas para para la web utilizan ya canvas para su funcionamiento, donde se puede destacar Bepin, un editor de código de Mozilla, o Google Wave. En la entrada de la Wiki sobre Canvas podemos encontrar diversos enlaces a sitios web con ejemplos de uso de este elemento del HTML 5.

```
<canvas id="micanvas" width="200" height="100">
  Este texto se muestra para los navegadores no compatibles con Canvas.
<br>
  Por favor, utiliza un navegador que soporte HTML5.
</canvas>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Dibujando2</title>
</head>
<body>
<h1>Dibujo: </h1>
  <input type="text" id="cajaTxt">
  <input type="button" value="Enviar" id="cajaBtn">
  <br><br>
  <canvas id="dib" width="300" height="300"></canvas>
  <p>-----</p>
  <script src="dibujo.js"></script>
</body>
</html>
```

## II. CONTENIDO

La herramienta CANVAS es parecida al elemento `<img>`, con la diferencia que esta no tiene los atributos `src` y `alt`. El elemento `canvas` tiene solo dos atributos - `width` y `height`. Ambos son opcionales y pueden ser definidos usando propiedades DOM.

Cuando los atributos ancho y alto no están especificados, el lienzo se inicializa con 300 pixels ancho y 150 pixels de alto. El elemento puede ser arbitrariamente redimensionado por CSS, pero durante el renderizado la imagen es escalada para ajustarse al tamaño de su layout. Si el tamaño del CSS no respeta el ratio del `canvas` inicial, este aparecerá distorsionado.

El atributo `id` no está especificado para el elemento `canvas` pero es uno de los atributos globales de HTML el cual puede ser aplicado a cualquier elemento; siempre se debe proporcionar un `id` porque esto hace que sea más fácil identificarlo en un script.

El elemento `canvas` puede ser estilizado como a cualquier imagen normal (`margin`, `border`, `background`, etc). Ya que estas reglas no afectan a lo dibujado sobre el `canvas`, cuando no se tienen reglas de estilo aplicadas al `canvas`, será completamente transparente.

`Canvas` consiste en una región dibujable definida en el código HTML con atributos de altura y ancho. El código JavaScript puede acceder a la zona a través de un conjunto completo de funciones similares a las de otras APIs comunes de dibujo 2D, permitiendo así que los gráficos sean generados dinámicamente. Algunos de los usos previstos incluyen construcción de gráficos, animaciones, juegos, y la composición de imágenes. Con la utilización de javascript también se puede implementar `canvas`:

```
var example = document.getElementById('example');
var context = example.getContext('2d');
context.fillStyle = 'red';
context.fillRect(30, 30, 100, 100);
```

**1) Incluir canvas en HTML.** Documento breve que presenta resultados originales preliminares o parciales de una investigación científica o tecnológica, que por lo general requieren de una pronta difusión.

El elemento tiene apertura y cierre y entre medias podemos escribir un texto que será lo que vean los usuarios que entren con navegadores que no soporten la etiqueta `CANVAS`.

Para especificar las características de este `canvas` tenemos varios atributos:

- Atributo `id`: Para asignarle un nombre único y luego referirnos a este `canvas` desde Javascript.

- Atributos `width` y `height`: Para indicar la anchura y altura del área del `canvas`.

Otros atributos se pueden colocar de manera opcional, como por ejemplo `style`, para indicar atributos de hojas de estilo para definir el aspecto del lienzo.

```
<canvas id="micanvas" width="200" height="100">
Este texto se muestra para los navegadores no compatibles con canvas.
<br>
Por favor, utiliza Firefox, Chrome, Safari u Opera.
</canvas>
```

**2) Manera alternativa para correr el código de canvas.** El elemento `canvas` se diferencia de un tag `<img>` en que, como los elementos `<video>`, `<audio>` o `<picture>`, es fácil definir contenido alternativo (fallback content) para mostrarse en navegadores viejos que no soporten el elemento `<canvas>`, como versiones de Internet Explorer previas a la versión 9 o navegadores de texto se debe proporcionar contenido alternativo para mostrar en estos navegadores.

Para Proporcionar contenido alternativo solo se debe insertar el contenido alterno dentro del elemento `canvas`. Los navegadores que no soportan `<canvas>` ignoran el contenedor y mostrarán el contenido indicado dentro de este. Navegadores que soporten `<canvas>` ignorarán el contenido en su interior (de las etiquetas), y mostrarán el `canvas` normalmente.

```
1 <canvas id="stockGraph" width="150" height="150">
2   current stock price: $3.15 +0.15
3 </canvas>
4
5 <canvas id="clock" width="150" height="150">
6   
7 </canvas>
```

**3) Contexto de renderización.** La herramienta crea un lienzo de dibujo fijado que expone uno o más contextos renderizados, los cuales son usados para crear y manipular el contenido mostrado.

El `canvas` está inicialmente en blanco. Para mostrar alguna cosa, un script necesita poder acceder al contexto a renderizar y dibujar sobre este. El elemento `canvas` tiene un método llamado `getContext()`, usado para obtener el contexto a renderizar y sus funciones de dibujo. `getContext()` toma un parámetro, el tipo de contexto.

```
1 var canvas = document.getElementById('tutorial');
2 var ctx = canvas.getContext('2d');
```

**4) Comprobación del soporte.** El contenido de regreso que es mostrado en navegadores los cuales no soportan `canvas`.

Para los Scripts puede también comprobarse su soporte desde la programación por un simple test para la presencia del método `getContext()`.

```
1 var canvas = document.getElementById('tutorial');
2
3 if (canvas.getContext){
4     var ctx = canvas.getContext('2d');
5     // drawing code here
6 } else {
7     // canvas-unsupported code here
8 }
```

### III. CONCLUSIONES

El elemento canvas de HTML5 es la funcionalidad más interesante que pueden utilizar los desarrolladores para crear aplicaciones realmente avanzadas para la Web sin necesidad de instalar complementos en el navegador , ya que canvas es un lienzo de mapa de bits dependiente de la resolución de pantalla que se puede utilizar para representar gráficos, imágenes de juegos o cualquier otra información de este tipo sobre la marcha. En otras palabras, canvas es un elemento de HTML5 que permite dibujar gráficos con Javascript. Que a su vez sirve para representar en la pantalla textos, imágenes, gráficos de todo tipo, rectángulos, líneas, gradientes y efectos de forma dinámica. El dibujo en el canvas se hace mediante el API Canvas 2D. Esta API contiene multitud de funciones con las que prácticamente se puede dibujar cualquier cosa sobre él.

### WEBGRAFIA

- [https://developer.mozilla.org/es/docs/Web/Guide/HTML/Canvas\\_tutorial](https://developer.mozilla.org/es/docs/Web/Guide/HTML/Canvas_tutorial)
- <https://www.arsys.es/blog/programacion/disenoweb/que-es-canvas/>
- <https://desarrolloweb.com/articulos/dibujo-api-canvas.html>
- <https://platzi.com/tutoriales/1050-programacion-basica/309-canvas-2-mas-funciones-y-un-poco-de-magia-codigo-comentado/>
- [https://es.wikipedia.org/wiki/Canvas\\_\(HTML\)](https://es.wikipedia.org/wiki/Canvas_(HTML))