CANVAS

Daniela Ospina Salamanca

*Facultad de ingenierias, Universidad Tecnológica, Pereira, Colombia*

Correo-e: daniela.ospina4@utp.edu.co

***Resumen*—**

**Canvas es un elemento computacional abarcado por el lenguaje de programación HTML que permite al usuario realizar gráficos mediante secuencias de comandos, también permite la renderización interpretada dinámica de gráficos 2D y mapas de bits, así como animaciones con estos gráficos. Se trata de un modelo de procedimiento de bajo nivel, que actualiza un mapa de bits y no tiene una gráfica de escena integrada.**

***Palabras clave—* HTML, Gráficos, Comandos**

***Abstract*—**

**Canvas is a computational element encompassed by the HTML programming language that allows the user to make graphs using scripts, also allows the dynamic interpreted rendering of 2D graphics and bitmaps, as well as animations with these graphics. This is a low-level procedural model, which updates a bitmap and does not have an integrated scene graph.**

***Key Word* —HTML, Graphics, Commands**

INTRODUCCIÓN

El elemento canvas puede definirse como un entorno para crear imágenes dinámicas. El propio elemento es relativamente simple. Lo único que hay que especificar al usarlo son sus dimensiones.

Sin embargo, el elemento <canvas> no tiene habilidad de dibujo en sí mismo, siendo sólo un contenedor para gráficos. Por esto es importante que el <canvas> tenga asignado un id, ya que habrá que manipularlo a través de JavaScript. Utilizaremos document.getElementById para encontrar el canvas.

Canvas fue introducido inicialmente por Apple para su uso dentro de su propio componente de Mac OS X Surgido en 2004, para empujar aplicaciones como widgets de Dashboard y el navegador Safari. Más tarde, fue estandarizado por el Grupo de Trabajo de Tecnologías de Aplicación de hipertexto Web (WHATWG por sus siglas en inglés) sobre las nuevas especificaciones propuestas para tecnologías web de última generación.

Para el usuario, todo son ventajas. No requiere ningún plugin adicional, sólo una un navegador que soporte HTML5 y hoy en día todos los navegadores importantes (Safari, Chrome, Firefox, Opera e Internet Explorer) soportan Canvas, desde hace ya unas cuantas versiones.

**Contexto de uso**

|  |  |
| --- | --- |
| Contenido permitido | Transparente, ya sea contenido estático o contenido dinámico |
| Omisión de etiquetas | Ninguna, deben estar presentes tanto las etiquetas de inicio como las de cierre |
| Elementos primarios permitidos | Cualquier elemento que acepte contenido dinámico o cualquier otro elemento que acepte contenido estático . |
| Documento normativoAtributos | HTML 5, sección 4.8.10 |

1. CONTENIDO

**1) Cómo se usa?**

El 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.

Cuando trabajamos con canvas, realmente toda la magia la haremos en JavaScript. Primero debemos referenciar el elemento canvas y adquirir su contexto (API).

* Para crear un canvas en HTML5 tenemos que utilizar el elemento CANVAS

<canvas>

Contenido alternativo

**<**/canvas>

* A un elemento CANVAS podemos darle un alto y ancho mediante los atributos height y width respectivamente. Así como darle un identificador mediante el atributo id.

<canvas height="300px" width="300px" id="micanvas"**><**/canvas>

En el caso de que no indiquemos las dimensiones, estas serán de 300x150.

* Una vez que tenemos nuestro "lienzo" vamos a dibujar algún gráfico. Para ello utilizaremos el lenguaje Javascript. Así lo primero que haremos será obtener la referencia sobre el CANVAS mediante un getElementById().

**var** canvas = document.getElementById("micanvas");

* Dentro de un CANVAS podemos volcar varios contextos. Lo que son conocidos como contextos de renderizado. Así podremos tener contextos 2D y 3D. Estos últimos todavía en fase experimental. El método .getContext() sobre el CANVAS nos permitirá crear un nuevo contexto.

**var** canvas = document.getElementById("micanvas");

**var** ctx = canvas.getContext("2d");

* Ahora ya pasaremos a dibujar gráficos. Así, con el manejo de una propiedad: .fillStyle y un método .fillRect() podemos crear rectángulos de colores. En el caso de .fillStyle recibe como parámetro un color definido en formato RGB y en el caso de .fillRect() las coordenadas de los vértices superior izquierdo, así como el alto y ancho.

**var** canvas = document.getElementById("micanvas");

**var** ctx = canvas.getContext("2d");

ctx.fillStyle = "rgb(255,0,0)";

ctx.fillRect(20,20,100,100);

ctx.fillStyle = "rgb(0,255,0)";

ctx.fillRect(60,60,140,140);

ctx.fillStyle = "rgb(0,0,255)";

ctx.fillRect(100,100,180,180);

Así nos quedará el código que insertará gráficos al crear nuestro primer canvas con HTML5.

**2**. **Ejemplos:**

* **El siguiente código crea un elemento canvas en una página HTML:**

**<canvas id="example" width="200" height="200">**

**Este texto se muestra si su navegador no soporta el lienzo (canvas) de HTML5.**

**</canvas>**

* **Utilizando JavaScript, usted puede dibujar en el lienzo (canvas):**

**Este código dibuja un rectángulo rojo en la pantalla.**

**var example = document.getElementById('example');**

**var context = example.getContext('2d');**

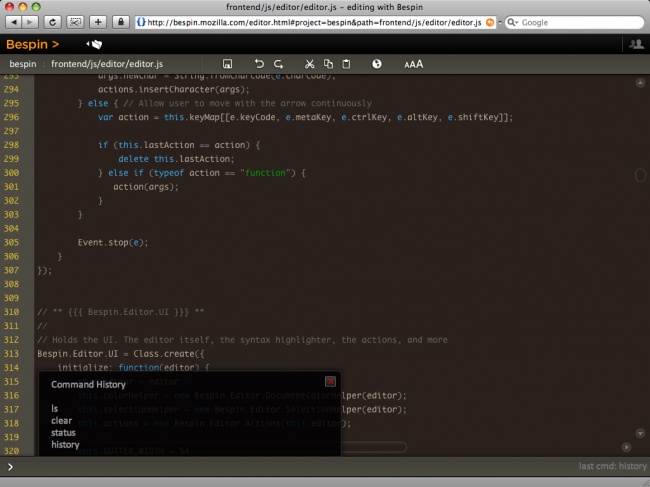
**context.fillStyle = 'red';**

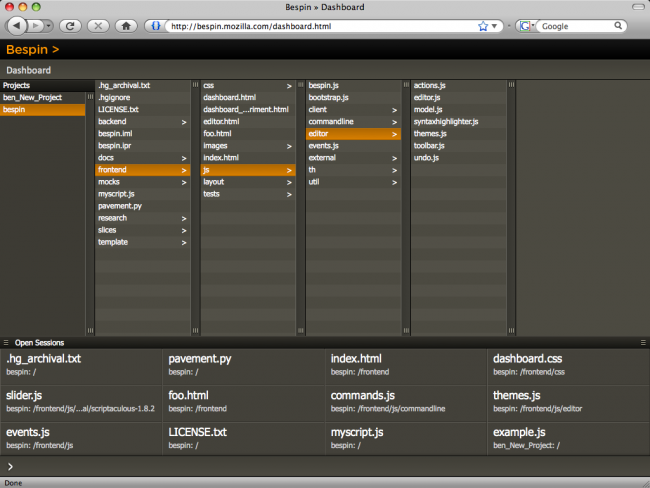
**context.fillRect(30, 30, 100, 100);**

**3) Aplicaciones**

El potencial de canvas reside en su habilidad para actualizar su contenido en **tiempo real**. Si usamos esa habilidad para responder a eventos de usuario, podemos crear herramientas y juegos que anteriormente a la nueva especificación hubiesen requerido de un plugin externo como Flash.

Se pueden hacer cosas **increibles** con canvas, una de las aplicaciones que más me han impresionado es Skywritter (Bespin) que después se integraría en Ace aunque éste último usa el DOM para dibujar el texto en lugar de canvas. Podemos ver unas screenshots de Skywritter (Bespin) en acción.





Como puedes comprobar, el potencial de canvas es **enorme** y no tiene nada que envidiar a plugins o frameworks JavaScript. Skywritter (Bespin) es muy potente e impresionante pero no es solo eso, es el mejor ejemplo de que ***no*** hacer con canvas.

¿Y eso por qué? te estarás preguntando. Pues porque el editor de Skywritter (Bespin) maneja el texto con el elemento canvas solo que **no es texto**, son una serie de formas que aparentan ser texto. Todo documento en la Web **debería de poder ser descrito** con un objeto **DOM**. El contenido dentro de canvas no puede ser representado como nodos de un objeto DOM y debemos huir de ese tipo de prácticas.

Algunos dispositivos como los lectores de pantallas y otro tipo de tecnología que provee de accesibilidad a personas discapacitadas necesitan tener acceso al DOMpara que el documento tenga sentido para ellos. Es decir, sin DOM no hay accesibilidad. La falta de accesibilidad de canvas puede definirse como una gran cagada de HTML5. Si el contenido de la web (por muy chulo que esté) no puede ser accedido por cualquier tipo de dispositivo facilitando así la accesibilidad, canvas estará cometiendo **los mismos errores** que Flash antes que él. Aunque por fortuna, ya hay gente trabajando en una **solución**.

.

**5) Conclusiones**

El nuevo elemento canvas nos ofrece una potencia desconocida hasta el momento en el desarrollo de páginas web, algunos frameworks JavaScript como ExtJS4 ya incorporan este elemento en sus rutinas de renderizado HTML. Ya hemos visto cuales son los problemas que se le imputan al elemento debido a su falta de accesibilidad, como con toda herramienta, debemos hacer un uso inteligente de ella e intentar no abusar y ser diligentes o convertiremos canvas en un nuevo elemento oscuro, inaccesible e insolidario como ciertos plugins existentes.

Aunque he de reconocer que lo que se puede hacer con él es una auténtica pasada. Esperemos que sus problemas se solucionen pronto puesto que es mucho lo que este nuevo elemento puede aportar al desarrollo web.

REFERENCIAS

1. Canvas.[Online]. Available: <https://developer.mozilla.org/es/docs/Web/HTML/Elemento/canvas#Contexto_de_uso>
2. Canvas (HTML). [Online]. Available: <https://es.wikipedia.org/wiki/Canvas_(HTML)>
3. O. Campos. (2011, Jun.). Introducción al elemento Canvas de HTML5 [Online].

Available: <https://www.genbeta.com/desarrollo/introduccion-al-elemento-canvas-de-html5>

1. J. M. Barquero García (2015, Sept.). ¿Qué es Canvas?. [Online]. Available: <https://www.arsys.es/blog/programacion/diseno-web/que-es-canvas/>
2. (2012, Feb.). Crear un Canvas en HTML5 [Online].

Available:

<http://lineadecodigo.com/html5/crear-un-canvas-en-html5/>

1. Canvas - Una introducción [Online].

Available:

<http://w3.unpocodetodo.info/canvas/introduccion.php>