HTML – JAVASCRIPT - CANVAS

HTML – JAVASCRIPT - CANVAS

Autor 1: Diego Alejandro Castro Cardona

*Ingeniería de Sistemas, Universidad Tecnológica de Pereira*

Correo-e: d.castro4@utp.edu.co

***Resumen*— HTML, siglas en inglés de HyperText Markup Language (‘lenguaje de marcas de hipertexto’), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.**

**JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,3​ basado en prototipos, imperativo, débilmente tipado y dinámico.**

**Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas4​ aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.**

**Desde el 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de javascript. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se liberó en julio del 2015.**

**JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.**

**Canvas (o "lienzo" traducido al español) es un elemento HTML incorporado en HTML5 que permite la generación de gráficos dinámicamente por medio del scripting. Entre otras cosas, permite la renderización interpretada dinámica de gráficos 2D y mapas de bits, asi 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, javascript, canvas, programación.**

***Abstract*— Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.**

**Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.**

**JavaScript often abbreviated as JS, is a high-level, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.**

**Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it, and major web browsers have a dedicated JavaScript engine to execute it.**

**As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities. It relies upon the host environment in which it is embedded to provide these features.**

**The canvas element is part of HTML5 and allows for dynamic, scriptable rendering of 2D shapes and bitmap images. It is a low level, procedural model that updates a bitmap and does not have a built-in scene graph; however through WebGL allows 3D shapes and images and so-on. HTML5 Canvas also helps in making 2D games.**

***Key Word* —html, javascript, canvas, programming.**

1. INTRODUCCIÓN

A continuación, se realizará una serie de ejemplos y definiciones que están relacionadas con los lenguajes HTML, JavaScript y Canvas.

Para empezar, debemos reconocer que solo JavaScript es un lenguaje de programación entre estos, HTML es un lenguaje marcado para la elaboración de páginas web y Canvas es un elemento de HTML que sirve para dibujar gráficos a través de scripts con el lenguaje de JavaScript.

También vamos a conocer las sintaxis de estos lenguajes, como podemos empezar a escribir código y codificarlo y usarlo en los ejemplos incluso mostraremos ejemplos basados en imágenes de ejecución del código y sus resultados.

1. CONTENIDO

**1- HTML.**

*“HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la siglas que corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto”.*

Html sirve para maquetar una pagina web, es decir, como van ordenados los elementos en esta. Cada elemento se encuentra encerrado dentro de etiquetas Éstas se caracterizan porque van dentro de los caracteres menor que < y mayor que >, como en este ejemplo:

<Aquí dentro va el nombre de la etiqueta>

* 1. **– Ejemplos.**

Aunque eso no es todo debido a que deben tener un inicio y un fin. El inicio de una etiqueta es normal, en otras palabras lleva los dos caracteres que mencioné anteriormente y dentro de estos se encuentra el nombre, sin embargo, existe una diferencia para el fin de la etiqueta ya que antes de escribir el nombre de la etiqueta debemos escribir una diagonal /. Como en este ejemplo:

<Inicio de la etiqueta>

</Fin de la etiqueta>

Existes ciertas excepciones con el fin de las etiquetas, ya que algunas usualmente sólo se utilizan con el inicio de la etiqueta, ejemplos de esto son las etiquetas BR (salto de línea), IMG (etiqueta para poner una imagen), entre otras.

* 1. **– Estructura del código.**

<html>  
<head>  
<meta charset="utf-8">  
 <title> Aquí va el título de la página</title>  
</head>  
<body>  
 <h1>Aquí va nuestro título y esta en la etiqueta <u>h1</u>.</h1>  
 <h2>Este es un subtítulo y se encuentra en la etiqueta <u>h2</u>.</h2>  
 <p>Este es un párrafo muy pequeño que se encuentra dentro de la etiqueta <b>p</b> de <i>html</i></p>  
</body>  
</html>

Para escribir código HTML se necesita un editor de texto tal como el Bloc de Notas, si embargo existen programas que facilitan la escritura de este tipo de código como lo son: Sublime Text o Notepad++.

Por ultimo, una vez teniendo listo el código para ser ejecutado se debe guardar con la extensión \*.html.

Ej: archivo.html

Y se ejecutaría en cualquier navegador web.

* 1. **– Imágenes de ejecución.**

CODIGO:

<html>

<head>

<meta charset="utf-8">

<title> Aquí va el título de la página</title>

</head>

<body>

<h1>Aquí va nuestro título y esta en la etiqueta <u>h1</u>.</h1>

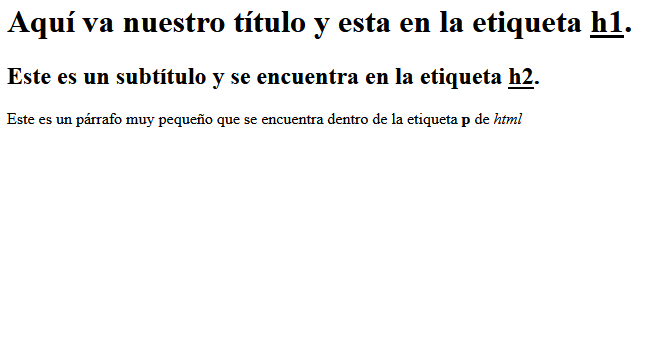
<h2>Este es un subtítulo y se encuentra en la etiqueta <u>h2</u>.</h2>

<p>Este es un párrafo muy pequeño que se encuentra dentro de la etiqueta <b>p</b> de <i>html</i></p>

</body>

</html>

RESULTADO:



CODIGO:

<HTML>

<HEAD>

<TITLE>Agencia de Marketing</TITLE>

</HEAD>

<BODY>

<A NAME=”arriba”>

<CENTER>

<IMG SRC=”img/logo.png”>

<IMG SRC=”img/marketing.gif”>

</CENTER>

Contenido de la página

<HR>

<ADDRESS>

Diseño realizado por

<FONT COLOR=”#800000″>

Ejemplos.

</FONT>

<BR>

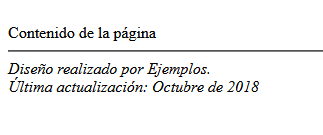
Última actualización: Octubre de 2018

</ADDRESS>

</BODY>

</HTML>

RESULTADO:



**2- JAVASCRIPT.**

JavaScript es un lenguaje de programación que te permite realizar actividades complejas en una página web — cada vez más una página web hace más cosas que sólo mostrar información estática — como mostrar actualizaciones de contenido en el momento, interactuar con mapas, animaciones gráficas 2D/3D etc.

Es un lenguaje de programación que te permite crear contenido nuevo y dinámico, controlar archivos de multimedia, crear imágenes animadas y muchas otras cosas más.

Para escribir código se necesita agregar la siguiente etiqueta al archivo HTML:

<script src="scripts/main.js"></script>

Dentro de esta etiqueta debe ir el código en javascript.

Ej:

var miTitulo = document.querySelector('h1');

miTitulo.innerHTML = 'Hello world!';

**2.1 – Ejemplos.**

En javascript, cada declaración debe terminar con un ;

Asi el programa reconoce cuando se comienza una nueva sentencia.

var helado = 'chocolate';

if (helado === 'chocolate') {

alert('¡Si, amo el helado de chocolate!');

} else {

alert('Awwww, pero mi favorito es el de chocolate...');

}

Nota: algunos puntos que se puede realizar con javascript son:

La asignación de variables, funciones, operaciones aritméticas, operaciones lógicas, loops, entre otras.

**2.2 – Estructura del código.**

<!DOCTYPE html>

<html>

<head>

<title>Ej JavaScript</title>

**<script type="text/javascript">**

document.querySelector('html').onclick = function() {

alert('Ouch! Deja de pincharme!');

}

**</script>**

</head>

<body>

</body>

</html>

**2.3 – Imágenes de ejecución.**

CODIGO:

<!DOCTYPE html>

<html>

<head>

<title>Ej JavaScript</title>

<script type="text/javascript">

document.querySelector('html').onclick = function() {

alert('Ouch! Deja de pincharme!');

}

</script>

</head>

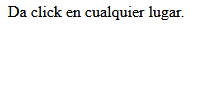
<body>

Da click en cualquier lugar.

</body>

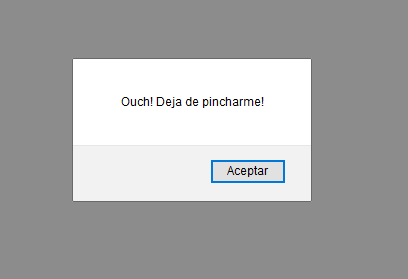
</html>

RESULTADO HTML:



RESULTADO JAVASCRIPT:

\*Hacemos click en cualquier lugar de la pagina y ocurre lo siguiente.\*



**3- CANVAS.**

Canvas (lienzo) es un elemento HTML que permite la creación de gráficos y animaciones de forma dinámica por medio de scripts. Sus aplicaciones son practicamente iniminaginables: crear juegos, interfaces, editores gráficos o efectos dinámicos, aplicaciones 3D, etc.

<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>

* 1. **– Ejemplos.**

Para empezar a usar canvas solo debemos crear la etiqueta dentro del body del html, a continuación, se crea un lienzo donde podemos empezar a dibujar.

Dentro de las etiqutas <canvas></canvas> simplemente va el texto que muestra si un navegar no es compatible con canvas.

El código para dibujar se encuentra dentro de las etiquetas de

<script></script>

Pero para que esto funcione debemos crear un id al canvas, tal que asi:

<canvas id=”mycanvas”></canvas>

Puede ser el nombre que quieras, además asignarle un tamaño al lienzo no está de más mal.

<canvas id=”mycanvas” width="200" height="100"></canvas>

//Recibimos el elemento canvas

var canvas = document.getElementById('mycanvas');

//Accedo al contexto de '2d' de este canvas, necesario para dibujar

var contexto = canvas.getContext('2d');

//Dibujo en el contexto del canvas

contexto.fillRect(50, 0, 10, 150);

* 1. **– Estructura del código.**

<html>

<head>

<title>Probando canvas</title>

<script>

window.onload = function(){

var elemento = document.getElementById('micanvas');

if (elemento && elemento.getContext) {

var contexto = elemento.getContext('2d');

if (contexto) {

contexto.fillRect(0, 0, 150, 100);

contexto.fillStyle = '#cc0000';

contexto.fillRect(10, 10, 100, 70);

}

}

}

</script>

</head>

<body>

<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>

</body>

</html>

* 1. **– Imágenes de ejecución.**

CODIGO:

<html>

<head>

<title>Probando canvas</title>

<script>

window.onload = function(){

var elemento = document.getElementById('micanvas');

if (elemento && elemento.getContext) {

var contexto = elemento.getContext('2d');

if (contexto) {

contexto.fillRect(0, 0, 150, 100);

contexto.fillStyle = '#cc0000';

contexto.fillRect(10, 10, 100, 70);

}

}

}

</script>

</head>

<body>

<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>

</body>

</html>

RESULTADO:



1. CONCLUSIONES

En conclusión, podemos crear páginas web dinámicas con variedades de funciones que permitan resolver problemas, usando los lenguajes explicados anteriormente.

REFERENCIAS

1. <https://translate.google.com/?hl=es>
2. <https://www.arsys.es/blog/programacion/diseno-web/que-es-canvas/>
3. <https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Qu%C3%A9_es_JavaScript>
4. <https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/JavaScript_basics>
5. <https://codigofacilito.com/articulos/que-es-html>