



UNIVERSIDAD DEL SUR

PROGRAMACION WEB

Inv.: HTML, CSS, JavaScript

Juan de Dios Pérez Collazo

4to cuatrimestre grupo A

Asesor: Ing. Jhonny Enrique Loria Vásquez

Notas del autor

Juan de Dios Pérez Collazo, Facultad de Ingeniería en Sistemas Computacionales,
Universidad del Sur

La correspondencia relacionada con esta investigación debe ser dirigido a Juan de Dios Pérez
Collazo

jlchillax@gmail.com

Universidad del Sur, Cancún-Quintana Roo, enero de 2023.

INDICE

¿Qué es el HTML?	3
Historia del HTML.....	3
¿Cómo funciona el HTML?	3
¿Qué es una etiqueta HTML?.....	4
Html5.....	5
Definición de estilos CSS	5
¿Cuáles son los 3 tipos de estilos CSS?.....	5
1. Interno	5
2. Externo	6
3. Inline	6
¿Qué es JavaScript?.....	6
¿Para qué se utiliza JavaScript?.....	7
¿Cómo funciona JavaScript?	7
Motor de JavaScript	8
JavaScript del lado del cliente	8
JavaScript del lado del servidor	8
¿Qué son las bibliotecas de JavaScript?	9
Visualización de datos	9
Manipulación de DOM	9
Formularios	9
Funciones matemáticas y de texto	9
¿Qué son los marcos de JavaScript?	9
Desarrollo de aplicaciones web y móviles	10
Desarrollo web con buena capacidad de respuesta	10
Desarrollo de aplicaciones del lado del servidor	10

¿Qué es el HTML?

HTML son las siglas en inglés de HyperText Markup Language, que significa Lenguaje de Marcado de HiperTexto. Se llama así al lenguaje de programación empleado en la elaboración de páginas Web, y que sirve como estándar de referencia para la codificación y estructuración de las mismas, a través de un código del mismo nombre (html).

El W3C o World Wide Web Consortium, organización que se dedica a la estandarización de los parámetros de la Red, emplea el html como el más importante lenguaje web, y al cual prácticamente todos los navegadores y exploradores se han adaptado. Es crucial, entonces, para el desarrollo y la expansión de la Red.

Este código opera en base a la diferenciación y ubicación de los distintos elementos que componen la página Web. Así, el código es liviano y meramente textual, pero contiene las direcciones URL de las imágenes, audios, videos y otros contenidos que serán recuperados por el navegador para ensamblar la página, así como las indicaciones para la representación gráfica y estética del texto que se encuentre en la misma.

El HTML ha atravesado diversas versiones y modificaciones, avanzando hacia un modelo de funcionamiento más eficiente y veloz, que obliga a las distintas empresas de software navegador a aplicar consecutivos parches a sus productos para ponerlos al día con cada nueva versión.

Historia del HTML

La primera versión de este código apareció en 1991 y fue escrita por Tim Berners-Lee (TBL), siendo poco más que un diseño inicial de 18 elementos, 13 de los cuales aún se conservan. Se lo consideró poco más que una variante del Lenguaje de Marcado General Estándar (SGML) en uso y consistente en un lenguaje de etiquetas, pero ya en 1993 se reconocieron sus virtudes y su potencia en la estandarización de los lenguajes de la Red.

Entonces se inició el trabajo en Html+, una versión más desarrollada, y en 1995 se obtuvo la tercera versión del estándar: HTML 3.0, cuyas sucesivas actualizaciones (3.1 y 3.2) fueron de gran éxito en los primeros navegadores populares de Netscape y Mosaic. En 1997 aparecería el HTML 4.0, ya como recomendación de la W3C, y finalmente en 2006 la versión más reciente, HTML 5.0.

¿Cómo funciona el HTML?

El lenguaje html opera en base a marcadores escritos (que aparecen entre comillas angulares: <html>), a partir de las cuales se cifra la apariencia y orden interno de una página web, así como los *scripts* o rutinas que operan dentro de ellas. Dicho código fuente hace de ADN de la página Web, diciéndole al navegador de dónde obtener los recursos para su representación y en qué orden, secuencia y modo establecerlos. Y siguiendo el código al pie de la letra, el navegador nos brinda la experiencia de la navegación Web.

Para ello HTML opera en base a un conjunto de componentes, como son:

- Elementos. Los ladrillos básicos del lenguaje HTML, sirven para representar el contenido y sus atributos, así como marcar los parámetros del propio lenguaje, como el punto de inicio de la cadena de comandos y el punto de cierre, o las necesidades especiales.
- Atributos. Las especificaciones respecto a valor, color, posición, etc. de los elementos incorporados en el código. Por lo general consisten en una serie de instrucciones lógicas o numéricas.

¿Qué es una etiqueta HTML?



Una etiqueta HTML tiene un valor específico en el conjunto.

Se llama etiquetas HTML a las **instrucciones mismas con que se compone el código**, es decir, las entradas rodeadas de comillas angulares y que tienen un valor específico en el conjunto, que será luego leído por el programa navegador y traducido en una página Web. Estas etiquetas deben abrir <> y luego cerrar </> cuando ya no se las necesite, en el orden correcto y la secuencia correcta para que no surjan errores.

Algunos ejemplos de etiquetas son:

- **<html>**. Comando que da inicio a la cadena de instrucciones que es el código html y que cierra con </html> al final del documento de programación.
- **<head>**. Que define la cabecera del documento html, que se vincula con el título de la ventana del navegador y contiene subelementos como <title> (título), <link> (para vincular a hojas de estilo o modelos estéticos), <meta> (para referir información sobre la autoría del código), etc.
- ****. Que hace referencia a imágenes y suele estar acompañado de la ruta en donde ésta se encuentra.
- **<a>**. Para introducir hipervínculos tanto internos como externos, mediante el atributo href y la dirección URL adonde el hipervínculo conducirá.
- **<div>**. Para introducir divisiones dentro de la página web.

Html5

HTML5 es la versión más reciente de este lenguaje de programación, **publicada en octubre de 2014 por el consorcio W3C**. Esta modernización del lenguaje reemplazó etiquetas en desuso por versiones actuales y saca provecho a las nuevas tecnologías y necesidades web como formularios, visores, grandes conjuntos de datos, etc.

Definición de estilos CSS

Los estilos CSS, son las **maneras en que aplicamos el código CSS, para darle formato visual a la interfaz de un sitio web**. Estos estilos definen la interacción entre los códigos HTML y CSS del sitio, mediante reglas que nos permiten controlar ese intercambio de información. La cantidad de reglas aplicadas puede ser infinita, ¡el cielo es el límite!

Tal vez todo esto te parezca un poco confuso y es porque aún no tienes bien en claro **qué es el código HTML y CSS**, ¡pues es lo primero que debes investigar antes de empezar a codear!

¿Cuáles son los 3 tipos de estilos CSS?

Existen 3 maneras de aplicar diseño visual al contenido y la interfaz de un sitio. Como mencionamos previamente, **esta elección define la forma en que interactúan el código CSS y el código HTML**.

Dicho esto, procedamos a analizar cómo aplicar estilos CSS en HTML:

1. Interno

El código CSS Interno o Incrustado, se utiliza **para añadir formato a cada página del sitio**, colocando la regla de formato <style> en el encabezado de cada archivo HTML. Analicemos en detalle sus ventajas y desventajas:

Ventajas del código CSS Interno:

- La principal ventaja de este estilo CSS, radica en la **agilidad de diseño cuando aplicamos formato a una sola página dentro del sitio**.
- Nos **permite diseñar en grupos**, en lugar de tener que agregar los mismos atributos de formato a los elementos una y otra vez.

Desventajas del código CSS Interno:

- Cuando tratamos con un **sitio web de varias páginas**, utilizar este estilo CSS es **poco práctico**, ya que debemos abrir cada archivo HTML que represente esas páginas y agregar o cambiar en la información del selector CSS en cada encabezado.
- La redacción de código en cada página **puede aumentar el tamaño de la página**.

- Al aumentar el tamaño de la página, también se **incrementa el tiempo de carga del sitio**.

2. Externo

El código CSS Externo, tiene el mismo formato que el CSS Interno, pero en vez de aplicar la regla <style> al encabezado de cada página, **se debe importar una hoja de estilos css y enlazar ese archivo de extensión .css** al mismo. Sabiendo esto, veamos sus ventajas y desventajas:

Ventajas del código CSS Externo:

- Podemos estilizar y realizar **cambios de manera global en todo el sitio**.
- Nos permite realizar **cambios con mayor velocidad**.
- **Mejora nuestro flujo de trabajo**.
- Tener el formato almacenado en un archivo paralelamente, **ayuda a que los motores de búsqueda puedan leer el código HTML con más agilidad y precisión**.
- **Mejora el SEO** del sitio.

Desventajas del código CSS Externo:

- Puede suceder que el navegador demore en reconocer la extensión .css del sitio, **incrementando su tiempo de carga**.

3. Inline

El CSS Inline, nos permite **modificar el formato de cada archivo HTML específicamente**, a diferencia de los estilos CSS Interno y Externo. En este caso, se aplica la regla de formato <style> a cada elemento HTML.

Ventajas del código CSS Inline:

- Su ventaja principal es la utilidad para **realizar cambios rápidos o de prueba** en tu sitio.
- Este estilo CSS es **ideal para cuando quieres modificar el formato de un elemento en particular**, distinto al resto del contenido.

Desventajas del código CSS Inline:

- Es **tedioso e ineficiente** al aplicar reglas de estilo CSS o etiquetas CSS a cada elemento dentro del sitio.
- **Genera desorden en la estructura** del código HTML.

¿Qué es JavaScript?

JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web. Como lenguaje de scripting del lado del servidor, se trata de una de las principales tecnologías de la World Wide Web. Por ejemplo, al navegar por Internet, en cualquier momento en el que vea un carrusel de imágenes, un menú desplegable “click-to-show” (clic

para mostrar), o cambien de manera dinámica los elementos de color en una página web, estará viendo los efectos de JavaScript.

¿Para qué se utiliza JavaScript?

Anteriormente, las páginas web eran estáticas, similares a las páginas de un libro. Una página estática mostraba principalmente información en un diseño fijo y no todo aquello que esperamos de un sitio web moderno. JavaScript surgió como una tecnología del lado del navegador para hacer que las aplicaciones web fueran más dinámicas. Por medio de JavaScript, los navegadores eran capaces de responder a la interacción de los usuarios y cambiar la distribución del contenido en la página web.

A medida que el lenguaje evolucionó, los desarrolladores de JavaScript establecieron bibliotecas, marcos y prácticas de programación y comenzaron a utilizarlo fuera de los navegadores web. En la actualidad, puede utilizar JavaScript para el desarrollo tanto del lado del cliente como del lado del servidor. En las siguientes subsecciones se presentan algunos casos de uso comunes:

Anteriormente, las páginas web eran estáticas, similares a las páginas de un libro. Una página estática mostraba principalmente información en un diseño fijo y no todo aquello que esperamos de un sitio web moderno. JavaScript surgió como una tecnología del lado del navegador para hacer que las aplicaciones web fueran más dinámicas. Por medio de JavaScript, los navegadores eran capaces de responder a la interacción de los usuarios y cambiar la distribución del contenido en la página web.

A medida que el lenguaje evolucionó, los desarrolladores de JavaScript establecieron bibliotecas, marcos y prácticas de programación y comenzaron a utilizarlo fuera de los navegadores web. En la actualidad, puede utilizar JavaScript para el desarrollo tanto del lado del cliente como del lado del servidor. En las siguientes subsecciones se presentan algunos casos de uso comunes:

¿Cómo funciona JavaScript?

Todos los lenguajes de programación funcionan mediante la traducción de sintaxis similar a la del inglés a código de máquina, que posteriormente el sistema operativo se encarga de ejecutar. JavaScript se clasifica principalmente como un lenguaje de scripting o interpretado. El código JavaScript es interpretado, es decir, directamente traducido a código de lenguaje de máquina subyacente mediante un motor de JavaScript. En el caso de otros lenguajes de programación, un compilador se encarga de compilar todo el código en código de máquina en un paso diferente. En consecuencia, todos los lenguajes de scripts son lenguajes de programación, pero no todos los lenguajes de programación son lenguajes de scripts.

Motor de JavaScript

Un motor JavaScript es un programa de computación que ejecuta código JavaScript. Los primeros motores de JavaScript eran verdaderos intérpretes, pero todos los motores modernos utilizan el método justo a tiempo o la compilación en tiempo de ejecución para mejorar el rendimiento.

JavaScript del lado del cliente

JavaScript del cliente se refiere a la forma en que JavaScript funciona en el navegador. En este caso, el motor de JavaScript está dentro del código del navegador. Todos los principales navegadores web incluyen sus propios motores de JavaScript incorporados.

Los desarrolladores de aplicaciones web escriben código JavaScript con diferentes funciones asociadas a varios eventos, como hacer clic con el ratón o situar el ratón sobre un elemento. Estas funciones realizan cambios en HTML y CSS.

A continuación, se muestra una perspectiva general del funcionamiento de JavaScript del lado del cliente:

1. El navegador carga una página web cuando recibe una visita.
2. Durante la carga, el navegador convierte la página y todos sus elementos, como los botones, las etiquetas y los cuadros desplegables, en una estructura de datos denominada modelo de objetos del documento (DOM).
3. El motor JavaScript del navegador convierte el código JavaScript en código intermedio. Se trata de un código intermediario entre la sintaxis de JavaScript y la máquina.
4. Diferentes eventos, como hacer clic con el ratón en un botón, desencadenan la ejecución del bloque de código JavaScript asociado. Posteriormente, el motor interpreta el código intermedio y realiza cambios en el DOM.
5. El navegador muestra el nuevo DOM.

JavaScript del lado del servidor

JavaScript del lado del servidor hace referencia al uso del lenguaje de codificación en la lógica del servidor de backend. En este caso, el motor de JavaScript se encuentra directamente en el servidor. Una función de JavaScript del lado del servidor puede acceder a la base de datos, realizar diferentes operaciones lógicas y responder a varios eventos desencadenados por el sistema operativo del servidor. La ventaja principal del scripting del lado del servidor es que admite un alto nivel de personalización de la respuesta del sitio web según sus requisitos, sus derechos de acceso y las solicitudes de información provenientes del sitio web.

¿Qué son las bibliotecas de JavaScript?

Las bibliotecas de JavaScript son colecciones de fragmentos de código escritos previamente que los desarrolladores web pueden volver a utilizar para realizar funciones estándar de JavaScript. El código de la biblioteca de JavaScript se conecta al resto de código del proyecto según sea necesario. Si se concibe el código de una aplicación JavaScript como una casa, las bibliotecas de JavaScript se asemejan a los muebles ya fabricados que los desarrolladores pueden utilizar para mejorar la funcionalidad de la casa.

A continuación, se muestran casos de uso comunes de bibliotecas de JavaScript:

Visualización de datos

La visualización de datos es fundamental para que los usuarios puedan ver estadísticas, por ejemplo, en el panel de administrador o en otros paneles y métricas de rendimiento.

Bibliotecas como Chart.js, ApexCharts y Algolia Places cuentan con funciones integradas que puede utilizar para crear aplicaciones web que muestren datos en gráficos y mapas.

Manipulación de DOM

Puede utilizar bibliotecas, como jQuery y Umbrella JS, para facilitar el desarrollo web, ya que brindan código para funciones estándar de sitios web, como animaciones de menús, galerías de imágenes, botones, lightboxes y mucho más.

Formularios

En todo desarrollo web se utilizan formularios para que los visitantes del sitio web se pongan en contacto con alguien, pidan productos y se inscriban en eventos. Algunas bibliotecas de JavaScript, como wForms, LiveValidation, Validanguage y qForms, simplifican las funciones de formulario, como la validación, el diseño, las condiciones y la transformación.

Funciones matemáticas y de texto

Varias aplicaciones web tienen que resolver ecuaciones matemáticas y procesar fechas, horas y texto. En lugar de enviar todas esas solicitudes al servidor, manejar algunas en el lado del cliente es más eficiente. Para ello, los desarrolladores web utilizan bibliotecas de JavaScript, como Date.js, Sylvester y JavaScript URL Library.

¿Qué son los marcos de JavaScript?

Al igual que las bibliotecas de JavaScript, los marcos de JavaScript son una colección de fragmentos de código escritos previamente que realizan diferentes funciones y se pueden volver a utilizar. Sin embargo, mientras que las bibliotecas de JavaScript son herramientas especializadas para el uso local, los marcos de JavaScript son un conjunto de herramientas completo que ayudan a dar forma y organizar cualquier aplicación web. Si piensa en el código

de aplicación de JavaScript como si fuera una casa, el marco de JavaScript sería el plano utilizado para crear la casa.

Estos son algunos ejemplos de casos de uso de los marcos de JavaScript:

Desarrollo de aplicaciones web y móviles

AngularJS es un marco que simplifica el desarrollo y las pruebas de las aplicaciones web, como las de comercio electrónico, las de tiempo real y las de video. React Native es otro marco que admite el desarrollo de aplicaciones móviles con renderizado nativo para iOS y Android.

Desarrollo web con buena capacidad de respuesta

Los sitios web con capacidad de respuesta ofrecen una experiencia de usuario consistente independientemente del dispositivo. Por ejemplo, las pantallas de móviles y tabletas son más pequeñas que las pantallas de equipos de escritorio y portátiles. Seguramente quiera que el sitio web muestre y presente los datos de manera precisa incluso en pantallas más pequeñas, sin, por ejemplo, recortar los extremos del sitio web. Al utilizar marcos como Bootstrap y Ember.js, los desarrolladores se pueden beneficiar de un diseño con capacidad de respuesta y personalizar fácilmente la apariencia visual de un sitio web en diferentes plataformas.

Desarrollo de aplicaciones del lado del servidor

Node.js es un marco de JavaScript del servidor, de código abierto, que ejecuta el código JavaScript fuera de un navegador. Los desarrolladores utilizan este marco para crear aplicaciones basadas en red del lado del servidor de manera escalable, rápida y fiable. Puede gestionar solicitudes HTTP y secuencias de datos, admitir sistemas de archivos y administrar varios procesos de back-end simultáneamente.