Marco Teórico

HTML

HTML, siglas en inglés de HyperText Markup Language (‘lenguaje de marcado 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.

Elementos:

Los elementos son la estructura básica de HTML. Los elementos tienen dos propiedades básicas: atributos y contenido. Cada atributo y contenido tiene ciertas restricciones para que se considere válido al documento HTML. Un elemento generalmente tiene una etiqueta de inicio y una etiqueta de cierre. Los atributos del elemento están contenidos en la etiqueta de inicio y el contenido está ubicado entre las dos etiquetas.

Atributos:

En su mayoría de los atributos de un elemento son pares nombre-valor, separados por un signo de igual y escritos en la etiqueta de comienzo de un elemento, después del nombre del elemento. El valor puede estar rodeado por comillas dobles o simples, aunque ciertos tipos de valores pueden estar sin comillas en HTML.

Etiquetas:

Las etiquetas HTML son fragmentos de código que nos permiten crear elementos HTML. Los elementos son la estructura básica de HTML. Dichos elementos tienen dos propiedades básicas: atributos y contenido. En este punto cabe destacar que los elementos no son etiquetas.

CSS

CSS (siglas en inglés de Cascading Style Sheets), en español hojas de estilo en cascada, es un lenguaje de diseño gráfico para definir y crear la presentación de un

documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML.
Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles.

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.

Sintaxis:

CSS tiene una sintaxis simple y usa un conjunto de palabras clave en inglés para especificar los nombres de varias propiedades de estilo. Una hoja de estilos consiste en una serie de reglas. Cada regla, o conjunto de reglas consisten en uno o más selectores, y un bloque de declaración.

Bloques de declaraciones:

Un bloque de declaraciones consiste en una lista de declaraciones unidas. Cada declaración consiste en una propiedad, dos puntos (:), y un valor. Si hay muchas declaraciones en un bloque, un punto y coma (;) es insertado para separar cada declaración.
Las propiedades son insertadas en el estándar CSS. Cada propiedad tiene un conjunto de posibles valores. Algunas propiedades afectan a cualquier elemento, otras solo a un grupo particular de elementos. Los valores pueden ser palabras clave, como "center" o "inherit", o valores numéricos, como 200px (200 píxeles) o 80% (80 por ciento del ancho de la ventana). Los valores de colores son especificados por medio de una palabra clave (ej. "red"), de valores hexadecimales (ej. #FF0000, pudiéndose abreviar como #F00), valores RGB en una escala del 0 al 255 (ej. rgb(255, 0, 0)).

Fuentes:

Los estilos CSS puede ser provistos desde varias fuentes. Esas fuentes pueden ser el navegador web, el usuario y el diseñador. La información del diseñador puede ser clasificada de las siguientes formas: inline, media type, importancia, especificidad del selector, orden de reglas, herencia y definición de propiedades. La información de los estilos CSS puede estar en un documento separado o puede estar embebido dentro de un documento HTML. Múltiples hojas de estilos pueden ser importadas al mismo tiempo.

Propiedades de posicionamiento:

Esta es una propiedad muy usada y a veces esto puede complicar algunas cosas al momento de visualizar varias divisiones y no sale en el lugar que se tenía en mente; hay 4 posibles valores para la propiedad position. Si un elemento está posicionado de una manera diferente a static, hay cuatro subpropiedades usadas para especificar posiciones y offsets: top, bottom, left y right.

Static

El valor por defecto a los elementos en el flujo normal.

Relative

El elemento el posicionado en el flujo normal, y luego movido relativamente a su posición normal. Los demás elementos son independientes del elemento movido relativamente.

Absolute

Especifica el posicionamiento absoluto. El elemento es posicionado en relación a su antecesor no estático más cercano.

Fixed

El elemento es posicionado absolutamente en una posición fija de la pantalla aunque el resto del documento se mueva.

PHP

PHP es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web.En la actualidad, la implementación de referencia de PHP es producida por The PHP Group. El código PHP suele ser procesado en un servidor web por un intérprete PHP implementado como un módulo, un daemon o como un ejecutable de interfaz de entrada común (CGI). En un servidor web, el resultado del código PHP interpretado y ejecutado —que puede ser cualquier tipo de datos, como el HTML generado o datos de imágenes binarias— formaría la totalidad o parte de una respuesta HTTP.

PHP puede ser desplegado en la mayoría de los servidores web y en todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. Migrar los servicios basados en PHP hacia las nuevas tecnologías que aparecen, supone un costo a justificar monetariamente (sobre todo, en términos de hardware y rendimiento), por ello, hablar si el número de sitios basados en PHP se ha visto reducido progresivamente en los últimos años, con la aparición de nuevas tecnologías como Node.js, Golang, ASP.NET, etc., o no, supone abrir un debate no carente de falacias y argumentos demagógicos. Es un hecho constatado que, en el mundo empresarial, solo se cambian las cosas cuando va a suponer una ventaja estratégica en el mercado. Las empresas carecen de un sentido que les polaricen los 'sentimientos', emergiendo una pauta de posicionamiento mercantilista a favor o en contra de algo solo por un sentido despectivo hacia ciertas tecnologías, tal como sucede con el mundo linux, windows y/o mac en el ámbito de los usuarios.

Sintaxis:

La sintaxis de PHP, se fundamenta en los principios de programación de C. El intérprete de PHP solo ejecuta el código que se encuentra entre sus delimitadores. Los delimitadores más comunes son <?php para abrir una sección PHP y ?> para cerrarla. El propósito de estos delimitadores es separar el código PHP del resto de código, como por ejemplo el HTML.

Las variables se prefijan con el símbolo del dólar ($) y no es necesario indicar su tipo. Las variables, a diferencia de las funciones, distinguen entre mayúsculas y minúsculas. Las cadenas de caracteres pueden ser encapsuladas tanto en dobles comillas como en comillas simples, aunque en el caso de las primeras, se pueden insertar variables en la cadena directamente, sin necesidad de concatenación.

Los comentarios se pueden escribir bien con dos barras (//) al principio de la línea, o con una almohadilla (#). También permite comentarios multi-línea encapsulados en /* */.

En cuanto a las palabras clave, PHP comparte con la mayoría de otros lenguajes con sintaxis C las condiciones con if, los bucles con for y while y los retornos de funciones. Como es habitual en este tipo de lenguajes, las sentencias deben acabar con punto y coma (;).

$_POST:

PHP $ _POST es una variable súper global de PHP que se utiliza para recopilar datos de formularios después de enviar un formulario HTML con method = "post". $ _POST también se usa ampliamente para pasar variables. Los datos del formulario se envían al archivo especificado en el atributo de acción de la etiqueta <form>. Ésta se envía a través del body del HTTP Request, por lo que no aparece en la URL.

- El método POST no tiene límite de cantidad de información a enviar.
- La información proporcionada no es visible, por lo que se puede enviar información sensible.
- Se puede usar para enviar texto normal así como datos binarios (archivos, imágenes...).
- PHP proporciona el array asociativo $_POST para acceder a la información enviada.

JavaScript:

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

Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas y 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.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Librerías:

En general, las librerías JavaScript son un código reutilizable que a menudo tiene un caso de uso principal. Las librerías proporcionan muchas funcionalidades estándar para que los desarrolladores no tengan que preocuparse por muchas funciones. Así, pueden usar estas para crear páginas web fácilmente utilizando componentes de la interfaz de usuario, utilidades de lenguaje, funciones matemáticas y más. Una librería consta de varias funciones, objetos y métodos, según el idioma. Además, las puedes incluir en un proyecto sin depender de una estructura en particular. Es decir, eres libre de usar una, dos o tantas librerías JavaScript como necesites.

En resumen, se puede decir que las librerías ayudan al programador a reciclar código y sobre todo en no perder tiempo que se podría utilizar en otra cosa que lo requiera como implementando una solución a un problema con demasiada lógica de por medio, por eso mismo es algo que es muy utilizado cuando se requiera trabajar con estos tipos de lenguaje que se debería implementar a menudo.

Funciones:

Las funciones son uno de los bloques de construcción fundamentales en JavaScript. Una función en JavaScript es similar a un procedimiento — un conjunto de instrucciones que realiza una tarea o calcula un valor, pero para que un procedimiento califique como función, debe tomar alguna entrada y devolver una salida donde hay alguna relación obvia entre la entrada y la salida. Para usar una función, debes definirla en algún lugar del ámbito desde el que deseas llamarla.

Una definición de función (también denominada declaración de función o expresión de función) consta de la palabra clave function, seguida de:

- El nombre de la función.

- Una lista de parámetros de la función, entre paréntesis y separados por comas.
- Las declaraciones de JavaScript que definen la función, encerradas entre llaves, { ... }.

Los parámetros primitivos (como un número) se pasan a las funciones por valor; el valor se pasa a la función, pero si la función cambia el valor del parámetro, este cambio no se refleja globalmente ni en la función que llama.

Si pasas un objeto (es decir, un valor no primitivo, como Array o un objeto definido por el usuario) como parámetro y la función cambia las propiedades del objeto, ese cambio es visible fuera de la función.

setInterval():

Ejecuta una función o un fragmento de código de forma repetitiva cada vez que termina el periodo de tiempo determinado. Devuelve un ID de proceso.
Sintaxis:
var procesoID = window.setInterval(función, intervaloDeTiempo[, parámetro1, parámetro2, ... , parámetroN]);
var procesoID = window.setInterval(código, intervaloDeTiempo);
Parámetros
Función:
La function que será ejecutada cada intervaloDeTiempo milisegundos.

Código:
Una sintaxis opcional permite introducir una cadena en lugar de una función, la cual es evaluada y ejecutada cada intervaloDeTiempo milisegundos. Se recomienda evitar esta sintaxis por la misma razón por la que el comando eval() conlleva problemas de seguridad.

Intervalo De Tiempo:
El tiempo en milisegundos (1/1000 de segundo, ms) que se debe esperar entre cada ejecución de la función o del código. Si el valor es menor que 10, se usará 10 en su lugar. El tiempo entre cada ejecución puede ser mayor al que indicamos, para mayor información puedes revisar el siguiente artículo: