

**Programación Web**

**Trabajo Practico 1**

**Nombre y apellido del alumno: Matias Hernan Braga**

**Turno: noche**

**Año: 2022**

1. ***¿Que es un lenguaje marcado y que tipo de lenguaje marcados existen?***

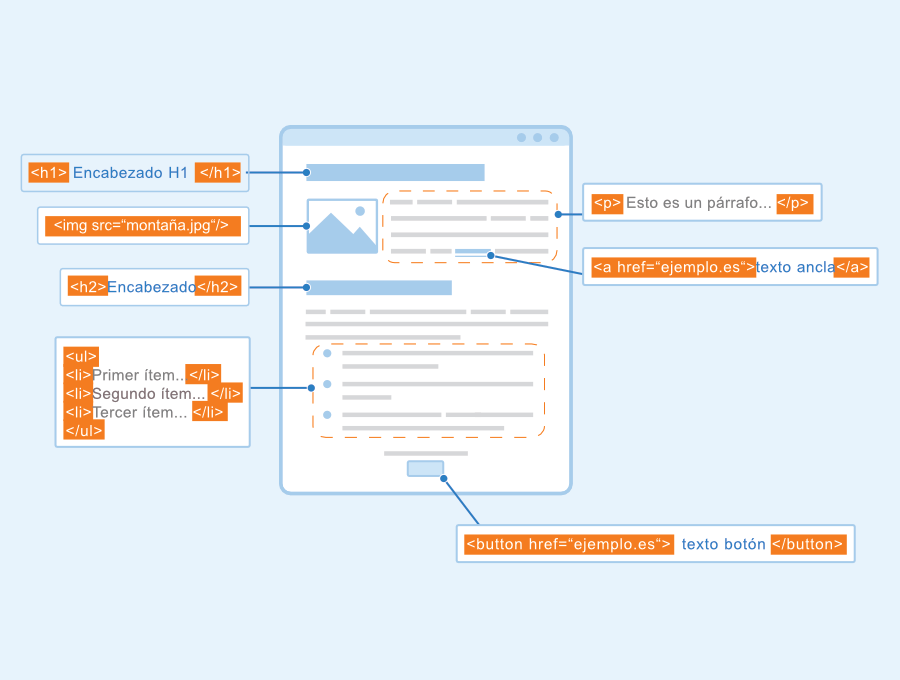
Un lenguaje de marcado o de marcas (Markup Language) es una manera de codificar o redactar un documento. Junto al texto se añade etiquetas (marcas) con información añadida de manera relativa a la estructura del texto o el formato de presentación.

Estos lenguajes dan la posibilidad de hacer claro la estructura de cierto documento y el contenido semántico del mismo. Cuando se observe las siglas en las web que contenga un ML, no es más que un lenguaje de marcado.

Las principales características de los lenguajes de marcado o marcas:

* Utilización del texto plano: al usarse como texto plano cualquier persona es capaz de leer y editar esa información. Los caracteres pueden escribirse en diferentes codificaciones (UTF-8, ASCII, entre otras).
* Es un lenguaje compacto: las etiquetas de marcas se unen con el contenido del mismo.
* Tiene independencia del dispositivo: al ser independiente del dispositivo permite enseñar el contenido.
* Facilidad de procesamiento: admite la creación de lenguajes especializadas según el tipo de documento que se necesita procesar.
* Flexibilidad: nos permite combinar con otros lenguajes.

Se suele diferenciar entre tres clases de lenguajes de marcado, aunque en la práctica pueden combinarse varias clases en un mismo documento.



*Marcado de presentación*

El marcado de presentación es aquel que indica el formato del texto. Este tipo de marcado es útil para maquetar la presentación de un documento para su lectura, pero resulta insuficiente para el procesamiento automático de la información. El marcado de presentación resulta más fácil de elaborar, sobre todo para cantidades pequeñas de información. Sin embargo resulta complicado de mantener o modificar, por lo que su uso se ha ido reduciendo en proyectos grandes en favor de otros tipos de marcado más estructurados.

*Marcado de procedimientos*

El marcado de procedimientos está enfocado hacia la presentación del texto, sin embargo, también es visible para el usuario que edita el texto. El programa que representa el documento debe interpretar el código en el mismo orden en que aparece. Por ejemplo, para formatear un título, debe haber una serie de directivas inmediatamente antes del texto en cuestión, indicándole al software instrucciones tales como centrar, aumentar el tamaño de la fuente, o cambiar a negrita. Inmediatamente después del título deberá haber etiquetas inversas que reviertan estos efectos. En sistemas más avanzados se utilizan macros o pilas que facilitan el trabajo.

Algunos ejemplos de marcado de procedimientos son nroff, troff, TeX. Este tipo de marcado se ha usado extensivamente en aplicaciones de edición profesional, manipulados por tipógrafos calificados, ya que puede llegar a ser extremadamente complejo.

*Marcado descriptivo*

El marcado descriptivo o semántico utiliza etiquetas para describir los fragmentos de texto, pero sin especificar cómo deben ser representados, o en qué orden. Los lenguajes expresamente diseñados para generar marcado descriptivo son el SGML y el XML.

Las etiquetas pueden utilizarse para añadir al contenido cualquier clase de metadatos. Por ejemplo, el estándar Atom, un lenguaje de sindicación, proporciona un método para marcar la hora «actualizada», que es el dato facilitado por el editor de cuándo ha sido modificada por última vez cierta información. El estándar no especifica cómo se debe representar, o siquiera si se debe representar. El software puede emplear este dato de múltiples maneras, incluyendo algunas no previstas por los diseñadores del estándar.

Una de las virtudes del marcado descriptivo es su flexibilidad: los fragmentos de texto se etiquetan tal como son, y no tal como deben aparecer. Estos fragmentos pueden utilizarse para más usos de los previstos inicialmente. Por ejemplo, los hiperenlaces fueron diseñados en un principio para que un usuario que lee el texto los pulse. Sin embargo, los buscadores los emplean para localizar nuevas páginas con información relacionada, o para evaluar la popularidad de determinado sitio web.

El marcado descriptivo también simplifica la tarea de reformatear un texto, debido a que la información del formato está separada del propio contenido. Por ejemplo, un fragmento indicado como cursiva (<i>texto</i>), puede emplearse para marcar énfasis o bien para señalar palabras en otro idioma. Esta ambigüedad, presente en el marcado presentacional y en el procedimental, no puede soslayarse más que con una tediosa revisión a mano. Sin embargo, si ambos casos se hubieran diferenciado descriptivamente con etiquetas distintas, podrían representarse de manera diferente sin esfuerzo.

1. ***¿cómo se conforma la estructura de un documento html. Enumere las principales etiquetas y su descripción?***

Un documento HTML debe tener siempre una estructura inicial, con algunas etiquetas y características obligatorias y otras recomendables. Ese esquema es el siguiente, donde se diferencian tres zonas principales:

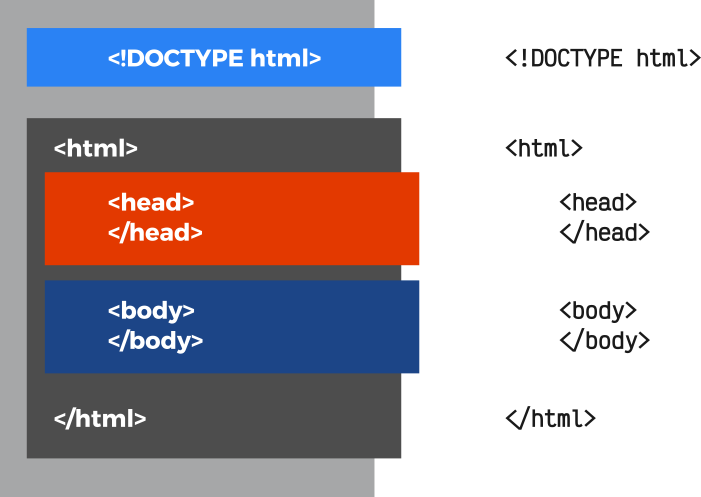
Tipo de documento: En la primera línea del documento HTML debe ir especificado siempre de que tipo de documento HTML se trata. No es absolutamente obligatorio indicarlo (de hecho, hay páginas webs que no lo tienen), pero eso trae algunas consecuencias que veremos más adelante.

Head: En el documento HTML hay una primera sección donde se especificarán todos los metadatos del documento, es decir, la parte «que no se ve» directamente en el navegador de forma visual.

Body: La segunda sección es el cuerpo de la página, donde si aparecen todos los elementos que aparecerán visualmente en el navegador del usuario.

Las dos últimas secciones deben estar dentro de la etiqueta <html>, que es la que debe contener cualquier tipo de etiqueta HTML (salvo el tipo de documento, que debe especificarse antes):

Estructura de un documento HTML



*Etiquetas:*

La gran mayoría están compuestas por:

* 1. Apertura y Cierre

La apertura y el cierre siempre llevan el nombre de la etiqueta HTML que usarás, y son las que sirven para delimitar el contenido de la misma.

* 1. Atributo y valor

El atributo son distintos ingredientes o modificaciones que puedes hacer a la etiqueta. Como por ejemplo colocarle un ID, especificar algún evento (como los casos de onsubmit y onclick), o modificar el aspecto visual (como cambiarle el color a un texto).

El valor es la especificación de estas modificaciones. ¿Qué debe suceder después del onsubmit? ¿Qué color debe tener el texto? ¿Qué ID tendrá la etiqueta?

* 1. Contenido

Es la información que queremos que sea afectada por la etiqueta con sus modificaciones (atributos).

* 1. Cierre

*Etiquetas de base*

<!doctype>

Indica la versión de HTML

<html>

La raíz del archivo

<head>

Engloba los metadatos

<title>

Titulo en la pestaña del navegador

<link>

Enlaza archivos CSS

<style>

Escribe CSS dentro del doc HTML

<Script>

Enlaza o escribe JavaScript en el doc HTML

<meta>

Características sobre la pagina

<body>

Elementos visibles en la pagina

*Etiquetas de texto*

<p>

Crea párrafos

<h1>,<h2>,<h3>…

Estructura los títulos y subtitulos, h1 es el mas importante de la gerarquia

<b>

Texto en negrita

<i>

Texto en cursiva

<u>

Texto subrayado

<a>

Agrega un link

<Strong>

Texto importante de la pagina

<blockquote>

Texto en formato de cita

<br>

Crea un salto de línea

*Etiquitas multimedia*

<img>

Inserta una imagen

<video>

Inserta un imagen

<inframe>

Inserta contenido de otras paginas

<audio>

Inserta un audio

*Etiquetas para estructurar el contenido*

<ul>

Crea una lista sin orden

<ol>

Crea una lista ordenada

<li>

Define los artículos en una lista

<div>

Secciona o agrupa etiquetas

<nav>

Crea el menú de navegación

<header>

Define la cabecera de la pagina

<footer>

Define el pie de pagina

*Etiquetas para crear tablas*

<table>

Crea una tabla

<tr>

Crea filas

<td>

Crea columnas

<th>

Crea el titulo de cada columna

<col>

Especifica las propiedades de las columnas

*Etiquetas de Formularios*

<form>

Crea un formulario

<input>

Campo para introducir datos

<label>

El titulo del input

<textarea>

Campo para introducir mucho texto

<button>

Un boton

1. ***¿Que es un Maquetado? Elabora un documento HTML con las etiquetas básicas.***

La maquetación web es el proceso de planificación, conceptualización y organización de contenido de una página web, es decir, asegura que los textos, las imágenes, los vídeos, como todo lo que quieras añadir a una página web, estén estructurados para crear una buena experiencia de usuario.

Hoy en día, la maquetación web va más allá de la estética para incluir la funcionalidad general de una página web, teniendo en cuenta la arquitectura de la información. Esto también incluye aplicaciones web, aplicaciones móviles y el diseño de interfaz de usuario.

Una buena maquetación web mantiene a los usuarios en el sitio web porque hace que la información sea fácilmente accesible e intuitiva de encontrar. Recuerda que un mal diseño puede hacer que los usuarios abandonen la página web porque no pueden encontrar lo que buscan.

Ejemplo de documento HTML con etiquetas básicas

<html>

<head>

<title>Ejemplo de una pagina en HTML</title>

</head>

<body>

<h1><b>Ejemplo de una pagina en HTML</b></h1>

<td>Esto es un ejemplo de texto</td>

</body>

</html>

1. ***¿Que es un lenguaje de estilos y que podemos controlar con él?***

Un lenguaje de estilo es una forma de controlar definiciones de formato (tipo de fuente, tamaño, color de la fuente, color de fondo, párrafos, etc) de las distintas etiquetas que forman una página HTML. En nuestro caso usamos el CSS.

1. ***¿Que tipo de selectores puede tener un archivo css? Realiza un ejemplo de cada caso***

Selector universal

Se usa para seleccionar todos los elementos dentro del documento html

Ejemplo:

\*{

font-size: 12px;

}

Selector de Etiqueta

Los selectores de etiqueta, también denominados selectores de tipo, son aquellos en los que se indica el nombre de la etiqueta HTML a la que se le asignará una propiedad CSS

Ejemplo:

body {

padding-top: 20px;

padding-left: 20px;

padding-right: 50px;

padding-bottom: 20px;

border-radius: 15px;

width: 300px;

position: relative;

left: 400px;

background-image: url('/Imagenes/FondoDeLibros.jpg');

height: auto;

width: auto;

}

Selectores de ID y de clase

Como ya sabes, los elementos HTML pueden tener clases o identificadores, las cuales se definen dentro de las etiquetas HTML mediante las estructuras class=»nombre-clase» e id=»nombre-id». La utilidad principal de las clases e identificadores es la de posibilitar la selección de elementos concretos de HTML mediante CSS.

Ejemplo:

<html lang="en">

<head>

<style>

#parrafo-1 {

color: black;

}

</style>

</head>

<body>

<p id="parrafo-1" class="mi-parrafo"></p>

</body>

</html

Selector de atributo

Este tipo de selectores permiten apuntar a elementos HTML que contengan un determinado atributo. Se definen, como mínimo, definiendo entre corchetes el nombre del atributo al que queremos señalar, con la nomenclatura [nombre-atributo].

Ejemplo:

<html lang="en">

<head>

<style>

[href]{

text-decoration: none;

color: orange;

}

</style>

</head>

<body>

<a href="https://carontestudio.com/">Enlace 1</a>

<a href="https://carontestudio.com/blog/">Enlace 2</a>

<a href="https://carontestudio.com/diseno-web/diseno-web-vitoria">Enlace 3</a>

</body>

</html>

1. ***Elabora un documento HTML que contenga un menú, un cuerpo y un pie de página, aplicando estilos personalizados***

<html>

<head>

<title>Ejemplo de una pagina en HTML</title>

</head>

<body>

<article>

<h1><b>Ejemplo de una pagina en HTML</b></h1>

<td>Esto es un ejemplo de texto</td>

</articule>

<footer>

<tr>© 2022 Todos los Derechos Reservados a Matias |

</footer>

</body>

</html>

1. ***¿A qué nos referimos cuando hablamos de estilos “inline”? Ejemplifique***

El CSS inline se utiliza para dar estilo a un elemento HTML específico. Para este tipo de estilo CSS, sólo tendrás que añadir el atributo style a cada etiqueta HTML, sin utilizar selectores.

Este tipo de CSS no es realmente recomendable, ya que cada etiqueta HTML debe ser estilizada individualmente. La gestión de tu sitio web puede resultar demasiado difícil si sólo utilizas CSS inline.

Sin embargo, el CSS inline en HTML puede ser útil en algunas situaciones. Por ejemplo, en los casos en los que no tienes acceso a archivos CSS o necesitas aplicar estilos para un solo elemento.

Ejemplo:

<html>

<head>

<title>Ejemplo de una pagina en HTML</title>

</head>

<body style="background-color:black;">

<article>

<h1 style="color:white;padding:30px;">

<b>Ejemplo de una pagina en HTML</b></h1>

<td style="color:white;">Esto es un ejemplo de texto</td>

</articule>

<footer style="color:white;">

<tr>© 2022 Todos los Derechos Reservados a Matias |

</footer>

</body>

</html>

1. ***¿Cómo se puede crear estilos en cascada? Realiza un ejemplo.***

Se puede crear lenguajes en cascada usando CSS

body {

padding-top: 20px;

padding-left: 20px;

padding-right: 50px;

padding-bottom: 20px;

border-radius: 15px;

width: 300px;

position: relative;

left: 400px;

background-image: url('/Imagenes/FondoDeLibros.jpg');

height: auto;

width: auto;

}

h2 {

color: #d31e1e;

font-family: Arial;

display: inline-block;

width: 250px;

border: 10px solid #371c09;

margin-block: auto;

padding-left: 20px;

padding-right: 50px;

padding-bottom: 20px;

background-color: #ffffff;

}

label {

color: #d31e1e;

font-family: Arial;

display: inline-block;

width: 200px;

text-align: left;

font-size: 20px;

font-weight: bold;

}

1. ***Como se crea un tabla en html y que atributos puede tener. Ejemplificar y aplicar diferentes estilos.***

Las tablas más sencillas de HTML se definen con tres etiquetas: <table> para crear la tabla, <tr> para crear cada fila y <td> para crear cada columna.

Ejemplo:

<html>

<head><title>Tabla de ejemplo</title></head>

<body>

<h1>Listado de cursos</h1>

<table>

<tr>

<td><strong>Curso</strong></td>

<td><strong>Horas</strong></td>

<td><strong>Horario</strong></td>

</tr>

<tr>

<td>CSS</td>

<td>20</td>

<td>16:00 - 20:00</td>

</tr>

<tr>

<td>HTML</td>

<td>20</td>

<td>16:00 - 20:00</td>

</tr>

<tr>

<td>Dreamweaver</td>

<td>60</td>

<td>16:00 - 20:00</td>

</tr>

</table>

</body>

</html>