

“Hyper Text markup lenguaje”:

sirve para dar la estructura principal de un sitio web dentro de un navegador

funciona con una estructura de arbol con una raiz principal de la cual deriban todas las partes de nuestro documento.

**DISCLAIMER:**

Todo el código existente en este material es pensando con fines didácticos y no necesariamente proponen la forma óptima de implementarlos en proyectos reales.

### ESTRUCTURA DEL RESUMEN:

El resumen está pensado para brindar un pantallazo general de HTML

No se abarcan todos los temas exhaustivamente.

#### GLOSARIO:

##### **ELEMENTO:**

Cada etiqueta html es un elemento customizable en CSS

##### **SHORTHAND:**

Un shorthand es un atajo de teclado que permite agrupar valores de varias propiedades con el fin de resumir la cantidad de código implementado

##### **ATRIBUTO O PROPIEDAD:**

Una propiedad o un atributo es una característica que un elemento de cierto tipo puede poseer.

##### **VALOR:**

Los valores sirven para darle un comportamiento diferente a cada atributo de cada elemento.

#### SIMBOLOS:



simboliza que la propiedad es un shorthand



Simboliza que la función, método o propiedad se usa para obtener un valor u otorgar un valor.

### ETIQUETAS-TAGS:

* Las etiquetas en HTML sirven para representar los diferentes elementos que podemos tener en nuestro sitio web.
* Hay una lista grande de etiquetas en HTML, las principales son obligatorias y las demás se van usando dependiendo del objetivo de la estructura.
* Las etiquetas pueden contener otras etiquetas dentro

La mayoria constan de 3 partes: apertura, contenido y cierre

<etiqueta>CONTENIDO</etiqueta>

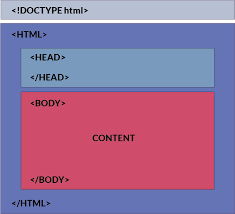
La **apertura** y el **cierre** son practicamente iguales:

Apertura: <NOMBRE>

Cierre: </NOMBRE>

### ETIQUETAS NECESARIAS:

Son imprescindibles para empezar a dar estructura a nuestro sitio web



##### DOCTYPE

indica al explorador web que el contenido debajo es de HTML 5, si se elimina el explorador web entrara en modo quirks “Quirks mode” interpretando todo el código de forma antigua y también generando los errores que existían antes.

##### HTML:

Es la etiqueta raíz de todo documento HTML

##### HEAD:

Es lo primero que se ejecuta al cargar un sitio web

Esta información no es explícitamente mostrada al usuario que ve la pagina

##### BOD**Y:**

Aquí se escribe todo el contenido visualmente explicito para el usuario final:

Titulos, subtitulos, parrafos, links, imágenes,etc…

### ETIQUETAS USUALES:

Hay una lista enorme de etiquetas html, vamos a resumir las que se usan prácticamente siempre al hacer cualquier sitio web

##### TITLE

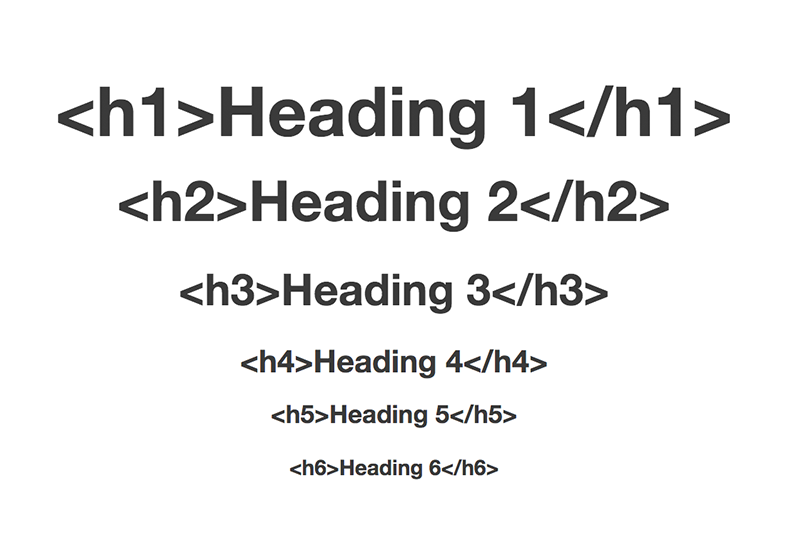
Contiene el título de la página web en la pestaña del navegador



#### HEADINGS:

Son diferentes medidas de títulos o encabezados en nuestra página web

Van desde el H1 al H6



#### ETIQUETAS DE TEXTO:

Tenemos diferentes etiquetas para los diferentes estilos de texto común, los cuales difieren de los headings por su funcionalidad

##### p:

contiene texto común

<p>CONTENIDO</p>

##### b:

<b>CONTENIDO</b>

**Contiene texto en negrita**

##### i:

*Contiene texto en itálica*

<i>CONTENIDO</i>

##### br

sirve para separar ítems; genera un salto de línea justo después de sí misma.

<br>

#### ETIQUETAS DE FUNCIONALIDAD:

##### <link>

especifica la relación entre el documento actual y un recurso externo.

<link>

##### span:

Permite seleccionar puntualmente texto para luego interactuar con el mismo

<span></span>

##### a:

permite tener texto clickeable para visitar un hipervínculo(enlace).

Podemos acceder a un hipervínculo interno o externo

los internos son archivos de nuestro servidor, archivos locales en nuestro pc. los externos son archivos alojados en otro servidor y se ingresa a ellos mediante el protocolo “https” (Protocolo seguro de transferencia de hipertexto)

#### MULTIMEDIA:

Es posible agregar contenido multimedia en nuestras páginas web.

##### img

Sirve para agregar imágenes en la página

<img>

##### Video:

Sirve para agregar videos en la página.

<video></video>

##### Audio:

Sirve para agregar audios en la página.

<audio></audio>

##### form:

Es la estructura semántica para agregar un formulario en una pagina

Sirven para interactuar con el usuario y recopilar información del mismo.

<form></form>

##### label:

coloca texto con un sentido práctico en los campos de un formulario.

Sirve para que el usuario pueda saber que escribir en los campos

<label></label>



# ATRIBUTOS:

Los atributos o también llamados propiedades sirven para agregar características al contenido las etiquetas.

van dentro de la etiqueta de apertura

cada atributo tiene diferentes ***valores***.

<tag atributo="valor"> CONTENIDO </tag>

## Selectores:

Los selectores nos sirven para referenciar elementos.

Cuando tenemos varios títulos, varias imágenes, o varios párrafos y queremos aplicar una acción en concreta a todos ellos, usamos selectores para interactuar con un grupo especifico. A continuación, veremos dos selectores.

##### Identificador:

Permite agregar un identificador único a una etiqueta.

<tag id="id\_name"> CONTENIDO </tag>

##### Clases:

Permite agregar una clase a una etiqueta.

<tag class="class\_name"> CONTENIDO </tag>

### ATRIBUTOS DE INPUTS:

#### Name

le adjudica un nombre al contenido del campo

Cuando nosotros enviemos este formulario aun servidor el mismo va a obtener una llave y un valor:

La llave es el nombre que le pusimos a todo el contenido del campo

y el valor es el contenido del campo

#### placeholder

Indica al usuario que tipo de dato tenemos que colocar dentro de nuestro input (campo)

#### type

caracteriza la información que va a ir en el campo input dependiendo del tipo que elijamos, por defecto es de tipo texto, pero hay muchas…

#### textarea

Crear un área de texto

#### Value:

Contiene el valor que estaría en un campo

## Metodos de envio en HTTP:

Cuando nosotros querramos enviar un formulario o cierta información a otra pagina, correo electrónico, etc… tenemos varias formas de hacerlo.

Dos formas muy conocidas son el método get y el método post

#### Metodo get:

El método get envia los datos en la URL misma, esto tiene el pro de la sencillez y la contra de la vulnerabilidad de los datos

#### Metodo post:

El método post en cambio oculta los datos de la URL y obtiene un punto mas de seguridad

pero en realidad aquí comprobamos que no es mas seguro porque no oculta los datos del todo.

Cuando se trata de seguridad en envio de datos, tenemos que recurrir a agregar lenguajes externos como JavaScript

La unica limitacion es que el metodo get admite solo 255 caracteres de URL por lo que si tenemos una url muy larga debido a un formulario de muchos campos , mejor usar el motodo post.

# METADATOS:

Los metadatos son información que describe datos, en teoría sirven a los servidores con respecto al SEO y también a quien quiere inspeccionar el código de la página, pero el usuario final, no llega a ver este contenido.

La etiqueta <meta> Sirve para trabajar con metadatos en html, va dentro de la sección HEAD

Lleva determinados atributos:

Si recordamos en los formularios anteriores la palabra contraseña se mostraba mal ya que la “ñ” es un carácter especial al igual que los acentos, esto se puede solucionar incluyendo el metadato de codificación universal: ósea:<meta charset="utf-8" >

Tenemos por lo general dos atributos mas que podemos usar en nuestros metadatos, que son el nombre del metadato y el contenido del mismo ,

ejemplos:

# Elementos:

Los elementos en html poseen diferentes caracteristicas, tienen atributos y valores puntuales y tienen una utilidad definida.

## button

Sirve para crear un botón con ciertos atributos y comportamientos:

<button></button>

### Type:

Define la acción del botón:

Por defecto es **submit**.

#### Reset:

Reinicia los campos del formulario

#### Submit

Va a enviar nuestro formulario a la ruta especificada en el campo **action** usando algún método indicado

## LISTAS:

##### ul:

Significa Unordered list Se utiliza para listas no ordenadas, ósea que los elementos de las mismas van a aparecer sin un orden especifico.

<ul></ul>

##### li

<li></li>

##### ol:

significa Ordered list , se utiliza para listas ordenadas.

<ol></ol>

Podemos indicar cual es el valor con el que queremos que empiece nuestra lista ordenada.

Por defecto empieza desde el 1 hacia adelante.

        <li value="50">Elemento 1</li>

        <li>Elemento 2</li>

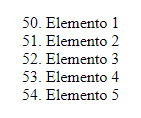
        <li>Elemento 3</li>

        <li>Elemento 4</li>

        <li>Elemento 5</li>

    </ol>

Se vería asi:



es posible anidar listas entre si

## TABLAS:

Una tabla se divide en filas y columnas cuyas pueden tener una cabecera y un pie

Sirven para agrupar datos generalmente de un mismo tipo.

##### Table:

Define una table

##### tr

significa table rows

define filas dentro de nuestra tabla

##### th

significa table header

define una cabecera dentro de una fila o columna

##### td

utiliza para definir elementos estándares dentro de nuestra tabla.

\*Algo importante a notar es que, si agregamos más elementos td de los th que hay, la tabla va a quedar desorganizada\*

\*Los elementos th están centrados a diferencia de los td que están alineados a la izquierda\*

### Thead

Sirve para agrupar elementos que están en la cabecera de nuestras tablas.

### Tbody

Sirve para poder identificar todas las filas que pertenezcan al cuerpo de nuestra tabla.

### Tfoot

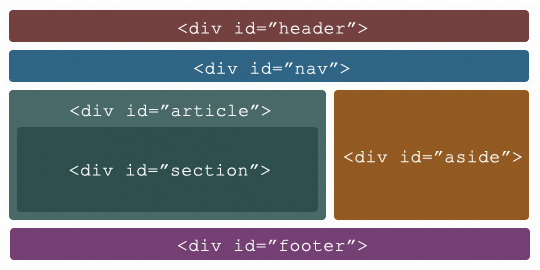
Agrega una celda en el pie de nuestra tabla.

\*El estilo de la tabla no se ven afectados con estas etiquetas, ya que las mismas sirven para que algunos exploradores web puedan habilitar un scroll del contenido separado de la cabecera y del pie de nuestra tabla\*

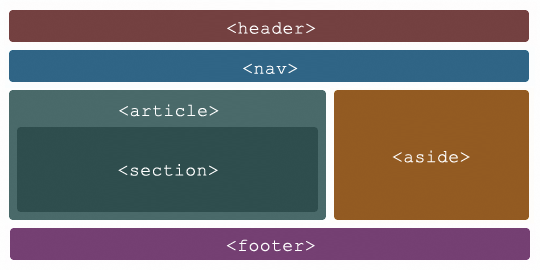
# HTML SEMANTICO:

Para lograr una legibilidad decente a la hora tanto de escribir nuestro código como de leerlo, nosotros hoy día contamos con atributos propios del mismo HTML para las diferentes partes que suelen tener presentes hoy en día las páginas web.

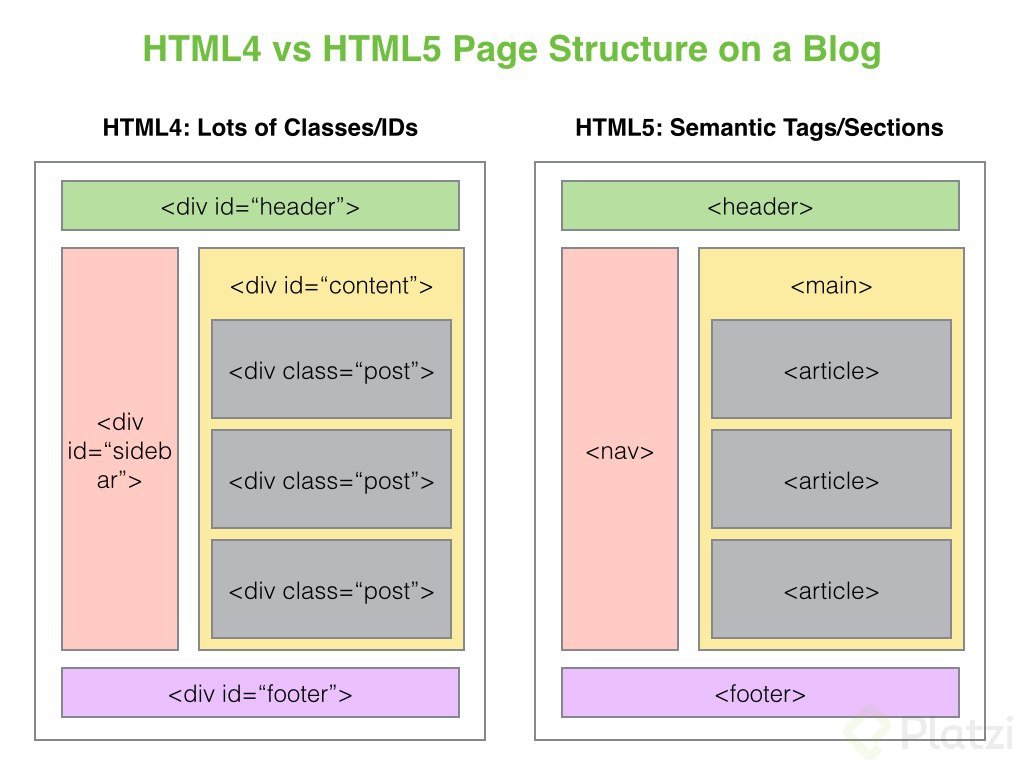
Pasando de esto:



A esto:



Como se puede observar la cantidad de palabras(codigo) que se escriben son menos y además son mas descriptivas y concretas.



La claridad que otorga esta semantica es incomparable con respecto a lo anterior, y por eso su necesidad , ademas trae otros beneficios a nivel social, como seo, posicionamiento mayor escalabilidad, etc.

##### header

Se utiliza para poder agrupar links, imágenes o logos en nuestra página.

<header></header>

##### div

Sirve para dividir la parte anterior y posterior separando el texto en una sección, pero estas etiquetas son usadas eventualmente en códigos viejos de html siendo hoy en dia reemplazadas por las etiquetas de section y article.

<div></div>

##### section

Se utiliza para definir secciones de nuestra aplicación. Donde podemos colocar las etiquetas que nosotros queramos

<section></section>

article

Se utiliza para poder definir contenido completamente independiente.

<article></article>

##### aside

Puede ir dentro de una etiqueta de articulo, y generalmente es lo que va Junto con el mismo, indicando contenido extra.

<aside></aside>

##### footer

Se utiliza para colocar el pie de página de nuestra página web

<footer></footer>

Se pueden tener múltiples etiquetas de footer, por lo general contienen el copyright de la aplicación, información de contacto, del sitio, links para volver al comienzo, e información relacionada con el autor, etc…