

CURSO DE HTML

<HTML>

<STYLE>

<SCRIPT>

PADDING

FUNCTION

MARGIN

<LINK>

TEXT-INDENT



Etiquetas Básicas



Ing. Andrés Felipe Liloy Lopez
IngenierosEnlaWeb.com

E-LIBRO CURSO DE HTML

TABLA DE CONTENIDO

| | |
|--|----|
| EL HTML: INTRODUCCIÓN..... | 3 |
| LAS MARCAS: (<i>Markup – ETIQUETAS – Meta Tags</i>) | 6 |
| ESTRUCTURA DE UNA PÁGINA WEB: | 7 |
| LA CABECERA – ENCABEZADO - DEL DOCUMENTO: | 7 |
| EL CUERPO DEL DOCUMENTO: | 9 |
| FORMATO DEL COLOR: | 10 |
| ORGANIZACIÓN DEL TEXTO | 10 |
| ESTILO LÓGICO: | 10 |
| ESTILO FÍSICO: | 11 |
| DIVISORES DEL TEXTO:..... | 12 |
| ELEMENTO DE UNA LISTA (<i>List Item</i>):..... | 14 |
| MENÚ DESPLEGABLE CON Y | 15 |
| ENLACES (<i>Hipervínculos</i>)..... | 16 |
| ENLACES O HIPERVÍNCULOS LOCALES: | 18 |
| ENLACES INTERNOS (<i>Anclas</i>) EN EL MISMO DOCUMENTO HTML: | 18 |
| ENLACES EXTERNOS:..... | 20 |
| ENLACES DE TEXTO QUE ABREN EN OTRA PESTAÑA: | 20 |
| VARIACIONES DE ESTILO EN LOS HIPERVÍNCULOS..... | 22 |
| BOTONES CON HIPERVÍNCULOS..... | 22 |
| ENLACE A UN ARCHIVO..... | 24 |
| OBJETOS INCRUSTADOS <EMBED> | 24 |
| ENLACE A CORREOS ELECTRÓNICOS | 25 |
| IMÁGENES..... | 25 |
| ETIQUETA | 26 |
| ANCHO (<i>Width</i>) Y ALTO (<i>Height</i>) | 26 |
| TABLAS <TABLE></TABLE>..... | 28 |
| EJEMPLOS USANDO TABLAS | 30 |
| JAVA SCRIPT | 32 |
| EJEMPLOS CON JAVASCRIPT: | 33 |
| JAVASCRIPT CÓMO LENGUAJE DE PROGRAMACIÓN | 34 |
| CSS - CASCADING STYLE SHEETS | 37 |
| PHP- HYPERTEXT PREPROCESSOR | 38 |
| INSERTAR VÍDEO Y SONIDO | 38 |
| VÍDEO DE YOUTUBE (<i>Autoejecutar</i>): | 39 |
| MÚSICA DE FONDO <AUDIO>: | 39 |
| LETREROS <MARQUEE></MARQUEE>: | 40 |
| BOTÓN DE WHATSAPP | 40 |
| BOTONES PARA REDES SOCIALES..... | 42 |
| FORMULARIOS | 44 |



Ing. Andrés Felipe Liloy López
andresliloy@ingenierosenlaweb.com

EL HTML: INTRODUCCIÓN

DEFINICIÓN Y ORÍGENES:

El **HTML** o *HyperText Markup Language* (se traduce como el Lenguaje de Marcas de HiperTexto), puede definirse como un sistema de lenguaje standard que permite integrar en un mismo documento, toda la estructura y código de una "Página Web" reuniendo o agrupando contenido de distinta naturaleza; cómo, texto, imagen, vídeo, animación, sonido y embeber otros objetos de código como por ejemplo JavaScript. Es mucho más sencillo que un lenguaje de programación, ya que está formado por una serie de marcas (denominadas etiquetas o *tags*) que se pueden escribir con el más simple de los procesadores de texto (bloc de notas). Una característica supremamente importante del lenguaje HTML es que permite crear enlaces que se conocen como "*Hipervínculos*" y se utilizan para enlazar o hacer saltos a otras páginas y/o redireccionar a otros contenidos.



Este lenguaje, creado por Tim Berners -Lee, ingeniero del CERN (*Conseil European pour la Recherche Nucleaire*), tiene sus raíces en el SGML (*Standard Generalized Markup Language*) que es un lenguaje de marcas mucho más amplio que sirve de fundamento para definir otros lenguajes de marcas para propósitos más concretos. Así, el HTML es una aplicación del estándar ISO 8879:1986 (SGML), que se formalizó en 1990 con la aparición del (3W) WWW - World Wide Web (*Red Mundial Amplia*).

VERSIONES Y EXTENSIONES:

Las distintas versiones han sido definidas por especificaciones técnicas, originalmente estructuradas mediante DTD (Document Type Definition), las cuales describían los elementos disponibles en el lenguaje y cómo debían usarse.

En los primeros años del desarrollo de la web, la IETF (Internet Engineering Task Force) y posteriormente el W3C (World Wide Web Consortium) lideraron la estandarización de HTML. Sin embargo, durante los años 90, compañías como Netscape y Microsoft incluyeron extensiones propias en sus navegadores, lo que generó fragmentación y problemas de compatibilidad. Estas extensiones buscaban mejorar la funcionalidad de HTML, pero no seguían necesariamente los estándares oficiales.

En mayo de 1996 se publicó HTML 3.2, una versión que intentó unificar las mejoras prácticas adoptadas por los principales actores de la industria. Fue apoyada por empresas como IBM, Microsoft, Netscape, Novell, SoftQuad, Spyglass y Sun Microsystems.

Posteriormente, se publicó HTML 4.01 (1999), una versión más robusta que introdujo mejoras en la separación entre contenido y presentación. No obstante, el desarrollo del lenguaje se estancó durante varios años, hasta el surgimiento de HTML5.

HTML en la actualidad: HTML5

La versión actual y recomendada del lenguaje es HTML5, cuyo desarrollo comenzó formalmente en 2008 y fue finalizado en 2014 por el W3C y WHATWG (Web Hypertext Application Technology Working Group). HTML5 no solo es una evolución del lenguaje, sino una plataforma completa para el desarrollo de aplicaciones web modernas. Entre sus principales características se encuentran:

- Eliminación del uso obligatorio de DTD (aunque se mantiene una declaración simplificada).
- Nuevos elementos semánticos como <article>, <section>, <nav>, <header> y <footer>.
- Soporte nativo para multimedia con etiquetas como <audio> y <video>.
- APIs integradas para almacenamiento local, gráficos 2D (canvas), geolocalización, arrastrar y soltar (drag and drop), entre otras.
- Mejora en la accesibilidad y estructura del contenido.

Para declarar un documento en HTML5, se utiliza una declaración simple al inicio del archivo:

<!DOCTYPE html>

Esta declaración es mucho más sencilla que las versiones anteriores, pero igual de efectiva para que los navegadores interpreten correctamente el documento como HTML5.



Tabla Histórica de Versiones desde HTML 2.0 hasta HTML 5

| Versión | Año | Características principales |
|----------------------|------------|---|
| HTML 2.0 | 1995 | Primer estándar formalizado por la IETF. |
| HTML 3.2 | 1996 | Integración de extensiones populares. |
| HTML 4.01 | 1999 | Separación de contenido y presentación. |
| XHTML 1.0 | 2000 | Versión basada en XML, más estricta. |
| HTML5 | 2014 (W3C) | Versión actual, moderna, y extensible. |
| HTML Living Standard | Desde 2017 | Mantenida por WHATWG, en constante actualización. |

SERVIDORES Y CLIENTES.

Las redes que forman Internet se organizan en **Servidores** y **Clientes**. Los primeros, administrados por expertos en informática, son ordenadores conectados permanentemente a Internet y que hospedan sus distintos recursos (páginas web, programas, etc). Los segundos, los clientes, son ordenadores conectados a servidores, y que pueden ser utilizados por usuarios con menos conocimientos informáticos. Cuando hablemos de Cliente nos referiremos a los programas instalados en este último tipo de ordenadores, en especial nos referiremos al navegador o *browser*.

(HTTP) UN PROTOCOLO "SIN ESTADO".

Para garantizar la comunicación entre los Servidores de alojamiento de contenido y los Clientes de la WWW, se utiliza el Protocolo **HTTP** - **HyperText Transfer Protocol** (Protocolo de Transferencia de Hipertexto). Es un protocolo, es decir, conjunto de reglas y procedimientos para transferir hipertexto.

¿Qué es transferir hipertexto?

El hipertexto es: texto con hipervínculos (enlaces) y este texto suele estar acompañado de imagen, animación, vídeo y sonido. Es decir, todos los recursos que puede contener una página web. Entonces, en HTTP "Protocolo de Transferencia de Hipertexto" permite transferir todos los recursos que contiene una página web y los transmite, vía web.

Se visualiza en el área de dirección del navegador web, al principio de la página, así:

http://

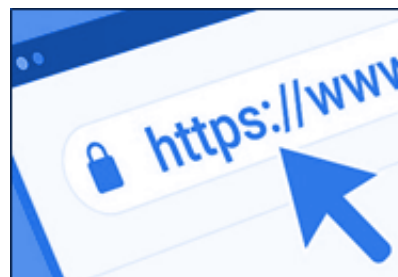
¿Cómo funciona el HTTP?

El Cliente (tu navegador) establece una conexión con el Servidor y le envía una petición, este último la entrega como respuesta en una (página web), y se cierra la comunicación entre ambas partes. Es decir, que, aunque el navegador mantenga el documento cargado con la respuesta o página, la conexión ya no permanece. Por ello se dice que el HTTP es un protocolo de transmisión *sin estado*, porque, no retiene la información sobre la conexión y una vez que la respuesta haya concluido, la comunicación también. Por eso al hacer un clic en otro enlace; se toma como otra petición nueva y si la comunicación se interrumpe; por eso es necesario volver a recargar con un *Refresh* o presionar la tecla [F5].

HTTP y HTTPS

El http sólo, no ofrece seguridad de contenido cifrado y es por eso que, para mayor seguridad, apareció el https (la S, significa seguridad). El https ofrece cifrado de seguridad **SSL/TLS**.

SSL – Secure Sockets Layer (los layer son capas de seguridad) y **TLS – Transport Layer Secure**. Cuando visualizamos en el área de dirección del navegador un candado negro y el protocolo **https://** podemos estar tranquilos que la página web o dirección cuenta con el cifrado y las capas de seguridad activadas.



Para las páginas de los bancos o páginas que soliciten información sensible (ejemplo datos de tarjetas, páginas de pago, etc.; es indispensable y casi que obligatorio, verificar que cuenten con este tipo de protocolo de seguridad.

EJECUCIÓN DE APLICACIONES EN EL SERVIDOR:

PROGRAMAS CGI. Sin embargo, los *estados* entre cliente y servidor debían ser gestionados para permitir enviar información a los servidores o acceder a bases de datos de forma interactiva. Fue, el **NCSA** (*National Center for Supercomputing Applications*), creador del navegador *MOSAIC*, quien desarrolló en sus servidores un interfaz denominado **CGI** (*Common Gateway Interface*). Gracias a este, una serie de programas instalados en el servidor podían interactuar con el cliente. Esto fue motivo suficiente para que se revisara el HTML, incorporándose en su versión 2 los **Formularios** (que veremos más adelante).

EJECUCIÓN DE APLICACIONES EN EL CLIENTE:

PROGRAMAS JAVA. Pero el futuro de Internet parece que pasa por un nuevo tipo de lenguaje de programación semejante al C++ y desarrollado por *SUN MICROSYSTEM*: el **JAVA**. Este es un robusto lenguaje de programación diseñado para crear aplicaciones que se pueden cargar a través de la Red para que posteriormente sean ejecutadas de forma local, es decir, en el propio cliente.

Las aplicaciones (*applets*) desarrolladas en este lenguaje se apoyan en una completa librería estandarizada, y se compilan para obtener un *código intermedio*, muy cercano a las instrucciones de la máquina, que es lo que se envía al cliente. Este incluye un intérprete que convierte el código intermedio al del ordenador con el que esté operando. De esta forma las aplicaciones JAVA son rápidas y pueden funcionar en cualquier ordenador (es decir, que es multiplataforma).

En la actualidad, en lugar de applets, se recomienda utilizar tecnologías web modernas como **JavaScript**, **HTML5** y **CSS3** para crear aplicaciones web interactivas acordes a la W3C.

LA W3C

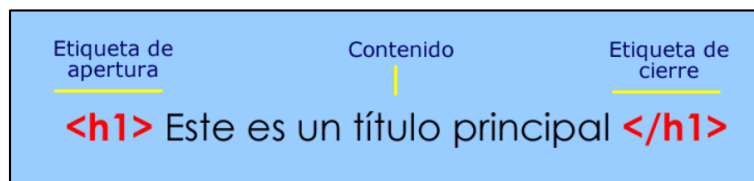
Es la organización internacional que desarrolla estándares y directrices para asegurar el crecimiento de la WWW – World Wide Web; desde su creación en 1994 por Tim Berners Lee, en el Instituto Tecnológico de Massachusetts – MIT, y en colaboración de la CERN (Organización Europea para la Investigación Nuclear) + DARPA (Agencia de Proyectos y Recursos de Avanzada).



LAS MARCAS: (Markup – ETIQUETAS – Meta Tags)

Como se ha indicado, el HTML es un sencillo lenguaje de programación que utiliza *marcas*. Estas consisten en una palabra rodeada por los símbolos: **menor que** ("**<**") y **mayor que** ("**>**") respectivamente. Como, por ejemplo: `<H1>`, `<HTML>`, `<HEAD>`, `<BODY>` `<BR>`.

Estas marcas (*de apertura*) normalmente están emparejadas con otras (*de cierre*). Las marcas de cierre se identifican por tener al principio una barra slash ("**/**") antes de la palabra de la etiqueta y siguiendo con el ejemplo anterior las etiquetas de cierre, serían: `</HTML>`, `</HEAD>` o `</BODY>`.



La etiqueta H1 es un heading (encabezado) de tamaño 1

Por tanto, haremos la siguiente distinción entre los elementos que forman una página web: las **Marcas Simples**, son aquellas que no necesitan estar emparejadas con marcas de cierre; sólo se utilizan; pero no se cierran. ejemplos: (`<BR>`, `<HR>`, etc.).

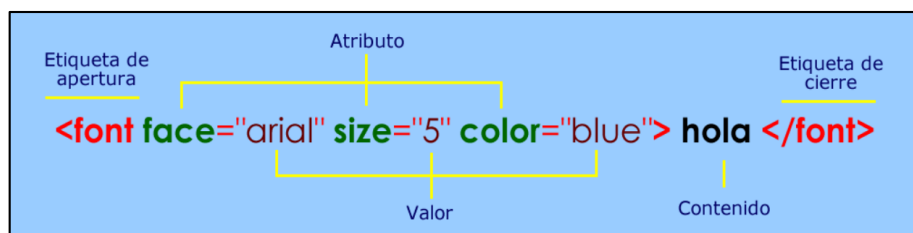
Ejemplo: `<BR>` ← Es salto de línea y no se cierra, sólo se utiliza.

`<HR>` ← Coloca una línea en pantalla.

Las otras marcas, debe estar emparejadas. Son las que utilizan **Delimitadores** o **Contenedores**, en ambos lados, es decir, que necesitan etiqueta de *apertura* y etiqueta de *cierre*, así: `<HTML>.. </HTML>`, `<HEAD> .. </HEAD>`, `<BODY> .. </BODY>`.

Algunas Marcas (en su momento de apertura) pueden ir acompañadas de varios **Atributos**, los cuales permiten complementar el comportamiento de la propia marca.

Ejemplo: `<FONT FACE="arial" SIZE="5" COLOR="blue">` el texto `</FONT>`



La etiqueta FONT hace referencia a la fuente o letra; y se le están aplicando los atributos: face (fuente), size (tamaño) y color

LAS ETIQUETAS NO RESPETAN MAYÚSCULAS, NI MINÚSCULAS

Por último, es importante tener en cuenta, que al escribir las marcas o etiquetas; no hay problema si se escriben en mayúscula, en minúscula o en ambas combinadas. El HTML reconoce los caracteres de la etiqueta y en este caso "*no es sensitivo*"; es decir, que reconoce el carácter en todas sus formas escritas.

Ejemplo: La etiqueta de título se puede escribir sin problema, de cualquiera de estas formas.

`<TITLE> </TITLE>` `<title> </title>` `<TitLe> </tItIe>`

CARACTERES ESPECIALES:

Existen una serie de Caracteres Especiales (*Character Entities* o *Escape Sequences*) cuya representación puede variar de un ordenador a otro o tienen un significado determinado en el HTML. Todos estos caracteres deberán comenzar por el símbolo *ampersand* (&) y terminar con un *punto y coma* (;).

En el caso del castellano, estos caracteres son la **ñ** (ñ), la **diéresis** ü (ü) y los **acentos**, que siempre son agudos (´ ê etc).

Los Caracteres Especiales del HTML son el propio símbolo *ampersand* & (&), el símbolo *menor que* < (<) y el símbolo *mayor que* > (>).

Estos caracteres y otros muchos (como ** ** que deja un espacio en blanco) pueden consultarse en el **APÉNDICE A**.

Por último, hay que añadir que, al contrario de lo que ocurre con las marcas, los Caracteres Especiales son "*caso sensitivo*", es decir, afectan las mayúsculas y minúsculas.



ESTRUCTURA DE UNA PÁGINA WEB:

Un documento HTML tiene la estructura indicada en el siguiente cuadro adjunto:

```
<HTML>
  <HEAD>
    <!-- Cabecera -->
    <TITLE>Título</TITLE>
  </HEAD>
  <BODY>
    Cuerpo del Documento.
  </BODY>
</HTML>
```

Es posible hacer COMENTARIOS dentro del documento sin que lleguen a visualizarse en el navegador. La marca que debemos usar es la siguiente:

<!-- texto en comentario -->

El comentario puede estar escrito con cualquier tipo de caracteres, pero no es conveniente incluir otras marcas en el mismo. Puede hacerse tanto en la cabecera como en el cuerpo del documento.

↑ Copia y pega todo este código dentro de un archivo bloc de notas, creado como .html nombre: **index.html** y al ejecutarlo, estarás viendo el **ejemplo en tu navegador web**.
NOTA: Lo importante es que este archivo NO quede con extensión (.TXT); sino, que debe quedar con extensión la (.HTML).

LA CABECERA – ENCABEZADO - DEL DOCUMENTO:

- **<HEAD></HEAD>**: Estas marcas delimitan la Cabecera de la página web y contiene información sobre el propio documento, también incluye la etiqueta de Título de la página. <TITLE></TITLE>. En algún momento dentro del Head, se incorporan los Style y Script.

Es por esto; que la etiqueta HEAD adquiere suma importancia, debido a que, en ella se incluyen varias etiquetas adicionales, complementos y hasta conexiones que permiten embeber código de otros lenguajes. A continuación, veremos algunas de estas otras etiquetas que se pueden incluir dentro de este HEAD:

- **<TITLE></TITLE>**: Delimitador del título de la página. Es un elemento necesario y especialmente útil para identificar el documento. Aunque no existe un límite, es recomendable no usar títulos con más de 64 caracteres.
- **<BASE>**: Dirección URL virtual del documento. Se emplea para especificar un directorio de base distinto del actual. Se suele usar para hacer pruebas con un documento y asociarle elementos propios de otro, y para poder cambiarlos de ubicación sin que dejen de funcionar los enlaces. Si se omite esta marca, se toma como dirección base la que se ha tomado para acceder al documento. Tiene un solo atributo y es obligatorio:

| Atributo | Función |
|--------------------|--------------------------|
| href ="URL" | Dirección del documento. |

- **<LINK>**: Se utiliza para definir las relaciones entre los distintos documentos, para definir barras de botones automáticamente o para crear áreas estáticas (*banners*).

| Atributo | Función |
|---|---|
| rel = <i>home, toc, index, glossary, copyright, up, next, previous, help, bookmarks, banner.</i> | Define la relación establecida entre el documento HTML y la dirección URL. |
| rev = <i>relación</i>
especifica
el | Es el <i>reverso</i> del anterior, es decir, la relación entre la dirección URL y documento HTML. |
| href ="URL" | Dirección del documento objeto de la relación. |
| title | Especifica el nombre de la relación. |

- **<META>**: Información especial de identificación del documento (llamada *meta-information*) no definida por otras marcas. Debe aparecer entre las primeras líneas del documento HTML. Es especialmente útil para la creación de **documentos dinámicos** desde el lado del cliente (*CLIENT-PULL* de *NETSCAPE*).

| Atributo | Función |
|---|--------------------------------------|
| name = <i>nombre</i> | Nombre asignado. |
| http-equiv = <i>comando HTTP</i> | Comando HTTP que se quiere ejecutar. |
| content = <i>contenido</i> | Contenido del atributo. |

Comandos <META> más usados:

Keywords. Palabras clave utilizadas en el SEO para referenciar la página web con estas palabras clave, digitadas en las búsquedas de Internet.

```
<META HTTP-EQUIV="Keywords" CONTENT="Cursos, Marketing, Colombia">
```

Description. Indica la descripción del sitio utilizada por los motores de búsqueda para encontrar las páginas web relacionadas.

```
<META NAME="Description" CONTENT="Capacitación y cursos de marketing digital" />
```

Refresh. Permite que el navegador web haga un refresh de la página cada 5 seg.

```
<META HTTP-EQUIV="Refresh" CONTENT="5;URL=http://www.hola.es/ini.html">
```

```
<META HTTP-EQUIV="Refresh" CONTENT="0" /> ← Sin redirección.
```

Tildes y Ñ

```
<META CHARSET="UTF-8"> → Permite que el HTML acepte las tildes y las ñ
```

- **<ISINDEX>**: Es una marca generada automáticamente por un programa instalado en un servidor. Sirve para establecer un mecanismo para realizar búsquedas en documentos y servidores. Tiene dos atributos (*href* y *prompt*).
- **<STYLE></STYLE>**: Es un contenedor reservado para el uso futuro de diferentes hojas de estilo. El estilo que se especifique en este lugar sustituirá al que tenga asignado por defecto el navegador, dando más libertad creativa al autor de páginas web.
- **<SCRIPT></SCRIPT>**: Reservado también para el uso futuro de programas ejecutables desde el navegador.

No obstante, *NETSCAPE* e *INTERNET EXPLORER* ya están haciendo uso de este tipo de contenedor al incluir en él el código fuente de aplicaciones desarrolladas en **JAVASCRIPT**, una versión aligerada del lenguaje **JAVA**.

A este respecto, y para aplicaciones desarrolladas en **JAVA**, se utiliza el contenedor **<APPLET></APPLET>**, que es el lugar donde se han de incluir sus códigos y que tienen los siguientes atributos:

Atributo	Función
code = <i>"nombre"</i>	Indica el nombre del archivo a ejecutar.
width = <i>número</i>	Define el ancho del resultado de la aplicación.
height = <i>número</i>	Define la altura del resultado de la aplicación.

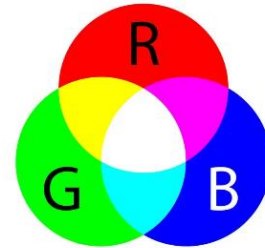
EL CUERPO DEL DOCUMENTO:

- **<BODY></BODY>**: Estas son las marcas que definen el contenido del documento HTML. Todas las que analicemos en las próximas páginas estarán incluidas dentro de este contenedor. De momento veamos sus atributos:

Atributo	Función
background = <i>"URL"</i>	Dirección de la imagen que se usará como fondo del documento. (H3, N1, M1)
bgcolor = <i>"#rrggbb"</i>	Valor RGB del color de fondo. (N1, M1)
bgcolor = <i>nombre de color</i>	Nombre del color usado como fondo. (N2, M1)
bgproperties = <i>fixed</i>	La imagen del fondo no se desplazará ni se multiplicará. (M1)
leftmargin = <i>número</i>	Margen del documento en pixels. (M1)
link = <i>"#rrggbb"</i>	Valor RGB del color usado en los enlaces no visitados. (N1, M1)
link = <i>nombre de color</i>	Nombre del color usado en el texto. (N2, M1)
text = <i>"#rrggbb"</i>	Valor RGB del color usado en el texto. (N1, M1)
text = <i>nombre de color</i>	Nombre del color usado en el texto. (N2, M1)
topmargin = <i>número</i>	Margen superior en pixels. (M1)
vlink = <i>"#rrggbb"</i>	Valor RGB del color usado en los enlaces visitados. (N1, M1)
vlink = <i>nombre de color</i>	Nombre del color usado en los enlaces visitados. (N2, M1)
alink = <i>"#rrggbb"</i>	Valor RGB del color usado en los enlaces activos. (N1, M2)
alink = <i>nombre de color</i>	Nombre del color usado en los enlaces activos. (N2, M2)

FORMATO DEL COLOR:

El formato más usado para definir el color es el **RGB** (*Red, Green, Blue*) que se basa en asignar un “peso” o valor a cada uno de los tres componentes del color: rojo, verde, azul. El código tiene pues el siguiente formato: **#RRGGBB**, y los valores de cada componente son números hexadecimales que van desde los extremos **00** (0%) a **FF** (100%) del color, pasando por niveles intermedios **33** (50%) entre otros: 66, 99, etc. Existen aplicaciones que asignan estos valores automáticamente.



Como ejemplo práctico asignaremos a continuación valores extremos a cada uno de los tres colores básicos, obteniendo los siguientes códigos y combinaciones de colores RGB:

Negro: #000000
Blanco: #FFFFFF
Gris: #333333

Rojo: #FF0000
Verde: #00FF00
Azul: #0000FF

Amarillo: #FFFF00 (rojo + verde)
Púrpura: #FF00FF (rojo + azul)
Celeste: #00FFFF (verde + azul)

Otra forma alternativa de asignar un color es indicando su nombre. El HTML 3.2 recoge una lista de 16 colores. Estos colores, que tienen su origen en el estándar soportado por la paleta VGA de Windows, son los siguientes: AQUA, BLACK, BLUE, FUCHSIA, GRAY, GREEN, LIME, MAROON, NAVY, OLIVE, PURPLE, RED, SILVER, TEAL, WHITE y YELLOW.

NOTA: En la última hoja de este libro, se anexa una tabla con 66 colores RGB para copiar y pegar.

ORGANIZACIÓN DEL TEXTO

ESTILO LÓGICO:

En el Estilo Lógico el control de la apariencia depende de la configuración del navegador. Por tanto, no se puede dar una indicación de cómo se visualiza el texto marcado con los siguientes elementos pues cada navegador utiliza una convención diferente, que además (a veces) puede ser modificada por el usuario. El Estilo Lógico es algo así como el **Libro de Estilo** que han de seguir las páginas *publicadas* en Internet.

- **<Hn></Hn>** (donde n=1,2,3,4,5,6): Cabeceras del texto (*Heading*). Suelen mostrarse en negrita y con tamaños de letra que varían desde muy grande (n=1) hasta muy pequeño (n=6). Fuerza un salto de línea.

Atributo	Función
align =left, right, center	Cabecera centrada (H3, M2)

Ejemplo: **<H1>** Aquí va un título importante de nivel uno **</H1>**

- **<DIV></DIV>**: Determina un bloque de texto o imágenes para ser tratado independientemente del resto. En el futuro esta marca podrá ser utilizada con Hojas de Estilo.

Atributo	Función
align =left, right, center	Alineación del texto. (H3, N1, M1)
nowrap N1, M1)	Suspende la división de palabras. (H3,
clear = left, right, center N1, M1)	Controla el borde de la imagen. (H3,

- **<BLOCKQUOTE></BLOCKQUOTE>**: Suele utilizarse para delimitar una Cita Larga, asignando automáticamente un margen al texto. Estas marcas pueden acumularse para aumentar dicho margen.
- ****: Enfatizado (*Emphasis*), normalmente se conoce como estilo físico de lo que llamamos letra “Cursiva”. Normalmente se utiliza también **<I></I>** de *Itálica*.

- **<CITE></CITE>**: Cita (*Citation*), se corresponde con el estilo físico cursiva.
- ****: Marcado (*Strong Emphasis*), se corresponde con el estilo físico "Negrilla". También para la negrita se utiliza directamente ** ** de Bold.
- **<CODE></CODE>**: Código (*Code*), normalmente para representar un elemento HTML. Suele mostrarse en un tipo de letra fijo y pequeño.
- **<SAMP></SAMP>**: Ejemplo (*Sample*). Se suele mostrar con un tipo fijo de pequeño tamaño.
- **<STRIKE></STRIKE>**: Normalmente se muestra como texto tachado. Normalmente se utiliza **<S> </S>** de Strikeout
- **<KBD></KBD>**: Carácter de teclado (*Keyboard Input*), se corresponde con el estilo físico <TT>. Normalmente se muestra en negrita con un tipo fijo.
- **<PLAINTEXT></PLAINTEXT>**: Normalmente se muestra con un tipo fijo.
- **<VAR></VAR>**: Variable. Suele mostrarse con un tipo fijo de tamaño pequeño.
- **<DFN></DFN>**: Definición. Normalmente se muestra en cursiva.
- **<ADDRESS></ADDRESS>**: Firmas y Direcciones. Se utiliza para escribir la fecha de finalización o actualización del documento, el nombre del autor y su dirección de correo electrónico. Se trata de una reminiscencia de la primitiva forma de trabajar en Internet, anterior a la masificación y difusión actual, y es por ello un rasgo que tiende a desaparecer pero que demuestra un alto nivel de *netiquette*.

ESTILO FÍSICO:

En el Estilo Físico es el autor del documento quien especifica la apariencia del texto. El autor del documento controla en mayor medida el diseño definitivo de la página. El efecto de estos estilos se acumula.

- ****: Negrita (*Bold*), el texto aparece con un tipo de letra más grueso.
- **<I></I>**: Cursiva (*Italic*), el texto se muestra ligeramente inclinado.
- **<U></U>**: Subrayado (*Underline*), se incluye una línea horizontal bajo cada letra. (N3)
- **<TT></TT>**: Texto de Terminal, se usa un tipo de letra no proporcional.
- **<S></S>**: Tachado (*Strikeout*). (H3, N3)
- ****: Subíndice (*Subscript*). (N2, H3)
- ****: Superíndice (*Superscript*). (N2, H3)
- **<BIG></BIG>**: Tamaño Grande (*Big*) de letra. (N2, H3)
- **<SMALL></SMALL>**: Tamaño Pequeño (*Small*) de letra. (N2, H3)
- **<PRE></PRE>**: Delimita un texto que se quiere mostrar con el formato propio del código fuente. Es decir que respeta el tipo de texto (*font*) proporcional, los espacios y los saltos de línea. Al finalizar un bloque delimitado con estas marcas se incorpora automáticamente un salto de línea.

Atributo	Función
width=número	Indica el tamaño máximo del texto.

- **<CENTER></CENTER>**: Delimitador que hace que el contenido se centre con respecto a los márgenes actuales.
- **<BASEFONT SIZE=n>** (donde $n=1,2,3,4,5,6,7$) Establece el tamaño base de la fuente de letra. En el *Navigator* de *NETSCAPE* el valor por defecto es 3, en el *Internet Explorer* de *MICROSOFT* es 2.
- ****: Delimitador para modificar la fuente de caracteres (*font*).

Atributo	Función
size = <i>n</i> ($n=1,2,3,4,5,6,7$)	Tamaño de la fuente.
size = $\pm n$ ($n=1,2,3,4,5,6,7$)	Incremento a disminución del tamaño de la fuente sobre el establecido en BASEFONT.
color =#rrggbb	Valor RGB del color del texto.
color = <i>nombre</i>	Nombre del color del texto. (M1)
face = <i>"nombre1, nombre2, nombre3"</i>	Establece el nombre de la fuente. Si se especifica más de un nombre se selecciona la primera disponible. (M1)

- **<PRE></PRE>**: Escribe el texto con el pre-formato original o tipo de letra propia del HTML.

ESTILO LÓGICO FRENTE A ESTILO FÍSICO:

Gran parte de la discusión sobre cómo debe diseñarse el World Wide Web, surge de las diferencias entre dos criterios: el europeo y el americano. La concepción europea indica que el medio debe orientarse principalmente hacia el *contenido* (defendiendo el **Estilo Lógico**), mientras que la americana, más comercial, lo concibe como un *fin* (defendiendo el **Estilo Físico**).

Para el sector europeo (abanderado por el *W3 Consortium*) el HTML es un lenguaje de marcas que definen la lógica del documento. De esta forma, es partidario de etiquetar un título como una *cabecera de nivel 1* (estilo lógico <H1></H1>) y que cada navegador lo interprete en función de su configuración, en lugar de especificar el tamaño de la fuente (estilo físico) forzando al navegador a presentarlo de esta forma.

Con el estilo lógico el autor de la página web pierde el control sobre el diseño final de la misma ya que diferentes navegadores con diferentes configuraciones producirán distintas presentaciones de una misma página. Sin embargo, el control pasa al usuario del navegador que lo ha configurado a su gusto. Así, un texto marcado como (*estilo lógico*) puede presentarse en negrita (equivalente a en el *estilo físico*) o en rojo si así decide configurar el usuario su navegador.

DIVISORES DEL TEXTO:

- **<P>**: Divisor de Párrafos (*Paragraph*). Normalmente es considerada como marca simple, pero en el futuro será un delimitador (<P></P>).

Atributo	Función
align = <i>left, right, center</i>	Alineación del párrafo. (H3)

- **
**: Salto de línea (*Line Break*). Al contrario del anterior, pueden acumularse.

Atributo	Función
clear = <i>left, right, all flotante</i>	Permite que el texto borde la <i>imagen</i>

- **<HR>**: Barra horizontal (*Horizontal Rule*) con cierto efecto de relieve.

Atributo	Función
align = <i>left, right, center</i>	Alineación del párrafo
size = <i>número</i>	Controla el grosor de la barra
width = <i>número</i>	Controla la anchura en <i>pixels</i>
width = <i>porcentaje</i>	Controla el porcentaje de pantalla
noshade	Elimina el efecto tridimensional de sombra

Para la demostración: Copia y pega todo este código dentro un documento de bloc de notas, creado como: **web01.html** y al ejecutarlo, estarás viendo el **ejemplo mi primera web, con varias etiquetas** explicadas hasta el momento.

```
<html>
<meta charset="UTF-8">
<HEAD>
<title>Mi Primera Web</title>

<script type="text/javascript">
  alert("Hola, soy un mensaje en Java Script dentro del HEAD, que se muestra antes de cargar la
  página");
</script>

</HEAD>

<BODY bgcolor="#B3E4FD" text="white">

<center>
<h1><font color="yellow">¿Quieres Aprender más sobre Marketing Digital?</font></h1>
<HR size="5" color="#FFFFFF" width="40%">
<br>
<HR color="#0000FF">
<br><br>
<font color="cyan">La importancia del Marketing es vital para sostener la imagen de
una Empresa o Negocio</font>
<br><br>
<EM>Texto con Cursiva</EM>
<br><br>
Nuestro curso anteriormente costaba: <font
color="yellow"><STRIKE>$150.000</STRIKE></font>, ahora lo adquieres por <font
color="yellow">$60.000</font>
</center>

<br><br>
Este es un texto importante alineado a la izquierda por defecto; es decir, que aquí ya no está
centrado.
<br><br>
<PRE>Aquí vemos el texto con el pre-formato o tipo de letra propia del HTML sin tipo de fuente
asignada.</PRE>
<br><br>
<HR size="5" color="red" width="15%">

</BODY>
</HTML>
```

[Ver ejemplo](#)

LISTA MARCADA (*Unordered List* o *Bulleted List*):

- ****: Delimita el contenido de una Lista Marcada.

ELEMENTO DE UNA LISTA (*List Item*):

- ****: Delimita (a partir del HTML 3.0) cada elemento de la lista. Se utiliza en Listas Marcadas, Numeradas, Menús y Directorios. El aspecto del *bullet*, la marca que precede a cada elemento de la lista, depende del **navegador** que estemos usando y del **nivel** de la lista. Sólo **NETSCAPE** cuenta con atributos para variar el aspecto de dicho *bullet*.

Ejemplo:

Tres puntos importantes

```
<UL>
  <li>Elemento 1 de la lista</li>
  <li>Elemento 2 de la lista</li>
  <li>Elemento 3 de la lista</li>
</UL>
```

[Ver el ejemplo](#)

LISTA NUMERADA (*Ordered List*):

- ****: Delimita el contenido de una Lista Numerada. Se trata de un concepto similar al anterior. Son listas precedidas por números de orden que empieza en uno (determinado por el atributo *start*) y crece para cada elemento del mismo nivel.

Atributo	Función
type =A,a,I,i	Estilo de numeración (A letras mayus), (a letras minus), (I en romanos mayus), (i en romanos minus).
start =valor (<i>por defecto 1</i>)	Valor en que inicial de la lista.

Ejemplo:

Numerando aspectos importantes

```
<OL>
  <li>Elemento de la lista</li>
  <li>Elemento de la lista</li>
  <li>Elemento de la lista</li>
</OL>
```

Usando letras

```
<OL type="A">
  <li>Elemento de la lista</li>
  <li>Elemento de la lista</li>
  <li>Elemento de la lista</li>
</OL>
```

En romanos

```
<OL type="I">
  <li>Elemento de la lista</li>
  <li>Elemento de la lista</li>
  <li>Elemento de la lista</li>
</OL>
```

[Ver el ejemplo](#)

LISTA DE DEFINICIONES (Definition List):

- **<DL></DL>**: Con esta apertura se ocupa la primera línea. Cada elemento de la lista queda alineado a la izquierda. Usando la (apertura con **<DT></DT>**) que ocupa la primera línea, y (apertura con **<DD></DD>**) ocupa la primera línea con tabulación.

Existen otros tipos de lista que no son reconocidos por todos los navegadores y los que lo hacen no suelen seguir el mismo criterio, por lo que no conviene usarlas. Estas son las **Listas Planas** o **Menús** (contenidos entre las marcas **<MENU></MENU>**), y los **Directorios** (delimitados por **<DIR></DIR>**) que son listas cuyos elementos, a diferencia de los anteriores, se distribuyen horizontalmente.

MENÚ DESPLEGABLE CON Y

Con las etiquetas anteriormente descritas se suelen hacer los menús desplegables usados en las páginas web. A continuación, encontrarás un ejemplo completo de un menú con opciones desplegables.

Crea un documento desde el bloc de notas, teniendo en cuenta que el archivo no puede tener la extensión .txt; sino la extensión .html

Sugerencia del nombre del archivo: **menu.html**

Copia y pega todo este código dentro del archivo bloc de notas, creado cómo: menu.html y al ejecutarlo, estarás viendo el **ejemplo tipo menú desplegable con submenú** de opciones.

```
<!DOCTYPE html>
<html>
<meta charset="UTF-8">
<head>
<title>Menú con Submenú</title>
<style>
/* Estilos básicos para el menú */
.menu {
  list-style: none;
  padding: 0;
  margin: 0;
  background-color: #f0f0f0;
  overflow: hidden;
}
.menu > li {
  float: left;
}
.menu li a {
  display: block;
  color: black;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
.menu li a:hover {
  background-color: #ddd;
}

/* Estilos para el submenú */
.submenu {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;

  list-style: none; /* ← Esto elimina las viñetas */
  padding: 0; /* ← Elimina sangría */
  margin: 0;
}
```

```

.submenu li {
  float: none;
}

.submenu a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
  text-align: left;
}

.submenu a:hover {
  background-color: #ddd;
}

.menu li:hover .submenu {
  display: block;
}

</style>
</head>

<body>

<ul class="menu">
<li><a href="#">Inicio</a></li>
<li>
  <a href="#">Productos</a>
  <ul class="submenu">
    <li><a href="#">Categoría 1</a></li>
    <li><a href="#">Categoría 2</a></li>
    <li><a href="#">Categoría 3</a></li>
  </ul>
</li>
<li>
  <a href="#">Servicios</a>
  <ul class="submenu">
    <li><a href="#">Servicio A</a></li>
    <li><a href="#">Servicio B</a></li>
  </ul>
</li>
<li><a href="#">Contacto</a></li>
</ul>

</body>
</html>

```

[Ver el ejemplo](#)

ENLACES (*Hipervínculos*)

Un enlace es una referencia a otro documento HTML (o página web) también puede ser un enlace para abrir otro tipo de objeto (archivo de sonido, gráfico, programa, etc.), vinculado por medio de un formato universalmente aceptado. Los enlaces, link o hipervínculos; probablemente son una de las características más importantes del lenguaje HTML y uno de los principales objetivos de interconexión de la WWW (World Wide Web), desde el contexto de su creación.

Sin importar a qué tipo de archivo u objeto, nos queramos vincular por medio del enlace; sólo hay una marca de etiqueta en HTML y es el comando `<A>` `</A>`.

- **`<A>` `</A>`:** A Link (*Un enlace*) sería la explicación de esta etiqueta tan importante; la cual, para su correcto funcionamiento, se le debe definir, el *atributo de referencia* "**href**" que se utiliza en la etiqueta de apertura, en esta se especifica la URL o la ruta correcta de búsqueda del objeto o archivo a hipervincular.

En medio de las etiquetas de apertura y de cierre, se debe colocar el texto o el letrero visible que aparecerá en pantalla subrayado en azul y al cual se le hará clic para activar el enlace o para hacer el salto a la referencia indicada (es decir, que este texto es el llamado hipervínculo).

Ejemplo (Link desde un texto):

```
<A href="https://www.gmail.com">texto del hipervínculo</A>
```

También en reemplazo del texto de hipervínculo, muchas veces utilizamos una imagen como hipervínculo para darle clic. En este caso entre las etiquetas de apertura y de cierre se coloca el código de etiqueta correspondiente a la imagen que contendrá el enlace o vínculo.

Ejemplo (Link desde una imagen):

```
<A href="https://www.eltiempo.com"></A>
```



Esta imagen puede ser estática con extensión (.JPG) (.PNG) o también puede ser una imagen animada con extensión (.GIF) en formato GIF Animado.

Ejemplo (Link desde una imagen animada):

```
<A href="https://www.youtube.com"></A>
```



Así se verían estos tres ejemplos:

```
<html>
<meta charset="UTF-8">
<body>

<br><br>
<A href="https://www.gmail.com">Abril Gmail</A>
<p/>
<A href="https://www.eltiempo.com"></A>
<p/>
<A href="https://www.youtube.com"></A>

<center><p/>
<h3>Mismos enlaces (abriendo en pestaña nueva):</h3>
<p/>
<A href="https://www.gmail.com" target="_blank">Abril Gmail</A>
<p/>
<A href="https://www.eltiempo.com" target="_blank"></A>
<p/>
<A href="https://www.youtube.com" target="_blank"></A>
</center>

</body>
</html>
```

[Ver el ejemplo](#)

ENLACES O HIPERVÍCULOS LOCALES:

Los HREF de referencias locales, pueden ir a buscar URL's con archivos u objetos que se encuentran en la misma **Carpeta raíz** de la página web ***index.html***

La otra opción es que también dentro de la carpeta raíz local; se crean subcarpetas para tener otros tipos de archivos más ordenados.

Es muy usual crear una subcarpeta llamada **/images** para contener todas las imágenes de la página, como se muestra en el cuadro. También se observa otra subcarpeta llamada **/enlaces**

Carpetas de la página

```
raiz/  
  index.html  
  menu.html  
  contacto.html  
raiz/images/  
  logo.jpg  
  producto01.jpg  
  producto02.jpg  
  animacion.gif  
raiz/enlaces/  
  productos.html  
  servicios.html
```

Ejemplo 01: Supongamos que desde el contenido de la página ***index.html*** se quiere tener un hipervínculo de texto que salte a la página de ***contacto.html***

```
<A HREF="contacto.html">Ir a nuestra información de contacto</A>
```

Ejemplo 02: Supongamos que desde el contenido de la página ***index.html*** se quiere tener un hipervínculo de texto que salte a la página de ***servicios.html*** que se encuentra en la subcarpeta **/enlaces**

```
<A HREF=" ../enlaces/servicios.html">Ir a nuestros servicios</A>
```

ENLACES INTERNOS (Anclas) EN EL MISMO DOCUMENTO HTML:

Para facilitar la navegación dentro de un documento con un contenido muy extenso; se pueden hacer hipervínculos dentro del mismo archivo o documento .HTML. A este tipo de enlaces dentro del mismo documento le llamamos saltos o hipervínculos entre "**Anclas**".

Ejemplo 01: Supongamos que dentro del contenido de la página ***colombia.html*** se quiere tener un hipervínculo que salte a ***Límites de Colombia***; que se encuentran mucho más abajo del mismo documento.

Primero: Creamos el ancla donde se encuentran los productos de aseo con un título en tamaño <H2>. Este <H2> es un tamaño de "*Heading2*" y se le agrega el **Atributo id**, en donde se coloca el nombre del ancla, así:

```
<H2 id="limites">Limites de Colombia</H2>
```

Segundo: Dentro del contenido de la misma página ***colombia.html*** se realiza el hipervínculo con el **HREF** dirigido directamente al ancla. El código de la etiqueta o quedaría, así:

```
<A HREF="#limites">Ver los Límites</A>
```

Ejemplo (Anclas en rojo):

```
<HTML>
<meta charset="UTF-8">
<BODY>

<H1>COLOMBIA</H1>

<h2 id="inicio">Aspectos importantes</h2>

<!-- Los enlaces para saltar a un ancla, inician con el signo # y el nombre del ancla -->

<OL>
  <li>Introducción</li>
  <li><A href="#limites">Límites</A></li>
  <li><A href="#sanandres">Archipiélago de San Andrés</A></li>
  <li><A href="#poblacion">Población</A></li>
</OL>

<p>Colombia, oficialmente República de Colombia, es un país soberano situado en la
región noroccidental de América del Sur.<br>
Se constituye en un Estado unitario, social y democrático de derecho, cuya forma de
gobierno es de república presidencialista; con dos cámaras legislativas (Cámara
de Representantes y Senado), cuyos miembros son elegidos por sufragio
directo, voluntario y secreto, con cuatro años de mandato.<br>

<br>

<!-- <strong></strong> otra forma de colocar negrilla distinta a <b></b> -->

Su ciudad más poblada es la Capital <strong>Bogotá</strong>. Es una república
organizada políticamente en treinta y dos departamentos descentralizados y el
Distrito Capital Bogotá, sede del Gobierno Nacional.</p>

<p> Incluyendo la isla de Malpelo, el cayo Roncador y quita sueño, el banco Serrana y
los archipiélagos de San Andrés, Providencia y Santa Catalina, el país abarca
una superficie terrestre de 1 141 748 km² y un dominio marítimo de 928 660
km², por lo que es el vigesimoquinto país más grande del mundo y el séptimo
más grande de América.</p>

<!--En el documento también se están utilizando varias imágenes -->

<p> </p>

<p>Reclama como mar territorial el área hasta las 12 millas náuticas de distancia,
manteniendo un diferendo limítrofe al respecto marítimo con Venezuela y
Nicaragua. </p>

<!-- En el documento se reconocen las anclas marcadas en rojo -->

<h2 id="sanandres">Archipiélago de San Andrés</h2>

<p>Hacia Nicaragua Colombia reclama como propios los Archipiélagos de San Andrés,
Providencia y Santa Catalina, siendo estas islas un destino predilecto por miles
de turistas nacionales e internacionales; por poseer el mar de los siete
colores.</p>

<p> </p>

<p><a href="#inicio">Volver al principio</a> </p>

<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>

<h2 id="limites">Límites de Colombia</h2>

<p>Limita al norte, con el océano Atlántico; al este, con Venezuela y Brasil; al sur con
Perú y Ecuador; y al oeste, con el océano Pacífico y Panamá.
Tiene costas en el océano Pacífico y acceso al Atlántico a través del mar Caribe, donde
posee diversas islas como el archipiélago de San Andrés, Providencia y Santa
Catalina. </p>

<p> </p>

<p><a href="#inicio">Volver al principio</a> </p>
```

```

<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>

<h2 id="poblacion">Población</h2>

<p>Es el vigesimoséptimo país más poblado del mundo, con una población de 53 millones de habitantes, además es la segunda nación con más hispanohablantes, después de México.

<p>Posee una población multicultural, la cual refleja la influencia de la colonización europea a gran escala, pueblos nativos y mano de obra africana, con oleadas migratorias provenientes de Europa y Oriente Medio durante los siglos XIX y XX.
</p>

<p>

<p> Otros aspectos importantes </p>

<OL type="A">
  <li>Cultura</li>
  <li>Política</li>
  <li>Deporte</li>
</OL>

<p><a href="#inicio">Volver al principio</a> </p>

<BODY>
<HTML>

```

[Ver el ejemplo](#)

Realiza este reto: Supongamos que dentro del contenido de otra de las páginas; por ejemplo, otra página llamada **deporte.html**; En esta página se describen varios deportes; pero en alguna parte, se destaca el deporte futbol (**Aquí se hace un ancla id="futbol"**) y queremos llegar a este contenido desde un enlace de la página principal **colombia.html**. El código o link, quedaría así:

```
<A HREF="deporte.html#futbol">Ir a los Productos de Aseo</A>
```

ENLACES EXTERNOS:

Los enlaces que hemos realizado hasta el momento son vínculos locales dentro de la misma carpeta de la página web. Ahora vamos realizar un hipervínculo a una página externa ubicada en la WWW (World Wide Web) y utilizando el **http://** (Protocolo de transferencia de hipertexto) y cuando tiene la **"s"** es a un servidor seguro **https://**

Ejemplo 01 (Abre link en la misma ventana):

```
<A href="https://www.youtube.com/@HistoryLA">Ir a History Channel</A>
```

ENLACES DE TEXTO QUE ABREN EN OTRA PESTAÑA:

Todos los enlaces o hipervínculos vistos hasta el momento abren o se ejecutan en la misma ventana o pestaña del navegador. Ahora lo que se quiere es que al hacer clic en el hipervínculo; este se ejecute y se abra en una pestaña nueva. Para esto se utiliza el **Atributo Target="_blank"**, el cual se utiliza en la etiqueta de apertura del enlace. Así:

Ejemplo 02 (Abre link en una ventana nueva):

```
<A href="https://www.youtube.com/@HistoryLA" target="_blank">Ver History Channel en pestaña nueva</A>
```

Así quedaría el código para los dos ejemplos anteriores; pero se le están agregando varios atributos, tales como:

- En la apertura del cuerpo del programa (body):
 - bgcolor** → Color de fondo de la página #E4C28F
 - style** → Define el estilo del tipo de fuente en verdana.
- Se utiliza el comando **** para aplicar negrilla a todo el texto.
- Se utiliza el comando **<center></center>** para centrar todo el contenido de la página.
- Se utiliza el atributo **target="_blank"** para que el link se abra en una pestaña nueva.
- Se está mostrando una **imagen** con el logo de history de color **transparente .png**
- La etiqueta **<p/>** genera un salto de párrafo y no necesita ser cerrada, sólo se utiliza y ya!
- La otra etiqueta es **
** que se utiliza para un salto de línea sencillo.
- La etiqueta **<meta charset="UTF-8">** permite que se acepten tildes y ñ.

```
<html>
<meta charset="UTF-8">
<head>

<style type="text/css">
<!--
a:link {
    color: orange;
}
a:visited {
    color: white;
}
a:hover {
    color: yellow;
}
-->
</style>

</head>

<body bgcolor="#E4C28F" style="font-family: verdana">

<strong>
<center>


<p/>
<A href="https://www.youtube.com/@HistoryLA">Ir a History Channel</A>
<p/>
<A href="https://www.youtube.com/@HistoryLA" target="_blank">Ver History Channel (en
    pestaña nueva)</A>

<center>
</strong>

</body>
</html>
```

[Ver el ejemplo](#)

En este ejemplo se utilizan **VARIACIONES DE ESTILO** para los hipervínculos y esto se hace dentro de la etiqueta **<HEAD></HEAD>** para que los enlaces cambien de color en toda la página.

- La etiqueta **<STYLE type="text/css"></STYLE>** estilo tipo texto. **Color en hipervínculos:**
 - a:link** → Color naranja para el **link inicial**.
 - a:visited** → Color blanco cuando el **link ya sea visitado**.
 - a:hover** → Color amarillo cuando pase el **mouse encima** (efecto hover).

VARIACIONES DE ESTILO EN LOS HIPERVÍNCULOS

CAMBIO DE COLOR EN LOS LINK

Es el mismo del ejemplo anterior; sólo que en el **evento a:hover** (al pasar el mouse) se está aplicando color de fondo amarillo.

```
<HEAD>
<style type="text/css">
<!--
a:link {
    color: BLACK;
}
a:visited {
    color: BLUE;
}
a:hover {
    color: RED; background-color:YELLOW;
}
-->
</style>
</HEAD>
```

[Ver el ejemplo](#)

NO SUBRAYAR LOS LINK

No coloca la línea de subrayado que tienen por defecto los link de texto.

```
<HEAD>
<style type="text/css">
<!--
a:link {
    color:BLUE;text-decoration:NONE
}
a:visited {
    color:BLUE;text-decoration:NONE
}
a:hover {
    color:RED;text-decoration:NONE
}
-->
</style>
</HEAD>
```

[Ver el ejemplo](#)

BOTONES CON HIPERVÍNCULOS

En varias ocasiones es muy usual no querer tener un enlace de texto o un enlace desde una imagen; sino, que se desea tener un botón con el hipervínculo. Aquí hay unos ejemplos de botones con hipervínculo:

Ejemplo 01:

```
<input onclick="window.open('http://www.gmail.com', '_blank');" type="button"
value="Ir a correo Gmail" >
```

[Ver el ejemplo](#)

Ir a correo Gmail

Ejemplo 02:

```
<a href="http://www.gmail.com" target="_blank" title="" style="display: inline-block;
padding: 8px 16px; background-color: white; color: rgb(0, 0, 139); border: 1px solid
red; border-radius: 6px; text-decoration: none;"
onmouseover="this.style.backgroundColor='#f0f0f0'"
onmouseout="this.style.backgroundColor='white'">
  Abrir GMAIL
</a>
```

[Ver el ejemplo](#)

Abrir GMAIL

En este botón ya tiene mayor presentación.

Ejemplo 03:

```
<a href="https://mail.google.com"
target="_blank"
title="Abrir Gmail"

style="
display:inline-block;
padding:8px 16px;
background-color:white;
color:darkblue;
border:1px solid red;
border-radius:6px;
text-decoration:none;
font-family:Verdana, Geneva, sans-serif;
font-size:16px;
font-weight:bold;
transition:all .25s ease;
"

onmouseover="
this.style.backgroundColor='#4DA6FF';
this.style.color='white';
this.style.transform='translateY(-4px)';
"

onmouseout="
this.style.backgroundColor='white';
this.style.color='darkblue';
this.style.transform='translateY(0px)';
">
 Abrir GMAIL
</a>
```

[Ver el ejemplo](#)

 **Abrir GMAIL**

Este último botón ya tiene mayor presentación, incluye icono y mayor interactividad al pasar el mouse.

ENLACE A UN ARCHIVO

Desde un documento HTML también se puede hacer referencia a otro tipo de recursos de Internet siguiendo una estructura similar a la anterior. Tan solo cambia el protocolo:

ENLACE A UN ARCHIVO PDF

- **FILE://** O **FILE:** Usado normalmente para hacer referencia a un archivo local en la misma raíz y misma carpeta del archivo html.

Ejemplo:

```
<a href="file:Archivo.pdf">Enlace a un Archivo PDF</a>
```

ABRIR ARCHIVO PDF DENTRO DE LA MISMA PÁGINA

En el ejemplo anterior el archivo .pdf se abre en una pestaña nueva; pero en ciertos momentos también se utiliza mucho que el archivo.pdf se abra empotrado o incrustado dentro de la misma página web y no por aparte; es decir, no en una pestaña ni en una ventana.

Para poder lograrlo se utiliza el comando para incrustar o embeber objetos.

OBJETOS INCRUSTADOS <EMBED>

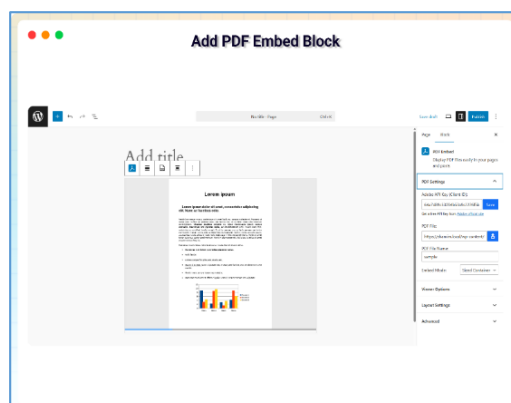
La marca **<EMBED>** permite incrustar objetos dentro de otros y que se ejecuten desde allí. Viene de una tecnología de conexión llamada Windows OLE (*Object Linking and Embedding*).

Atributo	Función
src = <i>"URL"</i>	Dirección del <i>objeto</i> a incorporar.
width = <i>anchura</i>	Anchura del <i>objeto</i> .
height = <i>altura</i>	Altura del <i>objeto</i> .
type = <i>application/pdf</i>	Tipo de archivo.

Ejemplo:

```
<html>
<body bgcolor="#E2EFED">
<center>
<embed src="archivo.pdf" width="70%" height="500px" type="application/pdf">
</center>
</body>
</html>
```

[Ver el Ejemplo](#)



ENLACE A CORREOS ELECTRÓNICOS

- **MAILTO:** Abre una ventana que permite enviar un mensaje a la dirección de **Correo Electrónico** que se escriba por defecto en el código.

Casi siempre se utiliza el mailto: "pero requiere tener configurado el **Microsoft Outlook**".

Ejemplo 01:

```
<a href="mailto:andresliloy@ingenierosenlaweb.com?subject=Asunto  
requiero un apoyo">Enviar correo con asunto por Outlook  
</a>
```

Cuando no se tiene Outlook o se quiera enviar "utilizando la **cuenta de Gmail**". Debes tener previamente abierta una cuenta de correo de Gmail.

Ejemplo 02:

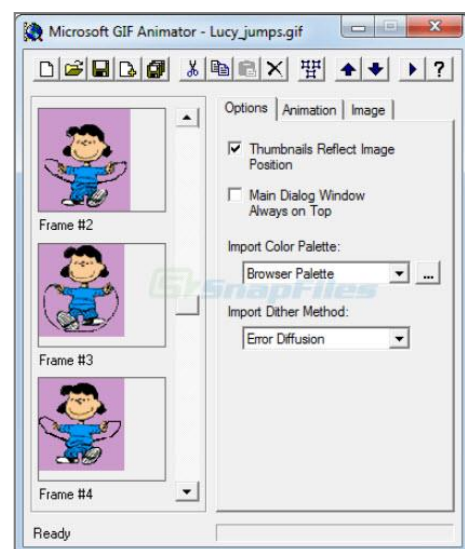
```
<html><body><br><br>  
<a  
href="https://mail.google.com/mail/?view=cm&fs=1&to=andresliloy@ingen  
ierosenlaweb.com&su=☺ Requiero un apoyo" target="_blank">Enviar correo  
desde Gmail  
</a>  
</body></html>
```

[Ver el ejemplo](#)

IMÁGENES

Existen varios formatos gráficos utilizables para el diseño de documentos HTML. Los más comunes son: GIF, PNG, JPEG y sus variantes (JPG, BMP, XMP, XBM).

GIF (*Graphics Interchange Format*), el más generalizado, es un formato comercial (creado por UNISYS para COMPUSERVE) y, por tanto, al no ser de uso público, de incierto futuro como estándar. Está basado en un mecanismo de compresión (LZW) muy sencillo, y utiliza como máximo 256 colores por pixel. Pero puede simular por aproximación (*dithering*) otros colores distintos de la paleta, combinando los valores de los *pixels* colindantes. Por último, permite la creación de imágenes con fondo transparente (sustituyendo uno de los colores por una máscara especial que muestra el fondo tras la imagen), la inclusión de extensiones multimedia (filmación o sonido -MIDI-) y la visualización gradual de la propia imagen. Es el formato apropiado para iconos o imágenes pequeñas con pocos colores. El formato GIF también puede contener movimiento y es así que también son muy conocidos los archivos denominados GIF Animados.



PNG (*Portable Network Graphic*), el formato de imagen PNG se lanzó en 1995. Al experto en TI Oliver Fromme se le ocurrió el nombre de PING, que más tarde se abrevió a PNG.

Los archivos PNG son la evolución del formato GIF, que ya cumplía ocho años cuando se lanzaron los PNG. Los archivos GIF presentaban algunos inconvenientes, como la necesidad de tener una licencia de patente y su limitada gama de tan solo 256 colores, que no se adaptaba a la constante mejora de la resolución de las pantallas de ordenador. Para evitar estos problemas, los archivos PNG se crearon sin patente y con una paleta de colores mucho más variada. Los PNG pueden ser archivos de imagen con fondo transparente y dependiendo el editor de imagen que se trabaje, también puede tener la versatilidad de manejar capas o layer y puede incluir formato vectorial. Herramientas como **Adobe Photoshop** y **Adobe Illustrator**, permiten crear este tipo de imágenes PNG



con las características ya descritas.

JPEG (*Joint Photographic Experts Group*), o su abreviatura **JPG** es de dominio público. Es un formato con mayor nivel de compresión haciendo que los archivos ocupen menos espacio y con una muy buena resolución que permite utilizar hasta 16'7 millones de colores proporcionando imágenes de muy buena calidad. Es el formato más difundido y utilizado para almacenar fotos e imágenes capturadas por cámaras fotográficas y celulares o dispositivos móviles.



**ETIQUETA ** Es la marca que se utiliza para incluir una imagen en un documento HTML. Es una marca simple que no se cierra; sólo se utiliza y ya!

Atributo	Función
src ="URL"	Dirección de la imagen. Es el único atributo obligatorio.
align =top, middle, bottom	Alineación vertical de la imagen con respecto a la línea de texto en la que se encuentra.
align =left, right	Alineación horizontal de la imagen. La inclusión de este atributo da lugar a una imagen flotante.
alt ="texto"	Texto que se mostrará como alternativa cuando ésta no se visualice.
border =tamaño	Grosor del borde de la imagen. (N1, M1)

ANCHO (*Width*) Y ALTO (*Height*):

width =ancho	Anchura de la imagen. (H3, N1, M1)
height =alto	Altura de la imagen. (H3, N1, M1)

La etiqueta **** se puede utilizar y combinar con los atributos anteriores, así:

Ejemplo 01:

```

```



Ejemplo 02: Imagen con borde especial

```

```



Ejemplo 03: Imagen con borde especial

```
style="padding: 3px 10px; border: PowderBlue 5px solid; border-radius: 20px;"
```



Ejemplo 04: Imagen con el borde sencillo

```
style="border:1px solid black;"
```



*Demostración: Copia y pega todo este código dentro del archivo bloc de notas, creado cómo: **web02.html** y al ejecutarlo, estarás viendo el **ejemplo segunda web, con varias etiquetas** explicadas hasta el momento.

```
<HTML>
<HEAD>
<title>Mi Segunda Web</title>

/* Ajusta el ancho de la pantalla para dispositivos móviles con escala sin zoom */
<meta name="viewport" content="width=device-width, initial-scale=1">

/* Le indica al navegador que reciba el formato utf-8 el cual acepta la ñ, tildes y símbolos */
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

/* Los Keywords son palabras clave, que sirven para el SEO en motores de búsqueda antiguos */
<meta name="keywords" content="marketing, capacitación, cursos " />

/* Para el SEO en motores de búsqueda más actualizados como Google */
<meta name="description" content="Capacitación y cursos de marketing digital" />

/* Asignar el favicon o imagen en la pestaña del navegador donde se visualiza la página */
<link rel="shortcut icon" href="https://www.archivosiw.com/img.png">

</HEAD>
```

```

<BODY bgcolor="black">
<center>

<h1><font color="yellow">¿Quieres Aprender más sobre Marketing Digital?</font></h1>
<HR size="5" color="#FFFFFF" width="40%">
<br>

/* Marquee es una etiqueta que desplaza un texto pasando de lado a lado en pantalla */
<H2><font color="cyan"><MARQUEE behavior="alternate" scrollamount="15">POR AQUÍ TE
AYUDAMOS "a Capacitarte en este Tema"</MARQUEE></font></H2>

/* Línea */
<HR color="#0000FF">
<br><br>
<font color="white">La importancia del Marketing es vital para sostener la imagen de una
Empresa o Negocio</font>
<br>

/* Imagen */

<br><br>

/* Hipervínculo desde texto */
<a href="https://online-shop.builderall.net/conferenciamk" target="_blank" >Tenemos para ti,
una CONFERENCIA importante</a>
<br><br>

/* Hipervínculo desde imagen */
<a href="https://drive.google.com/drive/folders/10_bZxDvA4MejmbYFhIOqbafSfCw5tmYY"
target="_blank">
<br><br>

</center>
</BODY>
</HTML>

```

[Ver el ejemplo](#)

TABLAS <TABLE></TABLE>

La creación de Tablas se contempló por primera vez en el HTML 2.0. Hasta entonces, la única manera que existía para tabular cosas en los documentos de hipertexto era a través de las marcas <PRE></PRE>. Pero los resultados dejaban mucho que desear.

Las Tablas, por tanto, son elementos creados esencialmente para mejorar el aspecto de las páginas web. Permiten la inclusión de enlaces e imágenes y la posibilidad de crear notas al margen, distribuir el texto en columnas o diseñar formularios.

Debido a las múltiples posibilidades que nos ofrecen estos elementos, se ha incluido una completa guía de ejemplos en el **APÉNDICE B**.

Las marcas que hacen posible las Tablas son las cinco siguientes:

```

<TABLE></TABLE>: Delimitador de la Tabla.
<TR></TR> (Table Row): Delimitador de Filas.
<TD></TD> (Table Data): Delimitador de Celdas.
<TH></TH> (Table Header): Delimitador de Cabeceras.
<CAPTION></CAPTION>: Delimitador del Título de la Tabla.

```

- **<TABLE></TABLE>**: Estas dos marcas delimitan los elementos de una Tabla. Es posible crear tablas dentro de otras tablas.

Atributo	Función
align = <i>left, right, center</i> (M1)	Alineación del texto de la tabla. (H3, M1)
border (N1, M1)	La tabla presenta los bordes. (H3, M1)
border = <i>número</i>	Control del grosor del borde. (N1)
cellspacing = <i>número</i> dos celdas defecto es 2. (N1)	Control del espacio insertado entre dentro de una tabla. Por
cellpadding = <i>número</i> la celda (N1)	Control del espacio entre el borde de y su contenido. Por defecto es 1.
width = <i>"número o porcentaje"</i> relativos del documento) (N1)	Control sobre el ancho de la tabla, en números absolutos (<i>pixels</i>) o (<i>porcentaje</i> sobre el ancho
bgcolor = <i>"#rrggbb"</i>	Color de fondo de la tabla. (N3, M2)

- **<TR></TR>**: Delimitador de una fila. Habrá tantas filas como delimitadores TR. Los atributos especificados aquí se convierten en los atributos por defecto para todas las celdas de la fila.

Atributo	Función
align = <i>left, right, center</i> (N1, M1)	Alineación del texto de la celda. (H3, M1)
valign = <i>top, middle, bottom, baseline</i> todas las celdas de verticalmente con respecto base. (H3, N1, M1)	Alineación vertical del texto. Con el valor <i>baseline</i> se especifica que la fila se alinearán a la misma
bgcolor = <i>"#rrggbb"</i>	Color de fondo de la fila. (N3)

- **<TD></TD>**: Delimitador de una celda. Sólo podrá aparecer dentro de los delimitadores TR. Por defecto estará alineados a la izquierda (**align**=*left*) y en el medio (**valign**=*middle*).

Atributo	Función
align = <i>left, right, center</i> (N1, M1)	Alineación del texto de la celda. (H3, M1)
valign = <i>top, middle, bottom, baseline</i> valor todas las celdas de verticalmente con respecto base. (H3, N1, M1)	Alineación vertical del texto. Con el valor <i>baseline</i> se especifica que la fila se alinearán a la misma
nowrap romperán la misma. (H3, N1)	Las líneas dentro de la celda no se para ajustarse al ancho de
colspan = <i>número</i> ocupa defecto es 1. (H3, N1)	Especifica el número de columnas que la celda. El valor por
rowspan = <i>número</i> ocupa la es 1. (H3, N1)	Especifica el número de filas que celda. El valor por defecto
width = <i>número o porcentaje</i> números (<i>porcentaje</i> sobre	Control sobre el ancho de la celda, en absolutos (<i>pixels</i>) o relativos el ancho de la tabla). (N1)
bgcolor = <i>"#rrggbb"</i>	Color de fondo de la celda. (N3)

- **<TH></TH>**: Delimitador de una cabecera. Es igual que el delimitador TD pero centrado (**align=center**) y en negrita (****).

Atributo	Función
align = <i>left, right, center</i> (H3, N1, M1)	Alineación del texto de la celda. (H3, N1, M1)
valign = <i>top, middle, bottom, baseline</i> valor todas las celdas de verticalmente con respecto base. (H3, N1, M1)	Alineación vertical del texto. Con el valor <i>baseline</i> se especifica que la fila se alinearán a la misma
nowrap romperán la misma. (H3, N1)	Las líneas dentro de la celda no se para ajustarse al ancho de
colspan = <i>número</i> ocupa defecto es 1. (H3, N1)	Especifica el número de columnas que la celda. El valor por
rowspan = <i>número</i> ocupa la es 1. (H3, N1)	Especifica el número de filas que celda. El valor por defecto
width = <i>número o porcentaje</i> números (porcentaje sobre	Control sobre el ancho de la celda, en absolutos (<i>pixels</i>) o relativos el ancho de la tabla). (N1)
bgcolor = <i>"#rrggbb"</i>	Color de fondo de la cabecera. (N3)

EJEMPLOS USANDO TABLAS

En este primer ejemplo:

- Tres filas **<TR></TR>**
- Tres columnas en negrilla **<TH></TH>** en la primera fila.
- Tres columnas **<TD><TD>** en las otras dos filas.

Ejemplo 01:

```
<TABLE BORDER="1">
  <TR>
    <TH>Enero</TH>
    <TH>Febrero</TH>
    <TH>Marzo</TH>
  </TR>
  <TR>
    <TD>Celda 1</TD>
    <TD>Celda 2</TD>
    <TD>Otra celda<br>Celda 3</TD>
  </TR>
  <TR>
    <TD>Celda 4</TD>
    <TD>y esta<br>es la celda 5</TD>
    <TD>Celda 6</TD>
  </TR>
</TABLE>
```

[Ver el ejemplo](#)

Enero	Febrero	Marzo
Celda 1	Celda 2	Otra celda Celda 3
Celda 4	y esta es la celda 5	Celda 6

Ejemplo 02:

```
<TABLE BORDER="1">
  <TR bgcolor="yellow">
    <TH>Enero</TH>
    <TH>Febrero</TH>
    <TH>Marzo</TH>
  </TR>
  <TR>
    <TD><font color="red">Celda 1</font></TD>
    <TD bgcolor="cyan">Celda 2</TD>
    <TD>Otra celda<br>Celda 3</TD>
  </TR>
  <TR>
    <TD><b>Celda 4</b></TD>
    <TD>y esta<br>es la celda 5</TD>
    <TD bgcolor="#ffcc66" align="center">Celda 6</TD>
  </TR>
</TABLE>
```

[Ver el ejemplo](#)

Enero	Febrero	Marzo
Celda 1	Celda 2	Otra celda Celda 3
Celda 4	y esta es la celda 5	Celda 6

Ejemplo 03:

```
<TABLE border="2">
  <TR>
    <TH>Enero</TH>
    <TH>Febrero</TH>
    <TH rowspan="2" align="center" bgcolor="cyan">Marzo</TH>
  </TR>
  <TR>
    <TD>Celda 1</TD>
    <TD>Celda 2</TD>
  </TR>
  <TR>
    <TD colspan="3" align="center" bgcolor="yellow">Texto combinado</TD>
  </TR>
</TABLE>
```

[Ver el ejemplo](#)

Enero	Febrero	Marzo
Celda 1	Celda 2	
Texto combinado		

JAVA SCRIPT

JavaScript ó (**JS**) es un *Lenguaje de Programación* para web, que se utiliza principalmente para acompañar a las páginas y/o aplicaciones web de interactividad, dinamismo y lógica. El lenguaje JavaScript es muy versátil y de propósito general en donde también se conectan otras herramientas desde sus líneas de comando, se ejecuta en el cliente y servidores web, permite hacer aplicaciones de análisis de datos, cálculos matemáticos y mucho más. Es uno de los pilares fundamentales de Internet, junto con el HTML y los CSS "Cascading Style Sheets" hojas de estilo en cascada que veremos en otro apartado de este curso.

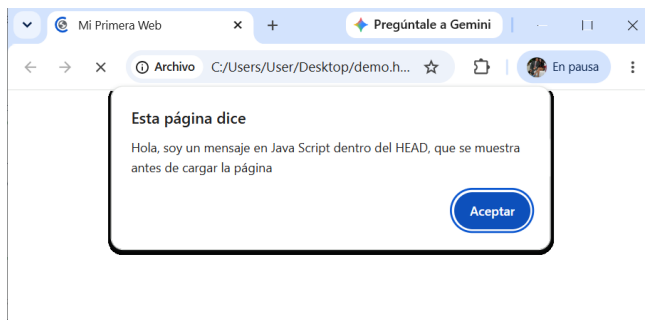
- **<SCRIPT TYPE="TEXT/JAVASCRIPT"></SCRIPT>**: Estas dos marcas delimitan el código JavaScript. Entre ellas sólo deben aparecer instrucciones propias de JavaScript.

Ejemplo 01:

Estructura para insertar un JavaScript dentro de HTML.

```
<script type="text/javascript">  
alert("Hola, soy un mensaje en Java Script dentro del HEAD, para que se muestre  
antes de cargar la página");  
</script>
```

En el ejemplo la instrucción permite ejecutar una ventana de alerta que se abriría antes de cargar la página completa en html. Se observa que el código termina con punto y coma. El botón [Aceptar] sale por defecto.

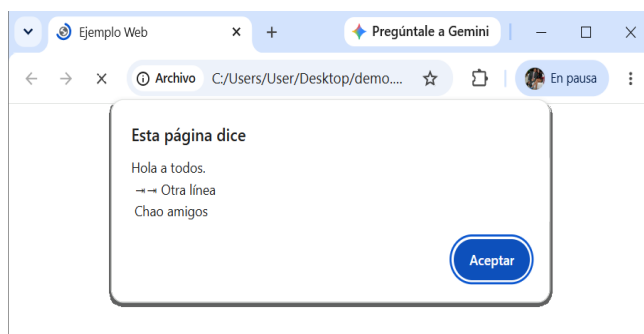


Ejemplo 02:

Alerta utilizando dos atributos. Salto de línea y salto de tabulación.

```
<script type="text/javascript">  
alert("Hola a todos.\n Otra línea\n \t\tChao amigos");  
</script>
```

Atributo	Función
\n	Salto de línea.
\t	Salto de tabulación.



EJEMPLOS CON JAVASCRIPT:

Ejemplo 03:

JavaScript con mensaje de alerta y manejo de tres viñetas aplicadas en los textos de las ventajas del Marketing.

****: Apertura y cierre de las viñetas.

****: La viñeta o puno de viñeta inicial en el texto.

```
<!DOCTYPE html>
<html>
<HEAD>
<title>Ejemplo Web</title>

<script type="text/javascript">
alert("Conceptos de Marketing.\n Unas ventajas.");
</script>

</HEAD>

<BODY>
<center><h1>MARKETING Y TRES VENTAJAS</h1></center>
<HR size="5" color="red" width="40%">
<br>
La importancia del Marketing es vital para sostener la imagen de
una Empresa o Negocio.
<br><br>

<b>Ventajas:</b>
<ul>
  <li> Aumento de visibilidad y ventas.
  <li> Posicionamiento y reputación.
  <li> Optimización de recursos y segmentación.
</ul>

<br>
</BODY>
</HTML>
```

[Ver el ejemplo](#)

Ejemplo 04:

JavaScript que permite llamar dos alerta desde dos botones [Si] y [No], al presionar el botón si sale alerta positiva y al presionar no sale alerta negativa.

```
<html>
<meta charset="UTF-8">
  <head>
    <script language="javascript">
      var ret;
      function returnfunction()
      {
        alert(ret);
      }
    </script>
  </head>
  <body>
    <form>
      <label id="QuestionToAsk" name="QuestionToAsk">¿Los
millennials se daptan a mejor a la tecnología?</label><p />
      <input type="button" value="Yes" name="yes"
onClick="ret=true;returnfunction()" />
      <input type="button" value="No"
onClick="ret=false;returnfunction()" />
    </form>
  </body>
</html>
```

[Ver el ejemplo](#)

JAVASCRIPT CÓMO LENGUAJE DE PROGRAMACIÓN

Demostraciones en JavaScript capturando datos, utilizando variables y realizando cálculos matemáticos o procesos para entender cómo funciona la lógica en su programación.

Ejemplo 01:

Pedir el nombre y el valor de venta de un producto cualquiera. Calcular el valor a pagar, teniendo en cuenta que al valor del producto se le aplicará un descuento del 5% menos y se le incrementará el IVA del 19% correspondiente a Colombia.

Valor del Producto

Información de la Venta

Nombre del Producto

Camisa

Valor del Producto

130000

Calcular

Limpiar

El producto es:

Camisa

Valor inicial:

\$130000

Con el descuento del 5%:

\$123500

IVA del 19%:

\$24700

El valor total a pagar es:

\$148200

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Valor Producto</title>

<script type="text/javascript">

  function valorpro()
  {
    var nompro = document.getElementById("nompro").value;
    var valor_ini = document.getElementById("valor_ini").value;

    var valor_dcto = eval(valor_ini) - (eval(valor_ini) * 0.05);
    var iva = (eval(valor_ini) * 0.19);
    var valor_total = eval(valor_dcto) + (eval(valor_ini) * 0.19);

    document.getElementById('print1').innerHTML = "El producto es: " + nompro;
    document.getElementById('print2').innerHTML = "Valor inicial: $" + valor_ini;
    document.getElementById('print3').innerHTML = "Con el descuento del 5%: $" + valor_dcto;
    document.getElementById('print4').innerHTML = "IVA del 19%: $" + iva;
    document.getElementById('print5').innerHTML = "El valor total a pagar es: $" + valor_total;
  }

</script>
</head>
```

```

<body>
  <h1 aling ="center">Valor del Producto</h1>
  <form>
    <fieldset style="width: 400px;">
      <legend><b>Información de la Venta</b></legend>
      <label><p>Nombre del Producto</p></label>
      <input type="text" id="nompro"/><p>
      <label>Valor del Producto</label>
      <input type="text" value="0" id="valor_ini"/><p>

      <input type="button" value="Calcular" onclick="valorpro();"/>
      <input type="reset" value="Limpiar"></br>

      <p id="print1">El producto es: </p>
      <p id="print2">Valor inicial: </p>
      <p id="print3">Con el descuento del 5%: </p>
      <p id="print4">IVA del 19%: </p>
      <p id="print5">El valor total a pagar es: </p>
    </fieldset>
  </form>

</body>
</html>

```

[Ver el ejemplo](#)

Ejemplo 02:

Un profesor pregunta cuantos hombres y mujeres hay en el salón de clase y con ese total, desea saber cuál es el porcentaje de hombres y cuál es el porcentaje de mujeres que hay en su grupo de estudiantes.

Total de Estudiantes

Información y Cálculos

¿Cuantos hombres hay?

¿Cuantas mujeres hay?

El total de estudiantes es: 80

Porcentaje de Hombres: 62.5%

Porcentaje de Mujeres: 37.5%

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Porcentaje de Estudiantes</title>

  <script type="text/javascript">
    function numest()
    {

```

```

var num_hom = document.getElementById("num_hom").value;
var num_muj = document.getElementById("num_muj").value;

var total = eval(num_hom) + eval(num_muj);
var porce_hom = (eval(num_hom) * 100) / eval(total);
var porce_muj = (eval(num_muj) * 100) / eval(total);

document.getElementById('print1').innerHTML = "El total de estudiantes es: "+total;
document.getElementById('print2').innerHTML = "Porcentaje de Hombres: "+porce_hom + "%";
document.getElementById('print3').innerHTML = "Porcentaje de Mujeres: "+porce_muj + "%";
}
</script>
</head>
<body>

<h1 aling="center">Total de Estudiantes</h1>
<form>
  <fieldset style="width: 400px;">
    <legend><b>Información y Cálculos</b></legend>
    <label><p>¿Cuántos hombres hay?</p></label>
    <input type="text" value="0" id="num_hom"/><p>
    <label>¿Cuántas mujeres hay?</label>
    <input type="text" value="0" id="num_muj"/><p>

    <input type="button" value="Calcular" onClick="numest();" />
    <input type="reset" value="Limpiar"></br>

    <p id="print1">El total de estudiantes es: </p>
    <p id="print2">Porcentaje de Hombres: </p>
    <p id="print3">Porcentaje de Mujeres: </p>
  </fieldset>
</form>

</body>
</html>

```

[Ver el ejemplo](#)

Ejemplo 03:

Realizar un programa que muestre los números del 0 hasta el número que se le indique.

Lista de Números

Información para el vector

Escriba el número final

El vector de números es: 0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10
- 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 -
24 - 25 - 26 - 27 - 28 - 29 - 30 -

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Realizar Vector</title>

  <script type="text/javascript">

```

```

function vector()
{
    var numMax = document.getElementById("numMax").value;
    var numeros = '';

    for (var i = 0; i <= numMax; i++) {
        numeros = numeros + i.toString() + ' - ';
    };

    document.getElementById('print1').innerHTML = "El vector de números es: " + numeros;
}
</script>
</head>

<body>
    <h1 aling ="center">Lista de Números</h1>
    <form>
        <fieldset style="width: 400px;">
            <legend><b>Información para el vector</b></legend>
            <label><p>Escriba el número final</p>
            <input type="text" value="0" id="numMax"/><p>

            <input type="button" value="Calcular" onClick="vector();"/>
            <input type="reset" value="Limpiar"/></br>

            <p id="print1">El vector de números es: </p>
        </fieldset>
    </form>

</body>
</html>

```

[Ver el ejemplo](#)

CSS - CASCADING STYLE SHEETS

CSS son las siglas en inglés de **Cascading Style Sheets**, que se traduce al español como **Hojas de Estilo en Cascada**. Es el lenguaje de diseño utilizado para controlar la apariencia visual, el formato y el diseño de los documentos (como HTML o XML) en una página web.

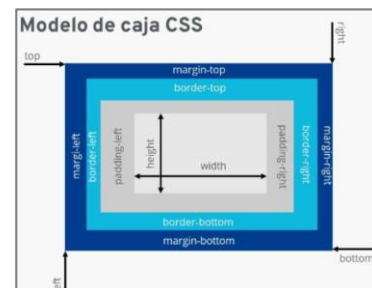
En el desarrollo web, CSS actúa como el "maquillaje" o la vestimenta de un sitio. Mientras que el HTML define la estructura básica y el contenido (los textos, los botones, el esqueleto), el CSS se encarga de darle estilo, color, fuentes atractivas, márgenes y adaptabilidad a diferentes tamaños de pantalla.

Funciones Principales:

- **Diseño visual:** Controla colores, fuentes, fondos y tamaños.
- **Estructura adaptable:** Permite crear diseños flexibles para celulares, tabletas y computadoras.
- **Separación de contenido:** Mantiene el diseño separado del código HTML para facilitar los cambios.

Conceptos Básicos:

- **Selectores:** Indican a qué elemento de HTML se le aplicará el estilo.
- **Propiedades:** Son las características que se van a cambiar (como el color o el margen).
- **Valores:** Definen cómo cambia esa propiedad (por ejemplo, el color rojo o un tamaño de 16 pixeles).



[Curso completo de CSS](#)

 [E-Libro de CSS](#)

PHP- HYPERTEXT PREPROCESSOR

PÁGINAS WEB DINAMICAS CON ACCESO A BASES DE DATOS:

PHP (cuyo nombre oficial recursivo es *PHP: Hypertext Preprocessor*) es un popular lenguaje de programación de código abierto que se ejecuta en el **servidor** para crear páginas web **dinámicas e interactivas**. Fue creado en 1994 por el programador Rasmus Lerdorf.

Cómo funciona

- **Del lado del servidor:** El código se procesa en el servidor web antes de enviarse en formato HTML al navegador del usuario.
- **Mezcla con HTML:** Se puede insertar de forma muy fácil dentro del código HTML usando etiquetas especiales (`<?php ... ?>`).
- **Conexión con datos:** Tiene una excelente compatibilidad con bases de datos como MySQL y PostgreSQL.

Para qué se utiliza

- **Gestores de contenido:** Impulsa plataformas famosas como WordPress, Drupal y Joomla.
- **Comercio electrónico:** Crea tiendas virtuales y procesa formularios o pagos de usuarios.
- **Sitios grandes:** Plataformas como Wikipedia y Facebook lo han usado en su infraestructura.



[Curso completo de PHP](#)

INSERTAR VÍDEO Y SONIDO

Utilizamos la etiqueta **<VIDEO>** para ejecutar un vídeo .MP4 el cual debemos tener cómo archivo en la misma carpeta donde se tenga en .html

VÍDEO EN FORMATO .MP4

Tenga en cuenta los siguientes atributos y la estructura correcta para visualizar el vídeo ejecutándose dentro de su página web.

Atributo	Función
src = <i>"URL"</i>	Dirección del <i>objeto</i> a incorporar.
width = <i>anchura</i>	Anchura del <i>objeto</i> .
height = <i>altura</i>	Altura del <i>objeto</i> .
controls	Activa los controles de reproducción.

```
<video width="640" height="360" controls>  
  <source src="tu-video.mp4" type="video/mp4">  
</video>
```

VÍDEO DE YOUTUBE (Autoejecutar):

La etiqueta <IFRAME> es utilizada para incrustar los vídeos de Youtube y que se reproduzcan directamente en nuestra página HTML. Para ello se pueden utilizar los siguientes atributos adicionales:

Atributo	Función
src ="URL"	Dirección del <i>objeto</i> a incorporar.
width =anchura	Anchura del <i>objeto</i> .
height =altura	Altura del <i>objeto</i> .
frameborder ="1 o 0"	Con borde o sin borde.
allowfullscreen	Muestra la opción para ampliar a pantalla completa.
src ="URL"	Dirección o link de Youtube.
mute =1	Quita el silencio.
&rel =0	
&autoplay =1	Activa el autoplay.

Ejemplo:

```
<html>
<body>
  <center>

  <iframe width="560" height="315" src=" https://www.youtube.com/watch?v=lo-
  g5ktoYRI&t=45s?mute=1&rel=0&autoplay=1" frameborder="0"
  allowfullscreen></iframe>

  </center>
</body>
</html>
```

[Ver el ejemplo](#)

Para dotar a las páginas web de filmaciones, MICROSOFT utiliza la marca con cinco nuevos atributos:

El formato compatible con esta marca es AVI, ya que es el único admitido por el visualizador interno de vídeo que viene incorporado en el *Internet Explorer*.

Esta marca, no obstante, es muy difícil de aplicar hoy en día debido a las limitadas condiciones de Internet (básicamente su actual lentitud) para transmitir archivos tan grandes como suelen ser las filmaciones en vídeo. Lo más habitual en la actualidad es cargar este tipo de archivos (a través de FTP) de forma comprimida, para visualizarlo posteriormente con una aplicación externa.

MÚSICA DE FONDO <AUDIO>:

Para que se ejecute la página web con música de fondo; se utiliza la etiqueta <AUDIO> y se configura con ciertos atributos especiales que activan los controles de reproducción. El archivo de sonido debe estar en la misma carpeta o raíz de archivo.html. Se recomienda sonido en **formato.mp3**

Atributo	Función
source src ="URL"	Dirección del archivo de sonido.
loop ="número"	Número de veces de repetición.
autoplay ="true"	El sonido se repetirá indefinidamente.
controls	Activa los controles de reproducción.

```
<audio id="player" autoplay loop controls>
<source src="archivo.mp3" type="audio/mpeg">
</audio>
```

LETREROS <MARQUEE></MARQUEE>:

Si queremos dar mayor dinamismo al texto de la página podemos crear letreros en desplazamiento. Para ello utilizaremos el delimitador <MARQUEE></MARQUEE>, que cuenta con los siguientes atributos:

```
<html>
<body>

<MARQUEE>Este es un letrero pasando de lado a lado</MARQUEE>

<MARQUEE behavior="alternate">Otro letrero rebotando de lado a
lado</MARQUEE>

<MARQUEE scrolldelay="150">Letrero con menos velocidad</MARQUEE>

</body>
</html>
```

[Ver el ejemplo](#)

Atributo	Función
behavior = <i>scroll, slide, alternate</i>	Comportamiento del letrero.
direction = <i>left, right</i>	Dirección del desplazamiento.
loop = <i>número</i>	Número de veces que se repite la visualización del letrero.
loop = infinite	La visualización del letrero se repetirá indefinidamente.
scrollamount = <i>número</i>	Número de pixels que cambian en de las fases de visualización.
scrolldelay = <i>número</i>	Tiempo (en milisegundos) entre una de cada las fases de visualización.
hspace = <i>número</i>	Anchura de los márgenes horizontales.
vspace = <i>número</i>	Anchura de los márgenes verticales.
height = <i>número o porcentaje</i>	Altura del elemento.
width = <i>número o porcentaje</i>	Anchura del letrero.
align = <i>top, bottom, middle</i>	Alineación vertical del letrero.
bgcolor = <i>#rrggbb</i>	Valor RGB del color del fondo.
bgcolor = <i>nombre</i>	Nombre del color del fondo.

BOTÓN DE WHATSAPP

Integrar el botón de WhatsApp en tu sitio web es clave para reducir la fricción en la comunicación. Permite a los usuarios iniciar un chat directo e instantáneo desde su dispositivo móvil o computadora. Esto genera mayor confianza, mejora la experiencia del cliente y aumenta significativamente la tasa de conversión y ventas.

Beneficios importantes:

Atención en tiempo real: Resuelve dudas al instante y ofrece soporte personalizado, lo que evita que el cliente abandone tu página y se vaya con la competencia.

Acercamiento directo: Facilita la comunicación desde el momento exacto en que el usuario navega por tu web.

Fácil implementación: Puedes añadirlo rápidamente usando widgets o enlaces directos, incluso automatizando mensajes de bienvenida según la sección de la página donde se encuentre el usuario.

Así podemos colocar entonces el botón de Whatsapp en nuestra página web.

IMPORTANTE: Este código lo vas a guardar dentro de una carpeta raíz y dentro de ella se deben crear dos elementos:

1. Una subcarpeta con el nombre **images** en donde se incluye la imagen o iconos de whatsapp. Desde este [link puedes descargar](#) el gif animado wp.gif
2. Crear este archivo **botonwp.html**



Wp.gif

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Boton Whatsapp</title>

<a
href="https://api.whatsapp.com/send?phone=573122584031&text=hola,%20tengo%20una%20
pregunta"
target="_blank"
class="whatsapp-float">

  
</a>

<style>
.whatsapp-float{
  position: fixed;
  top: 155px; /* Distancia desde arriba */
  right: 5px; /* Distancia desde la derecha */
  z-index: 999999; /* Siempre encima */
}

.whatsapp-float img{
  display: block;
}
</style>

</head>

<body>
Botón de Whatsapp ...
</body>
</html>
```

[Ver el ejemplo](#)

BOTONES PARA REDES SOCIALES

Incluir botones de redes sociales en tu página web es clave para aumentar el tráfico, viralizar tu contenido y construir una comunidad leal. Estos elementos actúan como puentes directos que conectan a tu audiencia con tus perfiles oficiales y facilitan que los usuarios compartan tu información.

Esto ayuda a que la página web y las redes sociales acumulen más:

Generación de tráfico y viralidad: Los botones para compartir permiten a los lectores difundir tus artículos, productos o noticias en sus propios perfiles. Esto amplía tu alcance y atrae nuevos visitantes orgánicos

Aumento de seguidores y alcance: Los botones de seguir conectan directamente a tus visitantes con tus redes (como Facebook, Instagram o X/Twitter), permitiéndote fidelizar a tu audiencia y mantener un contacto constante.

Mejora en el posicionamiento web (SEO): Aunque no es un factor de ranking directo, una mayor difusión en redes sociales aumenta la visibilidad de tu marca. Esto genera más visitas y enlaces externos, factores que Google toma en cuenta para posicionar mejor tu web.

Así podemos colocar entonces los botones de redes sociales desde nuestra página web.

IMPORTANTE: Este código lo vas a guardar dentro de una carpeta raíz y dentro de ella se deben crear dos elementos:

1. Una subcarpeta con el nombre **images** en donde se incluyen las imágenes o iconos de cada red social. Desde este [link puedes descargar](#) las imágenes.
2. Crear este archivo **redes.html**



facebook.png



google.png



instagram.png



twitter.png



youtube.png

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
<title>Documento sin título</title>
<style>body {
    font-family: Arial, Helvetica, sans-serif;
}

td {
    font-family: Arial, Helvetica, sans-serif;
}

th {
    font-family: Arial, Helvetica, sans-serif;
}

/* -----
Barra estática de redes sociales
-----*/

#social ul li a {
```

```

        background-color: white;
        width: 50px;
        padding: 0px;
        margin: 0px;
    }
    #social {
        position: fixed;
        left: -30px;
        top: 200px;
        z-index: 3000;
        list-style-type: none;
    }
    #social ul {
        list-style-type: none;
    }

    #social ul li a:hover{
        width: 100px;
        padding: 0px;
        margin-top: 0px;
        margin-right: 0px;
        margin-bottom: 0px;
        margin-left: 10px;
        -webkit-transition: all 1000ms 0s;
        -moz-transition: all 1000ms 0s;
        -ms-transition: all 1000ms 0s;
        -o-transition: all 1000ms 0s;
        transition: all 1000ms 0s;
    }
</style>
</head>
<body>

```

ESTA ES UNA PRUEBA DE BOTONES PARA REDES SOCIALES

<!--Barra lateral izquierda de redes sociales-->

```

<div id="social">
<ul>

```

```

<li><a href="https://www.facebook.com/ingenierosenlaweb" target="_blank"></a></li>
<li><a href="https://twitter.com/andresliloy" target="_blank"></a></li>
<li><a href="https://www.youtube.com/@andresliloy-IngenierosEnLaWeb"
target="_blank"></a></li>
<li><a href="https://www.instagram.com/ingenierosenlaweb" target="_blank"></a></li>
<li><a href="https://www.google.com/search?q=ingenieros+en+la+web"
target="_blank"></a></li>

```

```

</ul>
</div>

```

```

</body>
</html>

```

[Ver el ejemplo](#)

FORMULARIOS

Los Formularios (*forms*) se utilizan para la recepción y envío de datos desde una página Web. Dejando de lado aspectos más complejos como la programación de las aplicaciones (CGI), su instalación en el servidor y su manera de procesar la información, nos vamos a centrar exclusivamente en el diseño de los Formularios, cuyas marcas son las siguientes:

<FORM></FORM>: Delimitador de un formulario.
<INPUT>: Control de la entrada de datos.
<SELECT></SELECT>: Delimitador de una lista desplegable o de selección.
<OPTION>: Componente de una lista desplegable o de selección.
<TEXTAREA></TEXTAREA>: Delimitador de un campo amplio de texto.

- **<FORM></FORM>**: Estas marcas delimitan un formulario. Entre estas dos puede aparecer cualquier otra marca excepto otro **<FORM>**. Es decir, puede haber varios formularios en una página Web, pero nunca un formulario dentro de otro. Por otra parte, hay que tener en cuenta que los formularios no se diferencian visualmente del resto de la página. Por ello, si se quiere diferenciar, una posibilidad es colocar una marca **<HR>** antes y después del formulario. Estos son sus atributos:

Atributo	Función
action ="URL"	Dirección URL del programa que se ejecutará para procesar el formulario.
method = <i>get, post</i>	Método empleado para enviar y codificar la información introducida por el usuario.

El método (**METHOD**) utilizado dependerá de la forma en que trabaja el servidor:

GET es el método por defecto, y hace que el contenido del formulario sea añadido a la dirección URL.

POST es el método que suele recomendarse, y hace que el contenido del formulario sea enviado al servidor independientemente de la dirección URL. Junto a este atributo se puede añadir otro, denominado **ENCTYPE** (*encoding type*), que especifica el tipo de codificación del contenido del formulario.

- **<INPUT>**: Esta marca se utiliza para definir un elemento de entrada de datos dentro de un formulario. En algunos aspectos su tratamiento es semejante a la marca ****. Sus atributos son:

Atributo	Función
type ="text, password, checkbox, radio, submit, reset"	Tipo de entrada de datos.
name ="nombre"	Nombre del campo de entrada de datos.
value ="valor"	Valor que devuelve el control.
checked [={true false}]	Indica si el elemento está o no inicialmente seleccionado.
size ="caracteres"	Anchura de la ventana del control en caracteres.
size ="ancho, alto"	Anchura y altura de la ventana del control.
maxlength ="ancho"	Máximo número de caracteres permitidos para la entrada de datos.

No todos estos atributos son compatibles entre sí. Vamos a detenernos brevemente en ellos:

TYPE. Los distintos tipos de entrada de datos son los siguientes:

A. CAMPOS DE ENTRADA DE TEXTO:

- **Text.** Campo de entrada de una línea de texto. Es la opción por defecto.
- **Password.** Es un tipo de campo de entrada de texto en el que cada uno de los caracteres (letras y/o números) son representados como asteriscos.

Ejemplo:

```
<INPUT type="text" name="campo1" value="texto:" size="20"
maxlength="5">
```

B. CASILLEROS DE SELECCIÓN:

- **Checkbox.** Casillero de selección en forma de cuadrado. Se utiliza con variables que pueden tomar dos únicos valores. Por tanto, puede estar seleccionado (*on*) o ignorado (*off*).
- **Radio.** Casillero de selección en forma de círculo. Se utilizan cuando una misma variable está asociada a varios elementos, cada uno con un valor distinto. Por tanto, cada casillero puede estar seleccionado (*on*) o ignorado (*off*).

Ejemplo:

```
<INPUT type="checkbox" name="pregunta" value=1 checked>
```

C. BOTONES DE ACCIÓN:

- **Submit.** Es un botón que hace que los datos introducidos en el formulario sean enviados al servidor para su procesamiento.
- **Reset.** Es otro botón que borra todos los datos introducidos haciendo que el formulario vuelva a aparecer con sus datos originales.

Ejemplo:

```
<INPUT type="submit" value="ENVIAR
```

DATOS"> D. OTROS TIPOS DE ATRIBUTOS:

- **Hidden.** Es un almacén interno de datos. Es decir, el navegador no muestra su contenido (recogido en el atributo *value*) pero sí se lo transmite al programa CGI.

Ejemplo:

```
<INPUT type="hidden" name="browser" value="arena">
```

- **Image.** Es un tipo especial que funciona de forma parecida a los Mapas Sensibles *Servidor* y especialmente útil para la construcción de *barras de herramientas* divididas en varias imágenes. En lugar de un formulario se visualiza una imagen cuya ubicación se indica en el atributo *src*, y que puede alinearse con el atributo *align*.

Ejemplo:

```
<INPUT type="image" src="botones.gif" name="barra" align=right>
```

NAME . Es el nombre simbólico del campo de entrada de datos. Debe aparecer junto con el atributo *Type*, salvo que este sea *Submit* o *Reset*,

VALUE. Si se utiliza para un **campo de entrada de texto** (*text* o *password*) especifica el contenido por defecto del campo. Si se utiliza para **casilleros de selección** (*checkbox* o *radio*) especifica el valor del mismo cuando se selecciona (por defecto es "on"). Si se utiliza para **botones de acción** (*submit* o *reset*) especifica el nombre que aparecerá con los mismos.

CHECKED. Sólo se utiliza para indicar si los **casilleros de selección** (*checkbox* o *radio*) están seleccionados de antemano o no.

SIZE. Se usa para indicar el tamaño de los **campos de entrada de texto** (*text* o *password*). Su valor por defecto es 20. Se pueden conseguir campos de entrada de texto de varias líneas con *size=ancho,alto*, pero es más apropiado utilizar el delimitador `<TEXTAREA></TEXTAREA>`, como veremos más adelante.

MAXLENGTH. Este atributo sólo es apropiado para limitar el número de caracteres en **campos de entrada de texto** (*text* o *password*) de una sola línea. Si no aparece, por defecto es ilimitado.

- **<SELECT></SELECT>**: Estas marcas delimitan una lista desplegable o de selección, que suelen tener barras de desplazamiento (*scrollbar*). En un formulario, es decir, entre las marcas `<FORM></FORM>`, puede haber tantas listas desplegables como se quiera. Además, pueden entremezclarse texto u otras marcas (pero no otro formulario, como ya hemos dicho).

Atributo	Función
name ="nombre"	Nombre de la variable.
size =número visibles de (y, por tanto, su líneas). Si el tamaño es mayor que 1 se fuerza la visualización de una lista de selección.	Determina el número de elementos la ventana de visualización amplitud en
multiple Fuerza selección valor que tome el	Permite seleccionar varios elementos. la aparición de una lista de independientemente del atributo size .

Entre las marcas que delimitan una lista de selección sólo se permite otra, **<OPTION>**, que es la que le corresponde a cada componente de la lista, seguida del texto que se desee añadir.

Atributo	Función
selected seleccionada Pueden ser más de una.	Indica que la opción está inicialmente.
value ="valor"	Valor devuelto por la opción.

- **<TEXTAREA></TEXTAREA>**: Estas marcas delimitan un campo amplio de texto, es decir, con varias líneas y columnas. Es, por tanto, una variante del tipo texto (*type=text*) de la marca INPUT. Puede introducirse todo el texto que se desee, que será el texto por defecto que aparezca en el campo de entrada. La barra de desplazamiento (*scrollbar*) aparece automáticamente.

Atributo	Función
name ="nombre"	Nombre de la variable.
rows =número de vertical).	Número de filas (osea la altura o el número caracteres observados de forma
cols =número número horizontal).	Número de columnas (osea el ancho o de caracteres observados de forma

HTML bgcolor RGB

(66 ejemplos)

Complemento "TABLA DE COLORES RGB"

navy

#0000A8

#345ABC

blue

#6666FF

#0C6AA1

#6A6ABA

#A44FF9

#CF61FC

#E8EEEE

#FF00FF

#00A8A8

#8080C0

#6699FF

#55BBEA

#A3D5E4

#ADD2D3

#33E2E5

aqua

#B3E4FD

#E7E7EF

#E2EFED

#EDFEDF

#F0F7D2

purple

#B52E98

#E91998

#ED44AB

#EE7EBB

#EAA3BB

red

#F65F45

#555555

silver

#BBBAAA

#AAAAAA

gray

#545454

#BBB87E

#CCC102

#AAA111

#067120

green

olive

#AA9E7A

#DEB678

#A87654

#826446

#A45555

#AC003A

#CC2431

#EA744E

#A85400

maroon

#600000

black

#FFFFB4

#F0F7A0

#E8FF5A

yellow

#EEE31A

lime

#7EE34B

#FFCD9C

#FFD9C8

#DEB0AF