



Gobierno del  
**CHACO**

Ministerio  
de la Producción y el Desarrollo  
Económico Sostenible



**INFORMATARIO**



*Semana 2*

# INTRODUCCIÓN A HTML

*Qué es HTML | Historia | Etiquetas y atributos | Elementos | Texto | Marcado básico | Enlaces | URL | Listas | Imágenes | Tablas | Maquetación | Semántica y accesibilidad | Formularios.*





## Índice

¿Qué es HTML? .....	Página 2
Breve historia de HTML .....	Página 3
Especificación oficial .....	Página 4
Primer documento HTML .....	Página 6
Etiquetas y atributos .....	Página 8
Elementos HTML .....	Página 9
Texto .....	Página 10
Marcado básico de texto .....	Página 12
Enlaces .....	Página 13
URL .....	Página 14
Enlace absoluto y relativo .....	Página 15
Listas .....	Página 15
Imágenes .....	Página 16
Tablas .....	Página 17
Maquetación Web .....	Página 20
Posicionamiento .....	Página 21
Semántica y accesibilidad .....	Página 24
Formulario .....	Página 25
Conclusiones .....	Página 27
Créditos y atribuciones .....	Página 27
Bibliografía .....	Página 27

# ¿Qué es HTML?

**HTML es el lenguaje estándar para crear y diseñar páginas web. Utiliza etiquetas para estructurar texto, imágenes y otros contenidos para su visualización en un navegador de internet.**

## Introducción

En términos sencillos, HTML es el lenguaje que se utiliza para crear la mayoría de las páginas web en internet. Más específicamente, es el lenguaje con el que se “escriben” estas páginas. Aunque no es un lenguaje de programación propiamente dicho, se trata de un **lenguaje de marcado de hipertexto o lenguaje de etiquetas**.

Los diseñadores emplean HTML para elaborar sus páginas web. Los programas que utilizan generan páginas escritas en HTML, que los navegadores muestran a los usuarios tras interpretar su contenido.

Aunque HTML es un lenguaje informático utilizado por diseñadores, es muy sencillo de aprender y escribir. Las siglas HTML significan **Lenguaje de Marcado de Hipertexto**, en inglés **Hypertext Markup Language**.

Este lenguaje es un estándar reconocido a nivel mundial, cuyas normas son definidas por un organismo sin fines de lucro llamado World Wide Web Consortium (W3C). Dado que es un estándar aceptado por todas las empresas relacionadas con el mundo de internet, una misma página HTML se visualiza de manera “similar” en cualquier navegador, independientemente del sistema operativo.

*(...) una misma página HTML se visualiza de manera “similar” en cualquier navegador, independientemente del sistema operativo.*

## CÓMO FUNCIONA HTML EN UN NAVEGADOR





# Breve historia de HTML

***La historia completa de HTML es tan interesante como larga, por lo que a continuación te lo contamos de forma muy resumida para que tengas noción de sus inicios.***

## 1980

El físico Tim Berners-Lee, trabajador del CERN (Organización Europea para la Investigación Nuclear) propuso un nuevo sistema de "hipertexto" para compartir documentos. Los sistemas de "hipertexto" habían sido desarrollados años antes.

Mediante el hipertexto los usuarios accedían a la información relacionada con los documentos electrónicos que estaban visualizando. De cierta manera, los primitivos sistemas de "hipertexto" podrían asimilarse a los enlaces de las páginas web actuales.

Al finalizar el desarrollo del sistema de "hipertexto", Tim Berners-Lee lo presentó en una convocatoria organizada para desarrollar un sistema de "hipertexto" para Internet y, luego, en conjunto con el ingeniero de sistemas Robert Cailliau, presentaron la propuesta ganadora llamada WorldWideWeb (W3).

## 1993

El organismo IETF presentó la primera propuesta oficial para estandarizar HTML, pero no fue adoptada.

## 1995

El IETF creó un grupo de trabajo que publicó el primer estándar oficial de HTML, llamado HTML 2.0, el 22 de septiembre de ese año.

## 1996/ 1997

El W3C comenzó a publicar estándares de HTML. La versión HTML 3.2, lanzada el 14 de enero de 1997, fue la primera recomendación oficial e introdujo avances como applets de Java y texto alrededor de imágenes.

## 1998

HTML 4.0 se lanzó el 24 de abril de 1998 con mejoras significativas, como CSS, scripts, accesibilidad, tablas complejas y formularios.

## 1999

La última especificación oficial de HTML, HTML 4.01, se publicó el 24 de diciembre de 1999 como una revisión de HTML 4.0 sin cambios significativos.

## 2004

Después de HTML 4.01, el W3C se centró en XHTML. En 2004, Apple, Mozilla y Opera formaron WHATWG debido a la falta de interés del W3C en HTML.

## 2008

WHATWG se centró en el estándar HTML 5, con su primer borrador publicado en enero de 2008. La influencia de las empresas en WHATWG llevó al W3C a retomar la estandarización de HTML en marzo de 2007.

## HTML



*Logo de la última versión de HTML y la utilizada en la actualidad. Versión 5 - W3C*

Imagen provista por [www.w3.org](http://www.w3.org)

# Especificación oficial

**W3C establece normas para diseñadores de páginas web con especificaciones oficiales en inglés disponibles de forma gratuita en el enlace: <https://html.spec.whatwg.org/multipage/>**

## HTML y CSS

Originalmente, las páginas HTML solo tenían información de texto e imágenes. Con el desarrollo del estándar, se incorporaron detalles visuales como tipos de letra, colores y márgenes.

La llegada de tecnologías como JavaScript hizo que las páginas HTML incorporaran código para crear páginas web dinámicas.

HTML es flexible y permite ciertas irregularidades en la estructura del código, como errores que los navegadores a menudo ignoran al mostrar el sitio web. Los atributos y valores pueden estar en mayúsculas o minúsculas sin restricciones.

**CSS separa la presentación del contenido HTML.**

Es un lenguaje de diseño web que controla la apariencia de elementos como colores, fuentes, márgenes y estilos visuales.

## Características básicas

### Lenguajes de etiquetas

En los inicios de la informática, el reto era cómo almacenar información en archivos digitales. Se utilizaba la codificación de letras en números para archivos de texto sin formato.

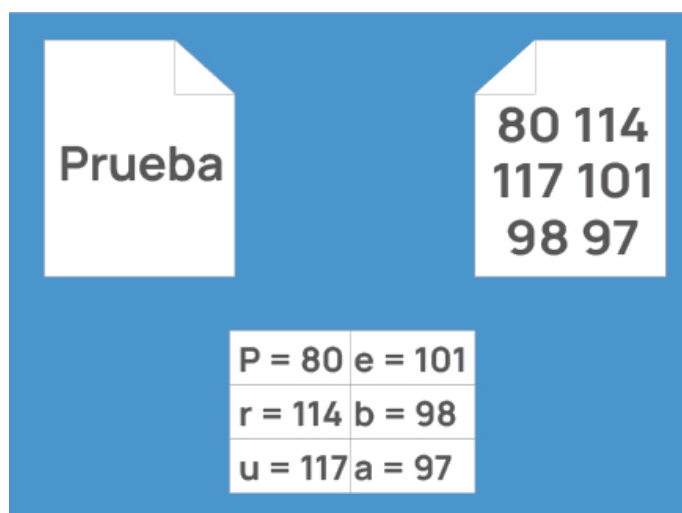
Para almacenar texto en un archivo electrónico se convierte cada carácter en un número, facilitando la recuperación del contenido mediante el proceso inverso.

La codificación de caracteres convierte caracteres en números utilizando páginas de código como ASCII.

Una vez resuelto el problema de almacenar el texto simple, se presentó el reto de almacenar los contenidos de texto con formato. En otras palabras, ¿cómo almacenar un texto en negrita? ¿Y un texto de color rojo? ¿Y otro texto azul, en negrita y subrayado?.

No es posible usar una tabla de conversión para textos sin formato, ya que hay infinitos estilos posibles. Una solución técnica es almacenar la información de formato dentro del archivo para indicar dónde comienza y termina cada estilo.

La solución para almacenar información con formato es simple: los archivos electrónicos guardan los contenidos y la información sobre su formato, como la división en párrafos o resaltado de palabras.





Para que lo comprendas mejor te brindamos un ejemplo:

```
<parrafo>
  Texto con una <importante>palabra</ importante> resaltada.
</parrafo>
```

Para este ejemplo, el principio de un párrafo se indica mediante la palabra `<parrafo>` y el final de un párrafo se indica mediante la palabra `</parrafo>`.

De la misma manera, para asignar más importancia a ciertas palabras del texto, se encierran entre `<importante>` y `</importante>`

El proceso de indicar las diferentes partes que componen la información se denomina marcar (markup en inglés). Cada una de las palabras que se emplean para marcar el inicio y el final de una sección se denominan etiquetas.

Aunque existen algunas excepciones, en general las etiquetas se indican por pares y se forman de la siguiente manera:

- Etiqueta de apertura: carácter `<`, seguido del nombre de la etiqueta (sin espacios en blanco) y terminado con el carácter `>`.
- Etiqueta de cierre: carácter `<`, seguido del carácter `/`, seguido del nombre de la etiqueta (sin espacios en blanco) y terminado con el carácter `>`

La estructura típica de las etiquetas HTML es:

```
<nombre_etiqueta> ... </nombre_etiqueta>
```

HTML es un lenguaje de etiquetas (también llamado lenguaje de marcado) y las páginas web habituales están formadas por cientos o miles de pares de etiquetas. De hecho, las letras "ML" de la sigla HTML significan "markup language", que es como se denominan en inglés a los lenguajes de marcado.

La principal ventaja de un lenguaje de etiquetas es que es muy sencillo de leer y escribir por parte de las personas y de los sistemas electrónicos. La principal desventaja es que puede aumentar mucho el tamaño del documento, por lo que en general se utilizan etiquetas con nombres muy cortos.

*HTML es un lenguaje de etiquetas (también llamado lenguaje de marcado) y las páginas web habituales están formadas por cientos o miles de pares de etiquetas. De hecho, las letras "ML" de la sigla HTML significan "markup language", que es como se denominan en inglés a los lenguajes de marcado.*

```
<etiqueta atributo = "valor" > contenido </etiqueta>
```

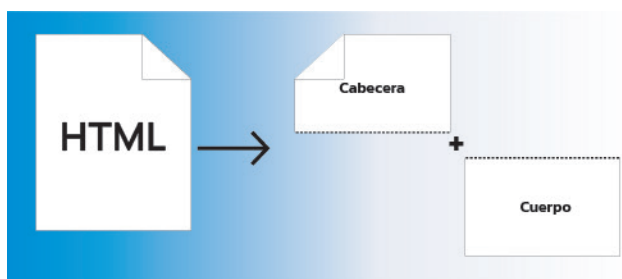
*Ejemplo de estructura de etiquetas en HTML.*

# Primer documento HTML

**Las páginas HTML se dividen en dos partes: la cabecera y el cuerpo. La cabecera incluye información sobre la propia página, como por ejemplo su título y su idioma. El cuerpo de la página incluye todos sus contenidos, como párrafos de texto e imágenes.**

## ¿Qué contienen estas partes?

El cuerpo (llamado body en inglés) contiene todo lo que el usuario ve en su pantalla y la cabecera (llamada head en inglés) contiene todo lo que no se ve (con la única excepción del título de la página, que los navegadores muestran como título en las pestañas de sus ventanas).



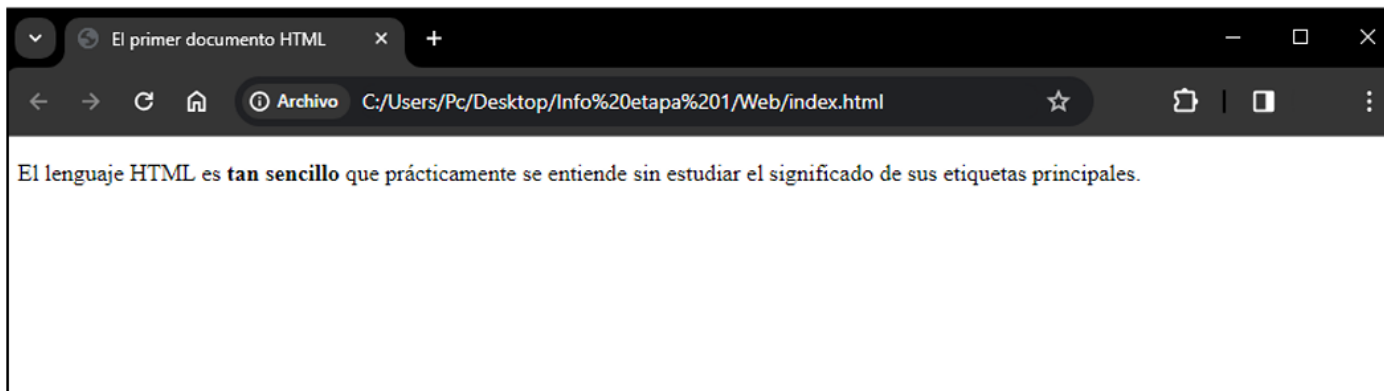
A continuación, te mostramos el código HTML de una página web muy sencilla:

```
<!DOCTYPE html>
<html>
<head>
  <title>El primer documento HTML</title>
</head>
<body>
  <p>El lenguaje HTML es <strong>tan
    sencillo</strong> que
    prácticamente se entiende sin estudiar el
    significado de sus etiquetas principales.</p>
</body>
</html>
```

## Probando el código:

para probar el código anterior, tenés que seguir estos pasos:

1. Abrí el editor de código (VSC - Visual Studio Code) y creá un nuevo documento HTML (generalmente usamos index.html para el archivo principal).
2. Copiá el código HTML mostrado anteriormente y pegalo tal cual en el archivo que creaste.
3. Guardá los cambios.
4. Si buscás el archivo y lo abrís mediante cualquier navegador web, vas a poder visualizarlo con el siguiente aspecto: (ver ejemplo1.html)





*Si ya estás viendo tu primera página HTML en el navegador, probá hacer clic derecho y elegir "inspeccionar" y vas a poder ver el código HTML de la página. De hecho, podés ver el código HTML de cualquier página de Internet mediante la opción "inspeccionar". Probá a ver el código HTML de tu página preferida y vas a ver cuantas etiquetas puede llegar a tener una página*

Volviendo al código HTML del primer ejemplo, es importante conocer las tres etiquetas principales de un documento HTML (`<html>`, `<head>`, `<body>`):

**`<!DOCTYPE html>`**: esta etiqueta al inicio de un archivo HTML indica que se está usando HTML5. Es crucial para que el navegador interprete correctamente el contenido y evite problemas de compatibilidad que afecten la presentación y funcionalidad del sitio web.

**`<html>`**: indica el comienzo y el final de un documento HTML. Ninguna etiqueta o contenido puede colocarse antes o después de la etiqueta `<html>` (con una sola excepción que es la que vimos en el ítem anterior).

En el interior de la etiqueta `<html>` se definen la cabecera y el cuerpo del documento HTML y todo lo que se coloque fuera de la etiqueta `<html>` se ignora.

El **atributo** comúnmente usado es "lang" para especificar el idioma, por ejemplo,

`<html lang="es">` para español y

`<html lang="es-ar">` para español de Argentina.

**`<head>`**: esta es la cabecera y contiene información sobre el propio documento HTML, como por ejemplo su título y los metadatos. Los contenidos indicados en la cabecera no son visibles para el usuario, con la excepción de la etiqueta `<title>`, que se utiliza para indicar el título del documento y que los navegadores lo visualizan en la pestaña del propio navegador.

Los metadatos más comunes a utilizar serán:

**`<meta charset="UTF-8">`**

Indica el formato de codificación de caracteres. Este es el estándar más usado.

**`<meta name="viewport" content="width=device-width, initial-scale=1.0">`**

Asegura que el sitio web optimice la visualización en dispositivos móviles y mejore la experiencia del usuario con el control de la escala y el ancho de la página.

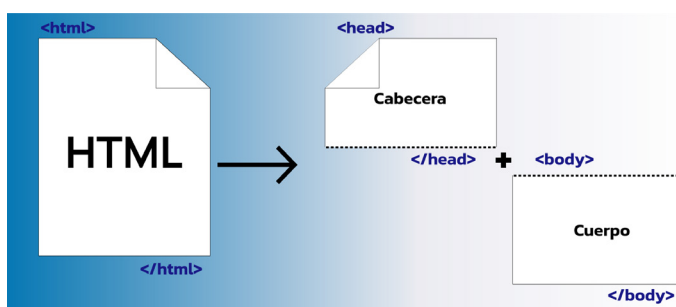
**`<title>Documento</title>`**

Mediante esta etiqueta, nos aseguramos de que se muestre el título correcto en la pestaña del navegador, dependiendo del documento HTML donde nos encontremos.

Hay muchas etiquetas más para metadatos, pero éstas son las más comunes, sin embargo, hay una más que es fundamental para que HTML reconozca la hoja de estilo que se va a usar, es decir, el código CSS de la página, y si no se tiene esta etiqueta, CSS, no se reconocería:

**`<link rel="stylesheet" href="estilo.css">`**

**`<body>`**: delimita el cuerpo del documento HTML. El cuerpo encierra todos los contenidos que se muestran al usuario (párrafos de texto, imágenes, tablas). En general, el `<body>` de un documento contiene cientos de etiquetas HTML, mientras que el `<head>` contiene unas





# Etiquetas y atributos

**HTML define 91 etiquetas que los diseñadores pueden utilizar para marcar los diferentes elementos que componen una página:**

## Las 91 etiquetas:

*a, abbr, acronym, address, applet, area, b, base, basefont, bdo, big, blockquote, body, br, button, caption, center, cite, code, col, colgroup, dd, del, dfn, dir, div, dl, dt, em, fieldset, font, form, frame, frameset, h1, h2, h3, h4, h5, h6, head, hr, html, i, iframe, img, input, ins, isindex, kbd, label, legend, li, link, map, menu, meta, noframes, noscript, object, ol, optgroup, option, p, param, pre, q, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var.*

A pesar de que se trata de un número de etiquetas muy grande, no es suficiente para crear páginas complejas. Algunos elementos como las imágenes y los enlaces requieren cierta información adicional para estar completamente definidos.

La etiqueta `<a>` por ejemplo, se emplea para incluir un enlace en una página. Utilizando sólo la etiqueta `<a>` no es posible establecer la dirección a la que apunta cada enlace. Como no es viable crear una etiqueta por cada enlace diferente, la solución consiste en personalizar las etiquetas HTML mediante cierta información adicional llamada **atributos**.

De esta forma, se utiliza la misma etiqueta `<a>` para todos los enlaces de la página y se utilizan los atributos para indicar la dirección a la que apunta cada enlace.

(ver [ejemplo2.html](#))

```
<!DOCTYPE html>
<html lang="es-ar">
<head>
  <title>Ejemplo de atributos en las etiquetas</title>
</head>
<body>
  <p> Los enlaces son fáciles de indicar:
  <a>Este enlace está incompleto. No tiene
  dirección de destino</a>.
  <a href="https://www.google.com">Este otro está
  completo con destino a Google</a>.
  </p>
</html>
```

Los enlaces son muy fáciles de indicar:

Soy un enlace incompleto, porque no tengo dirección de destino.

[Este otro enlace apunta a la página de Google.](#)

El primer enlace del ejemplo anterior no está completamente definido, ya que no apunta a ninguna dirección. El segundo, añade información adicional mediante un atributo llamado **href**. Los atributos se incluyen dentro de la etiqueta de apertura.

Cada etiqueta HTML tiene su propia lista de atributos, con valores específicos. Los atributos comunes a muchas etiquetas los vamos a explicar por separado para mayor claridad.

**Las etiquetas en HTML son como los contenedores que definen la estructura y el contenido de una página web, mientras que los atributos son como las instrucciones que le dan características específicas a esas etiquetas, como estilo o comportamiento.**



# Elementos HTML

**Además de etiquetas y atributos, HTML define el término elemento para referirse a las partes que componen los documentos HTML.**

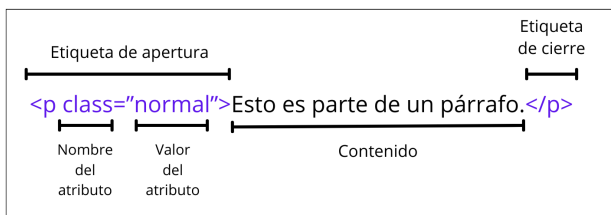
## Elementos y Etiquetas

Aunque en ocasiones se habla de forma indistinta de "elementos" y "etiquetas", en realidad un elemento HTML es mucho más que una etiqueta, ya que está formado por:

- Una etiqueta de apertura.
- Cero o más atributos.
- Texto encerrado por la etiqueta.
- Una etiqueta de cierre.

El texto encerrado por la etiqueta es opcional, ya que algunas etiquetas de HTML no pueden encerrar ningún texto.

El siguiente esquema muestra un elemento HTML, formado por una etiqueta `<p>`, atributos y contenidos de texto:



La estructura mostrada en el esquema anterior es un elemento HTML ya que comienza con una etiqueta de apertura (`<p>`), contiene cero o más atributos (`class="normal"`), dispone de un contenido de texto (Esto es un párrafo) y finaliza con una etiqueta de cierre (`</p>`). }

Por lo tanto, si una página web tiene dos párrafos de texto, la página contiene dos elementos y cuatro etiquetas (dos etiquetas `<p>` de apertura y dos etiquetas `</p>` de cierre). De todas formas, aunque estrictamente no son lo mismo, es habitual intercambiar las palabras "elemento" y "etiqueta".

El lenguaje HTML clasifica elementos en línea y de bloque. Los elementos de bloque ocupan toda la línea, mientras que los elementos en línea solo ocupan el espacio necesario. Los elementos de bloque pueden contener ambos tipos, pero los elementos en línea solo pueden contener texto o elementos en línea.

**Los elementos en línea definidos por HTML son:**

header, hgroup, hr, isindex, li, main, nav, noscript, ol, output, p, pre, section, table, tfoot, ul, video.

**Los siguientes elementos también se consideran que son de bloque:**

dt, frame-set, tbody, td, th, thead, tr.

**Los elementos en línea definidos por HTML son:**

a, abbr, acronym, b, basefont, bdo, big, br, cite, code, dfn, em, font, i, img, input, kbd, label, q, s, samp, select, small, span, strike, strong, sub, sup, textarea, tt, u, var.

**Los elementos de bloque definidos por HTML son:**

address, article, aside, audio, blockquote, canvas, center, dd, dir, div, dl, dlv, fieldset, figcaption, figure, footer, form, h1, h2, h3, h4, h5, h6.

**Los siguientes elementos pueden ser en línea y de bloque según las circunstancias:**

button, del, iframe, ins, map, object, script.

Cuando hablemos de maquetación profundiza-



# Texto

**Un gran porcentaje del contenido de las páginas HTML habituales está formado por texto, por este motivo, es muy importante conocer los elementos y etiquetas que define HTML para el manejo del texto.**

## Al hablar de texto, hablamos de contenido

El lenguaje HTML incorpora al tratamiento del texto muchas de las ideas y normas establecidas en otros entornos de publicación de contenidos.

De esta forma, HTML define etiquetas para estructurar el contenido en secciones y párrafos y define otras etiquetas para marcar elementos importantes dentro del texto.

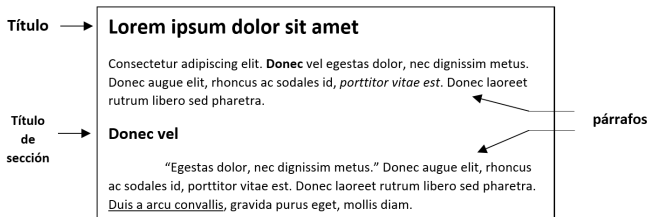
La tarea inicial del editor de contenidos HTML consiste en estructurar el texto original definiendo sus párrafos, titulares y títulos de sección, como se muestra en la siguiente imagen:

Texto original

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec vel egestas dolor, nec dignissim metus. Donec augue elit, rhoncus ac sodales id, porttitor vitae est. Donec laoreet rutrum libero sed pharetra.

Donec vel egestas dolor, nec dignissim metus. Donec augue elit, rhoncus ac sodales id, porttitor vitae est. Donec laoreet rutrum libero sed pharetra. Duis a arcu convallis, gravida

Texto estructurado



El anterior ejemplo se muestra la transformación de un párrafo con un texto simple en un párrafo cuyo texto contiene elementos marcados de forma especial. Así, algunas palabras del texto se muestran en negrita porque se

consideran importantes; otras palabras aparecen en cursiva, ya que se han marcado como destacadas e incluso una frase aparece tabulada y entre comillas, indicando que es una cita textual de otro contenido.

En las secciones siguientes se muestran todas las etiquetas que define HTML para estructurar y marcar el texto. Además, se hace una mención especial al tratamiento que hace HTML de los espacios en blanco y las nuevas líneas.

La forma más sencilla de estructurar un texto consiste en separarlo por párrafos. Además, HTML permite incluir títulos que delimitan cada una de las secciones.

## Párrafos

Una de las etiquetas más utilizadas de HTML es la etiqueta `<p>`, que permite definir los párrafos que forman el texto de una página.

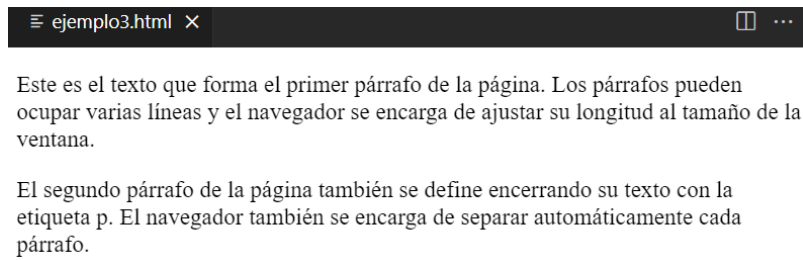
Para delimitar el texto de un párrafo, se encierra ese texto con la etiqueta `<p>`, como muestra el siguiente ejemplo:

`<p>`Este es el texto que forma el primer párrafo de la página.

Los párrafos pueden ocupar varias líneas y el navegador se encarga de ajustar su longitud al tamaño de la ventana.`</p>`

`<p>`El segundo párrafo de la página también se define encerrando su texto con la etiqueta `p`. El navegador también se encarga de separar automáticamente cada párrafo.`</p>`

(ver [ejemplo3.html](#))



Los párrafos creados con HTML son elementos de bloque, por lo que siempre ocupan toda la anchura de la ventana del navegador. Además, no tienen atributos específicos, pero sí que se les pueden asignar los atributos comunes de HTML básicos.

Solamente se debe encerrar con las etiquetas `<h1>`, ..., `<h6>` los títulos de cada sección.

El siguiente ejemplo muestra el uso de las etiquetas de título de sección (ver ejemplo4.html):

## Secciones

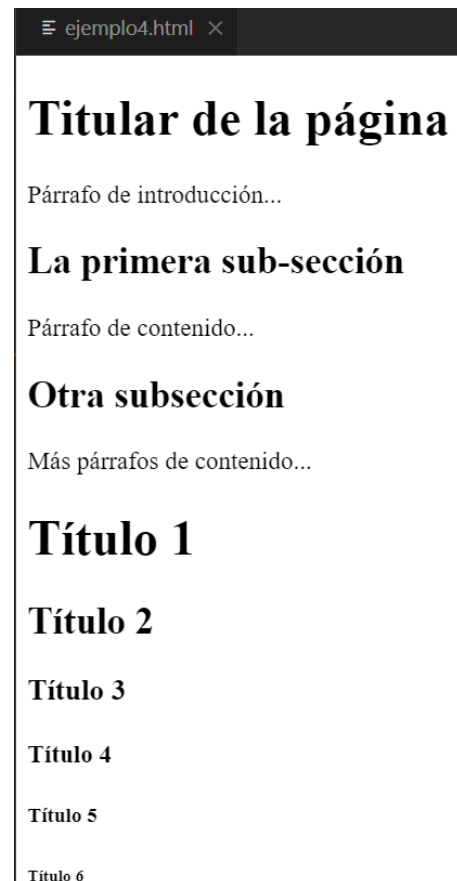
Las páginas HTML habituales suelen tener una estructura más compleja que la que se puede crear solamente mediante un párrafo. De hecho, es habitual que las páginas se dividan en diferentes secciones jerárquicas.

Los títulos de sección se utilizan para delimitar el comienzo de cada sección de la página. HTML permite crear secciones de hasta seis niveles de importancia. De esta forma, aunque una página puede definir cualquier número de secciones, sólo puede incluir seis niveles jerárquicos.

Las etiquetas que definen los títulos de sección son `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` y `<h6>`.

La etiqueta `<h1>` es la de mayor importancia y por tanto se utiliza para definir los titulares de la página. La importancia del resto de etiquetas es descendiente, de forma que la etiqueta `<h6>` es la que se utiliza para delimitar las secciones menos importantes de la página.

Las etiquetas `<h1>`, ..., `<h6>` definen títulos de sección, no secciones completas. Por este motivo, no es necesario encerrar los contenidos de una sección con su etiqueta correspondiente.



Los navegadores ajustan el tamaño de los títulos de las secciones según su importancia, con `<h1>` siendo el más grande y `<h6>` el más pequeño. Este estilo puede cambiarse con CSS.

# Marcado básico de texto

*Una vez estructurado el texto en párrafos y secciones, el siguiente paso es el marcado de los elementos que componen el texto.*

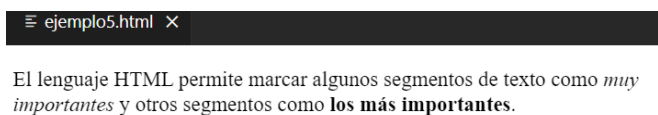
Los textos habituales están formados por elementos como palabras en negrita o cursiva, anotaciones y correcciones, citas a otros documentos externos, etc.

HTML proporciona varias etiquetas para marcar cada uno de los diferentes tipos de texto.

Entre las etiquetas más utilizadas para marcar texto se encuentran `<em>` y `<strong>`.

La etiqueta `<em>` marca un texto indicando que su importancia es mayor que la del resto del texto. La etiqueta `<strong>` indica que un determinado texto es de la mayor importancia dentro de la página.

Ejemplo (ver *ejemplo5.html*):



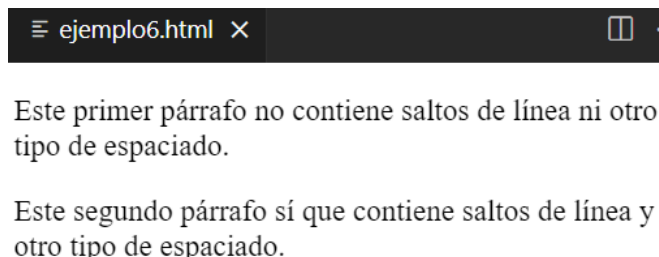
El lenguaje HTML permite marcar algunos segmentos de texto como *muy importantes* y otros segmentos como **los más importantes**.

Por defecto, los navegadores muestran los elementos `<em>` en cursiva para hacer evidente su importancia y muestran los elementos `<strong>` en negrita, para indicar que son los más importantes.

## Espacios en blanco y nuevas líneas

El aspecto más sorprendente del lenguaje HTML cuando se desarrollan los primeros documentos es el tratamiento especial de los "espacios en blanco" del texto. HTML considera espacio en blanco a los espacios en blanco, los tabuladores, los retornos (tecla return) y el carácter de nueva línea (tecla enter).

Ejemplo (ver *ejemplo6.html*):



Este primer párrafo no contiene saltos de línea ni otro tipo de espaciado.

Este segundo párrafo sí que contiene saltos de línea y otro tipo de espaciado.

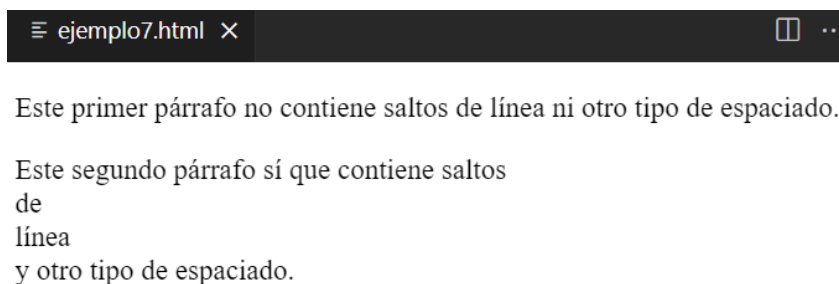
Los dos párrafos de la imagen anterior se ven idénticos, aunque el segundo párrafo incluye varios espacios en blanco y está escrito en varias líneas diferentes. La razón de este comportamiento es que HTML ignora todos los espacios en blanco sobrantes, es decir, todos los espacios en blanco que no son el espacio en blanco que separa las palabras.

No obstante, HTML proporciona varias alternativas para poder incluir tantos espacios en blanco y tantas nuevas líneas como sean necesarias dentro del contenido textual de las páginas.

## Nuevas líneas

Para incluir una nueva línea en un punto y forzar a que el texto que sigue se muestre en la línea inferior, se utiliza la etiqueta `<br>`. En cierta manera, insertar la etiqueta `<br>` en un determinado punto del texto equivale a presionar la tecla ENTER en ese mismo punto.

La etiqueta `<br>` es una de las pocas etiquetas especiales de HTML. La particularidad de `<br>` es que es una etiqueta vacía, es decir, no encierra ningún texto. De esta forma, la etiqueta debe abrirse y no hace falta cerrarla: `<br>`.

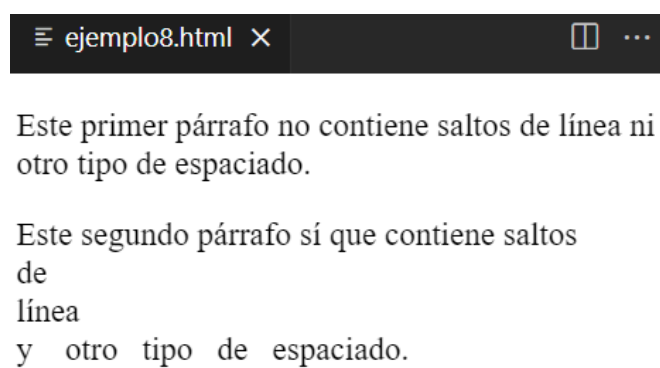


Utilizando la etiqueta `<br>` se puede rehacer el ejemplo anterior para que respete las líneas que forman el segundo párrafo (ver *ejemplo7.html*).

## Espacios en blanco

Para incluir espacios en blanco adicionales, se debe sustituir cada nuevo espacio en blanco por el texto `&nbsp;`; (es importante incluir el símbolo `&` al principio y el símbolo `;` al final).

Así, el código HTML del ejemplo anterior se debe rehacer para incluir los espacios en blanco adicionales (ver *ejemplo8.html*):



Cada texto `&nbsp;` solamente equivale a un espacio en blanco, por lo que se deben escribir tantos `&nbsp;` seguidos como espacios en blanco seguidos existan en el texto.

## Enlaces

La introducción del hipertexto fue fundamental para el éxito del lenguaje HTML, ya que permitió la creación de documentos interactivos que ofrecen información adicional cuando se solicita. El componente principal del hipertexto es el "hiperenlace", también conocido como "enlace web" o simplemente "enlace".

Los enlaces se emplean para establecer conexiones entre dos recursos. Aunque la mayoría de los enlaces conectan páginas web, también es factible enlazar otros recursos como imágenes, documentos y archivos.

Un aspecto a menudo pasado por alto en los enlaces es que están compuestos por dos extremos y una dirección. En otras palabras, el enlace se origina en un recurso y se dirige hacia otro. Cada uno de estos extremos se llama "*anchors*" en inglés, que se puede traducir como "*anclas*".

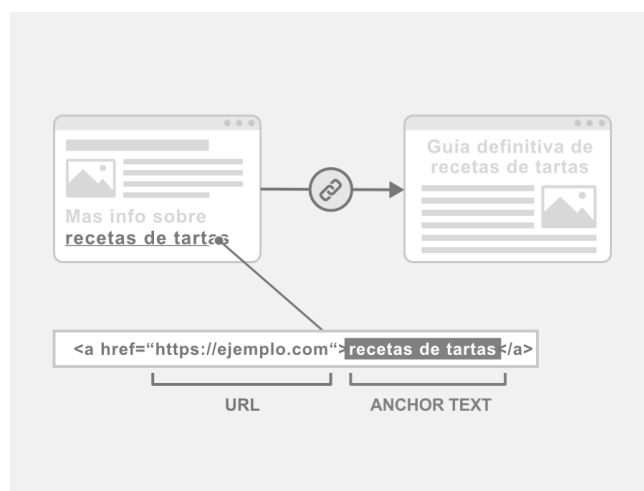


Imagen: Anchor text - Autor: Seobility -  
Licencia: CC BY-SA 4.0







## Enlace absoluto y relativo

Los enlaces en HTML se crean mediante la etiqueta `<a>`.

El atributo más importante de la etiqueta `<a>` es `href`, que se utiliza para indicar la URL a la que apunta el enlace. Cuando el usuario hace clic sobre un enlace, el navegador se dirige a la URL del recurso indicado mediante `href`.

### Enlace absoluto

Con la definición anterior, para crear un enlace que apunte a la página principal de Google solamente habría que incluir lo siguiente en un documento HTML:

```
<a href="https://www.google.com">Página principal de Google</a>
```

### Enlace relativo

Como el atributo `href` indica una URL, un enlace puede apuntar a cualquier tipo de recurso al que pueda acceder el navegador. El siguiente enlace apunta a un documento html que tenemos guardado de forma local, que se mostrará en el navegador cuando el usuario haga clic sobre el enlace:

```
<a href="ejemplo1.html">Documento HTML que nos lleva al primer ejemplo.</a>
```

De la misma forma, los enlaces pueden apuntar directamente a documentos PDF, Word, etc. (Ver ejemplo9.html)

## Listas

***En ocasiones, es posible agrupar determinadas palabras o frases en un conjunto de elementos que tienen más significado de forma conjunta.***

El lenguaje HTML define tres tipos diferentes de listas para agrupar los elementos: listas no ordenadas (se trata de una colección simple de elementos en la que no importa su orden), listas ordenadas (similar a la anterior, pero los elementos están numerados y por tanto, importa su orden) y listas de definición (un conjunto de términos y definiciones similar a un diccionario).

### Listas no ordenadas

Las listas no ordenadas son las más sencillas y las que más se utilizan.

las que más se utilizan.

Una lista no ordenada es un conjunto de elementos relacionados entre sí, pero para los que no se indica un orden o secuencia determinados. La etiqueta `<ul>` encierra todos los elementos de la lista y la etiqueta `<li>` cada uno de sus elementos.

El siguiente código HTML muestra un ejemplo sencillo de lista no ordenada (Ver ejemplo10.html).

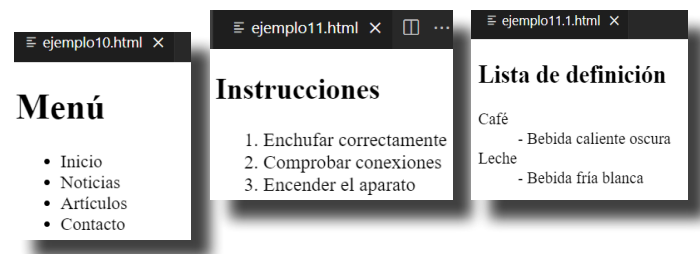
## Listas ordenadas

Son similares a las listas no ordenadas, pero se usan para presentar elementos en una secuencia específica, como pasos de instrucciones o índices de libros, donde el orden es crucial. Cuando se crea, por ejemplo, una lista con las instrucciones de un producto, es importante el orden en el que se realiza cada paso. Cuando se muestra un índice o tabla de contenidos en un libro, es importante el orden de cada elemento del índice.

En todos estos casos, la lista más adecuada es la lista ordenada, que se define mediante la etiqueta `<ol>`. Los elementos de la lista se definen mediante la etiqueta `<li>`, la misma que se utiliza en las listas no ordenadas. (Ver ejemplo11.html).

## Listas de definición

Las listas de definición apenas se utilizan en la mayoría de páginas HTML. Su funcionamiento es similar al de un diccionario, ya que cada elemento de la lista está formado por términos y definiciones. La etiqueta `<dl>` crea la lista de definición y las etiquetas `<dt>` y `<dd>` definen respectivamente el término y la descripción de cada elemento de la lista. (Ver ejemplo11.1.html).



## Imágenes

**Las imágenes son esenciales en las páginas web, dividiéndose en imágenes de contenido y de adorno.**

Las *imágenes de contenido* brindan información y complementan el texto, mientras que las *imágenes de adorno* se utilizan para propósitos estéticos, como bordes redondeados o íconos pequeños en listas.

Las imágenes de contenido se insertan en el código HTML con la etiqueta `<img>`, mientras que las imágenes de adorno no deben incluirse en el código HTML, sino mostrarse a través de hojas de estilo CSS.

### Atributos específicos

- **src** = "url" - Indica la URL de la imagen que se muestra.

- **alt** = "texto" - Descripción corta de la imagen.

- **longdesc** = "url" - Indica una URL en la que puede encontrarse una descripción más detallada de la imagen.

- **name** = "texto" - Nombre del elemento imagen.

- **height** = "unidad\_de\_medida" - Indica la altura con la que se debe mostrar la imagen (no es obligatorio que coincida con la altura original de la imagen).

- **width** = "unidad\_de\_medida" - Indica la anchura con la que se debe mostrar la imagen (no es obligatorio que coincida con la anchura original de la imagen).



Los dos atributos requeridos son *src* y *alt*. El atributo *src* es similar al atributo *href* de los enlaces, ya que establece la URL de la imagen que se va a mostrar en la página. Las URL indicadas pueden ser absolutas o relativas.

Un ejemplo sencillo para incluir una imagen:  
(Ver ejemplo12.html):

```

```



Como `<img>` es una etiqueta vacía, no tiene etiqueta de cierre, se cierra sola con el sím-

Los formatos utilizados generalmente son .jpg, .png, .gif o .svg, sin embargo, no existen restricciones sobre el formato a utilizar aunque te recomendamos mantenerte en estos formatos que son los que el navegador web reconoce completamente bien.

Otra cosa a tener en cuenta, es el tamaño de las imágenes, que, en este caso, es una imagen pequeña, pero si la imagen fuera grande, deberíamos manejar ese tamaño con los atributos *width* para el ancho o *height* para el alto. De esto hablaremos más adelante.

## Tablas

**Desde sus primeras versiones, HTML incluyó el soporte para crear tablas de datos en las páginas web. Además de ser sencillo, el modelo definido por HTML es muy flexible y bastante completo.**

Las tablas en HTML utilizan los mismos conceptos de filas, columnas, cabeceras y títulos que los que se utilizan en cualquier otro entorno de publicación de documentos (por

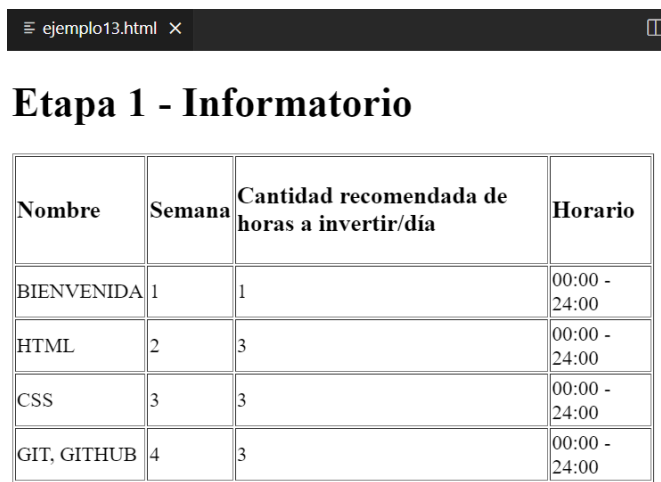
Las tablas de HTML pueden contener elementos simples, agrupaciones de filas y de columnas, cabeceras y pies de tabla, subdivisiones, múltiples y otros elementos complejos.

Etapa 1 - Informatario			
Nombre	Semana	Cantidad recomendada de horas a invertir/día	Horario
BIENVENIDA	1	1	00:00 - 24:00
HTML	2	3	00:00 - 24:00
CSS	3	3	00:00 - 24:00
GIT, GITHUB	4	3	00:00 - 24:00
PROYECTO OPTATIVO	-	3	00:00 - 24:00

## Tablas básicas

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

Ejemplo de una tabla sencilla.  
(Ver ejemplo13.html):



Nombre	Semana	Cantidad recomendada de horas a invertir/día	Horario
BIENVENIDA	1	1	00:00 - 24:00
HTML	2	3	00:00 - 24:00
CSS	3	3	00:00 - 24:00
GIT, GITHUB	4	3	00:00 - 24:00

La etiqueta `<table>` engloba todas las filas y columnas de la tabla.

Por otro lado, las etiquetas `<tr>` (de "table row" en inglés) definen cada fila de la tabla y contienen todas las columnas.

Finalmente, la etiqueta `<td>` (de "table data cell" en inglés) representa cada columna de las filas, aunque en realidad HTML no establece columnas, sino celdas de datos.

Al crear una tabla, es fundamental considerar primero las filas y luego las columnas. Esto se debe a que HTML procesa inicialmente las filas, por lo que las etiquetas `<tr>` aparecen antes que las etiquetas `<td>`.

Entre los diversos atributos disponibles para las celdas, los más comunes son `rowspan` y `colspan`, utilizados para elaborar tablas complejas como las que se presentan más ade-

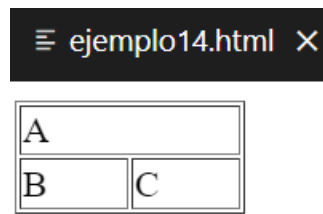
Por otro lado, el atributo `scope` es frecuentemente utilizado, especialmente en las celdas de cabecera.

En ocasiones, algunas celdas de la tabla funcionan como encabezados para otras celdas de la fila o columna. En estos casos, HTML incluye la etiqueta `<th>` (de "table header cell" en inglés) para señalar que una celda es la cabecera de otras.

Las tablas complejas suelen tener una estructura no convencional que combina múltiples columnas para crear una columna ancha o agrupa varias filas para formar una fila más alta que las demás.

Para combinar filas o columnas, se emplean los atributos `rowspan` y `colspan` respectivamente.

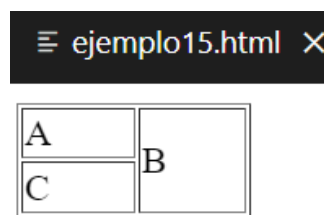
La siguiente imagen muestra una tabla compleja que ha fusionado dos columnas simples para formar una columna más ancha (Ver ejemplo14.html):



A	
B	C

La primera fila de la tabla está formada sólo por una columna, mientras que la segunda fila está formada por dos columnas.

De forma equivalente, si se quiere diseñar una tabla HTML que fusiona filas como la de la si-



A	B
C	

Utilizando los atributos **rowspan** y **colspan**, es posible diseñar tablas tan complejas como la que se muestra en el siguiente ejemplo. (Ver ejemplo16.html):

ejemplo16.html

A		B	
C	D	E	
F			
G	H	I	J

Las tablas pueden ser formateadas a partir de los atributos que nos ofrece la etiqueta `<table>`.

Estos atributos son:

- **align:** Alinea horizontalmente la tabla con respecto a su entorno. (El mismo atributo aplicado sobre la etiqueta `td` (fila) alinea el contenido de la celda).

Los valores posibles son: "right", "left", "center"  
Ejemplo: `<table align="center">`

- **background:** Nos permite colocar un fondo para la tabla a partir de un enlace a una imagen.

Ejemplo: `<table background="imagen.jpg">`

- **bgcolor:** Da color de fondo a la tabla.

Los colores en html se manejan a través del código de color hexadecimal.

Por ejemplo: `<table bgcolor="#ff0000">` o `<table bgcolor="red">`

- **border:** Define el número de pixeles del borde principal.

Las tablas por defecto no muestran el borde entre una celda y la otra, para hacer visible esa propiedad, debemos utilizar el atributo `border`.  
Ejemplo: `<table border="1">`

- **bordercolor:** Define el color del borde.

Una vez definido el atributo de borde, podemos

- **cellspacing:** Define el espacio entre los bordes (en pixeles).

Al igual que el `cellpadding`, la propiedad `cellspacing` es uniforme en toda la tabla.

Ejemplo: `<table cellspacing="5">`

- **height:** Define la altura de la tabla en pixeles o porcentaje.

Hay que tener en cuenta al utilizar este atributo que las tablas siempre priorizarán la estructura de contenido, es decir, si mi tabla tiene una altura de 100 px, pero el contenido supera este tamaño, la tabla se estirará para mostrar el contenido

completo.

Ejemplo: `<table height="200px">`

- **width:** Define la anchura de la tabla en pixeles o porcentaje.

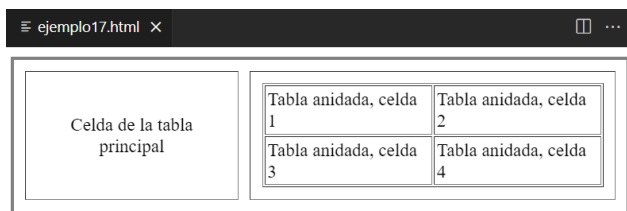
Ejemplo: `<table width="80%">`

### Tablas anidadas

De la misma forma que podemos anidar listas dentro de otras listas, el lenguaje HTML nos permite trabajar con tablas dentro de otras tablas.

Esto nos permite, por ejemplo, trabajar con diferentes atributos de **cellpadding**, uno para cada tabla.

Ejemplo de anidación de tablas. (Ver ejemplo17.html):



Celda de la tabla principal	<table> <tr> <td>Tabla anidada, celda 1</td><td>Tabla anidada, celda 2</td></tr> <tr> <td>Tabla anidada, celda 3</td><td>Tabla anidada, celda 4</td></tr> </table>	Tabla anidada, celda 1	Tabla anidada, celda 2	Tabla anidada, celda 3	Tabla anidada, celda 4
Tabla anidada, celda 1	Tabla anidada, celda 2				
Tabla anidada, celda 3	Tabla anidada, celda 4				

Esta posibilidad nos permite trabajar atributos de tabla independientes y nos brinda una forma más flexible de organizar el contenido.

*OBSERVACIÓN: Para profundizar en el tema de colores, podés obtener los códigos de color en la página oficial*

# Maquetación Web

***Todos los elementos HTML se consideran como cajas rectangulares, las cuales tienen dimensiones (alto y ancho) y características de comportamiento.***

## Modelo de cajas / box-model

Cuando los navegadores descargan el contenido HTML y CSS de las páginas web, aplican un procesamiento muy complejo antes de mostrar las páginas en la pantalla del usuario.

Para cumplir con el modelo de cajas "box-model", los navegadores crean una caja para representar a cada elemento de la página HTML.

Los factores que se tienen en cuenta para generar cada caja son:

- Las propiedades *width* y *height* de la caja (si están establecidas).
- El tipo de cada elemento HTML (elemento de bloque o elemento en línea).
- Posicionamiento de la caja (normal, relativo, absoluto o fijo).
- Las relaciones entre elementos (dónde se encuentra cada elemento, elementos descendientes, etc.).
- Otro tipo de información, como por ejemplo el tamaño de las imágenes y el tamaño de la ventana del navegador.

Veremos los cuatro tipos de posicionamientos definidos para las cajas y otras propiedades que afectan a la forma en la que se visualizan las cajas.

## Tipos de elementos

El estándar HTML clasifica a todos sus elementos en dos grandes grupos: elementos en línea y elementos de bloque.

**Los elementos de bloque ("block elements" en inglés)** siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea.

**Los elementos en línea ("inline elements" en inglés)** no empiezan necesariamente en nueva línea y sólo ocupan el espacio necesario para mostrar sus contenidos.

Debido a este comportamiento, el tipo de un elemento influye de forma decisiva en la caja que el navegador crea para mostrarlo.

**Los elementos en línea definidos por HTML son:**

`a`, `abbr`, `acronym`, `b`, `basefont`, `bdo`, `big`, `br`, `cite`, `code`, `dfn`, `em`, `font`, `i`, `img`, `input`, `kbd`, `label`, `q`, `s`, `samp`, `select`, `small`, `span`, `strike`, `strong`, `sub`, `sup`, `textarea`, `tt`, `u`, `var`.

**Los elementos de bloque definidos por HTML son:**

`dt`, `frame-set`, `tbody`, `td`, `th`, `thead`, `tr`.

**Los siguientes elementos pueden ser en línea y de bloque según las circunstancias:**

`iframe`, `ins`, `map`, `object`, `script`.

# Posicionamiento

***Los navegadores colocan automáticamente las cajas en las páginas HTML, pero CSS permite al diseñador ajustar su posición.***

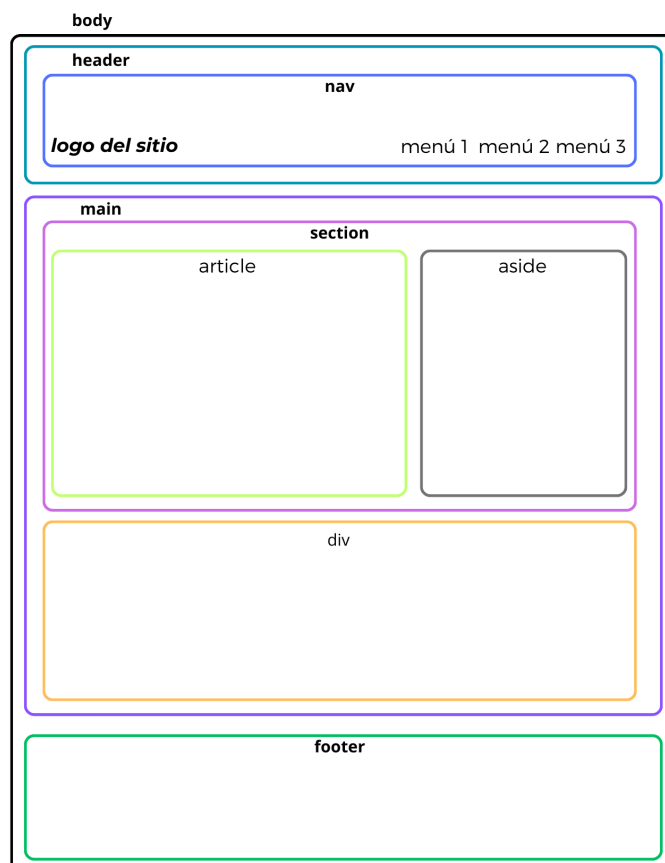
Utilizando las propiedades que proporciona CSS para alterar la posición de las cajas es posible realizar efectos muy avanzados y diseñar estructuras de páginas que de otra forma no serían posibles.

En esta semana no vamos a profundizar en los tipos de posicionamiento, ya que se abordarán a profundidad desde CSS en la siguiente semana, pero te brindamos estos conceptos para que tengas una idea de la estructura y maquetación de una página web.

En las primeras páginas de este documento hablamos de la estructura básica y las etiquetas utilizadas para crear un documento.

En la maquetación nos centramos en el contenido que hay dentro de `<body>` para crear la estructura visual.

Esta es la estructura que usamos en los videos y es la maqueta que comúnmente utilizarán los sitios web.



## Encabezado - header

Se muestra como un contenedor que reúne elementos introductorios o información relevante sobre lo que seguirá a continuación:

Dos principales usos de este elemento son:

- Encabezado de un sitio web, que puede incluir el logotipo, el menú de navegación, etc.
- Encabezado de ciertos bloques de información. Por ejemplo, al presentar la ficha técnica de un libro, en este encabezado se podría incluir el título del libro y una breve descripción

técnica, como introducción al resto de la información.

El elemento `<header>` se ha creado para evitar la necesidad de usar divs sin significado semántico.

## Pie de página - footer

El elemento footer es parecido al header ya que básicamente se ha incluido como agrupador de elementos al final de la estructura del sitio.



Por norma general los sitios web tienen un pie de página en el que se indican datos legales y de contacto, y esta podría ser la principal función, aunque no está pensado para que sea la única.

De la misma forma que el header, hay dos funciones principales para el elemento:

- **Pie de página del sitio**, donde se encuentran los datos legales, etc.
- **Pie de bloque de contenido**, donde incorporar información como la fecha, enlaces relacionados, tags o mapa del sitio.

### Menú de navegación - nav

Este elemento es el que agrupa los enlaces. Es un sección especial para los enlaces ya que incluiremos nuestro menú de navegación.

Se puede utilizar cualquier etiqueta para organizar los enlaces, lo recomendado es utilizar listas no ordenadas (`<ul>`).

### Menú principal - main

Representa el contenido principal del `<body>` de un documento o aplicación. El área prin

cipal del contenido consiste en el contenido que está directamente relacionado, o se expande sobre el tema central de un documento o la funcionalidad central de una aplicación. Este contenido debe ser único al documento, excluyendo cualquier contenido que se repita a través de un conjunto de documentos como enlaces de navegación, información de derechos de autor, logos del sitio y formularios de búsqueda (a menos, claro, que la función principal del documento sea un formulario de búsqueda).

### Secciones - section

Es una parte distintiva de un sitio web que se utiliza para agrupar y organizar contenido relacionado. A menudo se emplea para dividir el contenido en secciones significativas y fomentar una navegación más clara y estructurada para los usuarios. Al delimitar claramente cada sección, se facilita la comprensión del contenido y se mejora la experiencia general del usuario al interactuar con el sitio web. Es una herramienta importante para los diseñadores y desarrolladores web, ya que les permite crear sitios más coherentes y fáciles de navegar.



## Artículo, contenido - article

Es fundamental que el contenido del artículo sea relevante, informativo y bien estructurado, ya que esto no solo beneficia a los usuarios, sino que también mejora la visibilidad de la página en los motores de búsqueda. Por lo tanto, es importante dedicar tiempo y esfuerzo a la creación y optimización de los elementos clave de una página web para garantizar su efectividad y éxito en línea.

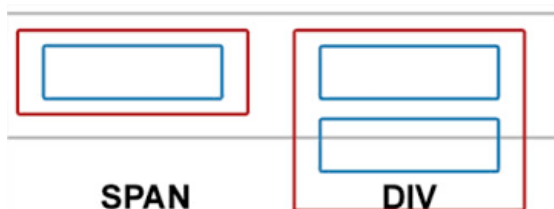
## Barra lateral - aside

Cualquier contenido que no esté relacionado con el objetivo primario de la página va en un `<aside>`.

Es el contenido que, en general, página a página dentro del sitio web se repite junto a la cabecera y el pie, y a su vez, los buscadores le darán menos peso a este bloque que al contenido dentro de `article`. Para darte una idea más clara, los sitios web de noticias utilizan mucho el `aside` para colocar a un costado (generalmente izquierdo) las últimas noticias o noticias relacionadas con la noticia que se está leyendo en el momento.

## Divisiones genéricas - div

Es el contenedor genérico para contenido. No tiene efecto en el contenido o en el diseño hasta que se le aplica un estilo usando CSS (por ej. se le aplica un estilo directamente o se aplica algún tipo de modelo de diseño como Flexbox a su elemento principal).



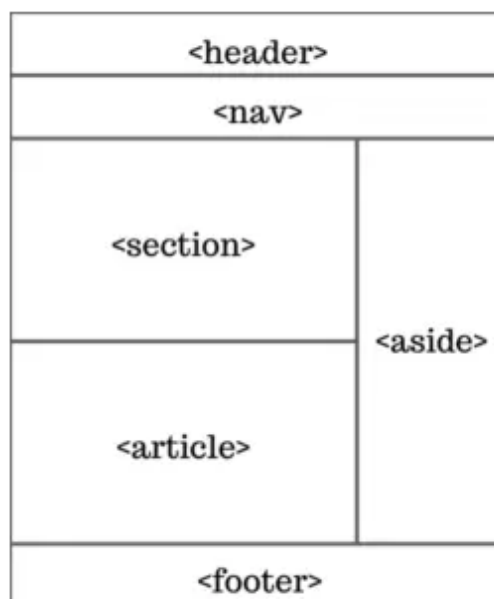
Es muy útil, y muy usado, cuando se sigue estructurando el sitio con mucha información pero que ya no es tan relevante como para ponerla en una etiqueta `section`.

## Títulos - h1 - h6

Estos elementos son los títulos de las diferentes secciones y cada `h` tiene un peso (ranking específico en base al número que tiene, del 1 (el más alto) al 6 (el más bajo)).

Los títulos de las diferentes secciones en un texto juegan un papel crucial en la organización y estructuración del contenido. Los encabezados `h1` a `h6` determinan la jerarquía de la información, asignando un peso específico a cada sección. Mientras que `h1` se considera el título más relevante y destacado, `h6` suele utilizarse para subtítulos de menor importancia.

Es importante elegir cuidadosamente el nivel de encabezado adecuado para cada sección, ya que esto ayuda a los lectores a navegar de manera más efectiva por el texto y comprender su estructura de forma clara y concisa.



# Semántica y accesibilidad

*Utilizar las etiquetas adecuadas para cada parte de nuestro sitio web, equivale a un mejor posicionamiento en el indexado de los buscadores, algo que también conocemos como SEO.*

## Posicionamiento en buscadores / Inclusión para todas las personas

Debemos tener en cuenta que con una buena y correcta estructura del sitio, vamos a lograr una mejor indexación, además, estamos facilitando la lectura del sitio a personas con discapacidad visual, ya que, si no se tienen en cuenta los elementos de forma semántica, el sitio no podrá ser correctamente leído por estas personas con necesidades especiales.

Una prueba que podés hacer para ver de lo que hablamos, es abrir el ejemplo de la página web que te dimos en los videos de esta semana, en un navegador web (lo ideal sería en Chrome), dar clic derecho en cualquier parte del sitio y elegir la opción de "Abrir en modo lectura".

Se abrirá un menú lateral en el navegador donde cada título, subtítulo o artículo, estará en formato de texto. Esto facilita la lectura del sitio, mediante voz sintetizada, para que una persona con necesidades especiales pueda recorrer el sitio, sin necesidad de tener que verlo específicamente.

Además, como te mencionamos, cuando se realice una búsqueda mediante un navegador web, se indexarán los títulos, subtítulos o el propio contenido, logrando así una mejor búsqueda de la página en internet.

## Etiquetas para accesibilidad

En el formulario que te dimos en los videos de esta semana, usamos unas etiquetas que son muy útiles cuando hablamos de temas de

accesibilidad, ya que cuando hablamos de formularios, requerimos que se introduzcan datos y si una persona no tiene capacidad visual, debemos ser específicos en los campos para dar el mayor detalle posible referido a los datos que necesitamos que se introduzcan.

### Etiqueta label for

"label" la utilizamos para poner un título al campo de ingreso de datos, sin embargo, con el atributo "**for**" marcamos en el código que el ingreso de lo que habrá en el campo será un nombre. De esta forma, cuando el usuario haga clic en la etiqueta, es decir, en el título, automáticamente el cursor, se posicionará en el campo a completar.

### Etiqueta label for

Es otra etiqueta que usamos para las descripciones.

Existen muchas más, pero se haría muy extenso dar un repaso por todas.

En la medida que tus proyectos requieran un nivel de detalle aún mayor, te irá surgiendo la necesidad de utilizar distintos tipos de etiquetas para dar una correcta estructuración del sitio y tener la mayor accesibilidad posible.

La recomendación para esto es que siempre busques del sitio oficial, w3.org, para profundizar en éstos temas en la medida que se requieran.

# Formulario

*Los formularios son muy útiles cuando queremos que el usuario nos contacte, ya sea para darnos una opinión, solicitar ayuda, comunicarse para contratar nuestros servicios, hacer un registro de usuario, etc.*

## Sintaxis básica

```
<form action="https://paginaweb.com" method="POST">
  <!--Acá van a ir los campos de ingreso de da-
  tos-->
</form>
```

Mediante la etiqueta **"form"** estamos habilitando controles para que el usuario pueda interactuar con el servidor.

Esto es útil para recolectar información y almacenarla, ya que cuando hablamos de un registro de usuario, por ejemplo, se deberán ingresar datos requeridos como nombre, apellido, email, etc.

Además, si nuestro formulario sirve de consultas, éstas quedarán registradas en nuestro servidor, e incluso, si ya tenemos registrados datos de usuarios, mediante el formulario, entre otros procesos, accedemos a los registros en el servidor para validar el usuario y poder hacer un login.

El atributo **"action"** especifica donde se enviarán los datos del formulario cuando se realiza el envío. Esto puede ser a una base de datos, otra página o algún lugar de almacenamiento que se utilizará para los datos.

El atributo **"method"** define como se enviarán los datos, en el cual hay dos posibles valores. Uno es **"post"** como el que se usó en el ejemplo y sirve para enviar los datos del formulario dentro del cuerpo del documento, sin mostrarlo en la url. Es decir, estos datos se enviarán, supongamos, a una base de datos y no serán vistos desde la url.

Otro método puede ser **"get"** con el cual los datos del formulario se agregan a la url como parámetros de consulta.

Por ejemplo,

<https://paginaweb.com/datos.php?nombre=Informatorio&email=informatorio@ejemplo.com>

En la mayoría de los casos usaremos **post**, ya que los datos del formulario tendrán información sensible que necesitamos que esté resguardada.

Además de otros atributos, el elemento **"form"** funciona como un contenedor para una variedad de controles que representan una amplia gama de elementos.

A continuación, se detallan estos controles:

**<button>**: al ser presionado, ya que funciona como un botón, puede ejecutar diferentes acciones según su atributo **"type"**.

Sus variantes de funcionamiento según el parámetro son:

**"submit"**: envía el formulario al procesador de formularios.

**"reset"**: restablece todos los controles del formulario a sus valores iniciales.

**"menu"**: muestra un menú.

**<fieldset>**: agrupa un conjunto de controles.

**<input>**: uno de varios controles según el valor del atributo **"type"**.

Estos incluyen:

**"hidden"**: un control oculto para enviar información al servidor, generalmente manejado por programas.

**"text"**: para la entrada de una sola línea de

"**search**": similar a "text" pero para búsquedas.  
 "**tel**": para proporcionar un número de teléfono.  
 "**url**": para ingresar una URL absoluta única.  
 "**email**": para editar direcciones de correo electrónico.  
 "**password**": para editar contraseñas de forma segura.  
 "**datetime**": para ingresar una fecha y hora específicas.  
 "**date**": para ingresar una fecha específica.  
 "**month**", "**week**", "**time**": controles para ingresar un mes, semana u hora específicos.  
 "**number**": para ingresar un número.  
 "**range**": para ingresar uno o dos números dentro de un rango.  
 "**color**": para ingresar un color.  
 "**checkbox**": para valores booleanos (verdadero/falso).  
 "**radio**": para seleccionar una única opción entre varias.  
 "**file**": para cargar archivos al servidor.  
 "**submit**": uno de los que más usaremos, ya que con él se enviarán los datos del formulario.  
 "**image**": sirve como submit, pero podemos poner una imagen en lugar del botón tradicional.  
 "**reset**": reinicia los valores del formulario.  
 "**button**": es un botón, pero no tiene una acción específica.

**<label>**: proporciona títulos a los controles del formulario.

**<output>**: muestra el resultado de cálculos realizados por la página o los usuarios basados en los controles del formulario.

**<select>**: para seleccionar una o más opciones de una lista.

**<textarea>**: para ingresar texto en varias líneas.

**<img>**: imagen utilizada como un mapa de imagen del lado del cliente.

Usar estos elementos de forma correcta, convierte a los formularios en una herramienta poderosa que permite recopilar información

sa de los usuarios y desarrollar aplicaciones complejas capaces de recopilar y procesar datos de diversas formas.

Mediante el ejemplo de código provisto en los videos de esta semana, vas a poder probar distintas formas de usar un formulario.

## Paginación

La paginación sirve para organizar nuestros documentos HTML. Los enlaces a cada documento HTML creado para nuestra página, generalmente se encuentran en nuestro menú de navegación, dentro de la etiqueta **<nav>**.

Con la paginación podemos segmentar el contenido completo del sitio web.

En los ejemplos de código que vimos en los videos de esta semana, se crearon 3 documentos HTML, los cuales, segmentaron el contenido del sitio, teniendo un documento principal (**index.html**) como contenido principal del sitio, donde para el ejemplo, contábamos las etapas del Informatario, teníamos una sección de artículos relacionados e información de los integrantes.

Todo esto, sin datos exactos o concretos, ya que usamos mucho texto de relleno. Sin embargo, para un proyecto real, el camino usual a seguir es ese.

En **index.html** contábamos con el contenido principal, en **nosotros.html**, contábamos acerca de nuestra historia y en **formulario.html**, realizábamos un ingreso de datos.

*Lo común entre todos los documentos HTML que utilizamos ese que el "**nav**" y el "**footer**" se repitan, es decir, estén presentes y, con el mismo contenido, en cada documento, como lo es en nuestro ejemplo de los tres documentos HTML creados.*

*La cantidad de documentos HTML de un sitio, va a estar sujeta el tipo de información que brinda nuestro sitio, pero hay que tener en cuenta ciertos aspectos, los cua*

*les son un tema de profundidad que a medida que transites en el mundo del desarrollo web, deberás ir aprendiendo para que la paginación favorezca al sitio y no al contrario.*

*Entonces, con esto podemos decir que la paginación tiene la característica de compartir información similar en cada documento HTML.*

## Conclusiones

---

En esta semana recorrimos los conceptos básicos de HTML, donde tratamos diversos temas, los cuales deberán abordarse con mayor profundidad en la medida que vayas desarrollando con HTML.

Con paciencia y mucha práctica vas a ir necesitando ampliar los conocimientos en diversos aspectos que hay que tratar a mayor detalle para el desarrollo web.

En este material y, complementando la serie de videos de esta semana, te dimos las herramientas fundamentales para comenzar tu carrera en el desarrollo web y la introducción a la programación.

La próxima semana se abordará CSS, por lo que en la medida que avances, vas a ver como el conjunto de conocimientos que vas adquiriendo, van a tener mayor sentido, como así también mayor dedicación de es

tudio para lograr dominar el mundo del desarrollo web, el mundo de la programación.

Hacemos hincapié en que hay que tener paciencia, ya que no se puede aprender todo de una vez, es con lectura, práctica y mucho estudio que se logran integrar todos los conocimientos.

No te desanimes si te cuesta entender algo, es común, no solo cuando empezás a programar, sino a lo largo de tu vida como programador. A medida que transcurre el tiempo y se estudia, y se practica más, el entendimiento de los nuevos conocimientos cuesta menos, pero como en cualquier conocimiento nuevo, tiene una curva de aprendizaje.

Y por último, recordá que la lectura y la práctica son tus mejores aliados en este viaje de aprendizaje.



## Créditos y atribuciones

---

\* **IMAGEN 1** <a href="https://www.freepik.es/vector-gratis/planificacion-desarrollo-interfaz-sitio-web-equipo-devops-personajes-planos-trabajando-ui-ux-diseno-contenido-creacion-software-desarrollo-web\_12083105.htm#page=6&position=10&from\_view=author&uuid=04b33207-47e4-4433-8187-3d2143f4e917">Imagen de vectorjuice</a> en Freepik

## Bibliografía

---

- \* w3c. (2024). <https://www.w3.org/>
- \* w3schools. (2024). <https://www.w3schools.com/>
- \* Damián De Luca. (2011). HTML5, Entienda el cambio y aproveche su potencial. 1ª Ed. Editorial Fox Andina.