

## RESEÑA GENERAL

**HTML** tiene una historia realmente corta, pero, para su poca vida, ha sufrido importantes cambios. Nació públicamente en un documento llamado **HTML Tags** (Etiquetas **HTML**), publicado por primera vez en Internet por **Tim Berners-Lee** en 1997 con objetivos de divulgación.

Él y su grupo crearon lo que por sus siglas en inglés se denomina **Lenguaje HTML** (**HyperText Markup Language**) o lenguaje de etiquetas de hipertexto, el protocolo **HTTP** (**HyperText Transfer Protocol**) y el sistema de localización de objetos en la web **URL** (**Uniform Resource Locator**).

Los programas que leen y representan las páginas web a los usuarios se llaman **navegadores**. Estos son los responsables de interpretar el código HTML y “**dibujar**” una página tal como ellos entiendan que deben hacer.

Esas reglas de representación no dependen de cada fabricante del navegador, sino que existe una organización llamada **W3C** que se encarga de definir el estándar que todos deben seguir a la hora de escribir e interpretar el código **HTML**. Estos estándares del **HTML** se conocen como “**Especificaciones**”, las cuales han ido apareciendo con el tiempo. El **HTML 5** es el último estándar en la actualidad, por lo cual será el que utilizemos en nuestro curso.

## INTRODUCCIÓN

El World Wide Web (WWW) es un sistema que contiene una cantidad de información casi infinita. Pero esta información debe estar ordenada de alguna forma de manera que sea posible encontrar lo que se busca. La unidad básica donde está almacenada esta información son las páginas Web. Estas páginas se caracterizan por contener texto, imágenes, animaciones... e incluso sonido y video.

Una de las características más importantes de las páginas Web es que son hipertexto. Esto quiere decir que las páginas no son elementos aislados, sino que están unidas a otras mediante los links o enlaces hipertexto. Gracias a estos enlaces el navegante de internet puede pulsar sobre un texto de una página para navegar hasta otra página. Será cuestión del programador de la página inicial decidir que palabras o frases serán activas y a donde nos conducirá pulsar sobre ellas.

## **HTML**

Entendiendo que las páginas web son hipertexto, aquí es donde entra HTML, el cual es el código que se utiliza para estructurar y desplegar una página web y sus contenidos. HTML es el lenguaje con el que se escribe el contenido de las páginas web. Las páginas web pueden ser vistas por el usuario mediante un tipo de aplicación llamada cliente web o más comúnmente "navegador", como vimos. Podemos decir por lo tanto que el HTML es el lenguaje usado para especificar el contenido que los navegadores deben representar a la hora de mostrar una página web.

Este lenguaje nos permite aglutinar textos, imágenes, enlaces... y combinarlos a nuestro gusto. La ventaja del HTML a la hora de representar el contenido en un navegador, con respecto a otros formatos físicos como libros o revistas, es justamente la posibilidad de colocar referencias a otras páginas, por medio de los enlaces hipertexto.

Cuando nos referimos al contenido queremos indicar párrafos, imágenes, listas, tablas y todo aquello que forma parte de "el qué". Nunca debemos usar HTML para definir cómo se debe de ver un contenido, si el texto debe tener color rojo, con una fuente mayor, o si se debe alinear a la derecha. Para especificar el aspecto que debe tener una web se usa un lenguaje complementario, llamado CSS.

## HTML LENGUAJE DE MARCADO

HTML no es un lenguaje de programación; es un lenguaje de marcado que define la estructura de tu contenido. Basa su sintaxis en un elemento base al que llamamos marca, tag o simplemente etiqueta. A través de las etiquetas vamos definiendo los elementos del documento, como enlaces, párrafos, imágenes, etc. Así pues, un documento HTML estará constituido por texto y un conjunto de etiquetas para definir la función que juega cada contenido dentro de la página. Todo eso le servirá al navegador para saber cómo se tendrá que presentar el texto y otros elementos en la página.

Existen etiquetas para crear negritas, párrafos, imágenes, tablas, listas, enlaces, etc. Así pues, aprender HTML es básicamente aprenderse una serie de etiquetas, sus funciones, sus usos y saber un poco sobre cómo debe de construirse un documento básico.

Es una tarea muy sencilla de afrontar, al alcance de cualquier persona, puesto que el lenguaje es muy entendible para los seres humanos.

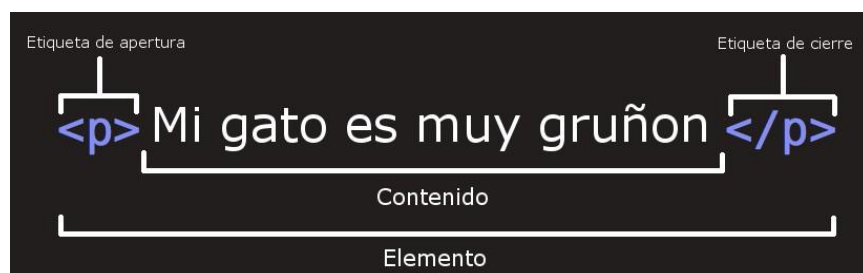
Por ejemplo, toma la siguiente línea de texto:

Mi gato es muy gruñón

Si quieres especificar que se trata de un párrafo, podrías encerrar el texto con la etiqueta de párrafo (<p>):

<p> Mi gato es muy gruñón </p>

## ANATOMÍA DE UNA ETIQUETA HTML



Las partes principales de la etiqueta completa llamada elemento son:

1. La etiqueta de apertura: consiste en el nombre de la etiqueta (en este caso, p), encerrado por comillas angulares (< >) de apertura y cierre. Establece dónde comienza o empieza a tener efecto la etiqueta, en este caso, dónde es el comienzo del párrafo.
2. La etiqueta de cierre: es igual que la etiqueta de apertura, excepto que incluye una barra de cierre (/) antes del nombre de la etiqueta. Establece dónde termina la etiqueta, en este caso dónde termina el párrafo.

3. El contenido: este es el contenido de la etiqueta, que en este caso es solo texto.
4. El elemento: la etiqueta de apertura, más la etiqueta de cierre, más el contenido equivale al elemento.

## ANIDAR ETIQUETAS

Puedes también colocar etiquetas dentro de otras etiquetas, esto se llama anidamiento. Si, por ejemplo, quieres resaltar una palabra del texto (en el ejemplo la palabra «muy»), podemos encerrarla en una etiqueta `<strong>`, que significa que dicha palabra se debe enfatizar:

```
<p> Mi gato es <strong>muy</strong> gruñón </p>
```

Debes asegurarte que las etiquetas estén correctamente anidadas: en el ejemplo, creaste la etiqueta de apertura del elemento `<p>` primero, luego la del elemento `<strong>`, por lo tanto, debes cerrar esta etiqueta primero, y luego la de `<p>`.

Las etiquetas deben abrirse y cerrarse ordenadamente, de forma tal que se encuentren claramente dentro o fuera el uno del otro. Si estos se encuentran solapados, el navegador web tratará de adivinar lo que intentas decirle, pero puede que obtengas resultados inesperados.

## ANATOMÍA DE UN DOCUMENTO HTML

Hasta ahora has visto lo básico de elementos HTML individuales, pero estos no son muy útiles por sí solos. Ahora verás cómo los elementos individuales son combinados para formar una página HTML entera.

Los documentos HTML van a ser archivos de texto con la extensión `.html` y tienen la siguiente anatomía:

```
<!DOCTYPE html>
<html lang="es-Ar">
  <head>
    <meta charset="utf-8">
    <meta charset="utf-8">
    <title>Mi página de prueba</title>
  </head>
  <body>
    <p> Hola Mundo </p>
  </body>
</html>
```

Tienes:

- `<!DOCTYPE html>`: el tipo de documento. Es un preámbulo requerido. Anteriormente, cuando HTML era joven (cerca de 1991/2), los tipos de documento actuaban como vínculos a un conjunto de reglas que el código HTML de la página debía seguir para ser considerado bueno, lo que podía significar la verificación automática de errores y algunas otras cosas de utilidad. Sin embargo, hoy día es simplemente un artefacto antiguo que a nadie le importa, pero que debe ser incluido para que todo funcione correctamente. Por ahora, eso es todo lo que necesitas saber.
- `<html></html>`: la etiqueta `<html>`. Esta etiqueta encierra todo el contenido de la página entera y, a veces, se le conoce como la etiqueta raíz (root element).
- `<lang>`: Es la etiqueta de inicio **`<html>`**. Esto sirve para especificar, mediante el atributo ***lang***, el idioma del contenido del elemento ***"html"***. En este caso español (Argentina). Usá siempre un atributo de idioma en el elemento ***html***. Este es heredado por todos los demás elementos, y así establecerás un idioma predeterminado para el texto en el head del documento. Si tu documento es ***HTML*** (es decir, servido como `text/html`), usá el atributo ***lang*** para establecer el idioma del documento o un rango de texto. ***ESTA ES UNA DE LAS POCAS ETIQUETAS DE HTML QUE NO NECESITA ETIQUETA DE CIERRE.***
- `<head></head>`: la etiqueta `<head>`. Esta etiqueta actúa como un contenedor de todo aquello que quieres incluir en la página HTML que no es contenido visible por los visitantes de la página. Incluye cosas como palabras clave (keywords), una descripción de la página que quieres que aparezca en resultados de búsquedas, código CSS para dar estilo al contenido, declaraciones del juego de caracteres, etc.
- `<meta charset="utf-8">`: esta etiqueta establece el juego de caracteres que tu documento usará en utf-8, que incluye casi todos los caracteres de todos los idiomas humanos. Básicamente, puede manejar cualquier contenido de texto que puedas incluir. No hay razón para no establecerlo, y puede evitar problemas en el futuro.
- `<title></title>`: la etiqueta `<title>` establece el título de tu página, que es el título que aparece en la pestaña o en la barra de título del navegador cuando la página es cargada, y se usa para describir la página cuando es añadida a los marcadores o como favorita.
- `<body></body>`: la etiqueta `<body>`. Encierra todo el contenido que desees mostrar a los usuarios web que visiten tu página, ya sea texto, imágenes, videos, juegos, pistas de audio reproducibles, y demás. Estos, delimitados a su vez por otras etiquetas como las que hemos visto.

## ELEMENTOS EN BLOQUE Y EN LÍNEA

El lenguaje HTML clasifica a todos los elementos en dos grupos: elementos en línea o inline elements y elementos en bloque o block elements. La diferencia entre ambos viene dada por el modelo de contenido, por el formato y la dirección.

Los elementos en bloque siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea, mientras que los elementos en línea sólo ocupan el espacio necesario para mostrar sus contenidos.

## ATRIBUTOS ETIQUETAS

Las etiquetas son la estructura básica del HTML. Estas etiquetas se componen y contienen otras propiedades, como son los atributos y el contenido.

HTML define un total de 91 etiquetas, de las cuales 10 se consideran obsoletas. Sin embargo, una etiqueta por sí sola a veces no contiene la suficiente información para estar completamente definida. Para ello contamos con los atributos: pares nombresvalor separados por "=" y escritos en la etiqueta inicial de un elemento después del nombre del elemento. El valor puede estar encerrado entre "comillas dobles" o 'simples'. Existen, también, algunos atributos que afectan al elemento por su presencia en la etiqueta de inicio.

Esta sería la estructura general de una línea de código en lenguaje HTML:

```
<etiqueta atributo1="valor1" atributo2="valor2">contenido</etiqueta>
```

```
<a href="http://www.enlace.com" target="_blank">Ejemplo de enlace</a>
```

Donde:

- <a> es la etiqueta inicial y </a> la etiqueta de cierre.
- href y target son los atributos.
- http://www.enlace.com y \_blank son las variables.
- Ejemplo de enlace es el contenido.

Nota: las etiquetas <img> y <a> las veremos en mayor profundidad más adelante.

## TIPOS DE ATRIBUTOS

Aunque cada una de las etiquetas HTML define sus propios atributos, encontramos algunos comunes a muchas o casi todas las etiquetas, que se dividen en cuatro grupos según su funcionalidad:

- Atributos básicos
- Atributos de internacionalización
- Atributos de eventos
- Atributos de foco

En esta guía solo vamos a ver los básicos. Ya que el resto de atributos son para el uso de otro lenguaje.

## ATRIBUTOS BÁSICOS

Los atributos básicos se utilizan en la mayoría de etiquetas HTML y XHTML, aunque adquieren mayor sentido cuando se utilizan hojas de estilo en cascada (CSS):

Atributo	Descripción
id="texto"	Establece un indicador único a cada elemento
class="texto"	Establece la clase CSS que se aplica a los estilos del elemento
style="texto"	Aplica de forma directa los estilos CSS de un elemento
title="texto"	Establece el título del elemento (Mejora la accesibilidad)

## SINTAXIS HTML

### LAS MAYÚSCULAS O MINÚSCULAS SON INDIFERENTES AL ESCRIBIR ETIQUETAS

En HTML las mayúsculas y minúsculas son indiferentes. Quiere decir que las etiquetas pueden ser escritas con cualquier tipo de combinación de mayúsculas y minúsculas. Resulta sin embargo aconsejable acostumbrarse a escribirlas en minúscula ya que otras tecnologías que pueden convivir con nuestro HTML (XML, por ejemplo) no son tan permisivas y nunca viene mal hacernos a las buenas costumbres desde el principio, para evitar fallos triviales en un futuro.

## COMENTARIOS EN HTML

En un documento HTML, los comentarios se escriben entre los caracteres "<!--" y "-->". Por ejemplo: <!--Esto es un comentario en HTML-->

## ETIQUETAS META

Las **metaetiquetas**, **etiquetas meta** o **elementos meta** (también conocidas por su nombre en inglés, **metatags** o **meta tags**) son etiquetas **HTML** que se incorporan en el encabezado (dentro del "**head**") de una página web y que resultan invisibles para un visitante normal, pero de gran utilidad para navegadores u otros programas que puedan valerse de esta información. Su propósito es el de incluir información (**metadatos**) de referencia sobre la página: *autor, título, fecha, palabras clave, descripción, etc.*

Se las puede definir como líneas de código que indican a los buscadores que indexan la página web por qué términos debe ser encontrada. Dependiendo de la utilización, caracterización y objetividad de dichos **meta**, más otros factores, se puede conseguir una excelente posición en el listado resultante de una búsqueda. Además, se encargan de añadir

información para facilitar el análisis de los archivos **HTML** y la gestión del contenido de una página web.

Los meta tags de **HTML** siguen a menudo la misma construcción: en primer lugar se define un **elemento** y en segundo lugar se asigna un **contenido**. De esto resulta el siguiente esquema:

<b>&lt;meta name="Nombre del elemento meta" content= "Contenido asignado"&gt;</b>
---

El atributo **name** indica el tipo de información y el atributo **content** indica el valor de dicha información. Por ejemplo, el siguiente código indica que la autora es "Jimena Carracedo", que la página trata sobre un **curso de HTML**, y especifica algunas **palabras clave** a ser consultadas por los buscadores.

<pre>&lt;html lang="es-Ar"&gt;   &lt;head&gt;     &lt;meta name="author" content="Jimena Carracedo"&gt;     &lt;meta name="copyright" content="Jimena Carracedo – Propietaria de los derechos de autor"&gt;     &lt;meta name="description" content="Curso de HTML"&gt;     &lt;meta name="keywords" content="código HTML, etiqueta, páginas web, curso"&gt;   &lt;/head&gt; &lt;/html&gt;</pre>
--

### **ATENCIÓN**

*Los documentos **HTML** están formados por elementos cuya gran mayoría deben escribirse delimitados por una etiqueta de inicio **<etiqueta>** y otra de fin **</etiqueta>**, tales como "**html**", "**head**", "**body**", "**title**" o "**p**". Sin embargo, existen otros elementos que no necesitan ser cerrados con la etiqueta de fin, como, por ejemplo, las etiquetas **<meta>**, **<br>** o **<img>**.*

### **SALTOS DE LÍNEA EN HTML**

Otra de las cosas importantes de conocer sobre la sintaxis básica del HTML es que los saltos de línea no importan a la hora de interpretar una página. Un salto de línea será simplemente interpretado como un separador de palabras, un espacio en blanco. Es por ello que para separar líneas necesitamos usar la etiqueta de párrafo, o la etiqueta BR que significa un salto de línea simple.

Esto es una línea

<br>

Esto es otra línea

Esto en una página se vería así



Esto es una línea  
Esto es otra línea

La etiqueta BR no tiene su correspondiente cierre. Es un detalle que quizás te haya llamado la atención.

## FORMATO DE PÁRRAFOS HTML

Previamente en nuestra guía habíamos visto la etiqueta `<strong>` que nos permitía darle formato a nuestro texto, más concreto ponerlo en negrita. Ahora veremos con más detalle las más ampliamente utilizadas y ejemplificaremos algunas de ellas posteriormente.

Formatear un texto pasa por tareas tan evidentes como definir los párrafos, justificarlos, introducir viñetas, numeraciones o bien poner en negrita, itálica, etc.

Hemos visto que para definir los párrafos nos servimos de la etiqueta `P` que introduce un salto y deja una línea en blanco antes de continuar con el resto del documento.

Podemos también usar la etiqueta `<br>`, de la cual no existe su cierre correspondiente, para realizar un simple salto de línea con lo que no dejamos una línea en blanco, sino que solo cambiamos de línea. Cabe destacar que la etiqueta `<br>`, no es la única etiqueta sin cierre.

Podéis comprobar que cambiar de línea en nuestro documento HTML sin introducir alguna de estas u otras etiquetas no implica en absoluto un cambio de línea en la página visualizada. En realidad, el navegador introducirá el texto y no cambiará de línea a no ser que esta llegue a su fin o bien lo especifiquemos con la etiqueta correspondiente.

## ALINEAR TEXTO

Los párrafos delimitados por etiquetas `P` pueden ser fácilmente justificados a la izquierda, centro o derecha especificando dicha justificación en el interior de la etiqueta por medio de un atributo "align". Recordemos que los atributos no son más que un parámetro incluido en el interior de la etiqueta que ayudan a definir el funcionamiento de la etiqueta de una forma más personalizada.

Es importante tener muy en cuenta lo siguiente, que ya hemos comentado anteriormente. El HTML se usa para definir el contenido. Por lo tanto, los atributos align que vamos a conocer a continuación se estarían metiendo en un terreno que no le corresponde al HTML, porque están definiendo la forma en la que un párrafo debe de representarse, su estilo, y no el contenido. Es importante señalarlo para aprender que estas cosas se deben hacer mediante el lenguaje CSS, que sirve para definir el estilo, la forma. Usamos este ejemplo también para reforzar el uso de los atributos de una manera más práctica.

Así, si deseásemos introducir un texto alineado a la izquierda escribiríamos:

```
<p align="left">Texto alineado a la izquierda</p>
```

Para una justificación al centro:



`<p align="center">Texto alineado al centro</p>`

Para alinear a la derecha:

`<p align="right">Texto alineado a la derecha</p>`

Esto en una página se vería así:

Texto alineado a la izquierda

Texto alineado al centro

Texto alineado a la derecha

Como ves, en cada caso el atributo align toma determinados valores que son escritos entre comillas. En algunas ocasiones necesitamos especificar algunos atributos para el correcto funcionamiento de la etiqueta. En otros casos, el propio navegador toma un valor definido por defecto. Para el caso de align, el valor por defecto es left.

## FORMATEO DE LETRA

Además de todo lo relativo a la organización de los párrafos, uno de los aspectos primordiales del formateo de un texto es el de la propia letra. Resulta muy común y práctico presentar texto resaltado en negrita, itálica y otros. Todo esto y mucho más es posible por medio del HTML a partir de multitud de etiquetas entre las cuales vamos a destacar algunas.

Pero antes de comenzar cabe hacer una reflexión sobre por qué son interesantes estas etiquetas y se siguen usando, a pesar que están entrando prácticamente en el terreno de CSS, ya que en la práctica están directamente formateando el aspecto de las fuentes. Son importantes porque las etiquetas en si no están para definir un estilo en concreto, sino una función de ciertas palabras dentro de un contenido.

## NEGRITA

Podemos escribir texto en negrita incluyéndolo dentro de las etiquetas strong y su cierre. Recordemos que ya la habíamos visto previamente.

`<p><strong> Texto en negrita </strong> y texto normal</p>`

Esto en una página se vería así:

### **Texto en negrita y texto normal**

## ITÁLICA

En este caso existen dos posibilidades, una corta: i y su cierre (italic) y otra un poco más larga: EM y su cierre. En esta guía vamos a usar, y en la mayoría de las páginas que veréis por ahí, os encontraréis con la primera forma sin duda más sencilla a escribir y a acordarse.

`<p><i> Texto en itálica </i> y texto normal</p>`

Esto en una página se vería así:

## *Texto en itálica y texto normal*

### **SUBRAYADO**

El HTML nos propone también para el subrayado la etiqueta: U (underlined). Sin embargo, el uso de subrayados ha de ser aplicado con mucha precaución dado que los enlaces hipertexto van, a no ser que se indique lo contrario, subrayados con lo que podemos confundir al lector y apartarlo del verdadero interés de nuestro texto.

Además, cabe decir que la etiqueta U se ha quedado obsoleta, debido a que es algo que realmente se debe hacer del lado del CSS, al ser básicamente un estilo.

<p><u> Texto subrayado</u> y texto normal</p>

Esto en una página se vería así:

## Texto subrayado y texto normal

### **ENCABEZADOS**

Existen otras etiquetas para definir párrafos especiales, que funcionaran como títulos de nuestra página. Son los encabezados o headings en inglés. Como decimos, son etiquetas que formatean el texto como un título, pero el hecho de que cambien el formato no es lo que nos tiene que preocupar, sino el significado en sí de la etiqueta. Es cierto que los navegadores asignan un tamaño mayor de letra y colocan el texto en negrita, pero lo importante es que sirven para definir la estructura del contenido de un documento HTML. Así los navegadores para ciegos podrán informar a los invidentes que esta es una división nueva de contenido y que su título es este o aquel. También motores de búsqueda sabrán interpretar mejor el contenido de una página en función de los títulos y subtítulos.

Hay varios tipos de encabezados, que se diferencian visualmente en el tamaño de la letra que utilizan. La etiqueta en concreto es la H1, para los encabezados más grandes, H2 para los de segundo nivel y así hasta H6 que es el encabezado más pequeño. Pero lo importante, insistimos es la estructura que denotan. Una página tendrá generalmente un encabezado de nivel 1 y dentro varios de nivel 2.

Luego, dentro de los H2 encontraremos si acaso H3, etc. Nunca debemos usar los encabezados porque nos formateen el texto de una manera dada, sino porque nuestro documento lo requiera según su estructura.

Los encabezados se verán de esta manera en la página:

# Encabezado de nivel 1

## Encabezado de nivel 2

### Encabezado de nivel 3

#### Encabezado de nivel 4

##### Encabezado de nivel 5

###### Encabezado de nivel 6

Los encabezados implican también una separación en párrafos, así que todo lo que escribamos dentro de H1 y su cierre (o cualquier otro encabezado) se colocará en un párrafo independiente.

Podemos ver cómo se presentan algunos encabezados a continuación.

```
<h1> Encabezado de nivel 1 </h1>
```

Los encabezados, como otras etiquetas de HTML, soportan el atributo align. Veremos un ejemplo de encabezado de nivel 2 alineado al centro, aunque repetimos que esto debería hacerse en CSS.

```
<h2 align="center"> Encabezado de nivel 2 </h2>
```

## LISTAS EN HTML

Las posibilidades que nos ofrece el HTML en cuestión de tratamiento de texto son realmente notables. No se limitan a lo visto hasta ahora, sino que van más lejos todavía. Varios ejemplos de ello son las listas, que sirven para enumerar y definir elementos.

Las listas originalmente están pensadas para citar, numerar y definir cosas a través de características, o al menos así lo hacemos en la escritura de textos. Sin embargo, las listas finalmente se utilizan para mucho más que enumerar una serie de puntos, en realidad son un recurso muy interesante para poder maquetar elementos diversos, como barras de navegación, pestañas etc.

Por ahora, trataremos las listas desde el punto de vista de su construcción y veremos los diferentes tipos que existen, y que podemos utilizar para resolver nuestras distintas necesidades a la hora de escribir textos en HTML.

Podemos distinguir dos tipos de listas HTML:

- Listas desordenadas
- Listas ordenadas

## LISTAS DESORDENADAS

Son delimitadas por las etiquetas UL y su cierre (unordered list). Cada uno de los elementos de la lista es citado por medio de una etiqueta LI (La LI tiene su cierre, aunque si no lo colocas el navegador al ver el siguiente LI interpretará que estás cerrando el anterior). La cosa queda así:

```
<p>Países del mundo</p>
```

```
<ul>
```

```
  <li>Argentina</li>
```

```
  <li>Perú</li>
```

```
  <li>Chile</li>
```

```
</ul>
```

Esto en una página se vería así:

#### Países del mundo

- Argentina
- Perú
- Chile

Podemos definir el tipo de viñeta empleada para cada elemento. Para ello debemos especificarlo por medio del atributo `type` incluido dentro de la etiqueta de apertura `UL`, si queremos que el estilo sea válido para toda la lista, o dentro de la etiqueta `LI` si queremos hacerlo específico de un solo elemento. La sintaxis es del siguiente tipo:

```
<ul type="tipo de viñeta">
```

Donde tipo de viñeta puede ser uno de los siguientes:

- `circle`
- `disc`
- `square`

Vamos a ver un ejemplo de lista con un cuadrado en lugar de un redondel, y en el último elemento colocaremos un círculo. Para ello vamos a colocar el atributo `type` en la etiqueta `UL`, con lo que afectará a todos los elementos de la lista.

```
<ul type="square">
```

```
  <li>Elemento 1 </li>
```

```
  <li>Elemento 2 </li>
```

```
  <li>Elemento 3 </li>
```

```
  <li type="circle">Elemento 4
```

```
</ul>
```

Esto en una página se vería así:

- Elemento 1
- Elemento 2
- Elemento 3
- Elemento 4

## LISTAS ORDENADAS

Las listas ordenadas sirven también para presentar información, en diversos elementos o ítems, con la particularidad que éstos estarán precedidos de un número o una letra para enumerarlos, siempre por un orden.

Para realizar las listas ordenadas usaremos las etiquetas OL (ordered list) y su cierre. Cada elemento será igualmente indicado por la etiqueta LI, que ya vimos en las listas desordenadas.

Pongamos un ejemplo:

`<p>Reglas de comportamiento en el trabajo</p>`

`<ol>`

`<li>El jefe siempre tiene la razón </li>`

`<li>En caso de duda aplicar regla 1 </li>`

`</ol>`

Esto en una página se vería así:

### Reglas de comportamiento en el trabajo

1. El jefe siempre tiene la razón
2. En caso de duda aplicar regla 1

Del mismo modo que para las listas desordenadas, las listas ordenadas ofrecen la posibilidad de modificar el estilo. En concreto nos es posible especificar el tipo de numeración empleado eligiendo entre números (1, 2, 3, ...), letras (a, b, c, ...) y sus mayúsculas (A, B, C, ...) y números romanos en sus versiones mayúsculas (I, II, III, ...) y minúsculas (i, ii, iii, ...).

Para realizar dicha selección hemos de utilizar, como para el caso precedente, el atributo type, el cual será situado dentro de la etiqueta OL. Los valores que puede tomar el atributo en este caso son:

- 1 Para ordenar por números
- a Por letras del alfabeto
- A Por letras mayúsculas del alfabeto
- i Ordenación por números romanos en minúsculas
- I Ordenación por números romanos en mayúsculas

Puede que en algún caso deseemos comenzar nuestra enumeración por un número o letra que no tiene por qué ser necesariamente el primero de todos. Para solventar esta situación,

podemos utilizar un segundo atributo, `start`, que tendrá como valor un número. Este número, que por defecto es 1, corresponde al valor a partir del cual comenzamos a definir nuestra lista. Para el caso de las letras o los números romanos, el navegador se encarga de hacer la traducción del número a la letra correspondiente.

Un ejemplo de todo esto sería:

```
<p>Ordenamos por números</p>
```

```
<ol type="1">
```

```
<li>Elemento 1 </li>
```

```
<li> Elemento 2 </li>
```

```
</ol>
```

```
<p>Ordenamos por letras</p>
```

```
<ol type="a">
```

```
<li>Elemento a </li>
```

```
<li> Elemento b </li>
```

```
</ol>
```

```
<p>Ordenamos por números romanos empezando por el 10</p>
```

```
<ol type="i" start="10">
```

```
<li>Elemento x </li>
```

```
<li> Elemento xi </li>
```

```
</ol>
```

Esto en una página se vería así:

Ordenamos por números

1. Elemento 1
2. Elemento 2

Ordenamos por letras

- a. Elemento a
- b. Elemento b

Ordenamos por números romanos empezando por el 10

- x. Elemento x
- xi. Elemento xi

## ANIDANDO LISTAS

Nada nos impide utilizar todas estas etiquetas de forma anidada como hemos visto en otros casos. De esta forma, podemos conseguir listas mixtas como, por ejemplo:

<p>Ciudades del mundo</p>

<ul>

<li>Argentina </li>

<ol>

<li>Buenos Aires </li>

<li>Bariloche </li>

</ol>

<li>Uruguay </li>

<ol>

<li>Montevideo </li>

<li>Punta del Este </li>

</ol>

</ul>

Esto en una página se vería así:

Ciudades del mundo

- Argentina
  1. Buenos Aires
  2. Bariloche
- Uruguay
  1. Montevideo
  2. Punta del Este

## ENLACES EN HTML

Hasta aquí, hemos podido ver que una página web es un archivo HTML en el que podemos incluir, entre otras cosas, textos formateados a nuestro gusto e imágenes (las veremos con detalle enseguida). Del mismo modo, un sitio web podrá ser considerado como el conjunto de archivos, principalmente páginas HTML e imágenes, que constituyen el contenido al que el navegante tiene acceso.

Sin embargo, no podríamos hablar de navegante o de navegación si estos archivos HTML no estuviesen debidamente conectados entre ellos y con el exterior de nuestro sitio por medio de enlaces hipertexto. En efecto, el atractivo original del HTML radica en la posible puesta en relación de los contenidos de los archivos introduciendo referencias bajo forma de enlaces que permitan un acceso rápido a la información deseada. De poco serviría en la red tener páginas aisladas a las que la gente no puede acceder y desde las que la gente no puede saltar a otras.

Un enlace puede ser fácilmente detectado por el usuario en una página. Basta con deslizar el puntero del ratón sobre las imágenes o el texto y ver cómo cambia de su forma original transformándose por regla general en una mano con un dedo señalador.

Adicionalmente, estos enlaces suelen ir, en el caso de los textos, coloreados y subrayados para que el usuario no tenga dificultad en reconocerlos.



## SINTAXIS DE UN ENLACE

Para colocar un enlace, nos serviremos de las etiquetas `a` y su cierre. Dentro de la etiqueta de apertura deberemos especificar asimismo el destino del enlace. Este destino será introducido bajo forma de atributo, el cual lleva por nombre "href".

La sintaxis general de un enlace es por tanto de la forma:

```
<a href="destino">contenido</a>
```

Siendo el "contenido" un texto o una imagen. Es la parte de la página que se colocará activa y donde deberemos pulsar para acceder al enlace. Por su parte, "destino" será una página, un correo electrónico o un archivo.

Por ejemplo, un enlace al home de Google, se vería así:

```
<a href="https://google.com.ar">Home de Google</a>
```

Ahora, si queremos que el contenido del enlace sea una imagen y no un texto, podremos colocar la correspondiente etiqueta `IMG` dentro de la etiqueta `a`. 

```
<a href="https://google.com.ar"></a>
```

Nota: veremos la etiqueta de imágenes más adelante.

## EL ASPECTO DE LOS ENLACES

Nosotros mediante el HTML, y las hojas de estilo CSS, podemos definir el aspecto que tendrán los enlaces en una página. Sin embargo, ya de manera predeterminada el navegador los destaca para que los podamos distinguir. Generalmente encontraremos a los enlaces subrayados y coloreados en azul, aunque esta regla depende del navegador del usuario y de sus estilos definidos como predeterminados.

## TIPOS DE ENLACES

Para estudiar en profundidad los enlaces tenemos que clasificarlos por su tipo, porque dependiendo ese tipo algunas cosas cambiarán a la hora de construirlos.

En función del destino los enlaces son clásicamente agrupados del siguiente modo:

- Enlaces locales: los que se dirigen a otras páginas del mismo sitio web.
- Enlaces remotos: los dirigidos hacia páginas de otros sitios web. Estos son los que vimos en el ejemplo anterior.

## ENLACES LOCALES

Como hemos dicho, un sitio web está constituido de páginas interconectadas, que se relacionan mediante enlaces de hipertexto. Para cumplir con esto es que vamos a utilizar los enlaces locales.

Los enlaces locales se tratan de un tipo de enlace mucho más común en el día a día del desarrollo. De hecho, es el tipo de enlace que más se produce en lo general. Estos enlaces locales nos permiten relacionar distintos documentos HTML que componen un sitio web.

Gracias a los enlaces locales podremos convertir varias páginas sueltas en un sitio web completo, compuesto de varios documentos.

Para crear este tipo de enlaces, hemos de usar la misma etiqueta A que ya conocemos, de la siguiente forma:

```
<a href="archivo.html">contenido</a>
```

## RUTAS DE LOS ENLACES

Hacer un enlace en si no es para nada complejo. No requiere muchas explicaciones con lo que ya hemos visto en la guía alcanza, sin embargo, hay que abordar con detalle un tema importante: las rutas de los enlaces. Como rutas nos referimos al destino del enlace, o sea, lo que ponemos en el atributo "href" y es importante que nos paremos aquí porque nos puede dar algunos problemas al desarrollar, sobre todo para las personas que puedan tener menos experiencia en el trabajo con el ordenador.

Por regla general, para una mejor organización, los sitios suelen estar ordenados por directorios. Estos directorios suelen contener diferentes secciones de la página, imágenes, scripts, estilos, etc. Es por ello que en muchos casos no nos valdrá con especificar el nombre del archivo, sino que tendremos que especificar además el directorio en el que nuestro archivo.html está alojado.

Para aquellos que no saben cómo mostrar un camino de un archivo, aquí van una serie de indicaciones que los ayudaran a comprender la forma de expresarlos. No resulta difícil en absoluto y con un poco de práctica lo harás prácticamente sin pensar.

1. Hay que situarse mentalmente en el directorio en el que se encuentra la página donde vamos a crear el enlace.
2. Si la página destino está en el mismo directorio que el archivo desde donde vamos a enlazar podemos colocar simplemente el nombre del archivo de destino, ya que no hay necesidad de cambiar de directorio.
3. Si la página de destino está en una carpeta o subdirectorio interior al directorio donde está el archivo de origen, hemos de marcar la ruta enumerando cada uno de los directorios por los que pasamos hasta llegar al archivo de destino, separándolos por el símbolo barra "/". Al final obviamente, escribimos el nombre del archivo destino.
4. Si la página destino se encuentra en un directorio padre (superior al de la página del enlace), hemos de escribir dos puntos y una barra "../" tantas veces como niveles subamos en la arborescencia hasta dar con el directorio donde está emplazado el archivo destino.
5. Si la página se encuentra en otro directorio no incluido ni incluyente del archivo origen, tendremos que subir como en la regla 3 por medio de ".." hasta encontrar un directorio que englobe el directorio que contiene a la página destino. A continuación, haremos como en la regla 2. Escribiremos todos los directorios por los que pasamos hasta llegar al archivo.

Imagina que tienes la siguiente estructura de carpetas y archivos. La que aparece en la siguiente imagen.



1) Para hacer un enlace a index.html

```
<a href="index.html">Ir a index.html</a>
```

2) Para hacer un enlace desde index.html hacia pagina1.html:

```
<a href="seccion1/paginas/pagina1.html">Ir a pagina1.html</a>
```

3) Para hacer un enlace desde pagina2 hacia pagina1:

```
<a href="../../seccion1/paginas/pagina1.html">Ir (también) a pagina1.html</a>
```

4) Para hacer un enlace desde pagina1 hacia pagina2:

```
<a href="../../seccion2/pagina2.html">Ir ahora a pagina2.html</a>
```

## IMÁGENES EN HTML

Sin duda uno de los aspectos más vistosos y atractivos de las páginas web es el grafismo. La introducción en nuestro texto de imágenes puede ayudarnos a explicar más fácilmente nuestra información y darle un aire mucho más estético. El abuso, no obstante, puede conducirnos a una sobrecarga que se traduce en una distracción para el navegante, quien tendrá más dificultad en encontrar la información necesaria.

El uso de imágenes también tiene que ser realizado con cuidado porque aumentan el tiempo de carga de la página, lo que puede ser de un efecto nefasto si nuestro visitante no tiene una buena conexión o si es un poco impaciente. Por ello es recomendable siempre optimizar las imágenes para Internet, haciendo que su tamaño en bytes sea lo mínimo posible, para facilitar la descarga, pero sin que ello comprometa mucho su calidad.

En esta guía no explicaremos como crear ni tratar las imágenes, únicamente diremos que para ello se utilizan aplicaciones como Paint Shop Pro, Photoshop o Gimp. Tampoco explicaremos las particularidades de cada tipo de archivo: GIF, JPG o PNG y la forma de optimizar nuestras imágenes.

La etiqueta que utilizaremos para insertar una imagen es IMG (image). Esta etiqueta no posee su cierre correspondiente y en ella hemos de especificar obligatoriamente el paradero de nuestro archivo gráfico mediante el atributo src (source).

La sintaxis queda entonces de la siguiente forma:

```

```

Para expresar el camino, lo haremos de la misma forma que vimos para los enlaces. Las reglas siguen siendo las mismas, lo único que cambia es que, en lugar de una página siendo el destino, el destino es un archivo gráfico. En el código anterior estamos enlazando un archivo con extensión .jpg, pero podrá ser otro tipo de archivo como .gif o .png.

Aparte de este atributo, indispensable obviamente para la visualización de la imagen, la etiqueta IMG nos propone otra serie de atributos de mayor o menor utilidad, que listamos a continuación:

## ATRIBUTO ALT

Dentro de las comillas de este atributo colocaremos una brevísima descripción de la imagen. Esta etiqueta no es indispensable, pero presenta varias utilidades. La sintaxis te quedaría de esta manera:

```

```

Primeramente, sirve para el posicionamiento de la página en buscadores. De los atributos alt el buscador puede extraer palabras clave y le ayuda a entender qué función o contenido tiene la imagen, y por lo tanto la página.

Otra utilidad importante la encontramos en determinadas aplicaciones, usadas por personas con discapacidad. Navegadores para ciegos, por ejemplo, no muestran las imágenes y por tanto los alt ofrecen la posibilidad de leerlas. Nunca está de más pensar en crear páginas accesibles.

Por último, durante el proceso de carga de la página y cuando la imagen no ha sido todavía cargada, el navegador podría mostrar esta descripción, con lo que el navegante se puede hacer una idea de lo que va en ese lugar. Si hubo problemas de conexión y no se pudo mostrar la imagen, también podría usarse ese alt para mostrar al menos su descripción.

En general podemos considerar como aconsejable el uso de este atributo, salvo para imágenes de poca importancia. Si la imagen es usada como cuerpo de un enlace todavía se hace más indispensable.

## ATRIBUTOS HEIGHT Y WIDTH

Estos atributos definen la altura y anchura respectivamente de la imagen en píxeles.

Aunque estas dimensiones forman parte del estilo de la imagen, y por tanto podrían ir en el CSS, todavía puede ser interesante definir las dentro del HTML. Esto, ya no es tan indispensable, puesto que muchos sitios creados con "Responsive Web Design" prefieren que las imágenes se adapten al tamaño de la pantalla donde se va a visualizar.

Todos los archivos gráficos poseen unas dimensiones de ancho y alto. Estas dimensiones pueden obtenerse a partir del propio diseñador gráfico o bien haciendo clic con el botón derecho sobre la imagen, vista desde el explorador de archivos de tu ordenador, para luego elegir "propiedades" o "información de la imagen" sobre el menú que se despliega.

Un ejemplo de etiqueta IMG con sus valores de anchura y altura declarados te quedaría así:

```

```

## IMÁGENES QUE SON ENLACES Y EL ATRIBUTO BORDER

Si un texto puede servir de enlace, una imagen puede cumplir la misma función:

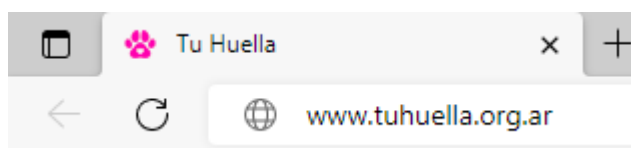
```
<a href="archivo.html"></a>
```

El problema de hacer esto en ciertos navegadores es que se crea un borde en la imagen, del mismo color que el color configurado para los enlaces, lo que suele ser un efecto poco deseado.

Sin embargo, en HTML podemos indicar que una imagen tenga borde. Mediante el atributo "border" se define el tamaño en píxeles del cuadro que rodea la imagen. De esta forma podemos recuadrar nuestra imagen si lo deseamos. No es algo que se use mucho, pero resulta particularmente útil cuando deseamos eliminar el borde que aparece cuando la imagen sirve de enlace. En dicho caso tendremos que especificar border="0".

## FAVICON O ÍCONO DE FAVORITOS

Se muestra una imagen de favicon a la izquierda del título de la página en la pestaña del navegador, como esta:



*Ejemplo de favicon*

El favicon aparecerá en tu lista de marcadores/favoritos y es esencial que coincida con el diseño de tu página y guarde una relación coherente con ella. Si tu página trata sobre una banda musical, sería extraño que el favicon sea la imagen de un pintor famoso, por ejemplo —excepto que se trate de La Oreja de Van Gogh.

Podés usar cualquier imagen que desees como su favicon. También puedes crear tu propio favicon en sitios como <https://www.favicon.cc>

Para agregar un favicon a tu sitio web, guardá tu imagen de favicon en el directorio raíz de su servidor web o creá una carpeta en el directorio raíz llamada imágenes y guardá tu imagen de favicon en esta carpeta. Un nombre común para una imagen de favicon es "favicon.ico".

A continuación, agregue un elemento <link> a tu archivo "index.html", después del elemento <title>, así:

```
<!DOCTYPE html>
```

```
<html lang="es-Ar">

<head>

  <title> Mi primera página </title>

  <link rel="icon" type="image/x-icon" href="/images/favicon.ico">

</head>

<body>

<h1> Este es un encabezado </h1>

<p> Este es un párrafo </p>

</body>

</html>
```

Ahora, guardá el archivo "index.html" y volvé a cargarlo en su navegador. La pestaña de tu navegador ahora debería mostrar tu imagen de favicon a la izquierda del título de la página.

**RECUERDA:** El enlace al favicon debe estar presente en cada documento HTML que conforme tu sitio web.

## EJERCICIOS DE APRENDIZAJE

1. Crear un archivo HTML, que contenga un encabezado <h1> seguido de un párrafo y un encabezado <h2> seguido de otro párrafo. Después, hacer que un párrafo se muestre en negrita y el otro en itálica. Utilizar saltos de línea si los consideran necesarios.
2. Ahora vamos a tener que centrar nuestros encabezados y nuestros párrafos alinearlos a la izquierda. A continuación, después del último párrafo vamos a tener que crear una lista ordenada de lo que queramos. Tendremos que mostrar la lista donde la enumeración sean letras del alfabeto.
3. Ahora crearemos otro archivo HTML en que crearemos una lista anidada de enlaces, deberá verse así:
  - Buscadores
    - Google
    - Bing
  - Redes sociales
    - Instagram
    - Twitter

---

Cada buscador y red social que sale en la lista deben ser links a las respectivas páginas.

4. Crear un nuevo archivo HTML en el que explicaremos la receta para hacer papas fritas. La página debería verse así:

## Papas fritas

Receta de papas fritas caseras



### Ingredientes

- 3 o 4 papas (300gr)
- Aceite
- Sal

### Elaboración (Pasos)

- Pelar las papas
- Cortalas en baston
- Calentar aceite en una sartén
- Cocinar hasta que esten doradas
- Removerlas del aceite y salar al gusto

5. Ahora vamos a crear una página web de nuestra banda favorita, vamos a mostrar un ejemplo con los Beatles:

## Los Beatles

Es una banda de rock formada en el año 1960 en Liverpool.



### Ingrantes

- Paul McCartney
- John Lennon
- Ringo Star
- George Harrison

Año	Disco
1965	Help!
1968	The Beatles
1969	Abbey Road
"Abbey Road fue su ultimo disco".	

La tabla tendrá bordes, así que investiga el atributo **border**.



6. Por último, vamos a crear una página que nos permita redirigirnos a nuestras páginas favoritas, pulsando sobre la imagen que represente cada web externa. Recuerda utilizar un favicon para poder distinguir mejor tu página de otras pestañas. Puedes reutilizar código del ejercicio 3.