



# Introducción

En las últimas décadas el alcance y el uso de internet se ha masificado inmensamente. Y en especial, en los últimos años se popularizaron las *web apps*, aplicaciones enteras que corren sobre un navegador web.

Estas *aplicaciones webs* tienen muchas ventajas sobre las aplicaciones tradicionales, pero para entender cómo funcionan, primero hay que empezar por lo básico.

En esta primera unidad vamos a ver cómo a partir de un archivo de texto plano, con cierta estructura, se puede “dibujar” distintos elementos en pantalla (como texto, imágenes, tablas, etc) dentro de un navegador. Spoiler: HTML

## Introducción a los lenguajes de marcado

Un lenguaje de marcado (o lenguaje de marcas) es una forma de codificar un texto que incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación.

Ejemplo: En Whatsapp existe un lenguaje de marcado muy muy simple, se puede poner el texto en *\*negrita\**, *\_cursiva\_* o *~tachada~* usando unos caracteres (asterisco, guion bajo, y la virgulilla o tilde de la ñ) de forma especial.

Existen muchos lenguajes de marcado, cada uno con un fin o uso específico. Pero todos hacen lo mismo, estructuran datos usando unas reglas muy específicas y predefinidas.

En el ejemplo de whatsapp, para que una frase esté en negrita se debe cumplir:

1. Empezar con un *\** (asterisco) pegado a una letra a la derecha
2. Terminar con otro *\** que no tenga una letra a su derecha.

Ejemplos:



- a. `* hola*` no se pone en negrita porque no se cumple la primera condición, hay un espacio al lado del primer asterisco.
- b. `*hola*chau` no resalta porque no se cumple la segunda condición, a la derecha del segundo `*` hay una letra.

Otros ejemplos de lenguajes de marcado son [Markdown](#), [YAML](#), [BBCode](#).



# Introducción al HTML

HTML es otro lenguaje de marcado, que significa **HyperText Markup Language** (Lenguaje de Marcas de HiperTexto), es la pieza más básica para la construcción de la web y se usa para definir el sentido y estructura del contenido en una página web.

Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. HTML se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

¿Para qué sirve?

- El HTML es el lenguaje de marcado estándar para crear páginas web.
- El documento HTML describe la estructura de una página web.
- El HTML consiste en una serie de elementos.
- Los elementos HTML le dicen al navegador cómo mostrar el contenido.
- Los elementos HTML etiquetan piezas del contenido como "esto es un encabezado", "esto es un párrafo", "esto es un enlace", etc.

En resumen, el navegador carga un archivo de texto a través de internet o desde un archivo local, el texto de este documento tiene un formato especial, descrito por el lenguaje HTML, que el navegador sabe interpretar para mostrarlo en pantalla.

## Historia del HTML

En 1980, el físico Tim Berners-Lee, un contratista del CERN, propuso y creó el prototipo de ENQUIRE, un sistema para que los investigadores del CERN utilizaran y compartieran documentos. En 1989, Berners-Lee escribió un memorando proponiendo un sistema de hipertexto basado en Internet. Berners-Lee especificó el HTML y escribió el software del navegador y del servidor a finales de 1990. Ese año, Berners-Lee y el ingeniero de sistemas de datos del CERN, Robert Cailliau, colaboraron en una solicitud conjunta de financiación, pero el proyecto no fue aprobado formalmente por el CERN.



La primera descripción de HTML disponible públicamente fue un documento llamado "Etiquetas HTML", mencionado por primera vez en Internet por Tim Berners-Lee a finales de 1991, en el que se describen 18 elementos que componen el diseño inicial, relativamente simple, de HTML. Once de estos elementos aún existen en HTML 4.

Berners-Lee consideraba el HTML una ampliación de [SGML](#) (un metalenguaje de marcado, es decir, un lenguaje para definir lenguajes), pero no fue formalmente reconocida como tal hasta la publicación a mediados de 1993, por la IETF (en español: Grupo de Trabajo de Ingeniería de Internet), de una primera proposición para una especificación del HTML: el borrador del Hypertext Markup Language de Berners-Lee y Dan Connolly, que incluía una Definición de Tipo de Documento SGML para definir la gramática. El borrador expiró a los seis meses, pero fue notable por su reconocimiento de la etiqueta propia del navegador Mosaic usada para insertar imágenes sin cambio de línea, que reflejaba la filosofía del IETF de basar estándares en prototipos con éxito. De la misma manera, el borrador competidor de Dave Raggett HTML+ (Hypertext Markup Format) (Formato de Marcaje de Hipertexto), de finales de 1993, sugería estandarizar características ya implementadas, como las tablas.

Después de que los borradores de HTML y HTML+ expiraran a principios de 1994, el IETF creó el HTML Working Group, que en 1995 completó "HTML 2.0", la primera especificación de HTML destinada a ser tratada como un estándar en el que deberían basarse las futuras implementaciones.

Desde 1996, el [World Wide Web Consortium](#) (W3C) ha mantenido las especificaciones del HTML. Sin embargo, en 2000, el HTML también se convirtió en una norma internacional (ISO/CEI 15445:2000). El HTML 4.01 se publicó a finales de 1999, y se publicaron nuevas erratas hasta 2001. En 2004, se comenzó a desarrollar el HTML5 en el [Web Hypertext Application Technology Working Group](#) (WHATWG), que se convirtió en un producto conjunto con el W3C en 2008, y se completó y estandarizó el 28 de octubre de 2014.

## Elemento HTML

HTML no es un lenguaje de programación, es un lenguaje de marcado que define la estructura de tu contenido. HTML consiste en una serie de elementos que usaremos para encerrar diferentes partes del contenido para que se vean o comporten de una determinada manera. Las etiquetas de encierre pueden hacer de una palabra o una imagen un hipervínculo a otro sitio, se pueden cambiar



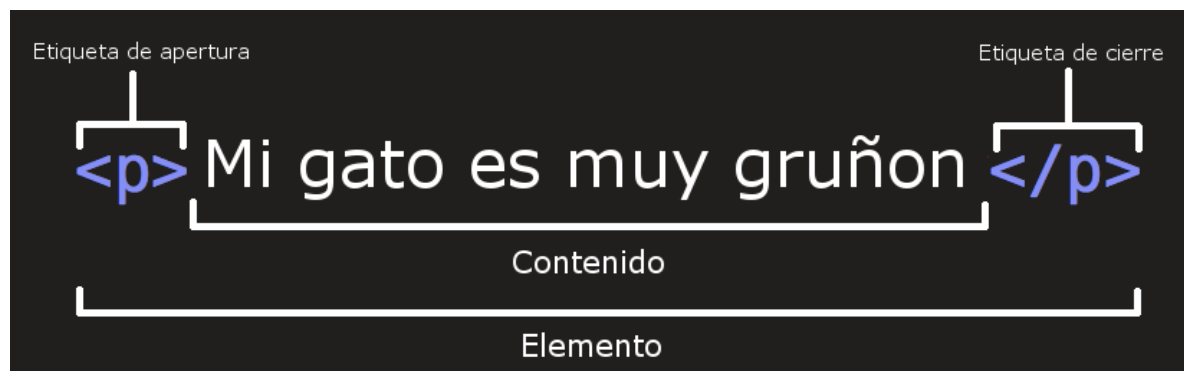
palabras a cursiva, agrandar o achicar la letra, etc. Por ejemplo, tomemos la siguiente línea de contenido:

```
Mi gato es muy gruñón
```

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

```
<p>Mi gato es muy gruñón</p>
```

## Anatomía de un elemento



Las partes principales del elemento son:

1. La **etiqueta de apertura**: consiste en el nombre del elemento (en este caso, `p`), encerrado por paréntesis angulares (`<` `>`) de apertura y cierre. Establece dónde comienza o empieza a tener efecto el elemento (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 el elemento (en este caso dónde termina el párrafo).
3. El **contenido**: este es el contenido del elemento, que en este caso es sólo texto.
4. El **elemento**: la etiqueta de apertura, más la etiqueta de cierre, más el contenido equivale al elemento.

Los elementos también pueden tener atributos:



Los atributos contienen información adicional acerca del elemento, la cual no queremos que aparezca en el contenido real del elemento. Aquí “class” es el nombre del atributo y “editor-note” el valor del atributo. En este caso, el atributo class permite darle al elemento un nombre identificativo, que se puede utilizar luego para apuntarle al elemento información de estilo y demás cosas (que veremos en las próximas clases).

Un atributo debe tener siempre:

1. Un espacio entre este y el nombre del elemento (o del atributo previo, si el elemento ya posee uno o más atributos).
2. El nombre del atributo, seguido por un signo de igual (=).
3. Comillas de apertura y de cierre, encerrando el valor del atributo.
4. Los atributos siempre se incluyen en la etiqueta de apertura de un elemento, nunca en la de cierre.

El nombre de un elemento dentro de una etiqueta no es sensible a mayúsculas. Esto quiere decir que se puede escribir en mayúsculas, minúsculas o una combinación. Por ejemplo, la etiqueta `<title>` puede ser escrita como `<Title>`, `<TITLE>` o de cualquier otra forma.

## Anidar elementos

También podemos colocar elementos dentro de otros elementos —esto se llama **anidamiento**—. Si, por ejemplo, si queremos resaltar una palabra del texto (en el ejemplo la palabra «muy»), podemos encerrarla en un elemento `<strong>`, que significa que dicha palabra se debe enfatizar:

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

Debemos asegurarnos que los elementos estén correctamente anidados: en el ejemplo de abajo, creamos la etiqueta de apertura del elemento `<p>` primero,



luego la del elemento `<strong>`, por lo tanto, debemos cerrar esta etiqueta primero, y luego la de `<p>`. Esto es **incorrecto**:

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

Los elementos 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.**

## Elementos vacíos

Algunos elementos no poseen contenido, y son llamados **elementos vacíos**. Tomemos, por ejemplo, el elemento `<img>`:

```

```

Posee dos atributos, pero no hay etiqueta de cierre `</img>` ni contenido encerrado. Esto es porque un elemento de imagen no encierra contenido. Su propósito es desplegar una imagen en la página HTML, en el lugar en que aparece.



## Estructura de un documento HTML

Hasta ahora hemos visto algunos elementos básicos de HTML, pero estos no son muy útiles por sí solos. Ahora veremos cómo los elementos individuales son combinados para formar una página HTML entera:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Titulo</title>
  </head>
  <body>
    <p>Mi gato es <strong>muy</strong> gruñón.</p>
  </body>
</html>
```

Tenemos:

- `<!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 no tiene mucha importancia, pero que debe ser incluido para que todo funcione correctamente. Por ahora, eso es todo lo que necesitas saber.
- `<html></html>` — el elemento `<html>`. Este elemento encierra todo el contenido de la página entera y, a veces, se le conoce como el elemento raíz (*root element*).
- `<head></head>` — el elemento `<head>`. Este elemento 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, 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">` — `<meta>`. Este elemento 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

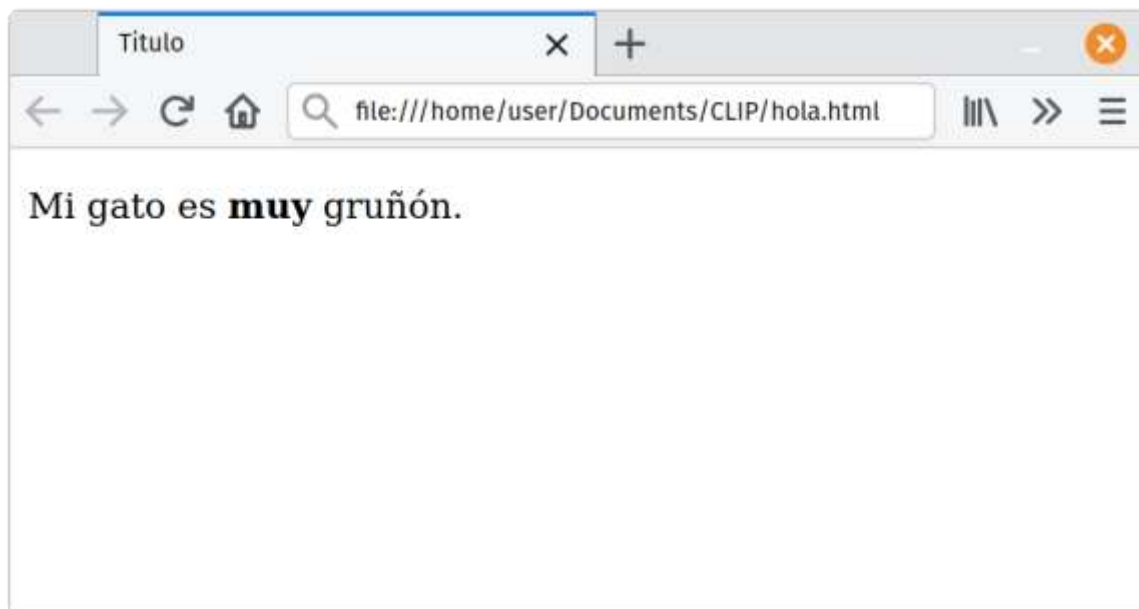




cualquier contenido de texto que puedas incluir. No hay razón para no establecerlo, y puede evitar problemas en el futuro.

- `<title></title>` — el elemento `<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.
- `<body></body>` — el elemento `<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.

Si guardamos el ejemplo en un archivo de texto llamado hola.html y lo abrimos con un navegador web, vamos a ver algo como esto:





## Más elementos HTML

Hasta ahora hemos visto la base y fundamentos del HTML, que forma tiene un documento y algunas etiquetas básicas. Ahora vamos a ver otras etiquetas muy populares y útiles.

### Encabezados

Los elementos de encabezado permiten especificar que ciertas partes del contenido son encabezados, o subencabezados del contenido. De la misma forma que un libro tiene un título principal, y que a su vez puede tener títulos por cada capítulo individual, y subtítulos dentro de ellos, un documento HTML puede tenerlos también. HTML posee seis niveles de encabezados, [<h1>](#)–[<h6>](#).

```
<h1>Mi título principal</h1>
<h2>Mi título de nivel superior</h2>
<h3>Mi subtítulo</h3>
<h4>Mi sub-subtítulo</h4>
```

### Párrafos

Como se explicó más arriba, los elementos [<p>](#) se utilizan para encerrar párrafos de texto; los usaremos frecuentemente para el marcado de contenido de texto regular.

```
<p>Este es un simple párrafo</p>
```

### Listas

Mucho del contenido web está dado por listas, así que HTML tiene elementos especiales para ellas. El marcado de listas se realiza siempre en al menos dos elementos. Los dos tipos de listas más comunes son las listas ordenadas y las desordenadas:

1. Las **listas desordenadas** son aquellas en las que el orden de los ítems no es relevante, como en una lista de compras. Estas son encerradas en un elemento [<ul>](#) (*unordered list*).



2. Las **listas ordenadas** son aquellas en las que el orden sí es relevante, como en una receta. Estas son encerradas en un elemento `<ol>` (*ordered list*).

Cada elemento de la lista se coloca dentro de un elemento `<li>` (list item).

```
<p>Lista de items:</p>
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

## Links

Los hipervínculos son una de las novedades más emocionantes que nos ofrece la web. De acuerdo que llevan formando parte de esta desde el principio, pero hacen que la web sea web: permiten enlazar nuestros documentos con cualquier otro documento (o recurso), hacer referencia a partes específicas de los mismos, podemos hacer las aplicaciones accesibles con una sencilla dirección web (al contrario de las apps nativas, que tiene que instalarse). Cualquier contenido web puede convertirse en un vínculo que, al pulsarlo (activarlo), dirija el navegador a la dirección web a la que apunte el enlace ([URL](#)).

Un enlace básico se crea incluyendo el texto (o cualquier otro contenido) que queramos convertir en un enlace en un elemento ancla `<a>`, dándole un atributo `href` (también conocido como **target** u **objetivo**) que contendrá la dirección web hacia dónde queremos que apunte el enlace. La *a* es la abreviatura de la palabra inglesa «anchor» («*ancla*»).

```
<a href="https://www.cba.gov.ar/programaclip/">CLIP</a>
```

Otro atributo que podemos añadir a los enlaces es el título (`title`); ha sido concebido para proporcionar información útil suplementaria sobre el destino, como el tipo de información que contiene la página o cuestiones que es necesario tener en cuenta sobre el mismo. Por ejemplo:



```
<p>Creo un enlace a  
<a href="https://www.cba.gov.ar/programaclip/"  
title="Programa de inserción laboral y capacitación en  
nuevas tecnologías">la página de inicio de CLIP</a>.  
</p>
```

El título de un enlace solo será visible al pasar el ratón por encima, lo que significa que los usuarios que naveguen usando los controles de sus teclados tendrán dificultades para acceder a la información proporcionada por el título. Si la información del título es verdaderamente importante para el uso de la página, deberemos presentarla de manera que sea accesible a todos los usuarios, por ejemplo incluyéndola como parte del texto del enlace.

## Bibliografía

- [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)
- <https://developer.mozilla.org/es/docs/Learn/HTML>
- <http://elies.rediris.es/elies24/fernandezvalmayor.htm>