

HTML



Lenguaje sencillo compuesto de elementos.

- Da significado a distintas piezas de texto
- Estructura un documento en secciones lógicas
- Incrusta contenido en una página

¿Qué es el HTML?

- Hypertext Markup Language
- No es un lenguaje de programación: es un lenguaje de marcado.
- Le dice a los navegadores web cómo estructurar las páginas.
- Consiste en una serie de elementos, que se pueden manipular para que aparezcan o actúen de una cierta manera.



Las etiquetas en HTML no son case sensitive. La mejor práctica es escribir todas las etiquetas en minúsculas por coherencia y legibilidad.

Anatomía de un elemento HTML

Las partes de un elemento son:

1. Etiqueta de apertura: es el nombre del elemento, encerrado entre <>. Esto marca donde comienza el elemento.
2. Contenido: contenido del elemento.
3. Etiqueta de cierre: lo mismo que la etiqueta de apertura, pero incluye un slash antes del nombre del elemento. Indica donde termina el elemento.

Elementos anidados

Se pueden poner elementos dentro de otros elementos. Para un anidamiento adecuado primero se debe cerrar el elemento interno antes de cerrar el externo.

Elementos de bloque y elementos en línea

Elementos de bloque

- Forman un bloque visible en la página
- Aparecerán en una línea nueva antes y después de cualquier contenido. Esto se debe al estilo CSS predeterminado que el navegador le aplica a este tipo de elementos
- Suelen ser elementos estructurales de la página
- Ejemplos: encabezados, párrafos, listas, menús, pies de página
- **Puede estar anidado dentro de otros elementos a nivel de bloque**

Elementos en línea

- Contenidos dentro de elementos de bloque
- Delimitan partes chicas del contenido

- No hará que aparezca una nueva línea en el documento
- Ejemplos: a, em, strong

Elementos vacíos

Algunos elementos solo consisten solo en una etiqueta, que se usa para incrustar algo en el documento.

Ejemplos: img, br.

Atributos

Contienen info extra sobre el elemento que no se mostrará en el contenido. Deben tener:

- Un espacio entre el nombre del elemento y el atributo, y entre cada atributo
- El nombre del atributo seguido por un signo de igual
- Un valor rodeado de comillas



Ejemplo <a>:

- href: indica la dirección a la que quiere que apunte el enlace
- title: suma info adicional sobre el enlace, que se mostrara cuando se haga hover sobre el link
- target: especifica el contexto de navegación que va a usar para mostrar el enlace

Atributos booleanos

Solo pueden tener un valor, y usualmente es el mismo nombre del atributo aunque también se puede omitir y solo poner el nombre del atributo. Ejemplo: disabled, que se usa para deshabilitar inputs.



Siempre envolver los valores de los atributos entre comillas para evitar errores. Mientras se sea consistente, se puede usar tanto comillas simples como dobles.

Anatomía de un documento HTML

1. `<!DOCTYPE html>` : antes, los doctypes servían como enlaces al conjunto de reglas que la página HTML debía seguir. Ahora se incluye para que todo funcione correctamente y es histórico.
2. `<html>` : envuelve todo el contenido, se lo conoce como elemento raíz.
3. `<head>` : actúa como contenedor para todos los parámetros que no serán visibles a los visitantes de la página. Por ejemplo - palabras clave, descripción de la página para mostrar en los resultados de búsqueda, hoja de estilo, etc.
4. `<meta charset="utf-8">` : establece que el documento usará la codificación UTF-8, para poder gestionar cualquier contenido de texto que pongas en el documento.
5. `<title>` : establece el título de la página que aparece en la pestaña del navegador.
6. `<body>` : contiene todo el contenido que quieras mostrar a los usuarios.



El intérprete de HTML siempre reduce cada secuencia de espacio en blanco a un único espacio al interpretar código.

Referencias a entidades: inclusión de caracteres especiales en HTML

Los caracteres <, >, ", ' y & son especiales.

Caracter	Equivalente en HTML
<	<
>	>
"	"
'	'
&	&

Comentarios

Se debe delimitar con los marcadores especiales `<!-- THIS IS A COMMENT -->`.

Metadatos en HTML

¿Qué hay en la cabecera HTML?

La cabecera HTML es el contenido del elemento `<head>`, y no se muestra en la página. La función de la cabecera es contener los metadatos del documento.

Añadir un título

El elemento `<title>` es un metadato que representa el título de todo el documento HTML, no del contenido del documento. También se utiliza cuando se marca la página como favorito, y en los resultados de las búsquedas.

El elemento `<meta>`

Los metadatos son datos que describen datos, y en HTML se introducen al documento con el elemento `<meta>`.

Añadir un autor y descripción

```
<meta name="author" content="Chris Mills">
```

- name especifica el tipo de metadato
- content especifica el contenido del metadato en si



Especificar un autor es útil para saber quien escribió la página. Incluir una descripción de la página puede mejorar el SEO.

Íconos personalizados

Se puede agregar en los metadatos referencias a iconos personalizados. El más común es favicon.

```
<!-- Guardar el favicon en el mismo directorio que index.html en formato .ico-->
<!-- Agregar esto al head -->
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
```

Aplicar CSS y JS a HTML

La manera más habitual es con los elementos `<link>` y `<script>`.

El elemento `<link>` siempre debe ir dentro del `<head>` de tu documento. Este toma dos atributos, `rel="stylesheet"`, que indica que es la hoja de estilo del documento, y `href`, que contiene la ruta al archivo de la hoja de estilo.

El elemento `<script>` también debería ir en el `head`, y debería incluir un atributo `src` con la ruta al JavaScript que quieres cargar, y `defer`, que básicamente le dice al navegador que cargue el JavaScript al mismo tiempo que el HTML de la página. → Este approach es bomb proof

Establecer el idioma principal

Esto se puede hacer añadiendo el atributo `lang` a la etiqueta de apertura del HTML.



- Los navegadores indizarán el doc HTML más efectivamente si se incluye el idioma.
- Es útil para personas con discapacidad visual que utilizan lectores de pantalla.

Fundamentos de texto en HTML

Crear hipervínculos



Una URL puede apuntar a archivos HTML, archivos de texto, imágenes, documentos de texto, archivos de audio o vídeo, y cualquier otra cosa que se pueda mostrar en la web. Si el navegador no sabe cómo manejar el archivo, te preguntará si lo quieres abrir (en cuyo caso la tarea de abrirlo y manejarlo se transferirá a la aplicación nativa instalada en el dispositivo) o si lo quieres descargar (en cuyo caso podrás ocuparte de él más tarde).

Anatomía de un enlace

Un enlace se crea incluyendo contenido que queramos convertir en un enlace usando un elemento `<a>`, dándole un atributo `href` (target).

Añadir info de asistencia con el atributo `title`

El título contiene info adicional sobre el enlace, como qué tipo de info contiene la página o cosas que debes tener en cuenta en el sitio web.

El título de un enlace solo será visible al pasar el ratón por encima, lo cual significa que los usuarios que naveguen usando los controles de sus teclados, o pantallas táctiles, tendrán dificultades para acceder a la información proporcionada por el título.

Convertir bloques de contenido en enlaces

Encerrar el elemento en un elemento `<a>`.

Primer acercamiento a URLs y rutas



Una localizadora uniforme de recursos (URL, de las iniciales en inglés de «*Uniform Resource Locator*») es simplemente una secuencia de caracteres de texto que definen donde está situado algo en la web.

Las URLs utilizan rutas para encontrar los archivos. Las rutas especifican dónde se encuentra el archivo que buscas dentro del sistema de archivos.



Para «Subir un nivel» utilizamos los dos puntos — `(..)`

Fragmentos de documento

Es posible apuntar hacia una parte concreta de un documento HTML en vez de a todo un documento. Para ello hay que asignar previamente un atributo `id` al elemento hacia el que apuntamos.

Para hacer referencia a este `id` concreto, lo añadiremos al final de la URL precedido por una almohadilla.

```
<h2 id="Dirección_de_envío">Dirección de envío</h2>
<p>¿Quieres mandarnos una carta? Aquí tienes nuestra <a href="contacts.html#Dirección_de_envío">Dirección de envío</a>.</p>
```

URLs absolutas y relativas

- **URL absoluta:** Hace referencia a una dirección definida por su ubicación absoluta en la web, esta incluye el protocolo y el nombre del dominio. Una URL absoluta siempre apuntará a la misma dirección, sin importar desde dónde se utilice.
- **URL relativa:** Hace referencia a una dirección que depende de la posición del archivo desde donde se utiliza. Una URL relativa hará referencia a diferentes direcciones según dónde se encuentre el archivo desde el cual se utiliza.

Buenas prácticas en el uso de enlaces

Se debe lograr que los enlaces sean accesibles para todo tipo de lectores:

- Incluir palabras claves al definir el texto de los enlaces para describir de forma efectiva el sitio al que apuntan
- No repetir la URL como parte del texto
- No escribir «*link*» o «*link a*» o «enlace» o «enlace a» en el texto del enlace porque es redundante.
- Redactar la etiqueta del enlace de manera breve y concisa
- Minimizar los casos en los que varias copias del mismo texto están vinculadas a diferentes lugares

Utilizar rutas relativas siempre que sea posible

1. Porque el código es más fácil de leer
2. Cuando se usa una ruta absoluta, el navegador busca la ubicación real del servidor en el DNS. Va a ese servidor y busca el archivo solicitado. Si se usa rutas relativas, el navegador busca el archivo en el mismo servidor. Si se usan siempre rutas absolutas cuando podrían usarse relativas estás haciendo que el navegador haga trabajo de más, reduciendo la performance de la web.

Utilizar el atributo download al enlazar una descarga

Sirve para hacer referencia a una descarga en lugar de a algo que abra el navegador.

Enlace a correo electrónico

Se consigue con el elemento `<a>` y el elemento `mailto:`:

```
<a href="mailto:nowhere@mozilla.org">Enviar correo electrónico a ninguna parte</a>
```

El atributo con la dirección de correo electrónico es opcional. Si lo omites y tu `href` simplemente es "mailto:", aparecerá una nueva ventana de correo saliente en el gestor de correo sin la dirección del destinatario.

Además de la dirección de correo electrónico, puedes proporcionar otra información. Los más utilizados son el «`subject`» (asunto), «`cc`» (con copia a) o «`bcc`» (copia oculta), y «`body`» (cuerpo del mensaje).



Los valores de cada campo deben tener la URL codificada, es decir, sin caracteres no imprimibles (caracteres invisibles, tabulaciones, retornos de carro y saltos de página) y espacios con codificación porcentual

Tablas HTML



Una tabla es un conjunto estructurado de datos distribuidos en filas y columnas (**datos tabulados**).

Una tabla HTML bien hecha debe mejorar la experiencia de los usuarios videntes e invidentes por igual.

Para crear tablas HTML se usa el elemento `<table>`.

No usar tablas HTML para hacer compaginaciones de páginas web

- Reducen la accesibilidad para usuarios con discapacidad visual
- Generan estructuras incorrectas
- No son responsive

Encabezados con elementos `<th>`

Los encabezados son una celda especial que van al principio de una fila o columna. Estos pueden agregarse con el elemento `<th>` que significa table header.

Junto con el atributo `scope` mejoran la accesibilidad de las tablas porque asocian cada encabezado con todos los datos de la misma fila o columna.

Celdas que abarcan varias filas y columnas

Los atributos `colspan` y `rowspan` nos permiten hacer que una celda abarque más de una columna o fila, respectivamente.

Estilo común a las columnas

Los elementos `<col>` y `<colgroup>` existen para especificar el estilo de una columna completa de datos. Los elementos `col` se especifican dentro de un contenedor `<colgroup>`, debajo de la apertura de `<table>`.

```
<table>
  <colgroup>
    <col>
    <col style="background-color: yellow">
  </colgroup>
```

Se definen dos tipos de columnas: la primera no tiene estilos particulares, esto se hace para que no se le apliquen los estilos de las que si customizamos. Si quisiéramos aplicar la información de estilo a ambas columnas, podríamos incluir un elemento `<col>` con un atributo `span` :

```
<colgroup>
  <col style="background-color: yellow" span="2">
</colgroup>
```