



HTML en profundidad

Ya hemos visto un pantallazo general sobre cómo funciona HTML, ahora veremos con más detalles ciertos aspectos del lenguaje.

Atributos booleanos

Está permitido usar atributos sin valores. Se denominan atributos booleanos y pueden tener únicamente un valor, que como norma general es el mismo que el nombre del atributo. Por ejemplo, el atributo `disabled` que se puede asignar a elementos de entrada de formularios si se desea desactivarlos para que el usuario no pueda introducir datos en estos.

```
<input type="text" disabled="disabled">
```

De manera abreviada, también es posible escribirlo como se describe a continuación:

```
<input type="text" disabled>
```

Caracteres especiales

En HTML, los caracteres `<`, `>`, `"`, `'` y `&` son caracteres especiales. Forman parte de la sintaxis HTML, pero ¿cómo se pueden incluir en el texto si realmente queremos hacer uso de un ampersand (&) o el signo menor que (<), y que el navegador no lo interprete como código?

Tenemos que utilizar caracteres de referencia, es decir, códigos especiales que representan caracteres. Cada carácter de referencia comienza con un signo de ampersand (&) y finaliza con un punto y coma (;).

Carácter literal	Equivalente de referencia de caracteres
------------------	---



<	<
>	>
"	"
'	'
&	&

Consideremos el siguiente ejemplo:

```
<p>En HTML, defines un párrafo con el elemento <p>.</p>  
<p>En HTML, defines un párrafo con el elemento &lt;p&gt;.</p>
```

Si lo probamos en el navegador, podemos ver que el primer párrafo ha salido mal porque el navegador cree que la segunda instancia de `<p>` comienza un párrafo nuevo. El segundo párrafo se ve bien porque hemos reemplazado `<` y `>` por sus referencias correspondientes.

Comillas en los atributos

Cuando observamos diferentes páginas web, podemos encontrarnos con todo tipo de estilos de etiquetado extraños, que incluyen valores de atributos sin comillas. Esto se permite en ciertas circunstancias, pero no siempre. Por ejemplo, si volvemos a revisar el ejemplo de enlace, sería posible escribir una versión básica con solo el atributo `href`, como este:

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



Sin embargo, las cosas no funcionarán cuando a este estilo le añadimos el atributo `title`:

```
<a href=https://www.cba.gov.ar/programaclip/ title=Página principal de CLIP>CLIP</a>
```

En este caso, el navegador interpretará que el atributo `title` corresponde a tres atributos: un atributo `title` con el valor `Página` y tres atributos booleanos: `principal`, `de` y `CLIP`. Obviamente, esto no es lo que queremos, porque causa errores o un comportamiento inesperado. Se aconseja incluir **siempre** las comillas de los atributos para evitar este tipo de problemas y producir un código más legible.

En HTML, se puede usar tanto comillas simples como dobles. Esto es un simple tema de estilo. Por ejemplo, ambas líneas de código son equivalentes.

```
<a href="http://www.example.com">Un enlace a mi ejemplo.</a>
<a href='http://www.example.com'>Un enlace a mi ejemplo.</a>
```

Pero no debemos mezclar ambos tipos de comillas. El ejemplo siguiente está mal:

```
<a href= http://www.example.com >Un enlace a mi ejemplo.</a>
```

Si usamos un tipo de comillas, podemos incluir el otro tipo de comillas para los valores de atributo sin que esto cree problemas:

```
<a href="http://www.example.com" title="¿Es 'divertido'?">Un
enlace a mi ejemplo.</a>
```

Sin embargo, si queremos añadir unas comillas dentro de otras del mismo tipo (ya sea simples o dobles), tendremos que los caracteres de referencia, como recién vimos. Por ejemplo, esto no va a funcionar:



```
<a href='http://www.example.com' title='¿Es divertido ?'>Un  
enlace a mi ejemplo.</a>
```

Así que tendremos que hacer lo siguiente:

```
<a href='http://www.example.com' title='¿Es  
&#39;divertido&#39;?'>Un enlace a mi ejemplo.</a>
```

Espacios en blanco

Los dos siguientes fragmentos de código son equivalentes:

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

No importa cuántos espacios en blanco se usen (se pueden incluir tanto caracteres de espacio como saltos de línea) el analizador de HTML reduce cada uno a un único espacio al representar el código. ¿Entonces por qué utilizar espacios en blanco? La respuesta está en la legibilidad: es mucho más fácil entender el código si está bien formateado que si está desordenado. En nuestro HTML cada elemento anidado tiene dos espacios de sangría más con respecto al que está por fuera. Depende de ti qué estilo de formato utilices (cuántos espacios para cada nivel de sangría, por ejemplo), pero lo normal es usar 2 o 4 espacios por cada nivel.



Comentarios

Como en la mayoría de los lenguajes de programación, en HTML hay un mecanismo para escribir comentarios en el código. Los comentarios son ignorados por el navegador y son invisibles para el usuario. Su propósito es permitir comentar el código para aclarar su funcionamiento, explicar lo que hacen las diferentes partes del código, etc. Esto es muy útil como recordatorio del trabajo que has hecho, si vuelves a trabajar con un código en el que no has trabajado durante meses o si entregas tu código a otra persona para que trabaje con él.

Para convertir en un comentario una sección de contenido de tu archivo HTML, debemos delimitarlo con los marcadores especiales `<!--` y `-->`, por ejemplo:

```
<p>No soy ningún comentario</p>
```

```
<!-- <p>¡Sí lo soy!</p> -->
```

Metadatos

Como vimos en el ejemplo de la clase pasa, en la cabecera (`<head>`) de un documento HTML es la parte que no se muestra en el navegador cuando se carga la página. Contiene información como el título (`<title>`) de la página, enlaces al CSS (si queremos aplicar estilo al contenido HTML con CSS, lo veremos en el siguiente módulo), enlaces para personalizar el favicon, y otros metadatos (como quién lo escribió o palabras claves importantes que describen el documento).

Repasemos el head del ejemplo de la clase pasada:

```
<head>
  <meta charset="utf-8">
  <title>Titulo</title>
</head>
```



Es una cabecera bastante simple, pero en páginas más grandes, la cabecera puede llegar a contener bastantes elementos. Nuestro objetivo no es ver cómo usar todo lo que podemos incluir en la cabecera, sino ver sólo los elementos principales para darte bases sólidas y el resto lo irás aprendiendo a medida que lo necesites. Empecemos.

Título

Ya hemos visto el elemento `<title>` en acción: se puede usar para poner un título al documento. Sin embargo, no hay que confundirlo con el elemento `<h1>`, que se utiliza para poner un encabezado de nivel superior al cuerpo de tu contenido. Esto también se conoce como el título de la página.

- El elemento `<h1>` aparece en la página cuando se carga en el navegador. En general debería haber uno solo por página, para especificar el título del contenido de tu página (el título de tu historia, el título de la noticia, o de lo que sea apropiado según el uso que le des).
- El elemento `<title>` es un metadato que representa el título de todo el documento HTML (no del contenido del documento).

Actividad: Copia y pega el código de la página title-example.html(1) en un archivo de texto nuevo en tu editor de código, luego guárdalo en un lugar conveniente. Ábrelo en el navegador y observa en qué lugares aparece “`<h1> element`” y “`<title> element`”.

Elemento `<meta>`

Los metadatos son datos que describen datos, y HTML tiene una forma «oficial» de introducir metadatos en un documento: el elemento `<meta>`. Por supuesto, el resto de elementos en el encabezado también se consideran metadatos.

Codificación de caracteres

Este elemento simplemente especifica la codificación de caracteres del documento: el conjunto de caracteres que el documento puede usar. [utf-8](#) es un conjunto universal de caracteres que incluye casi todos los caracteres de casi cualquier idioma humano. Existen otras codificaciones, como [ISO-8859-1](#) (el juego de caracteres para el alfabeto latino), pero normalmente veremos utf-8 ya que es un estándar.



```
<meta charset="utf-8">
```

Autor y descripción

Muchos elementos `<meta>` incluyen atributos `name` y `content`:

- `name` especifica el tipo de metadato del que se trata; es decir, qué tipo de información contiene.
- `content` especifica el contenido del metadato en sí.

Dos de esos metadatos que resultan útiles de incluir definen al autor de la página y proporcionan una descripción concisa. Veamos un ejemplo:

```
<meta name="author" content="Chris Mills">  
<meta name="description" content="El área de aprendizaje de  
MDN pretende proporcionar a los recién llegados a la web  
todo lo que deben saber para empezar a desarrollar páginas  
web y aplicaciones web.">
```

Especificar un autor resulta ventajoso por diversos motivos: es útil saber quién escribió la página y para poder ponerte en contacto con el autor si tienes preguntas sobre el contenido. Algunos sistemas de gestión de contenido tienen herramientas para extraer automáticamente la información del autor de la página y ponerla a disposición para tales fines.

Especificar una descripción que incluya palabras clave relacionadas con el contenido de tu página resulta útil porque tiene el potencial de hacer que la página aparezca más arriba en las búsquedas relevantes que efectúan los motores de búsqueda .

Metadatos específicos

Al navegar por la web también puedes encontrar otros tipos de metadatos. Muchas de las funciones que verás en los sitios web son creaciones propietarias diseñadas para proporcionar a ciertos sitios (como los sitios de redes sociales) información específica que puedan usar.



Por ejemplo, [Open Graph Data](#) es un protocolo de metadatos que Facebook inventó para proveer metadatos más ricos para los sitios web. Por ejemplo, la web de [MDN](#):

```
<meta property="og:image"
content="https://developer.mozilla.org/static/img/opengraph-
logo.72382e605ce3.png">
<meta property="og:title" content="MDN Web Docs">
<meta property="og:description" content="El sitio MDN Web Docs
proporciona información acerca de las tecnologías de la web abierta
incluyendo HTML, CSS, y APIs tanto [...]">
```

Un efecto de esto es que cuando desde Facebook enlazas a un MDN, el enlace aparece con una imagen y una descripción, lo que resulta una experiencia más enriquecedora para los usuarios.

Actividad: Comparte el link <https://developer.mozilla.org/es/> en alguna red social o mensajería instantánea y observa qué información aparece.

No vamos a ver todos los metadatos que existen, pero con esto ya van a poder entender y saber que significan estos atributos.

Favicon

El favicon (abreviatura de **favorite icons**, **icono de «favoritos»**, referido al uso que se le da en las listas de «favoritos» o de marcadores (**bookmarks**)).

El humilde favicon, que existe desde ya hace muchos, muchos años, fue el primer icono de este tipo, un icono de 16x16 píxeles usado en múltiples sitios. Para añadir un favicon a una página deberemos:

1. Guardarlo en el mismo directorio que la página **index** del sitio, en formato **.ico** (la mayoría de los buscadores admiten favicons en los formatos más comunes como **.gif** o **.png**, pero usar el formato **ICO** garantiza que funcionará hasta en Internet Explorer 6.)
2. Añadir a la cabecera del HTML (**<head>**) la siguiente línea:



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

Los navegadores modernos usan favicons en diversos lugares, como en la pestaña de la página que está abierta o en el panel de favoritos cuando la añades a tu lista de páginas favoritas.

Idioma

Por último, merece la pena mencionar que podemos (y realmente debemos) especificar el idioma de las páginas web. Esto se puede hacer añadiendo el atributo lang a la etiqueta de apertura del HTML, como se muestra abajo:

```
<html lang="en-US">
```

Esto resulta útil en muchos sentidos. Los motores de búsqueda van a indexar el documento HTML de modo más efectivo si establecemos el idioma (permitiendo, por ejemplo, que aparezca correctamente en los resultados del idioma especificado), y resulta útil para que las personas con discapacidad visual que utilizan los lectores de pantalla (por ejemplo, la palabra «six» existe tanto en francés como en inglés, pero su pronunciación es diferente).

El estándar [ISO 639-1](#) define estos códigos. Podemos encontrar más información sobre ello en [Etiquetas de idioma en HTML y XML](#).

Tablas

Hemos visto cómo representar listas con o sin orden. ¿Pero qué hay de un tabla?

Una tabla es un conjunto estructurado de datos distribuidos en filas y columnas (**datos tabulados**). Una tabla permite buscar con rapidez valores entre diferentes tipos de datos que indiquen algún tipo de conexión. Por ejemplo, una persona y su edad, o un día de la semana o el horario de una piscina municipal.

Por ejemplo:



Person	Age
Chris	38
Dennis	45
Sarah	29
Karen	47

Cuando se hace correctamente, incluso las personas ciegas pueden interpretar los datos de una tabla HTML. Una tabla HTML bien hecha debe mejorar la experiencia de los usuarios videntes e invidentes por igual.

Las tablas HTML están pensadas para utilizarse con datos tabulados. Así que no debemos usarlas para distribuir los elementos de la página (para lograr eso, hay que usar estilos CSS, que lo veremos en el siguiente módulo)

Vamos al código:

1. El contenido de cada tabla está delimitado entre estas dos etiquetas:
`<table></table>`.
2. El contenedor más pequeño dentro de una tabla es una celda, que se crea con un elemento `<td>` ('td' significa 'table data', datos de tabla). Por ejemplo:

```
<td>Hola, soy tu primera celda.</td>
```

3. Si queremos una fila de cuatro celdas, tenemos que copiar estas etiquetas tres veces. Por ejemplo:

```
<td>Hola, soy tu primera celda.</td>  
<td>Soy tu segunda celda.</td>  
<td>Soy tu tercera celda.</td>  
<td>Soy tu cuarta celda.</td>
```



Las celdas no se colocan una debajo de la otra, sino que se alinean automáticamente entre sí en la misma fila. Cada elemento `<td>` crea una sola celda, y juntas forman la primera fila. Cada celda que agregamos hace crecer la fila.

Para detener el crecimiento de esta fila y comenzar a colocar las celdas posteriores en una segunda fila, necesitamos usar el elemento `<tr>` ('tr' significa 'table row', fila de tabla). Por ejemplo:

```
<tr>
  <td>Hola, soy tu primera celda.</td>
  <td>Soy tu segunda celda.</td>
  <td>Soy tu tercera celda.</td>
  <td>Soy tu cuarta celda.</td>
</tr>
<tr>
  <td>Segunda fila, primera celda.</td>
  <td>Celda 2.</td>
  <td>Celda 3.</td>
  <td>Celda 4.</td>
</tr>
```

Esto debería dar como resultado una tabla similar a la siguiente:

Hola, soy tu primera celda.	Soy tu segunda celda.	Soy tu tercera celda.	Soy tu cuarta celda.
Segunda fila, primera celda.	Celda 2.	Celda 3.	Celda 4.



Encabezados

Ahora nos vamos a centrar en los encabezados de tabla: celdas especiales que van al comienzo de una fila o columna y definen el tipo de datos que contiene esa fila o columna (por ejemplo, las celdas «Person» y «Age» en el primer ejemplo).

Para reconocer los encabezados de la tabla como encabezados, tanto visual como semánticamente, puedes usar el elemento `<th>` ('th' significa 'table header', encabezado de tabla). Funciona exactamente igual que un `<td>`, excepto que denota un encabezado, no una celda normal.

Actividad: Edita la tabla de arriba para que la tabla tenga un encabezado.
--

Nota: Las tablas se verán con una estética no muy agradable a la vista, esto lo mejoraremos cuando aprendamos CSS.

Esto apenas es una introducción, pero las tablas HTML son mucho más potentes, revisa la bibliografía.



Imágenes

Con la etiqueta `` podemos hacer que el navegador muestre imágenes. Un ejemplo básico sería:

```

```

En algunos casos el navegador no puede mostrar la imagen, ya sea porque falló la conexión, no soporta el formato de la imagen, o se trata de un navegador no gráfico (incluye aquellos usados por personas con problemas de visión). En ese caso, el navegador la reemplazará con el texto definido en el atributo `alt`.

```

```

Hipervínculos

Ya hablamos sobre los links en la clase pasada, pero no mencionamos que podemos convertir cualquier contenido en un enlace. Si queremos, por ejemplo, convertir una imagen en un enlace, simplemente la ponemos entre los elementos `<a>` y ``.

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

</a>
```



Bibliografía

- <https://developer.mozilla.org/es/docs/Learn/HTML>
- <https://www.htmlquick.com/es/tutorials/tables.html>
- (1) <https://github.com/mdn/learning-area/blob/master/html/introduction-to-html/the-html-head/title-example.html>