

Definiéndolo de forma sencilla, *"HTML es lo que se utiliza para crear todas las páginas web de Internet"*. Más concretamente, HTML es el *lenguaje* con el que se *"escriben"* la mayoría de páginas web

El proceso de transformación de caracteres en secuencias de números se denomina **codificación de caracteres** y cada una de las tablas que se han definido para realizar la transformación se conocen con el nombre de **páginas de código**.

El archivo electrónico almacena tanto los contenidos como la información sobre el formato de esos contenidos. Si por ejemplo se quiere dividir el texto en párrafos y se desea dar especial importancia a algunas palabras, se podría indicar de la siguiente manera:

<parrafo>

Contenido de texto con <importante>algunas palabras</importante> resaltadas de forma especial.

</parrafo>

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 >

Así, 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

---

# HTML

---

sigla HTML significan "*markup language*", que es como se denominan en inglés a los *lenguajes de marcado*.

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.

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 de sus ventanas).

A continuación se muestra el código HTML de una página web muy sencilla:

```
<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>
```

Si ya estás viendo tu primera página HTML en el navegador, prueba a pulsar sobre el menú Ver > Código fuente y podrás ver el código HTML de la página que está cargada en el navegador.

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

- `<html>`: indica el comienzo y el final de un documento HTML. Ninguna etiqueta o contenido puede colocarse antes o después de la etiqueta `<html>`. 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.
- `<head>`: delimita la parte de la cabecera del documento. La cabecera contiene información sobre el propio documento HTML. 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 parte superior izquierda de la ventana del navegador.

- `<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>` no contiene más que unas pocas.

## Atributos

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

`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.`

De todas las etiquetas disponibles, las siguientes se consideran obsoletas y no se pueden utilizar:

`applet, basefont, center, dir, font, isindex, menu, s, strike, u.`

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.

Como no es viable crear una etiqueta por cada cosa diferente, la solución consiste en personalizar las etiquetas HTML mediante cierta información adicional llamada atributos.

De esta forma, se utiliza la misma etiqueta `x` para todas las `x` de la página y se utilizan los atributos para indicar qué cosa que diferencia cada `x`.

Ejemplo con enlaces :

```
<head>
```

```
<title>Ejemplo de atributos en las etiquetas</title>
```

```
</head>
```

```
<body>
```

```
<p>
```

```
    Los enlaces son muy fáciles de indicar:
```

```
    <a>Soy un enlace incompleto, porque no tengo dirección de destino</a>.
```

```
    <a href="http://www.google.com">Este otro enlace apunta a la página de Google</a>.
```

```
</p>
```

```
</body>
```

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

No todos los atributos se pueden utilizar en todas las etiquetas. Por ello, cada etiqueta define su propia lista de atributos disponibles. Además, cada atributo también indica el tipo de valor que se le puede asignar. Si el valor de un atributo no es válido, el navegador ignora ese atributo.

Los atributos comunes se dividen en cuatro grupos según su funcionalidad:

- Atributos básicos: se pueden utilizar prácticamente en todas las etiquetas HTML.
- Atributos para internacionalización: los utilizan las páginas que muestran sus contenidos en varios idiomas.
- Atributos de eventos: sólo se utilizan en las páginas web dinámicas creadas con JavaScript.
- Atributos de foco: relacionados principalmente con la accesibilidad de los sitios web.

## Atributos básicos

Los siguiente cuatro atributos básicos se pueden aplicar prácticamente a todas las etiquetas HTML:

Atributo	Descripción
<code>id = "texto"</code>	Establece un identificador único a cada elemento dentro de una página HTML

<code>class = "texto"</code>	Establece la clase CSS que se aplica a los estilos del elemento
<code>style = "texto"</code>	Establece de forma directa los estilos CSS de un elemento
<code>title = "texto"</code>	Establece el título a un elemento (mejora la accesibilidad y los navegadores lo muestran cuando el usuario pasa el ratón por encima del elemento)

Respecto al valor de los atributos `id` y `class`, sólo pueden contener guiones medios (-), guiones bajos (\_), letras y/o números, pero no pueden empezar por números. Además, los navegadores distinguen mayúsculas de minúsculas y no se recomienda utilizar letras como ñ y acentos, ya que no es seguro que funcionen correctamente en todas las versiones de todos los navegadores.

## Atributos para internacionalización

Estos atributos son útiles para aquellas páginas que muestran sus contenidos en varios idiomas y para aquellas que quieren indicar de forma explícita el idioma de sus contenidos:

Atributo	Descripción
<code>lang = "codigo de idioma"</code>	Indica el idioma del elemento mediante un código predefinido
<code>xml:lang = "codigo de idioma"</code>	Indica el idioma del elemento mediante un código predefinido
<code>dir</code>	Indica la dirección del texto (útil para los idiomas que escriben de derecha a izquierda)

En las páginas XHTML, el atributo `xml:lang` tiene más prioridad que `lang` y es obligatorio incluirlo siempre que se incluye el atributo `lang`. Los idiomas se indican mediante un código estandarizado (`es` para español, `en` para inglés, etc.)

## Atributos de eventos

Estos atributos sólo se utilizan en las páginas web que incluyen código JavaScript para realizar acciones dinámicas sobre los elementos de la página. Cada vez que el usuario pulsa una tecla, mueve su ratón o pulsa cualquier botón del ratón, se produce un evento dentro del navegador. Utilizando JavaScript y los siguientes atributos, es posible responder de forma adecuada a cada evento.

Atributo	Descripción	Elementos que pueden usarlo
<code>onblur</code>	Deseleccionar el elemento	<code>&lt;button&gt;</code> , <code>&lt;input&gt;</code> , <code>&lt;label&gt;</code> , <code>&lt;select&gt;</code> , <code>&lt;textarea&gt;</code> , <code>&lt;body&gt;</code>
<code>onchange</code>	Deseleccionar un elemento que se ha modificado	<code>&lt;input&gt;</code> , <code>&lt;select&gt;</code> , <code>&lt;textarea&gt;</code>
<code>onclick</code>	Pinchar y soltar el ratón	Todos los elementos
<code>ondblclick</code>	Pinchar dos veces seguidas con el ratón	Todos los elementos
<code>onfocus</code>	Seleccionar un elemento	<code>&lt;button&gt;</code> , <code>&lt;input&gt;</code> , <code>&lt;label&gt;</code> , <code>&lt;select&gt;</code> , <code>&lt;textarea&gt;</code> , <code>&lt;body&gt;</code>
<code>onkeydown</code>	Pulsar una tecla (sin soltar)	Elementos de formulario y <code>&lt;body&gt;</code>

onkeypress	Pulsar una tecla	Elementos de formulario y <code>&lt;body&gt;</code>
onkeyup	Soltar una tecla pulsada	Elementos de formulario y <code>&lt;body&gt;</code>
onload	La página se ha cargado completamente	<code>&lt;body&gt;</code>
onmousedown	Pulsar (sin soltar) un botón del ratón	Todos los elementos
onmousemove	Mover el ratón	Todos los elementos
onmouseout	El ratón "sale" del elemento (pasa por encima de otro elemento)	Todos los elementos
onmouseover	El ratón "entra" en el elemento (pasa por encima del elemento)	Todos los elementos
onmouseup	Soltar el botón que estaba pulsado en el ratón	Todos los elementos
onreset	Inicializar el formulario (borrar todos sus datos)	<code>&lt;form&gt;</code>
onresize	Se ha modificado el tamaño de la ventana del navegador	<code>&lt;body&gt;</code>
onselect	Seleccionar un texto	<code>&lt;input&gt;</code> , <code>&lt;textarea&gt;</code>

<code>onsubmit</code>	Enviar el formulario	<code>&lt;form&gt;</code>
<code>onunload</code>	Se abandona la página (por ejemplo al cerrar el navegador)	<code>&lt;body&gt;</code>

## Atributos de foco

Cuando el usuario selecciona un elemento en una aplicación, se dice que *"el elemento tiene el foco del programa"*. Si por ejemplo un usuario pincha con su ratón sobre un cuadro de texto y comienza a escribir, ese cuadro de texto tiene el foco del programa

Si el usuario selecciona después otro elemento, el elemento original pierde el foco y el nuevo elemento es el que tiene el foco del programa.

Los elementos de las páginas web también pueden obtener el foco de la aplicación (en este caso, el foco del navegador) y HTML define algunos atributos específicos para controlar cómo se seleccionan los elementos.

Atributo	Descripción
<code>accesskey = "letra"</code>	Establece una tecla de acceso rápido a un elemento HTML
<code>tabindex = "numero"</code>	Establece la posición del elemento en el orden de tabulación de la página. Su valor debe estar comprendido entre 0 y 32.767
<code>onfocus, onblur</code>	Controlan los eventos JavaScript que se ejecutan cuando el elemento obtiene o pierde el foco

Cuando se pulsa repetidamente la tecla del tabulador sobre una página web, el navegador selecciona de forma alternativa todos los elementos de la página que se pueden seleccionar (principalmente los enlaces y los elementos de formulario). El atributo `tabindex` permite alterar el orden en el que se



seleccionan los elementos, por lo que es muy útil cuando se quiere controlar de forma precisa cómo se seleccionan los campos de un formulario complejo.

Por su parte, el atributo `accesskey` permite establecer una tecla para acceder de forma rápida a cualquier elemento.

## Elementos HTML

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

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.

Por otra parte, el lenguaje HTML clasifica a todos los elementos en dos grupos: elementos en **línea** y elementos de **bloque**.

La principal diferencia entre los dos tipos de elementos es la forma en la que ocupan el espacio disponible en la página. Los elementos de bloque siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea, aunque sus contenidos no lleguen hasta el final de la línea. Por su parte, los elementos en línea sólo ocupan el espacio necesario para mostrar sus contenidos.

**Un elemento de bloque no puede aparecer dentro de un elemento en línea. En cambio, un elemento en línea puede aparecer dentro de un elemento de bloque y dentro de otro elemento en línea.**

## 3. Texto

HTML define etiquetas para estructurar el contenido en secciones y párrafos y define otras etiquetas para marcar elementos importantes dentro del texto

### 3.1. Estructurar

#### 3.1.1. 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.

<b>Etiqueta</b>	<code>&lt;p&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	-
<b>Tipo de elemento</b>	Bloque
<b>Descripción</b>	Delimita el contenido de un párrafo de texto

### 3.1.2. Secciones

Las páginas HTML es habitual que 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.

A continuación se muestra la definición formal de la etiqueta `<h1>`, siendo idéntica la definición del resto de etiquetas referidas a los títulos de sección:

<b>Etiqueta</b>	<code>&lt;h1&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>

Atributos propios	-
Tipo de elemento	Bloque
Descripción	Define los títulos de las secciones de mayor importancia de la página.

Al igual que la etiqueta `<p>`, las etiquetas de título de sección son elementos de bloque y no tienen atributos específicos.

## 3.2. 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 definición formal de estas dos etiquetas se muestra a continuación:

Etiqueta	<code>&lt;em&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	-
Tipo de elemento	En línea
Descripción	Realza la importancia del texto que encierra

<b>Etiqueta</b>	<code>&lt;strong&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	-
<b>Tipo de elemento</b>	En línea
<b>Descripción</b>	Realza con la máxima importancia el texto que encierra

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.

HTML permite indicar de forma clara el texto que ha sido eliminado y el texto que ha sido añadido a un determinado texto original. Las etiquetas utilizadas son `<ins>` y `<del>`, cuya definición formal es la siguiente:

<b>Etiqueta</b>	<code>&lt;ins&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	<ul style="list-style-type: none"><li>● <code>cite = "url"</code> - Indica la URL de la página en la que se puede obtener más información sobre el motivo por el que se realizó la modificación.</li><li>● <code>datetime = "fecha"</code> - Especifica la fecha y hora en la que se realizó el cambio</li></ul>

<b>Tipo de elemento</b>	Bloque y en línea
<b>Descripción</b>	Se emplea para marcar una modificación en los contenidos originales consistente en la inserción de un nuevo contenido

<b>Etiqueta</b>	<del>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	<ul style="list-style-type: none"><li>● <code>cite = "url"</code> - Indica la URL de la página en la que se puede obtener más información sobre el motivo por el que se realizó la modificación.</li><li>● <code>datetime = "fecha"</code> - Especifica la fecha y hora en la que se realizó el cambio</li></ul>
<b>Tipo de elemento</b>	Bloque y en línea
<b>Descripción</b>	Se emplea para marcar una modificación en los contenidos originales consistente en el borrado de cierto contenido

Las dos etiquetas cuentan con los mismos atributos específicos, que opcionalmente se pueden añadir para proporcionar más información sobre los cambios realizados. El atributo `cite` se emplea para indicar la dirección de un documento externo en el que se puede encontrar más información relacionada con la inserción o el borrado de texto. El atributo `datetime` puede utilizarse para indicar la fecha y la hora en la que se realizó cada cambio.

HTML define la etiqueta `<blockquote>` para incluir citas textuales en las páginas web. La definición de la etiqueta HTML con el nombre más largo se muestra a continuación:

Etiqueta	<code>&lt;blockquote&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	<ul style="list-style-type: none"><li>● <code>cite = "url"</code> - Indica la dirección de la página web original de la que se extrae la cita</li></ul>
Tipo de elemento	Bloque
Descripción	Se emplea para indicar que el texto que encierra es una cita textual de otro texto externo

Al igual que `<ins>` y `<del>`, la etiqueta `<blockquote>` permite indicar mediante el atributo `cite` la dirección de un documento del que se ha extraído la cita.

### 3.3. Marcado avanzado de texto

HTML incluye muchas otras etiquetas que permiten marcar más elementos del texto.

La etiqueta `<abbr>` marca las abreviaturas de un texto y se emplea para marcar las siglas o acrónimos del texto. Su definición es la siguiente:

Etiqueta	<code>&lt;abbr&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>

<b>Atributos propios</b>	<ul style="list-style-type: none"><li>● <code>title = "texto"</code> - Indica el significado completo de la abreviatura</li></ul>
<b>Tipo de elemento</b>	En línea
<b>Descripción</b>	Se emplea para marcar las abreviaturas del texto y proporcionar el significado de esas abreviaturas

En ambos casos, el atributo `title` se puede utilizar para incluir el significado completo de la abreviatura o sigla.

Por otra parte, en ocasiones resulta útil incluir la definición de una palabra extraña o cuyo uso está restringido a un entorno muy determinado. HTML incluye la etiqueta `<dfn>` para proporcionar al usuario la definición de todas las palabras para las que se considere apropiado. La definición formal de esta etiqueta se muestra a continuación:

<b>Etiqueta</b>	<code>&lt;dfn&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	<ul style="list-style-type: none"><li>● <code>title = "texto"</code> - Indica el significado completo del término</li></ul>
<b>Tipo de elemento</b>	En línea
<b>Descripción</b>	Se emplea para marcar las definiciones de ciertos términos y proporcionar el significado de esos términos

Por último, HTML incluye una etiqueta que se puede utilizar para marcar un texto como una citación:

Etiqueta	<cite>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	-
Tipo de elemento	En línea
Descripción	Se emplea para marcar una cita o una referencia a otras fuentes

En ocasiones, no está clara la diferencia entre `<cite>` y `<blockquote>`. El elemento `<cite>` marca el autor de la cita (persona, documento, etc.) y `<blockquote>` marca el contenido de la propia cita. En el siguiente ejemplo, `<blockquote>` encierra el contenido de una frase célebre y `<cite>` encierra el nombre de su autor:

### 3.4. Marcado genérico de texto

La etiqueta llamada `<span>` se emplea para marcar cualquier elemento que no se puede marcar con las otras etiquetas definidas.

La etiqueta `<span>` se visualiza por defecto con el mismo aspecto que el texto normal. Por tanto es habitual utilizar esta etiqueta junto con los atributos `id` y `class` para modificar posteriormente su aspecto con CSS:

```
<strong>Importante</strong>: si quiere ponerse en contacto con la empresa  
<acronym>ACME</acronym>, puede hacerlo en el teléfono <span class="telefono">900 555  
555</span> o a través de la dirección de correo electrónico <span  
class="email">contacto@acme.org</span>
```



La etiqueta `<span>` sólo se puede utilizar para encerrar contenidos y etiquetas en línea. Cuando se quieren estructurar elementos de bloque, se utiliza la etiqueta `<div>`, tal y como se verá en capítulos posteriores.

## 3.5. Espacios en blanco y nuevas líneas

HTML considera *espacio en blanco* a los espacios en blanco, los tabuladores, los retornos de carro y el carácter de nueva línea (`ENTER` o `Intro`).

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.

### 3.5.1. Nuevas líneas

Para incluir una nueva línea en un punto y forzar a que el texto que sigue se muestra 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` (o `Intro`) en ese mismo punto.

La definición formal de `<br>` se muestra a continuación:

<b>Etiqueta</b>	<code>&lt;br&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a>
<b>Atributos propios</b>	-
<b>Tipo de elemento</b>	En línea y etiqueta vacía
<b>Descripción</b>	Fuerza al navegador a insertar una nueva línea

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 cerrarse de forma consecutiva: `<br></br>`.

### 3.5.2. Espacios en blanco

La solución al problema de los espacios en blanco no es tan sencilla como el de las nuevas líneas. 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).

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. **(no usar, queda horrible)**

### 3.5.3. Texto preformateado

En ocasiones, es necesario mostrar los espacios en blanco de un texto que no se puede modificar. Se trata de un caso habitual cuando una página web debe mostrar directamente el texto generado por alguna aplicación.

En estos casos, se puede utilizar la etiqueta `<pre>`, que muestra el texto tal y como se ha escrito, respetando todos los espacios en blanco y todas las nuevas líneas. La definición formal de la etiqueta se muestra a continuación:

Etiqueta	<code>&lt;pre&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	-
Tipo de elemento	Bloque
Descripción	Muestra el texto que encierra tal y como está escrito (respetando los espacios en blanco)

Los elementos `<pre>` son *especiales*, ya que los navegadores les aplican las siguientes reglas:

- Mantienen todos los espacios en blanco (tabuladores, espacios y nuevas líneas)

- Muestra el texto con un tipo de letra especial, denominado "de ancho fijo", ya que todas sus letras son de la misma anchura
- No se ajusta la longitud de las líneas (las líneas largas producen un *scroll* horizontal en la ventana del navegador)

Esta última característica diferencia por completo a los párrafos de los elementos `<pre>`. Como se ha visto, los navegadores ajustan la anchura de los párrafos de texto para que ocupen todo el tamaño de la ventana. Sin embargo, los elementos `<pre>` se muestran tal y como son originalmente, por lo que una línea muy larga dentro de un elemento `<pre>` provoca que la anchura de la página sea superior a la anchura de la ventana del navegador.

Otra etiqueta relacionada con `<pre>` es la etiqueta `<code>`, que se utiliza para mostrar código fuente de cualquier lenguaje de programación. La definición formal de `<code>` es la siguiente:

Etiqueta	<code>&lt;code&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	-
Tipo de elemento	En línea
Descripción	Delimita el texto considerado un fragmento de código fuente

En la mayoría de páginas web, no tiene sentido utilizar la etiqueta `<code>`. Sin embargo, en muchas páginas web técnicas que incluyen listados de programas, trozos de código o etiquetas HTML, lo correcto es emplear la etiqueta `<code>`.

Al igual que sucede con los elementos `<pre>`, el texto encerrado por la etiqueta `<code>` se muestra con un tipo de letra especial de ancho fijo. Por el contrario, el elemento `<code>` no respeta los espacios en blanco ni las líneas, por lo que su comportamiento es similar a la etiqueta `<p>`. La última diferencia es que `<code>` es un elemento en línea, mientras que `<pre>` es un elemento de bloque.

### 3.6. Codificación de caracteres

Una consideración importante directamente relacionada con el texto de las páginas HTML es la codificación de los caracteres y la inserción de caracteres *especiales*. Algunos de los caracteres que se utilizan habitualmente en los textos no se pueden incluir directamente en las páginas web:

- Los caracteres que utiliza HTML para definir sus etiquetas (<, > y ") no se pueden utilizar libremente.
- Los caracteres propios de los idiomas que no son el inglés (ñ, á, ç, é, ï, etc.) pueden ser problemáticos dependiendo de la codificación de caracteres utilizada.

La solución a la primera limitación consiste en sustituir los caracteres reservados de HTML por unas expresiones llamadas *entidades HTML* y que representan a cada carácter:

Entidad	Carácter	Descripción	Traducción
&lt;	<	less than	signo de menor que
&gt;	>	more than	signo de mayor que
&amp;	&	ampersand	ampersand
&quot;	"	quotation mark	comillas
&nbsp;	(espacio en blanco)	non-breaking space	espacio en blanco
&apos;	'	apostrophe	apóstrofo

Por otra parte, los caracteres propios de los idiomas diferentes al inglés también pueden ser problemáticos. El motivo es que desde que se crea una página web hasta que llega al navegador del usuario, intervienen numerosos procesos:

- El diseñador crea la página web con su editor HTML
- Si se trata de una aplicación dinámica, el programador recorta la página HTML del diseñador y la mezcla con el resto del código de la aplicación
- El servidor web almacena las páginas HTML estáticas o el código de la aplicación web y sirve las páginas solicitadas por los usuarios.
- El usuario solicita y visualiza las páginas web a través de su navegador.

Si en todos los procesos anteriores se utiliza la misma codificación de caracteres, los caracteres propios de los idiomas se pueden escribir directamente. Sin embargo, muchas veces no es posible que todos los procesos involucrados utilicen la misma codificación de caracteres.

La solución más sencilla para asegurar que todos estos caracteres potencialmente problemáticos se van a visualizar correctamente en el navegador del usuario consiste en sustituir cada carácter problemático por su entidad HTML:

Entidad	Carácter	Descripción oficial
<code>&amp;ntilde;</code>	ñ	latin letter n with tilde
<code>&amp;Ntilde;</code>	Ñ	latin capital n letter with tilde
<code>&amp;aacute;</code>	á	a acute
<code>&amp;eacute;</code>	é	e acute
<code>&amp;iacute;</code>	í	i acute
<code>&amp;oacute;</code>	ó	o acute
<code>&amp;uacute;</code>	ú	u acute

<code>&amp;Aacute</code> ;	Á	A acute
<code>&amp;Eacute</code> ;	É	E acute
<code>&amp;Iacute</code> ;	Í	I acute
<code>&amp;Oacute</code> ;	Ó	O acute
<code>&amp;Uacute</code> ;	Ú	U acute
<code>&amp;euro</code> ;	€	euro

## 4. Enlaces

Los enlaces se utilizan para establecer relaciones entre dos recursos. Aunque la mayoría de enlaces relacionan páginas web, también es posible enlazar otros recursos como imágenes, documentos y archivos.

Una característica que no se suele tener en cuenta en los enlaces es que están formados por dos extremos y un sentido. En otras palabras, el enlace comienza en un recurso y apunta hacia otro recurso. Cada uno de los dos extremos se llaman "*anchors*" en inglés, que se puede traducir literalmente como "anclas".

### 4.1. URL

El acrónimo URL (del inglés *Uniform Resource Locator*) hace referencia al identificador único de cada recurso disponible en Internet. Las URL son esenciales para crear los enlaces, pero también se utilizan en otros elementos HTML como las imágenes y los formularios.

La URL de un recurso tiene dos objetivos principales:

- Identificar de forma única a ese recurso
- Permitir localizar de forma eficiente este recurso

En primer lugar, las URL permiten que cada página HTML publicada en Internet tenga un nombre único que permita diferenciarla de las demás. De esta forma es posible crear enlaces que apunten de forma inequívoca a una determinada página.

El segundo objetivo de las URL es el de permitir la localización eficiente de cada recurso de Internet. Para ello es necesario comprender las diferentes partes que forman las URL. Una URL sencilla siempre está formada por las mismas tres partes. Si por ejemplo se considera la siguiente URL:

```
http://www.librosweb.es/xhtml/capitulo4.html
```

Las partes que componen la URL anterior son:

- Protocolo (`http://`): el mecanismo que debe utilizar el navegador para acceder a ese recurso. Todas las páginas web utilizan `http://`. Las páginas web seguras (por ejemplo las de los bancos y las de los servicios de email) utilizan `https://` (se añade una letra `s`).
- Servidor (`www.librosweb.es`): simplificando mucho su explicación, se trata del ordenador en el que se encuentra guardada la página que se quiere acceder. Los navegadores son capaces de obtener la dirección de cada servidor a partir de su nombre.
- Ruta (`/xhtml/capitulo4.html`): *camino* que se debe seguir, una vez que se ha llegado al servidor, para localizar el recurso específico que se quiere acceder.

Por tanto, las URL no sólo identifican de forma única a cada recurso de Internet, sino que también proporcionan a los navegadores la información necesaria para poder llegar hasta ese recurso.

La mayoría de URL son tan sencillas como la URL mostrada anteriormente. No obstante, existen URL complejas formadas por más partes.

```
http://www.alistapart.com/comments/webstandards2008?page=5#42
```

Las cinco partes que forman la URL anterior son:

- Protocolo (`http://`)
- Servidor (`www.alistapart.com`)
- Ruta (`/comments/webstandards2008`)
- Consulta (`?page=5`): información adicional necesaria para que el servidor localice correctamente el recurso que se quiere acceder. Siempre comienza con el carácter `?` y contiene una sucesión de palabras separadas por `=` y `&`
- Sección (`#42`): permite que el navegador se posicione automáticamente en una sección de la página web. Siempre comienza con el carácter `#`

Como las URL utilizan los caracteres `:`, `=`, `&` y `/` para separar sus partes, estos caracteres están reservados y no se pueden utilizar libremente. Además, algunos caracteres no están reservados pero pueden ser problemáticos si se utilizan en la propia URL.

Si es necesario incluir estos caracteres reservados y especiales en una URL, se sustituyen por combinaciones de caracteres seguros. Esta sustitución se denomina *codificación* de caracteres y el servidor realiza el proceso inverso (*decodificación*) cuando le llega una URL con los caracteres codificados.

A continuación se muestra la tabla para codificar los caracteres más comunes:

Carácter original	Carácter codificado	Carácter original	Carácter codificado
/	%2F	?	%3F
:	%3A	@	%40
=	%3D	&	%26
"	%22	\	%5C
'	%27	~	%7E
(espacio en blanco)	%20	#	%23

Por otra parte, aunque desde hace tiempo ya es posible incluir en las URL caracteres de otros idiomas que no sean el inglés, aún no es completamente seguro utilizar estos caracteres en las URL. Si se utilizan letras como ñ, á, é o ç, es posible que algunos navegadores no las interpreten de forma correcta.

La solución consiste en codificar todos los caracteres que no existen en inglés. La siguiente tabla muestra la codificación de los caracteres más utilizados:

Carácter original	Carácter codificado	Carácter original	Carácter codificado
ñ	%F1	Ñ	%D1



á	%E1	Á	%C1
é	%E9	É	%C9
í	%ED	Í	%CD
ó	%F3	Ó	%D3
ú	%FA	Ú	%DA
ç	%E7	Ç	%C7

Teniendo en cuenta las dos tablas anteriores de codificación de caracteres, es fácil crear las URL correctas sin caracteres problemáticos

## 4.2. Enlaces relativos y absolutos

Cuando se pincha sobre algunos enlaces, el navegador abandona el sitio web para acceder a páginas que se encuentran en otros sitios. Estos enlaces se conocen como "enlaces externos". Sin embargo, la mayoría de enlaces de un sitio web apuntan a páginas del propio sitio web, por lo que se denominan "enlaces internos".

Además de internos/externos, la otra característica que diferencia a los enlaces (y por tanto, también a las URL) es si el enlace es absoluto o relativo. Las URL absolutas incluyen todas las partes de la URL (protocolo, servidor y ruta) por lo que no se necesita más información para obtener el recurso enlazado.

Las URL relativas prescinden de algunas partes de las URL para hacerlas más breves. Como se trata de URL incompletas, es necesario disponer de información adicional para obtener el recurso enlazado. En concreto, para que una URL relativa sea útil es imprescindible conocer la URL del origen del enlace.

Las URL relativas se construyen a partir de las URL absolutas y prescinden de la parte del protocolo, del nombre del servidor e incluso de parte o toda la ruta del recurso enlazado. Aunque las URL relativas pueden ser difíciles de entender para los que comienzan con HTML, son tan útiles que todos los sitios web las utilizan.

Las URL completas también se llaman URL absolutas, ya que el navegador no necesita disponer de información adicional para localizar el recurso enlazado. Si se utilizan siempre las URL absolutas, los enlaces están completamente definidos.

Una URL relativa es una versión abreviada de una URL absoluta. Su objetivo es eliminar todas las partes de la URL absoluta que se pueden *adivinar* a partir de la información de contexto de la página web.

la URL a la que se quiere enlazar utiliza el mismo protocolo y se encuentra en el mismo servidor que la página origen, por lo que la URL relativa puede prescindir de esas partes:

URL absoluta: `http://www.ejemplo.com/ruta1/ruta2/pagina2.html`

URL relativa: `/ruta1/ruta2/pagina2.html`

En el ejemplo anterior, las dos URL son equivalentes porque cuando no se indica el protocolo y el servidor de una URL, los navegadores suponen que son los mismos que los de la página origen. Por lo tanto, cuando el navegador encuentra la URL `/ruta1/ruta2/pagina2.html`, realiza el siguiente proceso:

1. La URL no es absoluta, por lo que se debe determinar la URL absoluta a partir de la URL relativa para poder cargar el recurso enlazado.
2. A la URL relativa le falta el protocolo y el servidor, por lo que se supone que son los mismos que los de la página origen (`http://` y `www.ejemplo.com`).
3. Se añaden las partes que faltan a la URL relativa para obtener la URL absoluta: `http://` + `www.ejemplo.com` + `/ruta1/ruta2/pagina2.html` = `http://www.ejemplo.com/ruta1/ruta2/pagina2.html`.

Aunque el ejemplo mostrado es el caso más sencillo de URL relativa, existen otros casos más avanzados en los que se prescinde de parte o toda la ruta del recurso que se enlaza. A continuación se muestran los cuatro tipos diferentes de URL relativas:

1) El origen y el destino del enlace se encuentran en el mismo directorio

Si desde una página web se quiere enlazar un recurso que se encuentra en el mismo directorio del servidor, la URL relativa puede prescindir de todas las partes de la URL absoluta salvo el nombre del recurso enlazado.

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>

Página enlazada	Página web llamada <code>pagina2.html</code> y que se encuentra en el mismo directorio
URL absoluta	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina2.html</code>
URL relativa	<code>pagina2.html</code>

Cuando el navegador encuentra una URL relativa que sólo consiste en el nombre de un recurso, supone que el protocolo, servidor y directorio del recurso enlazado son los mismos que los del origen del enlace.

2) El destino del enlace se encuentra cerca de su origen y en un nivel superior

En este caso, el recurso que se enlaza no está en el mismo directorio que el origen del enlace pero sí que está *cerca* y en algún directorio superior. La URL relativa debe indicar de alguna manera que es necesario *subir* un nivel en la jerarquía de directorios para llegar hasta el recurso.

Para indicar al navegador que debe subir un nivel, se incluyen dos puntos y una barra (`../`) en la ruta del recurso enlazado. De esta forma, cada vez que aparece `../` en una URL relativa, significa que se debe subir un nivel.

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>
Página enlazada	Página web llamada <code>pagina2.html</code> y que se encuentra en el directorio superior llamado <code>ruta2</code>
URL absoluta	<code>http://www.ejemplo.com/ruta1/ruta2/pagina2.html</code>

URL relativa	<code>../pagina2.html</code>
--------------	------------------------------

Cuando el navegador encuentra la URL relativa `../pagina2.html`, sabe que para encontrar el recurso enlazado (`pagina2.html`) tiene que subir un nivel desde el lugar en el que se encuentra esa URL relativa. La página que incluye esa URL se encuentra en el directorio `ruta1/ruta2/ruta3`, por lo que subir un nivel equivale entrar en el directorio `ruta1/ruta2`.

De la misma forma, si el destino se encuentra un par de niveles por encima, se debe incluir `../` dos veces seguidas:

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>
Página enlazada	Página web llamada <code>pagina2.html</code> y que se encuentra en el directorio superior llamado <code>ruta1</code>
URL absoluta	<code>http://www.ejemplo.com/ruta1/pagina2.html</code>
URL relativa	<code>../../pagina2.html</code>

Además de subir niveles, también se puede entrar en otros directorios para obtener los recursos:

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>

Página enlazada	Página web llamada <code>pagina2.html</code> y que se encuentra en un directorio llamado <code>ruta4</code> que se encuentra en la raíz del servidor
URL absoluta	<code>http://www.ejemplo.com/ruta4/pagina2.html</code>
URL relativa	<code>../../../../ruta4/pagina2.html</code>

Si se intentan subir más niveles de los que es posible, el navegador ignora todos los `../` sobrantes. Si la página que tiene el enlace es `http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html` y la URL relativa que se incluye es `../../../../../../../../pagina2.html`, el navegador realmente la interpreta como `../../../../pagina2.html`.

Como el objetivo de las URL relativas es crear URL más cortas y sencillas que las URL absolutas, este método sólo se puede utilizar cuando el origen y el destino se encuentran cerca, porque de otro modo la URL relativa se complica demasiado.

3) El destino del enlace se encuentra cerca de su origen y en un nivel inferior

Este caso es muy similar al anterior, pero más sencillo. Si el recurso enlazado se encuentra en algún directorio inferior al que se encuentra el origen, sólo es necesario indicar el nombre de los directorios a los que debe entrar el navegador.

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>
Página enlazada	Página web llamada <code>pagina2.html</code> y que se encuentra en un directorio inferior llamado <code>ruta4</code>
URL absoluta	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/ruta4/pagina2.html</code>

URL relativa	ruta4/pagina2.html
--------------	--------------------

De la misma forma, se pueden indicar varios directorios seguidos para que el navegador descienda jerárquicamente por la estructura de directorios:

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>
Página enlazada	Página web llamada <code>pagina2.html</code> y que se encuentra en un directorio inferior llamado <code>ruta6</code> que está dentro del directorio <code>ruta5</code> y que a su vez está dentro del directorio <code>ruta4</code>
URL absoluta	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/ruta4/ruta5/ruta6/pagina2.html</code>
URL relativa	<code>ruta4/ruta5/ruta6/pagina2.html</code>

#### 4) El origen y el destino del enlace se encuentran muy alejados

Cuando el origen y el destino de un enlace se encuentran muy alejados (pero en el mismo servidor) las URL relativas se pueden complicar en exceso. Aunque es posible utilizar `../` para subir por la jerarquía de directorios y se puede entrar en cualquier directorio indicando su nombre, las URL relativas que se obtienen son demasiado largas y complicadas.

En estos casos, lo más sencillo es indicar la ruta completa hasta el recurso enlazado comenzando desde la raíz del servidor web. Por lo tanto, estas URL relativas sólo omiten el protocolo y el nombre del servidor.

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>
Página enlazada	Página web llamada <code>pagina2.html</code> y que se guarda en un directorio llamado <code>ruta7</code> que se encuentra en la raíz del servidor
URL absoluta	<code>http://www.ejemplo.com/ruta7/pagina2.html</code>
URL relativa	<code>/ruta7/pagina2.html</code>

Cuando la URL relativa comienza por `/`, el navegador considera que es la ruta completa comenzando desde la raíz del servidor, por lo que sólo le añade el protocolo y el nombre del servidor origen.

A continuación se resumen los cuatro posibles casos de URL relativas y el procedimiento que sigue el navegador para convertirlas en URL absolutas:

Si la URL relativa...	El navegador la transforma en URL absoluta...
...sólo consiste en el nombre de un recurso	...añadiendo el protocolo, servidor y ruta completa del origen del enlace
...comienza por <code>../</code>	...añadiendo el protocolo y servidor del origen del enlace, subiendo un nivel en la jerarquía de directorios y añadiendo el resto de la ruta incluida en la URL relativa

...comienza por /	...añadiendo el protocolo y servidor del origen del enlace
En cualquier otro caso	...añadiendo el protocolo, servidor y ruta completa del origen del enlace, a la que se añade la ruta incluida en la URL relativa

## 4.3. Enlaces básicos

Los enlaces en HTML se crean mediante la etiqueta `<a>` (su nombre viene del inglés *"anchor"*, literalmente traducido como "ancla"). A continuación se muestra la definición simplificada de `<a>` y más adelante se muestra su definición completa:

<b>Etiqueta</b>	<code>&lt;a&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> , <a href="#">eventos</a> y <a href="#">foco</a>
<b>Atributos propios</b>	<ul style="list-style-type: none"> <li>● <code>name = "texto"</code> - Permite nombrar al enlace para que se pueda acceder desde otros enlaces</li> <li>● <code>href = "url"</code> - Indica la URL del recurso que se quiere enlazar</li> </ul>
<b>Tipo de elemento</b>	En línea
<b>Descripción</b>	Se emplea para enlazar todo tipo de recursos



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 pincha sobre un enlace, el navegador se dirige a la URL del recurso indicado mediante `href`. Las URLs de los enlaces pueden ser absolutas, relativas, internas y externas.

Un truco muy útil con los enlaces es el uso de URL relativas para poder volver al inicio del sitio web desde cualquier página web interior:

```
<a href="/">Volver al inicio</a>
```

El enlace anterior utiliza una URL relativa con una ruta que apunta directamente a la raíz del servidor. Para obtener la URL absoluta, el navegador añade el mismo protocolo y el mismo nombre de servidor de la página en la que se encuentra el enlace. El resultado es que cuando se pincha este enlace, siempre se vuelve al inicio del sitio web, independientemente de la página en la que se incluya el enlace.

El otro atributo básico de la etiqueta `<a>` es `name`, que permite definir enlaces dentro de una misma página web. Si una página es muy larga, puede ser útil mostrar enlaces de tipo "Saltar hasta la segunda sección", "Volver al principio de la página", etc.

Este tipo de enlaces son especiales, ya que la URL de la página siempre es la misma para todas las secciones y por tanto, debe añadirse otra parte a las URL para identificar cada sección.

El atributo `name` permite crear "enlaces vacíos" que hacen referencia a secciones dentro de una misma página. Una vez definidos los "enlaces vacíos", es posible crear un enlace que apunte directamente a una sección concreta de una página:

```
<!-- Enlace normal a la página -->
```

```
<a href="http://www.ejemplo.com/pagina1.html">Enlace a la página 1</a>
```

```
<!-- Enlace directo a la segunda sección de la página -->
```

```
<a href="http://www.ejemplo.com/pagina1.html#segunda_seccion">Enlace a la  
sección 2 de la página 1</a>
```

La sintaxis que se utiliza con estos enlaces es la misma que con los enlaces normales, salvo que se añade el símbolo `#` seguido del nombre de la sección a la que se apunta. Cuando el usuario pincha sobre uno de estos enlaces, el navegador accede a la página apuntada por la URL y baja directamente a la sección cuyo nombre se indica después del símbolo `#`.

También es posible utilizar este tipo de enlaces con URL relativas en una misma página. El siguiente ejemplo añade enlaces de tipo "Volver al inicio de la página" en varias secciones:

```
<a name="inicio"></a>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id  
ligula eu felis adipiscing ultrices. Duis gravida leo ut lectus. Praesent  
condimentum mattis ligula.</p>
```

```
<a href="#inicio">Volver al inicio de la página</a>
```

...

```
<p>Pellentesque malesuada. In in lacus. Phasellus erat erat, lacinia a,  
convallis eu, nonummy et, odio. Aenean urna elit, ultrices id, placerat  
varius, facilisis eget, dolor.</p>
```

```
<a href="#inicio">Volver al inicio de la página</a>
```

...

Los enlaces directos a secciones también funcionan con el atributo `id` de cualquier elemento. El siguiente ejemplo es equivalente al ejemplo anterior:

```
<h1 id="inicio">Título de la página</h1>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id  
ligula eu felis adipiscing ultrices. Duis gravida leo ut lectus. Praesent  
condimentum mattis ligula.</p>
```

```
<a href="#inicio">Volver al inicio de la página</a>
```

...

```
<p>Pellentesque malesuada. In in lacus. Phasellus erat erat, lacinia a,  
convallis eu, nonummy et, odio. Aenean urna elit, ultrices id, placerat  
varius, facilisis eget, dolor.</p>
```

```
<a href="#inicio">Volver al inicio de la página</a>
```

...

El nombre de la sección que se indica después del símbolo # puede utilizar el valor de los atributos `id` de cualquier elemento. De hecho, se recomienda utilizar los atributos `id` de los elementos ya existentes en la página en vez de crear "enlaces vacíos" de tipo `<a name="nombre_seccion"></a>`.

## 4.4. Enlaces avanzados

Incluir enlaces básicos mediante la etiqueta `<a>` es muy sencillo. Sin embargo, la definición completa de `<a>` es muy compleja, ya que dispone de varios atributos específicos importantes. A continuación se muestra la definición completa de `<a>`:

<b>Etiqueta</b>	<code>&lt;a&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> , <a href="#">eventos</a> y <a href="#">foco</a>

<b>Atributos propios</b>	<ul style="list-style-type: none"><li>● <code>name = "texto"</code> - Permite nombrar al enlace para que se pueda acceder desde otros enlaces</li><li>● <code>href = "url"</code> - Indica la URL del recurso que se quiere enlazar</li><li>● <code>hreflang = "codigo_idioma"</code> - Idioma del recurso enlazado</li><li>● <code>type = "tipo_de_contenido"</code> - Permite "avisar" al navegador sobre el tipo de contenido que se enlaza (imágenes, archivos, etc.) para que pueda prepararse en caso de que no entienda ese contenido</li><li>● <code>rel = "tipo_de_relacion"</code> - Describe la relación del documento actual con el recurso enlazado</li><li>● <code>rev = "tipo_de_relacion"</code> - Describe la relación del recurso enlazado con el documento actual</li><li>● <code>charset = "tipo_de_charset"</code> - Describe la codificación del recurso enlazado</li></ul>
<b>Tipo de elemento</b>	En línea
<b>Descripción</b>	Se emplea para enlazar todo tipo de recursos

#### 4.4.1. Idioma del enlace (hreflang)

El enlace puede indicar al navegador el idioma del recurso que se enlaza. Para establecer el valor del idioma, se utiliza un código estandarizado de dos letras. Además del idioma genérico, también se puede indicar una variación idiomática. Ejemplo de códigos de idioma más utilizados:

Código	Idioma	Variación idiomática
en	Inglés	-
en-AU	Inglés	Australia
en-US	Inglés	Estados Unidos
es	Español	-
es-AR	Español	Argentina
es-ES	Español	España
es-MX	Español	México

Otros códigos utilizados son: `fr` (francés), `de` (alemán), `it` (italiano), `nl` (holandés), `el` (griego), `pt` (portugués), `ar` (árabe), `he` (hebreo), `ru` (ruso), `zh` (chino), `ja` (japonés).

La lista completa de códigos de idioma está definida en [el estándar ISO 639](#).

#### 4.4.2. Tipo de contenido (type)

Se utiliza para notificar al navegador sobre el tipo de contenido que se enlaza. Se indica mediante una cadena de texto cuyos posibles valores también están estandarizados. Los valores de los contenidos más utilizados son los siguientes: `"text/html"` (páginas HTML), `"image/png"` (imágenes con formato PNG), `"image/gif"` (imágenes con formato GIF), `"text/css"` (hojas de estilo CSS), `"application/rss+xml"` (archivos RSS).

La lista completa de tipos de contenido se define en [los estándares RFC 2045 y RFC 2046](#).

### 4.4.3. Tipo de relación (rel y rev)

Los enlaces pueden proporcionar información adicional muy útil para los navegadores y para los motores de búsqueda como Google. Los atributos `rel` y `rev` permiten indicar la relación que la página actual tiene con la página a la que se enlaza (atributo `rel`) y la relación que tiene la página enlazada con la página actual (atributo `rev`).

Los tipos de relación definidos son los siguientes:

- `alternate` – Indica que es una versión alternativa al documento actual (puede ser una versión en otro idioma o una versión preparada para otro medio, como una impresora o un dispositivo móvil)
- `stylesheet` – Indica que se ha enlazado una hoja de estilos
- `start` – Indica que se trata del primer documento de una colección de documentos (por ejemplo el primer capítulo de un libro)
- `next` – Indica que es el documento que sigue al actual dentro de una secuencia lógica de documentos (por ejemplo, los capítulos de un libro)
- `prev` – Indica que es el documento que precede al actual dentro de una secuencia lógica de documentos (por ejemplo, los capítulos de un libro)
- `contents` – Indica que el recurso enlazado es el documento que contiene la tabla de contenidos de la colección de documentos (por ejemplo, el índice de un libro).
- `bookmark` – Establece el enlace actual como un "marcador" o "favorito". Un marcador es un enlace que constituye un punto de entrada muy importante dentro del documento.

La especificación oficial de HTML define la [lista completa de tipos de relaciones](#) que se pueden utilizar.

### 4.4.4. Codificación de caracteres (charset)

Además del idioma, tipo de contenido y relación del recurso que se enlaza, los enlaces también pueden indicar la codificación de caracteres que utiliza la página web enlazada.

Los valores que se pueden utilizar también están estandarizados y las codificaciones más utilizadas son `UTF-8` y `ISO-8859-1`, aunque existen decenas de códigos definidos (`ISO-10646-UCS-2`, `IBM852`, `Big5-HKSCS`, `windows-1252`, `HZ-GB-2312`).

El organismo IANA publica la [lista completa de codificaciones de caracteres disponibles](#).

## 4.5. Otros tipos de enlaces

HTML define las etiquetas `<script>` y `<link>` para enlazar recursos que se deben cargar automáticamente. Cuando el navegador encuentra alguna de estas dos etiquetas, descarga los recursos enlazados y los aplica a la página web.

La etiqueta `<script>` tiene dos modos de funcionamiento, ya que se emplea tanto para insertar un bloque de código JavaScript en la página como para enlazar un archivo JavaScript externo.

<b>Etiqueta</b>	<code>&lt;script&gt;</code>
<b>Atributos comunes</b>	-
<b>Atributos propios</b>	<ul style="list-style-type: none"><li>● <code>src = "url"</code> - Indica la dirección del archivo que contiene el código</li><li>● <code>type = "tipo_de_contenido"</code> - Permite "avisar" al navegador sobre el tipo de código que se incluye (normalmente JavaScript)</li><li>● <code>defer = "defer"</code> - El código no va a modificar el contenido de la página web</li><li>● <code>charset = "tipo_de_charset"</code> - Describe la codificación del código enlazado</li></ul>
<b>Tipo de elemento</b>	Bloque y en línea (también puede ser una etiqueta vacía)
<b>Descripción</b>	Se emplea para enlazar o definir un bloque de código (normalmente JavaScript)

Aunque la etiqueta `<script>` permite enlazar código de varios lenguajes de programación, el uso habitual de `<script>` consiste en enlazar un archivo JavaScript externo

El atributo `type` utilizado habitualmente para los archivos JavaScript es `"text/javascript"`. El atributo `src` es equivalente al atributo `href` de los enlaces creados con la etiqueta `<a>`. La URL indicada en el atributo `src` puede ser absoluta o relativa y externa o interna.

Además de enlazar un archivo JavaScript externo, la misma etiqueta `<script>` también permite incluir en la página web un bloque de código JavaScript:

Cuando se incluye código JavaScript en la propia página XHTML, se debe insertar dentro de una sección especial llamada `CDATA`. Para ello, el código JavaScript se debe encerrar entre `<![CDATA[` y `]]>`. Cuando el navegador encuentra una sección de este tipo, no procesa su contenido como si fuera XHTML y por tanto no tiene en cuenta los posibles errores de validación de XHTML.

La etiqueta `<script>` (tanto cuando enlaza, como cuando incluye directamente el código) puede aparecer en cualquier parte del documento HTML, aunque normalmente se incluye dentro de la cabecera de la página (`<head>...</head>`).

La segunda etiqueta de XHTML para enlazar recursos es `<link>`, que permite enlazar y relacionar la página con otros recursos externos.

<b>Etiqueta</b>	<code>&lt;link&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	<ul style="list-style-type: none"><li>● Los siguientes con el mismo significado que para la etiqueta "a": <code>charset</code>, <code>href</code>, <code>hreflang</code>, <code>type</code>, <code>rel</code> y <code>rev</code></li><li>● <code>media</code> = "tipo_de_medio" - Indica el medio para el que debe aplicarse la relación</li></ul>
<b>Tipo de elemento</b>	Etiqueta vacía
<b>Descripción</b>	Se emplea para enlazar y establecer relaciones entre el documento y otros recursos



Al contrario que `<script>`, la etiqueta `<link>` solamente se puede incluir dentro de la cabecera del documento. Se pueden añadir tantas etiquetas `<link>` como sean necesarias, pero siempre dentro de `<head>...</head>`.

El atributo `media` hace referencia al medio para el que es válida la relación con el recurso enlazado. Los medios disponibles también están estandarizados, siendo los más comunes `screen` para los contenidos mostrados en pantalla, `print` para las impresoras y `handheld` para los dispositivos móviles.

El uso habitual de la etiqueta `<link>` es el de enlazar las hojas de estilos CSS utilizadas por las páginas web

Es habitual establecer los atributos `rel` y `type` para indicar el tipo de recurso enlazado y su relación con la página web. La URL del recurso enlazado se indica en el atributo `href`, que admite tanto URL absolutas como relativas.

## 4.6. Ejemplos de enlaces habituales

### 4.6.1. Enlace al inicio del sitio web

```
<a href="/">Inicio</a>
```

Al pulsar el enlace anterior desde cualquier página web, se vuelve directamente a la página de inicio, *home* o página principal del sitio web.

### 4.6.2. Enlace a un email

```
<a href="mailto:nombre@direccion.com" title="Dirección de email para  
solicitar más información">  
Solicita más información  
  
</a>
```

Al pinchar sobre el enlace anterior, se abre automáticamente el programa de correo electrónico del ordenador del usuario y se establece la dirección de envío al valor indicado después de `mailto:`. La sintaxis es la misma que la de un enlace normal, salvo que se cambia el prefijo `http://` por `mailto:`:

La sintaxis de `mailto:` permite utilizarlo para otros ejemplos más complejos:

```
<!-- Envío del correo electrónico a varias direcciones a la vez -->
<a href="mailto:nombre@direccion.com,otro_nombre@direccion.com">Solicita
más información</a>

<!-- Añadir un "asunto" inicial al correo electrónico -->
<a href="mailto:nombre@direccion.com?subject=Solicitud de más
información">Solicita más información</a>

<!-- Añadir un texto inicial en el cuerpo del correo electrónico -->

<a href="mailto:nombre@direccion.com?body=Estaría interesado en solicitar
más información sobre sus productos">Solicita más información</a>
```

Todas las opciones anteriores se pueden combinar entre sí para realizar ejemplos más avanzados. Aunque el uso de `mailto:` puede parecer una ventaja, su uso está desaconsejado. Si se incluye una dirección de correo electrónico directamente en una página web, es muy probable que en poco tiempo esa dirección de email se encuentre llena de correo electrónico basura o "*spam*", ya que existen programas automáticos encargados de rastrear sistemáticamente todas las páginas web de Internet para encontrar direcciones de correo electrónico válidas.

La forma de mostrar las direcciones de correo electrónico en las páginas web consiste en incluir la dirección en una imagen o indicarla de forma que solamente los usuarios puedan entenderlo:

```
<p>La dirección de correo es <strong>nombre (arroba)
direccion.com</strong></p>
<p>La dirección de correo es
<strong>nombre_arroba_direccion.com</strong></p>
<p>La dirección de correo es
<strong>nombreQUITAESTO@direccion.com</strong></p>
<p>La dirección de correo es
<strong>nombre (ARROBA) direccion.com</strong></p>

<p>La dirección de correo es <strong>nombre @ direccion . com</strong></p>
```

### 4.6.3. Enlace a un archivo FTP

Para enlazar un archivo almacenado en un servidor FTP, la parte del protocolo de la URL debe cambiar de `http://` a `ftp://`:

```
<a href="ftp://ftp.ejemplo.com/ruta/archivo.zip" title="Archivo comprimido de los contenidos">
```

Descarga un ZIP con todos los contenidos

```
</a>
```

## 4.6.4. Enlazar varias hojas de estilos CSS

```
<link rel="stylesheet" type="text/css" href="/css/comun.css" />
```

```
<link rel="stylesheet" type="text/css" href="/css/secciones.css" />
```

## 4.6.5. Enlazar hojas de estilos CSS para diferentes medios

```
<link rel="stylesheet" type="text/css" href="/css/comun.css" media="screen, projection" />
```

```
<link rel="stylesheet" type="text/css" href="/css/impresora.css" media="print" />
```

```
<link rel="stylesheet" type="text/css" href="/css/movil.css" media="handheld" />
```

## 4.6.6. Enlazar el favicon

El *favicon* o icono para favoritos es el pequeño icono que muestran las páginas en varias partes del navegador. Dependiendo del navegador que se utilice, este icono se muestra en la barra de direcciones, en la barra de título del navegador y/o en el menú de favoritos/marcadores

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

Aunque en principio la imagen debería ser de tipo `.ICO` (formato gráfico de los iconos), algunos navegadores soportan favicons en otros formatos gráficos más habituales (como por ejemplo `.PNG`)

## 4.6.7. Enlazar un archivo RSS

```
<link rel="alternate" type="application/rss+xml" title="Resumen de todos los artículos del blog" href="/feed.xml" />
```

## 4.6.8. Enlazar hojas de estilos, favicon y RSS

En una misma página se pueden incluir varias etiquetas `<link>`, por lo que es habitual que las páginas enlacen hojas de estilos, favicon y archivos RSS de forma conjunta:

```
<link rel="stylesheet" type="text/css" href="/css/impresora.css" media="print" />
```

```
<link rel="stylesheet" type="text/css" href="/css/movil.css" media="handheld" />
```

```
<style type="text/css" media="screen,projection">
```

```
  @import '/css/main.css';
```

```
</style>
```

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

```
<link rel="alternate" type="application/rss+xml" title="Resumen de todos los artículos del blog" href="/feed.xml" />
```

## 4.6.9. Indicar que existe una versión de la página en otro idioma

```
<head>
```

```
<title>English tutorial</title>
```

```
<link lang="es" xml:lang="es" title="El tutorial en español" type="text/html"
rel="alternate" hreflang="es" href="http://www.ejemplo.com/tutorial/espanol.html" />

</head>
```

## 4.6.10. Indicar que existe una versión de la página preparada para imprimir

```
<head>

<link media="print" title="El tutorial en PDF" type="application/pdf" rel="alternate"
href="http://www.ejemplo.com/tutorial/documento.pdf" />

</head>
```

## 4.6.11. Indicar que existe una página que es índice de la página actual

```
<head>

<title>Tutorial - Capítulo 5</title>

<link rel="start" title="El índice del tutorial" type="text/html"
href="http://www.ejemplo.com/tutorial/indice.html" />

</head>
```

## 5. Listas

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).

## 5.1. Listas no ordenadas

Las listas no ordenadas son las más sencillas y 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.

<b>Etiqueta</b>	<code>&lt;ul&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	-
<b>Tipo de elemento</b>	Bloque
<b>Descripción</b>	Se emplea para definir listas no ordenadas

<b>Etiqueta</b>	<code>&lt;li&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	-
<b>Tipo de elemento</b>	Bloque
<b>Descripción</b>	Se emplea para definir los elementos de las listas (ordenadas y no ordenadas)

## 5.2. Listas ordenadas

Las listas ordenadas son casi idénticas a las listas no ordenadas, salvo que en este caso los elementos relacionados se muestran siguiendo un orden determinado. 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.

<b>Etiqueta</b>	<code>&lt;ol&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	-
<b>Tipo de elemento</b>	Bloque
<b>Descripción</b>	Se emplea para definir listas ordenadas

## 5.3. 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.

<b>Etiqueta</b>	<code>&lt;dl&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	-

<b>Tipo de elemento</b>	Bloque
<b>Descripción</b>	Se emplea para definir listas de definición

<b>Etiqueta</b>	<dt>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	-
<b>Tipo de elemento</b>	Bloque
<b>Descripción</b>	Se emplea para definir los términos de los elementos de una lista de definición

<b>Etiqueta</b>	<dd>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	-
<b>Tipo de elemento</b>	Bloque
<b>Descripción</b>	Se emplea para indicar las definiciones de los elementos de una lista de definición



## 6. Imágenes y objetos

### 6.1. Imágenes

Las imágenes son uno de los elementos más importantes de las páginas web. De hecho, prácticamente todas las páginas web contienen alguna imagen y la mayoría incluye decenas de imágenes. Dentro de las imágenes que se pueden incluir en una página HTML se deben distinguir dos tipos: las **imágenes de contenido** y las **imágenes de adorno**.

Las **imágenes de contenido** son las que proporcionan información y complementan la información textual. Las **imágenes de adorno** son las que se utilizan para hacer bordes redondeados, para mostrar pequeños iconos en las listas de elementos, para mostrar fondos de página, etc. Las imágenes de contenido se incluyen directamente en el código HTML mediante la etiqueta `<img>` y las imágenes de adorno no se deberían incluir en el código HTML, sino que deberían emplearse hojas de estilos CSS para mostrarlas.

A continuación se muestra la definición de la etiqueta `<img>`, utilizada para incluir las imágenes en las páginas HTML:

Etiqueta	<code>&lt;img&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>

<b>Atributos propios</b>	<ul style="list-style-type: none"> <li>● <code>src = "url"</code> - Indica la URL de la imagen que se muestra</li> <li>● <code>alt = "texto"</code> - Descripción corta de la imagen</li> <li>● <code>longdesc = "url"</code> - Indica una URL en la que puede encontrarse una descripción más detallada de la imagen</li> <li>● <code>name = "texto"</code> - Nombre del elemento imagen</li> <li>● <code>height = "unidad_de_medida"</code> - Indica la altura con la que se debe mostrar la imagen (no es obligatorio que coincida con la altura original de la imagen)</li> <li>● <code>width = "unidad_de_medida"</code> - Indica la anchura con la que se debe mostrar la imagen (no es obligatorio que coincida con la anchura original de la imagen)</li> </ul>
<b>Tipo de elemento</b>	En línea y etiqueta vacía
<b>Descripción</b>	Se emplea para incluir imágenes en los documentos

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. El atributo `alt` permite describir el contenido de la imagen mediante un texto breve. Las descripciones deben tener una longitud inferior a 1024 caracteres y son útiles para las personas y dispositivos discapacitados que no pueden acceder a las imágenes.

Cómo `<img>` es una etiqueta vacía, no tiene etiqueta de cierre. No obstante, para que la página XHTML sea válida, todas las etiquetas deben estar cerradas. Como ya se explicó anteriormente, para cerrar una etiqueta vacía se incluyen los caracteres `/>` al final de la etiqueta.

HTML no impone ninguna restricción sobre el formato gráfico que se puede utilizar en las imágenes, por lo que en principio la etiqueta `<img>` puede incluir cualquier formato gráfico existente. Sin

embargo, si la imagen utiliza un formato poco habitual, todos o algunos navegadores no serán capaces de mostrar esa imagen.

La recomendación es utilizar uno de los tres siguientes formatos gráficos que entienden todos los navegadores modernos: GIF, JPG y PNG. El formato PNG presenta el inconveniente de que los navegadores obsoletos como Internet Explorer 6 no muestran correctamente las imágenes con transparencias de 24 bits.

El atributo `longdesc` no se utiliza de forma habitual, pero permite indicar la URL en la que se puede encontrar más información sobre la imagen. Como el atributo `alt` sólo permite incluir descripciones de hasta 1024 caracteres, el atributo `longdesc` se emplea con las imágenes complejas que necesitan mucha información para ser descritas

Como norma general, se debe guardar todas las imágenes de contenido en un directorio especial independiente del resto de contenidos HTML. Además, el directorio siempre suele llamarse de la misma manera: `"imagenes"` o `"images"` en inglés.

Los atributos `width` y `height` se utilizan para indicar la anchura y altura con la que se muestran las imágenes, por lo que son los más contradictorios. Como ya se ha comentado, HTML estructura de forma correcta los contenidos de la página y CSS define el aspecto gráfico con el que se muestran los contenidos. En principio, la anchura y la altura con la que se muestra una imagen es parte de su aspecto gráfico, por lo que debería ser propio de CSS y no de XHTML.

Sin embargo, en la práctica no es viable establecer la anchura y altura de todas las imágenes de contenidos mediante CSS. Si el sitio web dispone de muchas imágenes, la sobrecarga de estilos diferentes que debería definir CSS sería contraproducente. Por este motivo, los atributos `width` y `height` son la excepción a la norma de que el código HTML no haga referencia al aspecto de los contenidos.

```

```

```

```

Si el valor del atributo `width` o `height` se indica mediante un número entero, el navegador supone que hace referencia a la unidad de medida píxel. Por tanto, en el ejemplo anterior, la primera foto se muestra con una anchura de 200 píxel y una altura de 350 píxel.

También es posible indicar la anchura y altura en forma de porcentaje. En este caso, el porcentaje hace referencia a la altura/anchura del elemento en el que está contenida la imagen. Si la imagen no se encuentra dentro de ningún otro elemento, hace referencia a la anchura/altura total de la página.

```
<div>

  

</div>
```

## 6.2. Mapas de imagen

Un mapa de imagen permite definir diferentes zonas "*pinchables*" dentro de una imagen. El usuario puede pinchar sobre cada una de las zonas definidas y cada una de ellas puede apuntar a una URL diferente. Siguiendo el ejemplo anterior, una sola imagen que muestre un mapa de todos los continentes puede definir una zona diferente para cada continente. De esta forma, el usuario puede pinchar sobre la zona correspondiente a cada continente para que el navegador muestre la página que contiene los viajes disponibles a ese destino.

Las zonas o regiones que se pueden definir en una imagen se crean mediante rectángulos, círculos y polígonos. Para crear un mapa de imagen, en primer lugar se inserta la imagen original mediante la etiqueta `<img>`. A continuación, se utiliza la etiqueta `<map>` para definir las zonas o regiones de la imagen. Cada zona se define mediante la etiqueta `<area>`.

<b>Etiqueta</b>	<code>&lt;map&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	<ul style="list-style-type: none"> <li> <code>name = "texto"</code> - Nombre que identifica de forma única al mapa definido (es obligatorio indicar un nombre único) </li> </ul>
<b>Tipo de elemento</b>	Bloque y en línea
<b>Descripción</b>	Se emplea para definir mapas de imagen

Etiqueta	<area>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> , <a href="#">eventos</a> y <a href="#">foco</a>
Atributos propios	<ul style="list-style-type: none"> <li>● <code>href = "url"</code> - URL a la que se accede al pinchar sobre el área</li> <li>● <code>nohref = "nohref"</code> - Se emplea para las áreas que no son seleccionables</li> <li>● <code>shape = "default   rect   circle   poly"</code> - Indica el tipo de área que se define (toda la imagen, rectangular, circular o poligonal)</li> <li>● <code>coords = "lista de números"</code> - Se trata de una lista de números separados por comas que representan las coordenadas del área. Rectangular = <code>X1,Y1,X2,Y2</code> (coordenadas X e Y del vértice superior izquierdo y coordenadas X e Y del vértice inferior derecho). Circular = <code>X1,Y1,R</code> (coordenadas X e Y del centro y radio del círculo). Poligonal = <code>X1,Y1,X2,Y2,...,XnYn</code> (coordenadas de los vértices del polígono. Si las últimas coordenadas no son iguales que las primeras, se cierra automáticamente el polígono uniendo ambos vértices)</li> </ul>
Tipo de elemento	Etiqueta vacía

Descripción	Se emplea para definir las distintas áreas que forman un mapa de imagen
-------------	---

Si una imagen utiliza un mapa de imagen, debe indicarlo mediante el atributo `usemap`. El valor del atributo debe ser el nombre del mapa de imagen definido en otra parte del mismo documento HTML.

Las áreas se definen mediante el atributo `shape` que indica el tipo de área y `coords` que es una lista de coordenadas cuyo significado depende del tipo de área definido. El enlace de cada área se define mediante el atributo `href`, con la misma sintaxis y significado que para los enlaces normales.

## 6.3. Objetos

Además de las imágenes, HTML permite incluir en las páginas web otros elementos mucho más complejos, como *applets* de Java y vídeos en formato QuickTime o Flash. La mayoría de este tipo de contenidos no los interpreta el navegador directamente, sino que hace uso de pequeños programas llamados *plugins* y que se encargan de tratar con este tipo de elementos complejos.

La etiqueta `<object>` es la que permite "embeber" o incluir en las páginas HTML cualquier tipo de contenido complejo:

Etiqueta	<code>&lt;object&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	<ul style="list-style-type: none"><li>● <code>data = "url"</code> - Indica la URL de los datos que utiliza el objeto</li><li>● <code>classid</code>, <code>codebase</code>, <code>codetype</code> - Información específica que depende del tipo de objeto</li><li>● <code>type</code> - Indica el tipo de contenido de los datos</li><li>● <code>height = "unidad_de_medida"</code> - Indica la altura con la que se debe mostrar el objeto</li></ul>

	<ul style="list-style-type: none"> <li>● <code>width = "unidad_de_medida"</code> - Indica la anchura con la que se debe mostrar el objeto</li> </ul>
<b>Tipo de elemento</b>	Bloque y en línea
<b>Descripción</b>	Se emplea para embeber objetos en los documentos

El atributo `data` se emplea para indicar la URL del recurso que se va a incluir. El atributo `type` indica el tipo de contenido de los datos del objeto. Los posibles valores de `type` están estandarizados y coinciden con los del atributo `type` de la etiqueta `<a>` que se explicó anteriormente.

El propio estándar de HTML incluye ejemplos de uso de esta etiqueta. Incluir un vídeo en formato MPEG. También se pueden incluir varias versiones alternativas de un mismo contenido. Así, si el navegador no es capaz de interpretar el formato por defecto, puede optar por cualquiera de los otros formatos alternativos

A los objetos también se les puede pasar información adicional en forma de parámetros mediante la etiqueta `<param>`:

<b>Etiqueta</b>	<code>&lt;param&gt;</code>
<b>Atributos comunes</b>	<code>id</code>
<b>Atributos propios</b>	<ul style="list-style-type: none"> <li>● <code>name = "texto"</code> - Indica el nombre del parámetro</li> <li>● <code>value = "texto"</code> - Indica el valor del parámetro</li> </ul>
<b>Tipo de elemento</b>	Etiqueta vacía

Descripción	Se emplea para indicar el valor de los parámetros del objeto
-------------	--

Las etiquetas `<param>` siempre se incluyen en el interior de las etiquetas `<object>`

Uno de los principales inconvenientes de `<object>` es la forma de incluir vídeos en formato Flash en las páginas HTML. Si se utiliza el siguiente código:

```
<object data="nombre_video.swf" type="application/x-shockwave-flash"></object>
```

El elemento anterior es correcto desde el punto de vista técnico, pero provoca que algunos navegadores como Internet Explorer no visualicen el vídeo hasta que se ha descargado completamente. Si se trata de un vídeo largo, esta solución no es válida para el usuario.

Por este motivo, se utiliza una solución alternativa para incluir vídeos Flash en las páginas HTML: el uso de la etiqueta `<embed>`. Aunque esta solución funciona correctamente, no se trata de una solución válida desde el punto de vista del estándar de XHTML, por lo que las páginas que incluyan esta solución no pasarán correctamente el proceso de validación.

Etiqueta	<code>&lt;embed&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	<ul style="list-style-type: none"><li>● <code>src = "url"</code> - Indica la URL del archivo u objeto que se incluye en la página</li><li>● <code>type = "tipo_de_contenido"</code> - Indica el tipo de contenido del objeto (flash, quicktime, java, etc.)</li><li>● <code>height = "unidad_de_medida"</code> - Indica la altura con la que se debe mostrar el objeto</li><li>● <code>width = "unidad_de_medida"</code> - Indica la anchura con la que se debe mostrar el objeto</li></ul>



Tipo de elemento	Bloque
Descripción	Se emplea para embeber objetos en los documentos

Este es el motivo por el que los sitios web más populares de vídeos en formato Flash proporcionan un código similar al siguiente para incluir sus vídeos en las páginas HTML:

```
<object width="425" height="350">

  <param name="movie" value="http://www.youtube.com/v/MsH0rBWCYjs"></param>

  <param name="wmode" value="transparent"></param>

  <embed src="http://www.youtube.com/v/MsH0rBWCYjs"
type="application/x-shockwave-flash" wmode="transparent" width="425"
height="350"></embed>

</object>
```

Una vez más, se debe tener en cuenta que la solución anterior de utilizar la etiqueta `<embed>` es correcta desde el punto de vista del usuario (no tiene que esperar a que el vídeo se descargue completamente para poder verlo) pero no es una solución técnicamente válida, ya que la etiqueta `<embed>` no es parte del estándar XHTML.

## 7. 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

A pesar de que las tablas HTML son fáciles de comprender y utilizar, son uno de los elementos más polémicos de HTML. El problema de las tablas es que no siempre se utilizan adecuadamente. Aunque parezca obvio, las tablas se deben utilizar para mostrar información tabular

### 7.1. Tablas básicas

---

# HTML

---

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.

La etiqueta `<table>` encierra todas las filas y columnas de la tabla. Las etiquetas `<tr>` (del inglés "table row") definen cada fila de la tabla y encierran todas las columnas. Por último, la etiqueta `<td>` (del inglés "table data cell") define cada una de las columnas de las filas, aunque realmente HTML no define columnas sino *celdas de datos*.

Al definir una tabla, se debe pensar en primer lugar en las filas que la forman y a continuación en las columnas. El motivo es que HTML procesa primero las filas y por eso las etiquetas `<tr>` aparecen antes que las etiquetas `<td>`.

Etiqueta	<code>&lt;table&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	<ul style="list-style-type: none"><li>● <code>summary = "texto"</code> - Permite describir el contenido de la tabla (lo utilizan los buscadores y las personas discapacitadas)</li></ul>
Tipo de elemento	Bloque
Descripción	Se emplea para definir tablas de datos

Etiqueta	<code>&lt;tr&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	-

Tipo de elemento	Bloque
Descripción	Se emplea para definir cada fila de las tablas de datos

Etiqueta	<td>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	<ul style="list-style-type: none"><li>● <code>abbr = "texto"</code> - Permite describir el contenido de la celda (empleado sobre todo en los navegadores utilizados por personas discapacitadas)</li><li>● <code>headers = "lista_de_id"</code> - Indica las celdas que actúan como cabeceras para esta celda (los títulos de las columnas y filas). Se indica como una lista de valores del atributo "id" de celdas</li><li>● <code>scope = "col, row, colgroup, rowgroup"</code> - Indica las celdas para las que esta celda será su cabecera. Ej: <code>scope="col"</code> indica que esta celda es la cabecera de todas las demás celdas que están en la misma columna</li><li>● <code>colspan = "numero"</code> - Número de columnas que ocupa esta celda</li><li>● <code>rowspan = "numero"</code> - Número de filas que ocupa esta celda</li></ul>
Tipo de elemento	Bloque

Descripción	Se emplea para definir cada una de las celdas que forman las filas de una tabla, es decir, las columnas de la tabla
-------------	---

De todos los atributos disponibles para las celdas, los más utilizados son `rowspan` y `colspan`, que se emplean para construir tablas complejas como las que se verán más adelante. Entre los demás atributos, sólo se utiliza de forma habitual el atributo `scope`, sobre todo con las celdas de cabecera que se ven a continuación.

Normalmente, algunas de las celdas de la tabla se utilizan como cabecera de las demás celdas de la fila o de la columna. En este caso, HTML define la etiqueta `<th>` (del inglés *"table header cell"*) para indicar que una celda es cabecera de otras celdas.

Etiqueta	<code>&lt;th&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	<ul style="list-style-type: none"><li>● <code>abbr = "texto"</code> - Permite describir el contenido de la celda (empleado sobre todo en los navegadores utilizados por personas discapacitadas)</li><li>● <code>headers = "lista_de_id"</code> - Indica las celdas que actúan como cabeceras para esta celda (los títulos de las columnas y filas). Se indica como una lista de ID de celdas</li><li>● <code>scope = "col, row, colgroup, rowgroup"</code> - Indica las celdas para las que esta celda será su cabecera. Ej: <code>scope="col"</code> indica que esta celda es la cabecera de todas las demás celdas que están en la misma columna</li><li>● <code>colspan = "numero"</code> - Número de columnas que ocupa esta celda</li></ul>

	<ul style="list-style-type: none"> <li>● <code>rowspan = "numero"</code> - Número de filas que ocupa esta celda</li> </ul>
Tipo de elemento	Bloque
Descripción	Se emplea para definir las celdas que son cabecera de una fila o de una columna de la tabla

Los atributos de la etiqueta `<th>` son idénticos que los atributos definidos para la etiqueta `<td>`. En este caso, el atributo más utilizado es `scope`, que permite indicar si la celda es cabecera de la fila o de la columna (`<th scope="row">` y `<th scope="col">` respectivamente).

Por otra parte, HTML define la etiqueta `<caption>` para establecer la leyenda o título de una tabla. La etiqueta debe colocarse inmediatamente después de la etiqueta `<table>` y cada tabla sólo puede incluir una etiqueta `<caption>`.

Etiqueta	<code>&lt;caption&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	-
Tipo de elemento	En línea
Descripción	Se emplea para definir la leyenda o título de una tabla

## 7.2. Tablas avanzadas

Algunas tablas complejas están formadas por más elementos que filas y celdas de datos. Así, es común que las tablas más avanzadas dispongan de una sección de cabecera, una sección de pie y

varias secciones de datos. Además, también es posible agrupar varias columnas de forma lógica para poder aplicar estilos similares a un determinado grupo de columnas.

Las partes que componen las tablas complejas se definen mediante las etiquetas `<thead>`, `<tbody>` y `<tfoot>`. La cabecera de la tabla se define con la etiqueta `<thead>`, el pie de la tabla se define mediante `<tfoot>` y cada sección de datos se define con una etiqueta `<tbody>`.

<b>Etiquetas</b>	<code>&lt;thead&gt;</code> <code>&lt;tbody&gt;</code> <code>&lt;tfoot&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	-
<b>Tipo de elemento</b>	Bloque
<b>Descripción</b>	Se emplean para agrupar varias filas en una cabecera (thead) un pie (tfoot) o una sección (tbody) de una tabla

Cada tabla puede contener solamente una cabecera y un pie, pero puede incluir un número ilimitado de secciones. Si se define una cabecera y/o un pie, las etiquetas `<thead>` y/o `<tfoot>` deben colocarse inmediatamente antes que cualquier etiqueta `<tbody>`.

Aunque al principio resulta extraño, el elemento `<tfoot>` siempre se escribe antes que cualquier elemento `<tbody>` en el código HTML. De hecho, si la etiqueta `<tfoot>` aparece después de un elemento `<tbody>`, la página no se considera válida.

La etiqueta `<tbody>` permite realizar agrupaciones de filas, pero en ocasiones se necesitan agrupar columnas. Aunque su uso no es muy común, HTML define dos etiquetas similares para agrupar columnas: `<col>` y `<colgroup>`.

La etiqueta `<col>` se utiliza para asignar los mismos atributos a varias columnas de forma simultánea. De esta forma, la etiqueta `<col>` no agrupa columnas, sino que sólo asigna atributos comunes a varias columnas.

Por otra parte, la etiqueta `<colgroup>` se emplea para agrupar de forma estructural varias columnas de la tabla. La forma habitual de indicar el número de columnas que abarca la agrupación es utilizar el atributo `span`, que establece el número de columnas de cada agrupación.

El uso de las etiquetas `<col>` y `<colgroup>` no está muy extendido, debido a que la mayoría de navegadores no soportan muchas de sus funcionalidades.

## 8. Formularios

### 8.1. Formularios básicos

Los formularios más sencillos se pueden crear utilizando solamente dos etiquetas: `<form>` y `<input>`.

La etiqueta `<form>` encierra todos los contenidos del formulario (botones, cuadros de texto, listas desplegables) y la etiqueta `<input>` permite definir varios tipos diferentes de elementos (botones y cuadros de texto).

<b>Etiqueta</b>	<code>&lt;form&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>

Atributos propios	<ul style="list-style-type: none"> <li>● <code>action = "url"</code> - Indica la URL que se encarga de procesar los datos del formulario</li> <li>● <code>method = "POST o GET"</code> - Método HTTP empleado al enviar el formulario</li> <li>● <code>enctype = "application/x-www-form-urlencoded o multipart/form-data"</code> - Tipo de codificación empleada al enviar el formulario al servidor (sólo se indica de forma explícita en los formularios que permiten adjuntar archivos)</li> <li>● <code>accept = "tipo_de_contenido"</code> - Lista separada por comas de todos los tipos de archivos aceptados por el servidor (sólo para los formularios que permiten adjuntar archivos)</li> <li>● Otros: <code>accept-charset</code>, <code>onsubmit</code>, <code>onreset</code></li> </ul>
Tipo de elemento	Bloque
Descripción	Se emplea para insertar un formulario en la página

La mayoría de formularios utilizan sólo los atributos `action` y `method`. El atributo `action` indica la URL de la aplicación del servidor que se encarga de procesar los datos introducidos por los usuarios. Esta aplicación también se encarga de generar la respuesta que muestra el navegador.

El atributo `method` establece la forma en la que se envían los datos del formulario al servidor. Este atributo hace referencia al método HTTP, por lo que no es algo propio de HTML. Los dos valores que se utilizan en los formularios son `GET` y `POST`. De esta forma, casi todos los formularios incluyen el atributo `method="get"` o el atributo `method="post"`.

Al margen de otras diferencias técnicas, el método `POST` permite el envío de mucha más información que el método `GET`. En general, el método `GET` admite como máximo el envío de unos 500 bytes de información. La otra gran limitación del método `GET` es que no permite el envío de archivos adjuntos con el formulario. Además, los datos enviados mediante `GET` se ven en la barra de direcciones del



navegador (se añaden al final de la URL de la página), mientras que los datos enviados mediante `POST` no se pueden ver tan fácilmente.

## 8.2. Elementos de formulario

Los elementos de formulario como botones y cuadros de texto también se denominan "*campos de formulario*" y "*controles de formulario*". La mayoría de controles se crean con la etiqueta `<input>`, por lo que su definición formal y su lista de atributos es muy extensa:

<b>Etiqueta</b>	<code>&lt;input&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> , <a href="#">eventos</a> y <a href="#">foco</a>

<b>Atributos propios</b>	<ul style="list-style-type: none"> <li>● <code>type = "text   password   checkbox   radio   submit   reset   file   hidden   image   button"</code> - Indica el tipo de control que se incluye en el formulario</li> <li>● <code>name = "texto"</code> - Asigna un nombre al control (es imprescindible para que el servidor pueda procesar el formulario)</li> <li>● <code>value = "texto"</code> - Valor inicial del control</li> <li>● <code>size = "unidad_de_medida"</code> - Tamaño inicial del control (para los campos de texto y de password se refiere al número de caracteres, en el resto de controles se refiere a su tamaño en píxel)</li> <li>● <code>maxlength = "numero"</code> - Máximo número de caracteres para los controles de texto y de password</li> <li>● <code>checked = "checked"</code> - Para los controles checkbox y radiobutton permite indicar qué opción aparece preseleccionada</li> <li>● <code>disabled = "disabled"</code> - El control aparece deshabilitado y su valor no se envía al servidor junto con el resto de datos</li> <li>● <code>readonly = "readonly"</code> - El contenido del control no se puede modificar</li> <li>● <code>src = "url"</code> - Para el control que permite crear botones con imágenes, indica la URL de la imagen que se emplea como botón de formulario</li> <li>● <code>alt = "texto"</code> - Descripción del control</li> </ul>
<b>Tipo de elemento</b>	<p>En línea y etiqueta vacía</p>
<b>Descripción</b>	<p>Se emplean para insertar un control en un formulario</p>

A continuación se muestran ejemplos para los diez controles que se pueden crear con la etiqueta `<input>`.

## 8.2.1. Cuadro de texto

Se trata del elemento más utilizado en los formularios. En el caso más sencillo, se muestra un cuadro de texto vacío en el que el usuario puede escribir cualquier texto

El atributo `type` diferencia a cada uno de los diez controles que se pueden crear con la etiqueta `<input>`. Para los cuadros de texto, su valor es `text`. El atributo `name` es el más importante en los campos del formulario. De hecho, si un campo no incluye el atributo `name`, sus datos no se envían al servidor. El valor que se indica en el atributo `name` es el nombre que utiliza la aplicación del servidor para obtener el valor de este campo de formulario.

Cuando el usuario pulsa el botón de envío del formulario, el navegador envía los datos a una aplicación del servidor para que procese la información y genere una respuesta adecuada. En el servidor, la aplicación que procesa los datos debe obtener en primer lugar toda la información introducida por el usuario. Para ello, utiliza el valor del atributo `name` para obtener los datos de cada control del formulario.

Como el valor del atributo `name` se utiliza en aplicaciones programadas, es esencial ponerse de acuerdo con el programador de la aplicación, no se debe modificar su valor sin modificar la aplicación y no se deben utilizar caracteres problemáticos en programación (espacios en blanco, acentos y caracteres como ñ o ç).

El atributo `value` se emplea para establecer el valor inicial del cuadro de texto. Si se crea un formulario para insertar datos, los cuadros de texto deberían estar vacíos. Por lo tanto, o no se añade el atributo `value` o se incluye con un valor vacío `value=""`. Si por el contrario se crea un formulario para modificar datos, lo lógico es que se muestran inicialmente los datos guardados en el sistema. En este caso, el atributo `value` incluirá el valor que se desea mostrar: `<input type="text" name="nombre" value="Juan Pérez" />`

Si no se especifica un tamaño, el navegador muestra el cuadro de texto con un tamaño predeterminado. El atributo `size` permite establecer el tamaño, en caracteres, con el que se muestra el cuadro de texto. Su uso es imprescindible en muchos formularios, en los que algunos campos como la dirección deben mostrar más caracteres de lo normal (`<input size="100" ...>`) y otros campos como el código postal deben mostrar menos caracteres de lo normal (`<input size="5" ...>`).

Además de controlar el tamaño con el que se muestra un cuadro de texto, también se puede limitar el tamaño del texto introducido. El atributo `maxlength` permite establecer el máximo número de caracteres que el usuario puede introducir en un cuadro de texto. Su uso es imprescindible para campos como el código postal, el número de la Seguridad Social y cualquier otro dato con formato predefinido y limitado.

Por último, el atributo `readonly` permite que el usuario pueda ver los contenidos del cuadro de texto pero no pueda modificarlos y el atributo `disabled` deshabilita un cuadro de texto de forma que el usuario no pueda modificarlo y además, el navegador no envía sus datos al servidor.

## 8.2.2. Cuadro de contraseña

La única diferencia entre este control y el cuadro de texto normal es que el texto que el usuario escribe en un cuadro de contraseña no se ve en la pantalla. En su lugar, los navegadores ocultan el texto utilizando asteriscos o círculos, por lo que es ideal para escribir contraseñas y otros datos sensibles.

Contraseña `<br/>`

```
<input type="password" name="contrasena" value="" />
```

Cambiando el valor del atributo `type` por `password` se transforma el cuadro de texto normal en un cuadro de contraseña. Todos los demás atributos se utilizan de la misma forma y tienen el mismo significado.

## 8.2.3. Checkbox

Los checkbox o "*casillas de verificación*" son controles de formulario que permiten al usuario seleccionar opciones individualmente. Aunque en ocasiones se muestran varios checkbox juntos, cada uno de ellos es completamente independiente del resto. Por este motivo, se utilizan cuando el usuario puede activar y desactivar varias opciones relacionadas pero no excluyentes.

El valor del atributo `type` para estos controles de formulario es `checkbox`. Como se muestra en el ejemplo anterior, el texto que se encuentra al lado de cada *checkbox* no se puede establecer mediante ningún atributo, por lo que es necesario añadirlo manualmente fuera del control del formulario. Si no se añade un texto al lado de la etiqueta `<input />` del *checkbox*, el usuario sólo ve un pequeño cuadrado sin ninguna información relativa a la finalidad de ese *checkbox*.

El valor del atributo `value`, junto con el valor del atributo `name`, es la información que llega al servidor cuando el usuario envía el formulario.

Si se quiere mostrar un *checkbox* seleccionado por defecto, se utiliza el atributo `checked`. Si el valor del atributo es `checked`, el *checkbox* se muestra seleccionado. En cualquier otro caso, el *checkbox* permanece sin seleccionar. Aunque resulta redundante que el nombre y el valor del atributo sean idénticos, es obligatorio indicarlo de esta forma porque los atributos en XHTML no pueden tener valores vacíos

## 8.2.4. Radiobutton

Los controles de tipo `radiobutton` son similares a los controles de tipo `checkbox`, pero presentan una diferencia muy importante: son mutuamente excluyentes. Los `radiobutton` se utilizan cuando el usuario solamente puede escoger una opción entre las distintas opciones relacionadas que se le presentan. Cada vez que se selecciona una opción, automáticamente se deselecta la otra opción que estaba seleccionada.

El valor del atributo `type` para estos controles de formulario es `radio`. El atributo `name` se emplea para indicar los `radiobutton` que están relacionados. Por lo tanto, cuando varios `radiobutton` tienen el mismo valor en su atributo `name`, el navegador sabe que están relacionados y puede deselectar una opción del grupo de `radiobutton` cuando se seleccione otra opción.

## 8.2.5. Botón de envío de formulario

La mayoría de formularios dispone de un botón para enviar al servidor los datos introducidos por el usuario:

```
<input type="submit" name="buscar" value="Buscar" />
```

El valor del atributo `type` para este control de formulario es `submit`. El navegador se encarga de enviar automáticamente los datos cuando el usuario pincha sobre este tipo de botón. El valor del atributo `value` es el texto que muestra el botón. Si no se establece el atributo `value`, el navegador muestra el texto predefinido `Enviar consulta`.

## 8.2.6. Botón de reseteo del formulario

Se trata de un botón especial que borra todos los datos introducidos por el usuario y devuelve el formulario a su estado original:

```
<input type="reset" name="limpiar" value="Borrar datos del formulario" />
```

El valor del atributo `type` para este control de formulario es `reset`. Cuando el usuario pulsa este botón, el navegador borra toda la información introducida y muestra el formulario en su estado original. Si el formulario no contenía originalmente ningún valor, el botón de `reset` lo vuelve a mostrar vacío. Si el formulario contenía información, el botón `reset` vuelve a mostrar la misma información original.

Como es habitual en los botones de formulario, el atributo `value` permite establecer el texto que muestra el botón. Si no se utiliza este atributo, el navegador muestra el texto predefinido del botón, que en este caso es `Restablecer`.

## 8.2.7. Ficheros adjuntos

Los formularios también permiten adjuntar archivos para subirlos al servidor. Aunque desde el punto de vista de HTML y del navegador no existe ninguna limitación sobre el número, tipo o tamaño total de los archivos que se pueden adjuntar, todos los servidores añaden restricciones por motivos de seguridad.

Fichero adjunto

```
<input type="file" name="adjunto" />
```

El valor del atributo `type` para este control de formulario es `file`. El navegador se encarga de mostrar un cuadro de texto donde aparece el nombre del archivo seleccionado y un botón que permite navegar por los directorios y archivos del ordenador del usuario.

Si se incluye un control para adjuntar archivos, es obligatorio añadir el atributo `enctype` en la etiqueta `<form>` del formulario. El valor del atributo `enctype` debe ser `multipart/form-data`, por lo que la etiqueta `<form>` de los formularios que permiten adjuntar archivos siempre es:

```
<form action="..." method="post" enctype="multipart/form-data">
```

```
</form>
```

## 8.2.8. Campos ocultos

Los campos ocultos se emplean para añadir información oculta en el formulario:

```
<input type="hidden" name="url_previa" value="/articulo/primer.html" />
```

El valor del atributo `type` para este control de formulario es `hidden`. Los campos ocultos no se muestran por pantalla, de forma que el usuario desconoce que el formulario los incluye. Normalmente los campos ocultos se utilizan para incluir información que necesita el servidor pero que no es necesario o no es posible que la establezca el usuario.

## 8.2.9. Botón de imagen

El aspecto de los botones de formulario se puede personalizar por completo, ya que incluso es posible utilizar una imagen como botón:

```
<input type="image" name="enviar" src="accept.png" />
```

El valor del atributo `type` para este control de formulario es `image`. El atributo `src` indica la URL de la imagen que debe mostrar el navegador en lugar del botón normal.

Su principal ventaja es que permite personalizar por completo la estética de los botones y mostrarlos con un aspecto homogéneo en todos los navegadores. El principal inconveniente es que ralentiza la carga del formulario y que si se quiere modificar su aspecto, es necesario crear una nueva imagen.

## 8.2.10. Botón

Algunos formularios complejos necesitan botones más avanzados que los de enviar datos (`type="submit"`) y resetear el formulario (`type="reset"`). Por ese motivo, el estándar HTML/XHTML define un botón de tipo genérico:

```
<input type="button" name="guardar" value="Guardar Cambios" />
```

El valor del atributo `type` para este control de formulario es `button`. Si pruebas a pulsar un botón de este tipo, verás que el navegador no hace nada: no envía los datos al servidor y no borra los datos introducidos. Este tipo de botones sólo son útiles si se utilizan junto con el lenguaje de programación JavaScript. Si la página incluye código JavaScript, los botones de este tipo se pueden programar para que realicen cualquier tarea compleja cuando se pulsa sobre ellos.

## 8.3. Formularios avanzados

Utilizando solamente las etiquetas `<form>` y `<input>` es posible diseñar la mayoría de formularios de las aplicaciones web. No obstante, HTML define algunos elementos adicionales para mejorar la estructura de los formularios creados.

La siguiente imagen muestra un formulario que agrupa sus elementos y añade etiquetas a cada campo para mejorar su estructura:

La etiqueta `<fieldset>` agrupa campos del formulario y la etiqueta `<legend>` asigna un nombre a cada grupo.

<b>Etiqueta</b>	<code>&lt;fieldset&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	-
<b>Tipo de elemento</b>	Bloque
<b>Descripción</b>	Se emplea para agrupar de forma lógica varios campos de un formulario

<b>Etiqueta</b>	<code>&lt;legend&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	-
<b>Tipo de elemento</b>	En línea
<b>Descripción</b>	Se emplea para definir el título o leyenda de un conjunto de campos de formulario agrupados con la etiqueta <code>fieldset</code>

A continuación se muestra el código HTML del formulario correspondiente a la imagen anterior y que hace uso de `<fieldset>` y `<legend>` para agrupar los campos del formulario

La etiqueta `<fieldset>` agrupa todos los controles de formulario a los que encierra. El navegador muestra por defecto un borde resaltado para cada agrupación. La etiqueta `<legend>` se incluye



dentro de cada etiqueta `<fieldset>` y establece el título que muestra el navegador para cada agrupación de elementos.

Por otra parte, todos los controles de formulario salvo los botones presentan una carencia muy importante: no disponen de la opción de establecer el título o texto que se muestra junto al control. En el código HTML del ejemplo anterior, el nombre de cada campo se incluye en forma de texto normal, sin ninguna relación con el campo al que hace referencia.

Afortunadamente, el lenguaje HTML incluye una etiqueta denominada `<label>` y que se utiliza para establecer el título de cada campo del formulario. Su definición formal es la siguiente:

<b>Etiqueta</b>	<code>&lt;label&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	<ul style="list-style-type: none"><li>● <code>for = "id_de_elemento"</code> - Indica el ID del campo del formulario para el que este elemento es su título</li><li>● Otros: <code>accesskey</code>, <code>onfocus</code> y <code>onblur</code></li></ul>
<b>Tipo de elemento</b>	En línea
<b>Descripción</b>	Se emplea para definir el título o leyenda de los campos definidos en un formulario

El único atributo que suele utilizarse con la etiqueta `<label>` es `for`, que indica el identificador (atributo `id`) del campo de formulario para el que esta etiqueta hace de título.

En el anterior ejemplo, el nombre de los campos de formulario se incluía mediante un texto normal:

Nombre `<br/>`

```
<input type="text" name="nombre" value="" />
```

Apellidos `<br/>`

```
<input type="text" name="apellidos" value="" />
```

```
DNI <br/>
```

```
<input type="text" name="dni" value="" size="10" maxlength="9" />
```

Utilizando la etiqueta `<label>`, cada campo de formulario puede disponer de su propio título:

```
<label for="nombre">Nombre</label> <br/>
```

```
<input type="text" id="nombre" name="nombre" value="" />
```

```
<label for="apellidos">Apellidos</label> <br/>
```

```
<input type="text" id="apellidos" name="apellidos" value="" />
```

```
<label for="dni">DNI</label> <br/>
```

```
<input type="text" id="dni" name="dni" value="" size="10" maxlength="9" />
```

La principal ventaja de utilizar `<label>` es que el código HTML está mejor estructurado y se mejora su accesibilidad. Además, al pinchar sobre el texto del `<label>`, el puntero del ratón se posiciona automáticamente para poder escribir sobre el campo de formulario asociado. Este comportamiento es especialmente útil para los campos de tipo `radiobutton` y `checkbox`.

## 8.4. Otros elementos de formulario

La etiqueta `<input>` permite crear diez tipos diferentes de controles de formulario. Sin embargo, algunas aplicaciones web utilizan otros elementos de formulario que no se pueden crear con `<input>`. Las listas desplegables y las áreas de texto disponen de sus propias etiquetas (`<select>` y `<textarea>` respectivamente).

Las áreas de texto son útiles cuando se debe introducir una gran cantidad de texto, ya que es mucho más cómodo de introducir que en un campo de texto normal

La definición formal de la etiqueta `<textarea>` es:

Etiqueta	<textarea>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> , <a href="#">eventos</a> y <a href="#">foco</a>
Atributos propios	<ul style="list-style-type: none"> <li>● <code>rows = "numero"</code> - Número de filas de texto que mostrará el textarea</li> <li>● <code>cols = "numero"</code> - Número de caracteres que se muestran en cada fila del textarea</li> <li>● Otros: name, disabled, readonly, onselect, onchange, onfocus, onblur</li> </ul>
Tipo de elemento	En línea
Descripción	Se emplea para incluir un área de texto en un formulario

Los atributos más utilizados en las etiquetas `<textarea>` son los que controlan su anchura y altura. La anchura del área de texto se controla mediante el atributo `cols`, que indica las *columnas* o número de caracteres que se podrán escribir como máximo en cada fila. La altura del área de texto se controla mediante `rows`, que indica directamente las filas de texto que serán visibles.

El principal inconveniente de los elementos `<textarea>` es que el lenguaje HTML no permite limitar el número máximo de caracteres que se pueden introducir. Mientras los elementos `<input type="text">` disponen del atributo `maxlength`, las áreas de texto no disponen de un atributo equivalente, por lo que sólo es posible limitar el número de caracteres mediante su programación con JavaScript.

Los tres tipos de listas desplegables se definen con la misma etiqueta `<select>` y cada elemento de la lista se define mediante la etiqueta `<option>`:

Etiqueta	<select>
----------	----------

Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	<ul style="list-style-type: none"><li>● <code>size = "numero"</code> - Número de filas que se muestran de la lista (por defecto sólo se muestra una)</li><li>● <code>multiple = "multiple"</code> - Si se incluye, se permite seleccionar más de un elemento</li><li>● Otros: <code>name</code>, <code>disabled</code>, <code>onchange</code>, <code>onfocus</code>, <code>onblur</code></li></ul>
Tipo de elemento	En línea
Descripción	Se emplea para incluir una lista desplegable en un formulario

Etiqueta	<code>&lt;option&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	<ul style="list-style-type: none"><li>● <code>selected = "selected"</code> - Indica si el elemento aparece seleccionado por defecto al cargarse la página</li><li>● <code>value = "texto"</code> - El valor que se envía al servidor cuando el usuario elige esa opción</li><li>● Otros: <code>label</code>, <code>disabled</code></li></ul>

Tipo de elemento	-
Descripción	Se emplea para definir cada elemento de una lista desplegable

La inmensa mayoría de listas desplegables que utilizan las aplicaciones web son simples, por lo que el código HTML habitual de las listas desplegables es:

```
<label for="so">Sistema operativo</label> <br/>
<select id="so" name="so">
  <option value="" selected="selected">- selecciona -</option>
  <option value="windows">Windows</option>
  <option value="mac">Mac</option>
  <option value="linux">Linux</option>
  <option value="otro">Otro</option>
</select>
```

La etiqueta `<select>` define la lista y encierra todas las opciones que muestra la lista. Cada una de las opciones de la lista se define mediante una etiqueta `<option>`. El atributo `value` de cada opción es obligatorio, ya que es el dato que se envía al servidor cuando el usuario envía el formulario. Para seleccionar por defecto una opción al mostrar la lista, se añade el atributo `selected` a la opción deseada.

Por otra parte, las listas desplegables permiten agrupar sus opciones de forma que el usuario pueda encontrar fácilmente las opciones cuando la lista es muy larga

La etiqueta `<optgroup>` permite agrupar opciones relacionadas dentro de una lista desplegable. Su definición formal se muestra a continuación:

Etiqueta	<code>&lt;optgroup&gt;</code>
----------	-------------------------------

<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
<b>Atributos propios</b>	<ul style="list-style-type: none"> <li>● <code>label = "texto"</code> - Texto que se muestra como título de la agrupación de opciones</li> <li>● Otros: disabled, selected</li> </ul>
<b>Tipo de elemento</b>	-
<b>Descripción</b>	Se emplea para definir una agrupación lógica de opciones de una lista desplegable

El único atributo que suele utilizarse con la etiqueta `<optgroup>` es `label`, que indica el nombre de cada agrupación. Los navegadores muestran de forma destacada el título de cada agrupación, de forma que el usuario pueda localizar más fácilmente la opción deseada.

## 9. Estructura y layout

Los capítulos anteriores muestran las decenas de etiquetas XHTML disponibles para marcar y estructurar cada elemento individual de las páginas web: tablas, listas, enlaces, párrafos, imágenes, etc. Aunque combinando esas etiquetas es posible crear cualquier página web, no es posible hacer que las páginas muestren estructuras complejas.

La mayoría de páginas HTML disponen de estructuras complejas formadas por varias columnas de contenidos y otro tipo de divisiones. Utilizando exclusivamente HTML no es posible crear estas estructuras complejas, ya que es imprescindible emplear las hojas de estilos CSS.

No obstante, los estilos de CSS necesitan la ayuda de HTML/XHTML para crear los diseños más avanzados. En concreto, el código HTML se encarga de agrupar los elementos de la página en diferentes divisiones en función de su finalidad: la zona de la cabecera de la página, la zona de contenidos, una zona lateral para el menú y otras secciones menores, la zona del pie de página, etc.

Para agrupar los elementos que forman cada zona o división de la página se utiliza la etiqueta `<div>`:

Etiqueta	<code>&lt;div&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	-
Tipo de elemento	Bloque
Descripción	Agrupar elementos de bloque

El nombre de la etiqueta `div` tiene su origen en la palabra *división*, ya que esta etiqueta define zonas o divisiones dentro de una página HTML. En cualquier caso, casi todos los diseñadores web utilizan la palabra "capa" para referirse a una "división". Aunque se trata de un error grave (las capas se crean mediante una propiedad de CSS llamada `z-index`) es preferible seguir llamando "capas" a las zonas definidas con la etiqueta `<div>` para poder entenderse con el resto de diseñadores.

Las páginas web complejas que están bien diseñadas utilizan decenas de etiquetas `<div>`. Con mucha diferencia, los atributos más utilizados con esta etiqueta son `id` (para identificar la capa de forma única) y `class` (para aplicar a la capa estilos CSS).

No se va a profundizar en el proceso de diseñar una página web mediante `<div>`, ya que no es posible diseñar una página web compleja utilizando elementos `<div>` sin utilizar hojas de estilos CSS.

Por último, si observas el código HTML de algunas páginas web complejas, verás que la mayoría utilizan los mismos nombres para identificar sus divisiones. Los nombres más comunes, y sus equivalentes en inglés, se muestran a continuación:

- `contenedor` (`wrapper`) suele encerrar la mayor parte de los contenidos de la página y se emplea para definir las características básicas de la página: su anchura, sus bordes, imágenes laterales, si se centra o no respecto de la ventana del navegador, etc.
- `cabecera` (`header`) que incluye todos los elementos invariantes de la parte superior de la página (logotipo, imagen o banner, cuadro de búsqueda superior, etc.)
- `contenido` (`content`) engloba el contenido principal del sitio (la zona de noticias, la zona de artículos, la zona de productos, etc. dependiendo del tipo de sitio web)
- `menu` (`menu`) se emplea para agrupar todos los elementos del menú lateral de navegación de la página

- `pie` (`footer`) que incluye todos los elementos invariantes de la parte inferior de la página (aviso de copyright, política de privacidad, términos de uso, etc.)
- `lateral` (`sidebar`) se emplea para agrupar los elementos de las columnas laterales y secundarias de la página.

## 10. Metainformación

Las páginas y documentos HTML incluyen más información de la que los usuarios ven en sus pantallas. Estos datos adicionales siempre están relacionados con la propia página, por lo que se denominan *metainformación* o *metadatos*. La metainformación siempre se incluye en la sección de la cabecera, es decir, dentro de la etiqueta `<head>`.

Aunque la metainformación más conocida y utilizada es el título de la propia página, se puede incluir mucha otra información útil para los navegadores y para los buscadores. En las próximas secciones se explica cómo incluir la metainformación y se introduce un concepto relacionado llamado `DOCTYPE`.

### 10.1. Estructura de la cabecera

Como ya se explicó anteriormente, las páginas XHTML se dividen en dos partes denominadas cabecera y cuerpo. La sección de la cabecera está formada por todas las etiquetas encerradas por la etiqueta `<head>`:

<b>Etiqueta</b>	<code>&lt;head&gt;</code>
<b>Atributos comunes</b>	<a href="#">internacionalización</a>
<b>Atributos propios</b>	<ul style="list-style-type: none"><li>• <code>profile = "url"</code> - Especifica la URL del perfil o perfiles que utilizan los metadatos</li><li>• <code>lang = "codigo_de_idioma"</code> - Especifica el idioma principal de los contenidos de la página</li></ul>
<b>Tipo de elemento</b>	-



Descripción	Define la cabecera del documento HTML
-------------	---------------------------------------

La cabecera típica de una página HTML completa presenta la siguiente estructura:

```
<head>

  <!-- Zona de etiquetas META -->

  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />


  <!-- Zona de título -->

  <title>El título del documento</title>


  <!-- Zona de recursos enlazados (CSS, RSS, JavaScript) -->

  <link rel="stylesheet" href="#" type="text/css" media="screen" />

  <link rel="stylesheet" href="#" type="text/css" media="print" />


  <link rel="alternate" type="application/rss+xml" title="RSS 2.0" href="#" />


  <script src="#" type="text/javascript"></script>

</head>
```

La etiqueta `<title>` establece el título de la página. Los navegadores muestran este título como título de la propia ventana del navegador. Los buscadores utilizan este título como título de sus resultados de búsqueda.

Por tanto, el valor de `<title>` no sólo es importante para los usuarios, sino que también es importante para que los usuarios encuentren las páginas a través de los buscadores. Un error común de muchos sitios web consiste en mostrar un mismo título genérico en todas sus páginas. Cada página debe mostrar un título corto, adecuado, único y que describa inequívocamente los contenidos de la página.

Las páginas XHTML deben tener definido un título y sólo uno, por lo que todas las páginas web deben incluir obligatoriamente una etiqueta `<title>`, cuya definición formal se muestra a continuación:

<b>Etiqueta</b>	<code>&lt;title&gt;</code>
<b>Atributos comunes</b>	<a href="#">internacionalización</a>
<b>Atributos propios</b>	<ul style="list-style-type: none"><li>● <code>lang = "codigo_de_idioma"</code> - Especifica el idioma principal del título de la página</li></ul>
<b>Tipo de elemento</b>	-
<b>Descripción</b>	Define el título del documento HTML

Por último, la etiqueta `<head>` permite definir en el atributo `profile` la URL de un documento externo que contiene el perfil que siguen los metadatos de la cabecera. Los blogs creados con el programa WordPress incluyen por ejemplo el siguiente perfil en su cabecera:

```
<head profile="http://gmpg.org/xfn/11">
```

```
...
```

```
</head>
```

El documento `http://gmpg.org/xfn/11` es un perfil que define atributos adicionales para establecer la relación entre sitios web.

## 10.2. Metadatos

Una de las partes más importantes de la metainformación de la página son los metadatos, que permiten incluir cualquier información relevante sobre la propia página.

La especificación oficial de HTML no define la lista de metadatos que se pueden incluir, por lo que las páginas tienen libertad absoluta para definir los metadatos que consideren adecuados. La etiqueta empleada para la definición de los metadatos es `<meta>`.

Etiqueta	<meta>
Atributos comunes	<a href="#">internacionalización</a>
Atributos propios	<ul style="list-style-type: none"><li>● <code>name = "texto"</code> - El nombre de la propiedad que se define (no existe una lista oficial de propiedades)</li><li>● <code>content = "texto"</code> - El valor de la propiedad definida (no existe una lista de valores permitidos)</li><li>● <code>http-equiv = "texto"</code> - En ocasiones, reemplaza al atributo "name" y lo emplean los servidores para adaptar sus respuestas al documento</li><li>● <code>scheme = "texto"</code> - Indica el esquema que se debe emplear para interpretar el valor de la propiedad</li></ul>
Tipo de elemento	-
Descripción	Permite definir el valor de los metadatos que forman la metainformación del documento

Los metadatos habituales utilizan solamente los atributos `name` y `content` para definir el nombre y el valor del metadato:

```
<meta name="autor" content="Juan Pérez" />
```

No obstante, algunas etiquetas `<meta>` muy utilizadas hacen uso del atributo `http-equiv`. Este atributo se utiliza para indicar que el valor establecido por este metadato puede ser utilizado por el servidor al entregar la página al navegador del usuario. El siguiente metadato indica al servidor que el contenido de la página es código HTML y su codificación de caracteres es UTF-8:

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

El atributo `scheme` no suele utilizarse, aunque permite proporcionar información de contexto para que el navegador interprete correctamente el valor del metadato. En el siguiente ejemplo, el atributo `scheme` indica al navegador que el valor del metadato hace referencia al código ISBN:

```
<meta scheme="ISBN" name="identificador" content="789-1392349610">
```

Aunque no existe una lista oficial con los metadatos que se pueden definir, algunos de ellos se utilizan en tantas páginas que se han convertido prácticamente en un estándar. A continuación se muestran los metadatos más utilizados:

Definir el autor del documento:

```
<meta name="author" content="Juan Pérez" />
```

Definir el programa con el que se ha creado el documento:

```
<meta name="generator" content="WordPress 2.8.4" />
```

Definir la codificación de caracteres del documento:

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
```

Definir el copyright del documento:

```
<meta name="copyright" content="librosweb.es" />
```

Definir el comportamiento de los buscadores:

```
<meta name="robots" content="index, follow" />
```

Definir las palabras clave que definen el contenido del documento:

```
<meta name="keywords" content="diseño, css, hojas de estilos, web, html" />
```

Definir una breve descripción del sitio:

```
<meta name="description" content="Artículos sobre diseño web, usabilidad y  
accesibilidad" />
```

La etiqueta que define la codificación de los caracteres (`http-equiv="Content-Type"`) se emplea prácticamente en todas las páginas y las etiquetas que definen la descripción (`description`) y las palabras clave (`keywords`) también son muy utilizadas.

## 11.1. Comentarios

Normalmente, los diseñadores y programadores incluyen comentarios para marcar el comienzo y el final de las secciones de las páginas, para incluir avisos y notas para otros diseñadores o para incluir explicaciones sobre la forma en la que se ha creado el código HTML.

Aunque los comentarios no se muestran por pantalla y por tanto son invisibles para los usuarios, sí que se descargan con el código HTML de la página. Por este motivo, nunca debe incluirse información sensible o confidencial en los comentarios.

La sintaxis de los comentarios es la siguiente:

Apertura del comentario: `<!--`

Contenido del comentario: (cualquier texto)

Cierre del comentario: `-->`

El siguiente ejemplo muestra el uso de los comentarios HTML para indicar el comienzo y final de cada sección. Recuerda que los comentarios no se muestran por pantalla y que no influyen en la forma en la que se ven las páginas:

```
<!-- Inicio del menú -->  
<div id="menu">  
<ul>  
  <li>...</li>  
  <li>...</li>  
  <li>...</li>  
  <li>...</li>  
</ul>  
<!-- Fin del menú -->
```

```
<!-- Inicio de la publicidad -->
<div id="publicidad"> ... </div>
<!-- Fin de la publicidad -->
```

Los comentarios de HTML pueden ocupar tantas líneas como sea necesario. Sin embargo, los comentarios no se pueden anidar, es decir, no se puede incluir un comentario dentro de otro comentario

## EXTRAS :

**<nav>** → El elemento de sección de navegación

El elemento HTML `<nav>` representa una sección de una página cuyo propósito es proporcionar enlaces de navegación, ya sea dentro del documento actual o a otros documentos. Ejemplos comunes de secciones de navegación son menús, tablas de contenido e índices.

**<footer>** → footer

Contexto de uso

El *Elemento HTML Footer* (`<footer>`) representa un pie de página. Un pie de página típicamente contiene información acerca del autor de la sección, datos de derechos de autor o enlaces a documentos relacionados.

**<input type="date">** →

Los elementos de `<input>` del tipo `date` crean campos de entrada que permiten a los usuarios proporcionar una fecha, bien a través de un cuadro de texto que valida automáticamente el contenido, o bien mediante una interfaz especial de selección de fechas. El valor resultante incluye el año, el mes y el día, pero no la hora. Los tipos de entrada `time` y `datetime-local` admiten entradas de hora y de hora y fecha.

Es posible establecer un valor predeterminado para la entrada al incluir una fecha en el atributo `value`, de esta manera:

```
<input id="date" type="date" value="2017-06-01">
```

## Atributos adicionales

Tenemos atributos adicionales tales como:

Attribute	Description
<a href="#">max</a>	La fecha máxima aceptada.
<a href="#">min</a>	La fecha minima aceptada.
<a href="#">step</a>	El intervalo a usar, como cuando das click en las flechas, de cuanto en cuanto va avanzar o retroceder.

## max

La última fecha a aceptar. Si el [value](#) ingresado en el elemento está después de esta fecha, el elemento cae [constraint validation](#). Si el valor del max no es un valor válido, sigue el formato yyyy-MM-dd, donde el elemento no tiene máximo valor.

Este valor debe ser especificado mayor o igual especificado por min attribute.

## min

La mínima fecha aceptada; Una fecha menor que esta caerá [constraint validation](#). Si el valor es min el atributo no es válido y esa el formato yyyy-MM-dd, entonces el elemento no tiene valor mínimo.

Este valor debe especificarse como menor o igual máx attribute

**<caracteres>** →

Entidad	Carácter	Descripción oficial
<code>&amp;ntilde</code> <code>e;</code>	ñ	latin letter n with tilde
<code>&amp;Ntilde</code> <code>e;</code>	Ñ	latin capital n letter with tilde
<code>&amp;aacute</code> <code>e;</code>	á	a acute
<code>&amp;eacute</code> <code>e;</code>	é	e acute
<code>&amp;iacute</code> <code>e;</code>	í	i acute
<code>&amp;oacute</code> <code>e;</code>	ó	o acute
<code>&amp;uacute</code> <code>e;</code>	ú	u acute
<code>&amp;Aacute</code> <code>e;</code>	Á	A acute
<code>&amp;Eacute</code> <code>e;</code>	É	E acute



<code>&amp;Iacute e;</code>	Í	I acute
<code>&amp;Oacute e;</code>	Ó	O acute
<code>&amp;Uacute e;</code>	Ú	U acute
<code>&amp;euro;</code>	€	euro

## **<header>** →

El *elemento de HTML Header* (`<header>`) representa un grupo de ayudas introductorias o de navegación. Puede contener algunos elementos de encabezado, así como también un logo, un formulario de búsqueda, un nombre de autor y otros componentes.

## **<article>** → article

El *Elemento article de HTML* (`<article>`) representa una composición auto-contenida en un documento, página, una aplicación o en el sitio, que se destina a distribuir de forma independiente o reutilizable, por ejemplo, en la indicación. Podría ser un mensaje en un foro, un artículo de una revista o un periódico, una entrada de blog, un comentario de un usuario, un widget interactivo o gadget, o cualquier otro elemento independiente del contenido.

## **<figcaption>**

El elemento **HTML** `<figcaption>` representa un subtítulo o leyenda asociado al contenido del elemento padre `<figure>`, pudiendo ser colocado como primer o último hijo. Es importante destacar que el elemento `<figcaption>` es opcional. (debajo fotos)

## <thead> →

El elemento HTML <thead> define una sección de encabezado en una tabla ([elemento HTML table](#)). Cuando está presente, debe contener una o más filas que se supone, proveen información de encabezado para las celdas de datos.

## <caption> →

Es el encargado de darle un título descriptivo a las tablas.

## <código> → `&lt;t;html&gt;`

## <mostrar texto tal y como es> → **<pre>**

El Elemento HTML <pre> (o *Texto HTML Preformateado*) representa texto preformateado. El texto en este elemento típicamente se muestra en una fuente fija, no proporciona, exactamente como es mostrado en el archivo. Los espacios dentro de este elemento también son mostrados como están escritos.

<code>&lt;strong&gt;</code>	Fragmento de texto <b>importante</b> o <b>palabras clave</b> .
<code>&lt;em&gt;</code>	Fragmento de texto <i>enfaticado</i> respecto a la frase que lo contiene.

<code>&lt;mark&gt;</code>	Fragmento de texto resaltado, simulando estar marcado con rotulador amarillo.
<code>&lt;i&gt;</code>	Fragmento de texto con voz o tono alternativo al resto.
<code>&lt;b&gt;</code>	Fragmento de texto sin importancia destacable ( <i>finés utilitarios</i> ).
<code>&lt;u&gt;</code>	Fragmento de texto para nombres propios o escritura incorrecta intencionada ( <u>sic</u> ).
<code>&lt;s&gt;</code>	Fragmento de texto inexacto o que ya no es relevante. ( <del>errores o inexactitudes</del> )
<code>&lt;span&gt;</code>	Fragmento de texto sin significado (útil para seleccionar).
<code>&lt;cite&gt;</code>	Fragmento de texto con el título de un trabajo creativo: obras, libros...

<code>&lt;sup&gt;</code>		Superíndice (24 <sup>2</sup> ).
<code>&lt;sub&gt;</code>		Subíndice (24 <sub>2</sub> ).
<code>&lt;small&gt;</code>		Anotaciones menores pequeñas puntualizaciones.
<code>&lt;q&gt;</code>	<code>cite</code>	Cita o frase extraída de otro contexto.
<code>&lt;dfn&gt;</code>	<code>title</code>	Definición (término que posteriormente será definido).
<code>&lt;abbr&gt;</code>	<code>title</code>	Abreviatura o acrónimo.

<code>&lt;kbd&gt;</code>		<b>Entrada de información del usuario (<u>generalmente, combinación de teclado</u>).</b>
--------------------------	--	--

<code>&lt;samp&gt;</code>		Salida de información de un programa informático.
<code>&lt;var&gt;</code>		Variable (contexto matemático o informático).
<code>&lt;time&gt;</code>	<code>datetime</code>	Indica una fecha/hora legible para humanos, con formato para máquinas opcional.
<code>&lt;data&gt;</code>	<code>value</code>	Información equivalente orientado a máquinas.
<code>&lt;code&gt;</code>		Fragmento de código fuente ( <i>en línea</i> ).

<code>&lt;div&gt;</code>		Capa o división utilizado para agrupar varias etiquetas HTML.
<code>&lt;p&gt;</code>		Define un párrafo de texto ( <i>con sus etiquetas HTML para texto</i> ).
<code>&lt;pre&gt;</code>		Establece un texto preformateado ( <i>respetando espacios y saltos de línea</i> ).

# HTML

<code>&lt;blockquote&gt;</code>	<code>cite</code>	Agrupación y características de una cita ( <i>autor, fuente, etc...</i> ).
<code>&lt;main&gt;</code>		Contenedor para englobar la parte principal de la página.
<code>&lt;hr&gt;</code>		Indica una separación temática del texto.

<code>&lt;ul&gt;</code>		Define una lista sin orden. Se trata de la etiqueta contenedora.
<code>&lt;ol&gt;</code>	<code>start, reversed, type</code>	Define una lista numerada (con orden). Etiqueta contenedora.
<code>&lt;li&gt;</code>	<code>value</code>	Define un ítem de la lista.

<code>&lt;dl&gt;</code>		Define una lista de descripciones. Es la etiqueta contenedora.
-------------------------	--	--

<code>&lt;dt&gt;</code>	<b>Término de la descripción. Contiene el nombre o término a describir.</b>
<code>&lt;dd&gt;</code>	<b>Descripción o valor asociado al término. Pueden existir varios por término.</b>

<code>&lt;a&gt;</code>	<b>Etiqueta para hacer referencia a otras páginas web o recursos como documentos o archivos.</b>
------------------------	--

<code>href</code>	<u><b>URL</b></u>	<b>Enlace al documento que se quiere cargar. Atributo obligatorio.</b>
<code>download</code>	<u><b>nombre.ext</b></u>	<b>Descarga el enlace (href) en lugar de abrirlo. Si se indica valor, se renombra.</b>
<code>target</code>	<code>_blank</code>	<b>Abre el enlace en una nueva pestaña.</b>
	<code>_self</code>	<b>Abre el enlace en la pestaña o <code>&lt;iframe&gt;</code> actual.</b>

	<a href="#">_parent</a>	Abre el enlace en el documento padre.
	<a href="#">_top</a>	Abre el enlace en el documento raíz (padre global).
	<u><a href="#">nombre</a></u>	Abre el enlace en el <code>&lt;iframe&gt;</code> con el nombre especificado.
<a href="#">rel</a>	<a href="#">alternate</a>	Indica que el enlace es una versión alternativa (idioma o formato diferente).
	<a href="#">author</a>	Indica que el enlace es la web del autor de la página actual.
	<a href="#">bookmark</a>	El enlace es un <a href="#">permalink</a> a la sección de la página actual.
	<a href="#">help</a>	Indica que el enlace ofrece ayuda para la página actual.
	<a href="#">license</a>	Indica que la página actual está cubierta por la licencia referenciada.
	<a href="#">prev</a>	Indica que el enlace es la parte previa del documento actual.



	<a href="#">next</a>	Indica que el enlace es la parte siguiente del documento actual.
	<a href="#">nofollow</a>	Indica que el enlace no está supervisado por el autor del sitio web.
	<a href="#">noreferrer</a>	El navegador no envía la página de procedencia al visitar el enlace.
	<a href="#">prefetch</a>	El navegador precarga el enlace (es probable que se haga clic)
	<a href="#">search</a>	Indica que el enlace es una página para realizar búsquedas.
	<a href="#">tag</a>	Asocia una etiqueta al enlace especificado, en relación al documento actual.
<a href="#">hreflang</a>	<i><b>Idioma</b></i>	<b>Código ISO 639-1</b> del idioma del documento.
<a href="#">type</a>	<i><b>Tipo</b></i>	<b>Tipo de contenido MIME</b> al que se enlaza. Atributo no obligatorio.

tacos : Entonces al ser una palabra en un párrafo se haría con un span y a está añadirle un class o un id, dependiendo de lo que te pida no?

<code>&lt;iframe&gt;</code>	<code>src, srcdoc, name, width, height</code>	Permite incrustar contenido externo en «vivo».
-----------------------------	---	--

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/ZK9iP5PEhSU"
frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture"
allowfullscreen></iframe>
```

Al detallar este código se puede observar que se encuentra dentro de las etiquetas `<iframe>` y `</iframe>` las cuales son utilizadas para incorporar contenido de otra página HTML en la página actual.

Asimismo, presenta los siguientes atributos los cuales pueden ser modificados de acuerdo a las necesidades:

- `src`: indica la dirección del video que se quiere incluir a la página.
- `width` y `height`: especifican el tamaño de la ventana de video.
- `frameborder`: sirve para indicar si se quiere que la ventana tenga borde o no.
- `allowfullscreen`: permite elegir si se quiere ver en pantalla completa.
- `allow`: especifica una política de características para el `<iframe>`.

Ya con el código dentro del archivo HTML, es posible pasarle parámetros a la URL del video (en caso de necesitarlo). Para ello, hay que agregarlos a la URL del video, justo después del ID del video (el conjunto de caracteres que se muestra después de "embed/").

Parámetros que se le pueden pasar a los vídeos de YouTube

- **autoplay.** - Se utiliza para que el video se reproduzca automáticamente. Si el valor se fija en uno "1" se reproducirá automáticamente, mientras que si se fija en cero (0), no lo hará (valor por defecto).
- **start.** - Se utiliza para que el video empiece a reproducirse en un punto específico. Para ello hay que fijar el tiempo (en segundos) en el cual se desea que comience a reproducirse.
- **controls.** - Permite especificar la conducta de los controles del video. Si se fija su valor en cero (0) los controles no se mostrarán; si se fija en uno (1), se mostrarán los controles (valor por defecto).

### Ejemplo

Si se desea que el video comience en el minuto 2:00, el código de inserción debe ser de la siguiente manera:

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/ZK9iP5PEhSU?start=120"
frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture"
allowfullscreen></iframe>
```

En HTML5 se introduce la interesante posibilidad de **mostrar videos directamente** desde nuestro navegador. De hecho, si arrastramos un video a la ventana del navegador, veremos que comienza a reproducirse en él. Para poder insertar videos en nuestras páginas HTML tenemos que utilizar la etiqueta `<video>`, que junto a la etiqueta `<source>` podremos utilizar estas capacidades multimedia de HTML5.

La etiqueta `<video>` tiene varios atributos a nuestra disposición:

src	<u>Dirección URL</u>	Video a reproducir. Obligatoria si actua como etiqueta contenedora.
poster	<u>Dirección URL</u>	Muestra una imagen a modo de presentación.
preload	<b>auto</b>   <b>metadata</b>   <b>none</b>	Indica como realizar la precarga del video.
mediagroup	<u>nombre</u>	Establece un nombre para un grupo de contenidos multimedia.
autoplay		Comienza a reproducir el video automáticamente.
loop		Vuelve a iniciar el video cuando finaliza su reproducción (bucle).

<code>muted</code>		Establece el video sin sonido (silenciado).
<code>controls</code>		Muestra los controles de reproducción. Por defecto no se muestran.
<code>width</code>	<u>tamaño</u>	Indica el tamaño de ancho del video.
<code>height</code>	<u>tamaño</u>	Indica el tamaño de alto del video.

Un primer ejemplo muy básico para colocar un video en nuestra página web:

```
<video src="video.mp4" width="640" height="480"></video>
```

Sin embargo, esto mostrará el primer fotograma del video, con un tamaño de 640x480, pero se verá como una imagen, ya que no muestra los controles del video y tampoco tiene la autoreproducción activada. Podríamos solucionarlo indicando los atributos **controls** o **autoplay**.

## Videos alternativos

Si utilizamos la etiqueta `<video>` como etiqueta contenedora, podemos incluir etiquetas `<source>` en su interior para proporcionar formatos alternativos y tener mayor compatibilidad con otros navegadores y navegadores antiguos que no soporten HTML5:

```
<video width="640" height="480">

  <source src="video.mp4" type="video/mp4" />

  <source src="video.webm" type="video/webm" />

  <source src="video.ogv" type="video/ogg" />

  

</video>
```

En este ejemplo, los navegadores no mostrarán todos los contenidos a la vez, sino que seguirán el siguiente procedimiento:

- Intenta mostrar el primer formato (MP4). Si el navegador no soporta este formato, salta al siguiente.
- Intenta mostrar el segundo formato (WEBM). Si el navegador no soporta este formato, salta al siguiente.
- Intenta mostrar el tercer formato (OGV). Si el navegador no soporta este formato, salta al siguiente.

- Si se trata de un navegador que no soporta HTML5, intentará mostrar la imagen.
- Si se trata de un navegador de terminal de texto (o *sin capacidades gráficas*), mostrará el texto "Su navegador no soporta contenido multimedia".

De esta forma tenemos soporte completo para todo tipo de dispositivos.

## Acceso a segundos concretos

Utilizando los [fragmentos multimedia](#) se pueden conseguir algunas acciones interesantes, como por ejemplo especificar el momento concreto del video (o *audio*) en el que se quiere empezar a reproducir o terminar de reproducir. Veamos unos ejemplos:

<!-- Ejemplo 1 -->

```
<video autoplay controls src="video.mp4#t=15"></video>
```

<!-- Ejemplo 2 -->

```
<video autoplay controls src="video.mp4#t=25,45"></video>
```

En el primer caso, reproducimos el video a partir del segundo **15** del mismo, mientras que en el segundo caso, reproducimos el video a partir del segundo **25** y terminará de reproducirse en el segundo **45**.