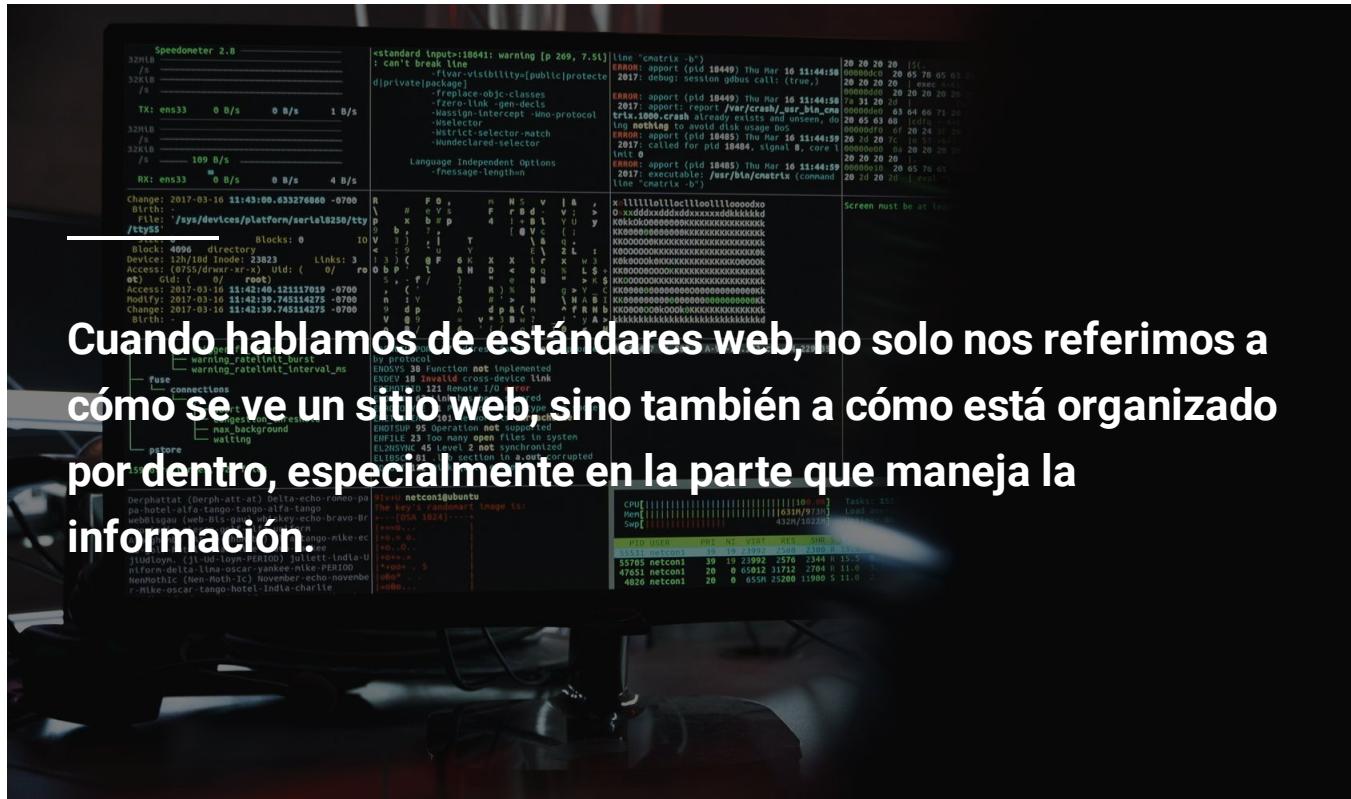


-  Utilización de lenguajes de marcado en entornos web
-  Evolución del HTML
-  Estructura de un documento HTML5
-  Etiquetas y atributos de HTML5
-  Hojas de estilo CSS
-  Autoevaluación

Utilización de lenguajes de marcas en entornos web



Ambos aspectos son muy importantes. Estos estándares web son como reglas que hacen que publicar en Internet sea más ordenado. Antes, crear y compartir cosas en la web era un poco caótico.

Para poner orden, surgieron organizaciones que establecen reglas que deben seguirse al crear y compartir contenido en la web. Una de las más importantes es el W3C (World Wide Web Consortium).

El W3C tiene como objetivo hacer que las tecnologías web más importantes funcionen bien juntas. Esto se llama interoperabilidad web. Básicamente, quieren que cualquier software que uses para acceder a la web funcione sin problemas con todas las tecnologías web.

Así que, gracias a estos estándares y organizaciones, la web se vuelve un lugar más organizado y fácil de entender.

Entre los estándares web más conocidos están:

- HTML, HyperText Markup Language, lenguaje de marcas de hipertexto.
- XML, eXtensible Markup Language, lenguaje de marcas extensible.
- CSS, Cascading Style Sheets, hojas de estilo en cascada.
- JavaScript-.

Evolución del HTML

HTML (HyperText Markup Language) es un lenguaje estándar de marcas ampliamente utilizado para la creación de páginas web, definiendo la estructura del contenido, como son el texto, imágenes y videos.

Es reconocido por todos los navegadores, mostrando de manera consistente el aspecto de la página, independientemente del sistema operativo.

Desde sus primeras versiones hasta HTML5, el desarrollo estaba a cargo del W3C, pero en los años noventa, el W3C inició nuevas recomendaciones bajo el estándar XML llamado XHTML. La versión HTML5, desarrollada en 2004 por el grupo WHATWG, marcó un hito. Aunque HTML5 se convirtió en el estándar líder y dinámico para la web, continúa evolucionando para adaptarse a las

nuevas tecnologías, siendo compatible con dispositivos móviles desde su concepción en 2011, cuando el W3C y WHATWG separaron sus caminos definitivamente.

La siguiente tabla muestra la evolución de las diferentes versiones del HTML:

HTML	Creado por Tim Berners-Lee al comienzo de los años noventa. Fue el primer sistema de hipertexto para compartir documentos.
HTML 2.0	En 1995, el organismo IETF (Internet Engineering Task Force) organiza un grupo de trabajo de HTML y publica el primer estándar oficial HTML 2.0.
HTML 3.2	En 1997 se publica la versión 3.2 que es la primera recomendación de HTML publicada por el W3C. Esta versión incorpora la utilización de los applets de Java.
HTML 4.0	En 1998, W3C publica la versión 4.0, que incorpora la utilización de las hojas de estilo CSS, añade mejoras en la accesibilidad, la opción de incluir guiones ó scripts en las páginas web, optimiza las tablas y los formularios, etcétera.
HTML 4.01	En 1999, el W3C publica la última especificación oficial de HTML, la 4.01,

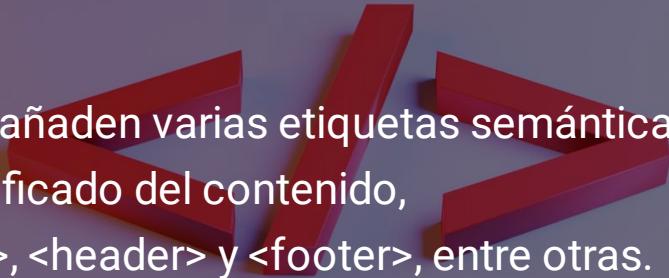
	<p>que es una actualización de la versión HTML 4.0 y no incluye novedades significativas. En este punto, W3C paraliza el proceso de estandarización del HTML.</p>
HTML 5	<p>En 2014, WHATWG publica la versión 5 de HTML.</p>

En cuanto a XHTML, cuya primera versión se publicó en 2000, combinó la sintaxis de HTML 4.01 con XML, solucionando la incompatibilidad de HTML 4.01 con las herramientas basadas en XML. La versión actual, XHTML 2.0, enfrenta problemas, como la falta de nuevas funcionalidades, la ausencia de retrocompatibilidad con las páginas existentes, una sintaxis estricta y demandas elevadas para los navegadores.

A pesar de su evolución, XHTML 2.0 no cumplió con las expectativas, y con la llegada de HTML5, que acepta todo lo permitido en XHTML, pero no viceversa, el W3C abandonó el proyecto, ya que HTML5 se convirtió en el nuevo estándar de la web.



Estructura de un documento HTML5



En HTML5 se añaden varias etiquetas semánticas nuevas que indican el significado del contenido, como `<article>`, `<header>` y `<footer>`, entre otras.

Una página web básica se compone generalmente de tres tipos de código:

- HTML5, aportando los elementos estructurales.
- CSS, aplicando estilo a estos elementos.
- JavaScript, interactuando con esos elementos.

Toda página HTML5 debe tener un **prólogo** y un **ejemplar**, y lo mismo ocurre con cualquier documento XML.

Prólogo

Un documento HTML5 debe tener una declaración de tipo de documento, en la que se indica que es de tipo HTML. Con esta declaración se activa el modo estándar del navegador web. El navegador sabe el tipo de documento que va a mostrar y las incorporaciones de HTML5 son interpretadas siempre que sea posible, o ignoradas si no las soporta.

```
<!DOCTYPE html>
```

Ejemplar

El ejemplar es el propio documento HTML5 que está delimitado por las marcas `<html>` y `</html>`.
El elemento raíz o nodo raíz es siempre `<html>`.

No es obligatorio pero se puede añadir el atributo lang, que define el idioma del contenido del documento que se está creando, en este caso 'es', por español.

```
<html lang="en">  
    ...  
</html>
```

El ejemplar consta de dos secciones: la **cabecera** y el **cuerpo**.

Cabecera

El elemento **head**(cabecera) contiene los metadatos del documento. El contenido de la cabecera no se muestra en la página pero se incluye información sobre la página: título, descripción, autoría, etcétera.

También se pueden incluir enlaces a otros contenidos para completar la presentación del contenido del documento, como una hoja de estilo asociada o scripts. Por ejemplo:

```
<head>  
<meta charset="utf-8" />  
    <title>TEXTO</title>  
    <link rel="stylesheet" type="text/css" href="EstilosBirtLH.css" />  
</head>
```

En la cabecera se pueden incluir los siguientes elementos:

meta

—

Permite incluir metadatos y posicionar la web en los buscadores. También se puede utilizar para indicar la codificación, y de esa forma las palabras con acentos no mostrarán símbolos extraños en el navegador. Por ejemplo:

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

style

—

Permite definir estilos en línea. No es necesaria si se va a utilizar una hoja de estilo externa mediante la etiqueta `<link>`. Por ejemplo:

```
<style type="text/css">  
.....  
</style>
```

title

—

Identifica el documento y se muestra en la pestaña de título del navegador. Por ejemplo:

```
<title>Una página de BirtLH</title>
```

link

—

Para vincular recursos externos como hojas de estilo ó código Javascript, imágenes, etc. Por ejemplo:

```
<link rel="stylesheet" href="EstilosBirtLH.css" />
```

script

—

Permite incluir un script en la página web. Hay dos opciones para hacerlo:

a) el código está en un archivo como por ejemplo:

```
<script src="archivo.js" type="text/javascript"></script>
```

b) el código se inserta en este punto:

```
<script type="text/javascript">Código del script</script>
```

Cuerpo

Contiene información de la página que se mostrará en el navegador. Está delimitado entre las etiquetas `<body>` y `</body>`.

En el siguiente punto se detallan los diferentes elementos que se pueden incluir en el cuerpo de un documento HTML.



En HTML, las etiquetas y los atributos están definidos previamente y en XML los define el programador.



Etiquetas y atributos de HTML5

```
<!DOCTYPE html PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html dir="ltr" xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Hello World</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <meta name="keywords" content="HTML5, CSS3, JavaScript"/>
    <meta name="description" content="A simple Hello World application using HTML5, CSS3, and JavaScript."/>
    <meta name="content-language" content="es"/>
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <link href="style.css" type="text/css" rel="stylesheet"/>
    <script src="script.js" type="text/javascript">

```

Hay páginas que necesitan complementar las etiquetas "type" y "value" con valores predefinidos con atributos para su correcta visualización.

```
</head>
<body>
    <div class="banner">
        <div style="margin:0 auto; width:300px; border:1px solid black; padding:5px; text-align:center; background-color:#f0f0f0; font-size:14px; font-weight:bold; color:#000080; text-decoration:none; text-transform:uppercase; letter-spacing:1px; border-radius:5px; transition:background-color 0.3s ease-in-out; position:relative; z-index:1;">
            <a href="#" style="color:#000080; text-decoration:none; font-size:14px; font-weight:bold; text-transform:uppercase; letter-spacing:1px; position: absolute; top: -15px; left: -15px; background-color:#f0f0f0; width:30px; height:30px; border-radius:50%; display:flex; align-items:center; justify-content:center; transition:background-color 0.3s ease-in-out; z-index:1;">
                <span style="font-size:10px; font-weight:bold; color:#000080; position: absolute; top: 0; left: 0; transform: rotate(-45deg); transform-origin: top left; opacity:0.5; z-index:-1;">X</span>
            </a>
        </div>
        <div style="margin-top:10px; text-align:center; font-size:14px; font-weight:bold; color:#000080; text-decoration:none; text-transform:uppercase; letter-spacing:1px; position:relative; z-index:1;">
            <a href="#" style="color:#000080; text-decoration:none; font-size:14px; font-weight:bold; text-transform:uppercase; letter-spacing:1px; position: absolute; top: -15px; left: -15px; background-color:#f0f0f0; width:30px; height:30px; border-radius:50%; display:flex; align-items:center; justify-content:center; transition:background-color 0.3s ease-in-out; z-index:1;">
                <span style="font-size:10px; font-weight:bold; color:#000080; position: absolute; top: 0; left: 0; transform: rotate(-45deg); transform-origin: top left; opacity:0.5; z-index:-1;">X</span>
            </a>
        </div>
    </div>

```

Atributos

Una etiqueta HTML5 puede tener sus propios atributos, pero hay algunos que son comunes a varias etiquetas. En general, los atributos se pueden agrupar según su funcionalidad en:

Atributos básicos

Se pueden utilizar en la mayoría de etiquetas HTML.

Atributo	Descripción
title = "texto"	Establece el título del elemento HTML que es mostrado por los navegadores cuando el usuario pasa el ratón por encima del elemento.
id = "texto"	Establece un identificador único a un elemento.
style = "texto"	Aplica al elemento HTML el estilo "texto",
class = "texto"	Establece los estilos de la clase CSS al elemento.

Atributos de internacionalización

Se utilizan en páginas que muestran sus contenidos en varios idiomas ó en las que se quiere indicar explícitamente el idioma de sus contenidos.

Atributo	Descripción
lang="codigo"	Indica el idioma del elemento. Por ejemplo: lang="es"

Atributos de eventos

Se utilizan en páginas web que incluyen código JavaScript para realizar acciones dinámicas sobre los elementos de la página.

Para información detallada de los atributos visita el siguiente [enlace](#).

- Atributos de eventos de ventana: son desencadenados para el objeto de ventana (se aplica a la etiqueta <body>)
- Atributos de eventos de formulario: son desencadenados por acciones dentro de un formulario HTML (se aplica a casi todos los elementos HTML, pero se usa más en elementos de formulario).
- Atributos de eventos de teclado: son desencadenados al pulsar el teclado.
- Atributos de eventos del ratón: son desencadenados por las acciones del ratón.
- Atributos de eventos al arrastrar elementos: son desencadenados al arrastrar elementos.
- Atributos de eventos del portapapeles: son desencadenados al copiar ó cortar y pegar elementos.
- Atributos de eventos de medios: son desencadenados por medios como vídeos, imágenes y audio.

Atributos de foco

Se llama foco cuando un control o elemento del documento ha sido seleccionado. Cuando un usuario pincha con su ratón sobre un cuadro de texto y comienza a escribir, ese cuadro de texto tiene el foco del programa, llamado "focus" en inglés.

Atributo	Descripción
accesskey="letra"	Establece una tecla de acceso rápido a un elemento HTML.
tabindex="numero"	Establece la posición del elemento en el orden de tabulación de la página. Su valor debe estar comprendido entre 0 y 32767.

Atributo	Descripción
onfocus	Controlan los eventos JavaScript que se ejecutan cuando el elemento obtiene el foco.
onblur	Controlan los eventos JavaScript que se ejecutan cuando el elemento 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. Aunque la tecla de acceso rápido se establece mediante HTML, la combinación de teclas necesarias para activar ese acceso rápido depende del navegador. En el navegador Internet Explorer se pulsa la tecla ALT + la tecla definida; en el navegador Firefox se pulsa Alt + Shift + la tecla definida; en el navegador Opera se pulsa Shift + Esc + la tecla definida; en el navegador Safari se pulsa Ctrl + la tecla definida.

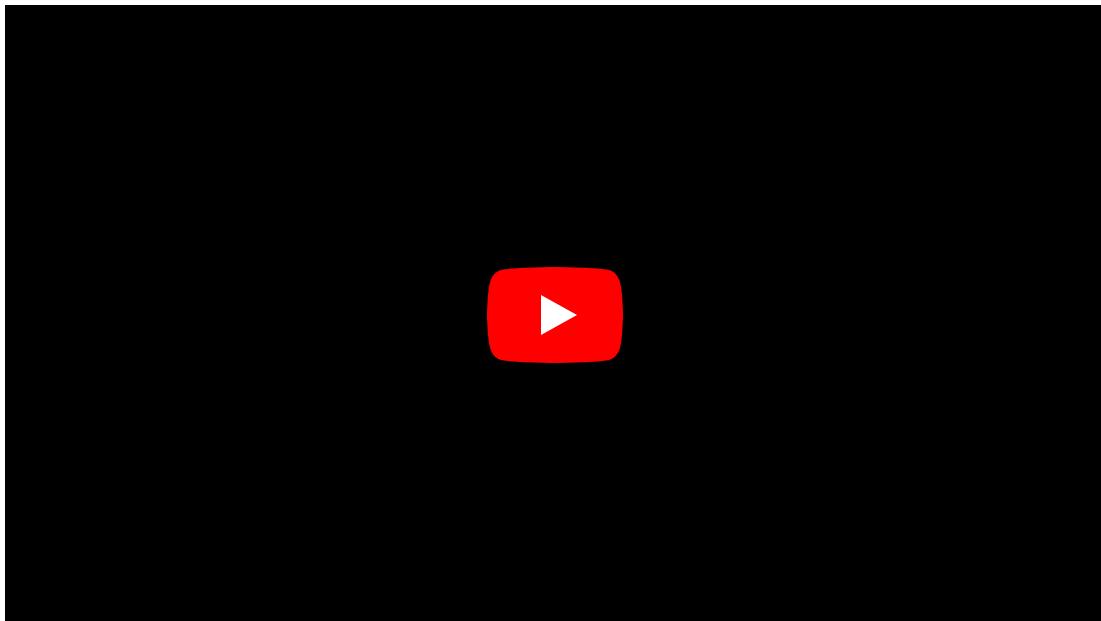
Elementos HTML5

En HTML5 y XML, además de etiquetas y atributos, se utiliza el término **elemento** para referirse a las partes que componen un documento HTML.

En general, un elemento HTML consta de:

- Una etiqueta de apertura.

- Cero o más atributos.
- Texto encerrado por la etiqueta. Es variable, ya que no todas las etiquetas pueden **encerrar texto**.
- Una etiqueta de cierre. No siempre es necesaria.



En HTML existen dos tipos de elementos contenedores:

ELEMENTOS EN LÍNEA

ELEMENTOS EN BLOQUE

En inglés, Inline elements. Ocupan el espacio necesario para mostrar sus contenidos. Cada elemento en línea se colocará a continuación del elemento anterior, sin introducir ningún salto de línea. Su contenido puede ser texto u otros elementos en línea.

Por ejemplo, los enlaces y el texto resaltado son elementos en línea. Es decir, de forma predeterminada, los elementos en línea no comienzan con la nueva línea. Se utiliza `` para aplicar estilo al texto o agrupar elementos en línea.

Son elementos en línea, entre otros: `<a>`, `<abbr>`, `<acronym>`, ``, `<bdo>`, `<big>`, `
`, `<button>`, `<cite>`, `<code>`, `<dfn>`, ``, `<i>`, ``, `<input>`, `<kbd>`, `<label>`, `<map>`, `<object>`, `<output>`, `<q>`, `<samp>`, `<script>`, `<select>`, `<small>`, ``, ``, `<sub>`, `<sup>`, `<textarea>`, `<time>`, `<tt>`, `<var>`.

ELEMENTOS EN LÍNEA

ELEMENTOS EN BLOQUE

En inglés, Block-level elements. Definen bloques de contenido coherente, ya sea texto o imágenes e incorpora un salto de línea antes y después del mismo. Es decir, ocupan todo el ancho y pueden contener elementos en línea, así como otros elementos a nivel de bloque.

Por ejemplo, encabezados, párrafos, listas ó imágenes son elementos en bloque. Con `<div>`, agruparemos otros elementos HTML para diseñar la página web y añadir estilo utilizando las hojas de estilo.

Son elementos en bloque, entre otros: `<address>`, `<article>`, `<aside>`, `<blockquote>`, `<canvas>`, `<dd>`, `<div>`, `<dl>`, `<dt>`, `<fieldset>`, `<figcaption>`, `<figure>`, `<footer>`, `<form>`, `<h1>`-`<h6>`, `<header>`, `<hr>`, ``, `<main>`, `<nav>`, `<noscript>`, ``, `<p>`, `<pre>`, `<section>`, `<table>`, `<tfoot>`, ``, `<video>`.

Sin embargo, hay elementos que se pueden comportar como elementos en línea o en bloque según las circunstancias utilizando la propiedad **display** de CSS.

Elementos de la estructura básica del documento

La estructura básica de un documento HTML contiene las siguientes etiquetas:

Elemento	Descripción
html	Las etiquetas <html> y </html> indican la apertura y el cierre de un documento HTML, y no debe añadirse nada antes o después de ellas excepto el DOCTYPE.
head	Las etiquetas <head> y </head> indican la apertura y el cierre de la cabecera del documento y la información que contiene no se muestra en el navegador, a excepción del contenido de la etiqueta <title>.
body	Las etiquetas <body> y </body> delimitan el cuerpo del documento HTML, y su contenido se muestra en el navegador.

Elementos que dan formato al texto

Para dar formato al texto las principales etiquetas:

Elemento	Descripción
p	Para delimitar párrafos.
hi	Para los encabezados. Donde "i" es un valor entre 1 y 6 para indicar que el tamaño de la letra del encabezado sea mayor cuanto menor sea el valor de "i". Solo se utilizan en títulos de párrafos.
hgroup	Para agrupar un conjunto de uno o más encabezados.
header	Puede contener logotipos, encabezados hi, menús de navegación nav, secciones, artículos...

Elemento	Descripción
br	Para introducir un salto de línea.
b	Para el texto en negrita.
i em	Para el texto en itálica o cursiva.
u	Para el texto subrayado.
cente r	Para centrar el texto.
sup	Superíndice.
sub	Subíndice
hr	Para introducir una línea horizontal
com entario	El texto del comentario no se muestra en el navegador y se escribe <!-- comentario -->.

Elementos de listas

—

La estructura básica de un documento HTML contiene las siguientes etiquetas:

Elemento	Descripción
li	Delimita un elemento de una lista.
ul	Delimita los elementos que forman una lista desordenada. Anida elementos "li".
ol	Delimita los elementos que forman una lista ordenada. Anida elementos "li".
dd	Delimita una definición de una lista de descripción.
dt	Delimita un término de una lista de descripción.
dl	Delimita los elementos que forman una lista de descripción

Un ejemplo de las diferentes listas:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplos de las tres listas</title>
  </head>
  <body>
    <h3>Una lista desordenada: Bebidas</h3>
    <ul>
      <li>Agua</li>
      <li>Cerveza</li>
      <li>Vino</li>
    </ul>
    <h3>Una lista ordenada: Menú</h3>
    <ol>
      <li>Primer plato</li>
      <li>Segundo plato</li>
      <li>Postre</li>
    </ol>
    <h3>Una lista de descripción: Tareas</h3>
    <dl>
      <dt>Revisar correo</dt>
      <dd>Contestar sin falta</dd>
    </dl>
  </body>
</html>
```

```
<dt>Compras semanales</dt>
<dd>Falta la comida para el finde</dd>
<dt>Documentación</dt>
<dd>Reclamar gastos de hipoteca</dd>
</dl>
</body>
</html>
```

Elementos de tablas

Los elementos para definir una tabla serían los siguientes:

Elemento	Descripción
table	Para delimitar el contenido de una tabla.
tr	Para delimitar una fila de la tabla dentro de una tabla.
td	Para delimitar una celda de la tabla dentro de una fila.
thead	Para definir la fila de cabecera de la tabla.
th	Para delimitar una celda de la cabecera.
tfoot	Para definir la fila de pie de la tabla.
colgroup	Para agrupar columnas de la tabla.
tbody	Para agrupar filas de la tabla.
dl	Delimita los elementos que forman una lista de descripción

Un ejemplo de tabla:

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Un ejemplo de tabla</title>
  </head>
  <body>
    <h3>Menú</h3>
    <table border = "1">
      <thead>
        <tr><th>Plato</th><th>Sugerencia</th></tr>
      </thead>
      <tbody align="left">
        <tr><td>Primero</td><td>Ensalada</td></tr>
        <tr><td>Segundo</td><td>Pollo</td></tr>
        <tr><td>Postre</td><td>Fruta</td></tr>
      </tbody>
    </table>
  </body>
</html>
```

Elementos de formularios

El formulario permite obtener información del usuario. Un formulario contiene un conjunto de campos que el usuario rellena en una página web y se envía a la URL indicada.

Los elementos para definir y gestionar formularios son los siguientes:

Elemento	Descripción
form	Para delimitar la definición de todos los campos del formulario.
input	Para definir una caja de texto para textos cortos.
textarea	Para definir una caja de texto para textos largos.

Elemento	Descripción
select	Para definir un menú desplegable que permite elegir de una lista de opciones.
option	Para delimitar las opciones de un menú desplegable.
button	Para definir un botón donde, a diferencia de los botones hechos con input, se permite introducir en el botón otros elementos de HTML, como imágenes.
label	Para definir una etiqueta de un campo del formulario.

En la etiqueta de apertura <form> se deben indicar los atributos básicos:

- action="" .Para indicar la acción a realizar al enviar el formulario. En general, se indica el nombre de un archivo alojado en el servidor que se encargará de procesar la información.
- method="post" ó method="get". Para determinar el método de transferencia de las variables con la información introducida en el formulario. El método "post" envía los datos de forma no visible y el método "get", en cambio, adjunta los datos a la URL a la que se redirige.

Un ejemplo de un formulario:

```
<form action="http://www.lmsgi.com/UD2/formulario.php" method="POST">
    <!-- Campo1 del formulario -->
    <!-- Campo2 del formulario -->
</form>
```

Principales etiquetas para formularios

Entre las principales etiquetas para definir los campos de un formulario, tenemos:

Elemento	Atributo	Descripción
input	name=""	Nombre que se le da al campo y no es visible en el navegador. Solamente se utiliza para diferenciar los campos al enviar la información al servidor.
input	type=""	Define el tipo de datos del campo que puede ser: text, password, checkbox, radio, file, hidden, submit, reset.
input	pattern=""	Para utilizar expresiones regulares en el contenido insertado por el usuario en los inputs del formulario. Por ejemplo: <code><input type="text" name="nombre" pattern="[A-Za-z]{4-6}></code>
input	required	El navegador no permite el envío del formulario si el campo está vacío. Puede ser utilizado en <textarea> y en muchos elementos <input>, excepto en los tipos hidden, image o botones como submit. Por ejemplo: <code><label for="nombre">Nombre</label> <input id="nombre" name="nombre" type="text" required/></code>
label	for=""	Las etiquetas se utilizan para añadir un texto o descripción de los campos de un formulario. Se utiliza la etiqueta <label> con el atributo for para indicar el nombre (atributo name) del campo asociado. Por ejemplo: <code><label for="campo1">Etiqueta</label> <input name="campo1" type="text"/></code>
select	name=""	Para definir el nombre del campo.
select	size=""	Especifica el número de opciones visibles a la vez. Si no se indica nada o se le da valor uno, se presentará como un menú desplegable. Para valores mayores de uno aparece como una lista con barra de desplazamiento.

Elemento	Atributo	Descripción
select	multiple	Para seleccionar más de una opción a la vez.
option	value=""	Para indicar el valor que se envía con el formulario.
option	selected	Para indicar la opción seleccionada por defecto. Si no se indica, aparece como seleccionado el primer elemento de la lista.

Otros elementos

—

Otros elementos de HTML5 que podemos utilizar son:

Elemento	Descripción
img	Para Insertar una imagen donde es obligatorio utilizar el atributo src para especificar el path o ubicación del archivo de imagen que se quiere insertar. Con alt se escribe un texto alternativo y dar dimensiones con los atributos height y width. Por ejemplo: <code></code>
br	Para insertar una línea en blanco. No tiene etiqueta de cierre.
div	Para agrupar elementos de bloque (párrafos, encabezados, listas, tablas, otras divisiones, etcétera). Si no se da estilo al div, el navegador no muestra nada. No

Elemento	Descripción
	hace falta definir divisiones cuando el grupo es de un solo elemento.
video	<p>Para incrustar vídeos en la página web. Puede reproducirse con autoplay, en bucle con loop, controlando la reproducción con controls, etc. Por ejemplo:</p> <pre data-bbox="393 540 1269 572"><video src="UnVideo.mp4" width="640" height="480" controls></video></pre>
audio	<p>Para insertar archivos de audio. Por ejemplo:</p> <pre data-bbox="393 688 1139 720"><audio id="player" src="UnArchivo.omp3" controls></audio></pre>

Etiquetas semánticas

En HTML5 se introduce el concepto de **esquema** (outline) que consiste en estructurar el contenido de la página en secciones y bloques de contenido.

Hasta HTML4, el contenido se jerarquizaba utilizando los encabezados "h1", y se siguen utilizando en HTML5, pero ahora se definen de forma explícita nuevas secciones y los títulos "h1" son relativos a cada una de esas secciones. De esta forma desaparece el posible problema de no respetar la jerarquía de títulos "h1". En concreto, las etiquetas `<article>`, `<section>`, `<nav>` y `<aside>` definen secciones y reinician los títulos "h1" que contienen.

En general, las etiquetas semánticas de HTML5 se utilizan para estructurar los documentos HTML para identificar cada bloque de la página, evitando usar etiquetas `<div>` como en versiones anteriores de HTML. Con las etiquetas semánticas, el navegador conoce cuál es la estructura de la página y la procesa correctamente. Estas etiquetas estructurales son:

- **header**: Contiene un grupo de elementos introductorios (encabezados, logos, menús globales) ó de navegación específicos de una sección de la página web ó de su totalidad. No tiene por qué ser único en un documento, se puede incluir un header por cada sección. No hay que confundirlo con el elemento head, que proporciona los metadatos para el documento. Puede ir dentro de body si es encabezado de toda la página y en section, article, nav ó aside si el encabezado es solo para esos elementos.
- **section**: Identifica las partes o zonas temáticas de los contenidos dentro de una página web. Es decir, la información que contiene debe estar relacionada de alguna forma. Una sección puede tener

subsecciones, y utilizando elementos "hi" se sigue estructurando la página para jerarquizar el contenido, aunque la recomendación de HTML5 es utilizar el esquema (outline). Hasta HTML4 se utilizaba para esta función el elemento div, pero existe una sutil diferencia entre section y div. Cuando se crean bloques cuyo contenido está relacionado se usa <section>, y cuando estos bloques responden a diseño o necesidades de estructura se usa <div>.

- **article:** Define un contenido de un tema específico dentro de una página web. Su característica principal es que puede distribuirse de forma independiente. Se utiliza en mensajes de foro, comentarios enviados por los usuarios, artículos de periódicos, revistas y blogs. Es decir, contenidos que pueden publicarse o imprimirse de forma independiente. Una página web puede contener uno ó varios artículos de temas relacionados ó no. También se pueden anidar. Un elemento article puede contener secciones, y viceversa.
- **aside:** Agrupa contenido separado del contenido que acompaña. Suele aparecer al lado del contenido. Ejemplo típico son los anuncios que aparecen en los laterales, contenidos en barras laterales, citas, enlaces externos, etcétera. Puede estar dentro de body, en cuyo caso es contenido secundario en relación con la página web completa. Si va dentro de section ó article es contenido secundario en relación con el elemento en el que esté incluido. Es el contenido propiamente, y su relación con otros elementos, lo que lo define.
- **nav:** Agrupa enlaces de navegación a las diferentes secciones de la página web o a otras páginas. Una página web puede contener varios nav, y su significado dependerá del elemento header, body ó aside que lo contenga. Sus usos suelen ser: en una barra de navegación principal de la página web, para definir un conjunto de enlaces que apuntan a diferentes partes de la página actual ó para una zona de búsqueda utilizada como mecanismo de navegación dentro de la página. Si aparecen enlaces a pie de página web no se utilizará el elemento nav.
- **footer:** Agrupa enlaces hacia otros sitios, secciones de la página o información general de la página, como son el autor, las licencias y el copyright, entre otras. Puede ir dentro de body si es pie de toda la página web y en section, article, nav o aside si es pie solo para esos elementos.
- **main:** Establece el contenido principal del documento y que es único. Todo aquello que se repite en todas las páginas del sitio web (menús, pie de página, barras laterales, etcétera) no va en main. No tiene que haber más de un main por página y no puede ir dentro de header, article, aside, footer y nav.

Es importante tener en cuenta que, utilizando las etiquetas semánticas, cada sección tiene su propio esquema de documento y no se necesitan los div, aunque se pueden seguir utilizando. Por otra parte, la jerarquía de títulos es relativa a la sección o artículo. De esa forma, además, se puede mover una sección completa a un documento diferente manteniendo el mismo esquema.

Las etiquetas semánticas no tienen un estilo predeterminado que el navegador vaya a asignar directamente. Es decir, no se colocan en un determinado sitio de la página web. Por ejemplo, aunque header significa 'cabecera' no por eso el navegador va a colocar el elemento en la parte superior del documento ó, en el caso de footer, no lo colocará en la parte inferior. Es el desarrollador quien tiene que diseñar o establecer el estilo que se quiera aplicar a los elementos del documento HTML. Y esto lo hace mediante el uso de CSS.

Un ejemplo de la estructura de una página web en HTML5 utilizando etiquetas semánticas:

body

header

nav

aside

main

section

article

article

section

footer

Hojas de estilo CSS

```
<!DOCTYPE html PUBLIC "-//W3C//DTD
<html dir="ltr" xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Hello World</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <meta name="keywords" content="HTML, CSS, JavaScript"/>
    <meta name="description" content="A simple Hello World page"/>
    <meta name="content-language" content="es"/>
    <link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>
<div class="banner">
    <div style="margin:0 auto; width:300px; border:1px solid black; padding:10px; text-align:center">
        <h1>Hello World</h1>
        <p>This is a simple example of a CSS styled page.</p>
    </div>
</div>
</body>
```

CSS puede ser utilizado tanto en HTML como en XML. Sin embargo, no todos los navegadores soportan por igual el uso de hojas de estilo CSS.

Introducción

CSS (Cascading Style Sheets, en español "Hojas de estilo en cascada") puede ser utilizado tanto en HTML en general (HTML5 en particular) y en XML.

El W3C (World Wide Web Consortium) es el organismo que elabora las recomendaciones para las hojas de estilo y que son un estándar para los navegadores.

En resumen, las ventajas de utilizar CSS son:

- Los usuarios pueden configurar su propia hoja de estilo, mejorando la accesibilidad.
- Una misma página web puede vincularse a hojas de estilo diferentes según las necesidades y una misma hoja de estilo puede ser utilizada en diferentes sitios web.
- El uso de hojas de estilo simplifica y reduce el tamaño del código HTML.

La versión actual de CSS es la 3, CSS3 que, entre otras opciones, incluye soporte para selectores adicionales, sombras, esquinas redondeadas, múltiples fondos, animaciones y transparencias.

Para conocer el soporte del navegador a cada uno de los selectores de CSS se puede utilizar la herramienta online llamada [caniuse](#) que es de código abierto y tiene una herramienta de búsqueda que, al introducir una propiedad, parámetro ó característica dice exactamente qué navegadores la soportan y qué versiones. Además, esa compatibilidad la expresa también en porcentaje, ya que podría ser parcialmente compatible.

Cómo usar CSS en un documento

Las hojas de estilo se pueden añadir al código HTML de tres formas:

ESTILO EN LÍNEA

ESTILO INTERNO

ESTILO EXTERNO

El estilo en línea solo permite aplicar un estilo a un solo elemento HTML.

Por ejemplo:

```
<p style="color:blue">Mi pequeño párrafo</p>
```

Donde style se define como atributo del párrafo y se mostrará el texto "Mi pequeño párrafo" en azul.

ESTILO EN LÍNEA

ESTILO INTERNO

ESTILO EXTERNO

El estilo interno permite aplicar un solo estilo a toda la página web. Se define en la etiqueta `<head>` de la página web utilizando la etiqueta `<style>`.

Por ejemplo:

```
<head>
  <style>
    body {
      background-color: orange;
    }
    h1 {
      color: blue;
      padding: 60px;
    }
  </style>
</head>
```

ESTILO EN LÍNEA

ESTILO INTERNO

ESTILO EXTERNO

En este caso, el archivo de estilo CSS creado puede ser utilizado en diferentes páginas HTML. El archivo puede crearse con cualquier editor de texto y debe tener extensión .css.

De esta forma sí que existe una separación real entre el código HTML y el estilo CSS. El vínculo con la hoja de estilo externa se hace en la sección <head>.

Por ejemplo:

```
<head>
  <meta charset=utf-8" />
  <title>Mi pàgina de BirtLH</title>
  <link rel="stylesheet" href="estilosBirtLH.css" />
</head>
```

Donde el contenido del archivo de estilosBirtLH.css podría ser:

```
p {color: blue;}
```

Los atributos de link son:

- **rel**: Indica el tipo de relación existente entre el archivo que se referencia con la etiqueta link y el documento HTML donde se carga. En el ejemplo se indica stylesheet porque la relación es de hoja de estilos, es decir, el archivo que se referencia es una hoja de estilos y se aplica al documento HTML.
- **href**: Indica la URL absoluta o relativa del archivo externo referenciado.
- **type**: indica el tipo MIME del archivo referenciado. En este caso, y al ser una hoja de estilos, será type="text/css", pero HTML5 no lo necesita ya que, por defecto, una hoja de estilo es text/css.

Sintaxis CSS

Una hoja de estilo consta de uno o más estilos, y cada uno de ellos se llama **regla**.

Cada regla consta de:

- **Selector**: para seleccionar los elemento/s HTML a los que se aplica dicha regla. No se deben repetir los nombres de selectores, para evitar confusión.
- Una llave de apertura {.

- **Declaración:** donde se definen los estilos concretos que se aplican a los elementos y si consta de varias propiedades, irán separadas entre sí por un punto y coma (;). Consta de:
 - **Propiedad:** que indica la característica de un atributo del elemento.
 - Dos puntos (:).
 - **Valor:** define el nuevo valor del atributo.
- Una llave de cierre }.

Por ejemplo, en la regla **p {color: blue;};**:

- **p** es el selector que indica que se selecciona el elemento párrafo.
- **color: blue** es la declaración que aplica el valor blue(azul) a la propiedad color.



Propiedades de CSS

Hay que tener en cuenta que las unidades de medida que se pueden utilizar en CSS son de dos tipos:

- Absolutas:
 - milímetros (mm)
 - centímetros (cm): un cm son 10 mm
 - pulgadas (in): una pulgada son 2,54 cm
 - puntos (pt): un punto son 1/72 in
- Relativas:
 - element (em): hace referencia al tamaño en puntos de la letra que se está utilizando.
Si se utiliza una tipografía de 12 puntos, 1 em equivale a 12 puntos.
 - pixels (px) es relativo a la resolución del monitor.

Las propiedades se clasifican en los siguientes grupos:

Propiedades de color y fondo

La siguiente tabla enumera las principales propiedades de color y fondo usados en CSS.

Elemento	Descripción y valores
color	Para definir el color del texto. Lo admiten casi todas las etiquetas de HTML. Valor es un nombre de color(en inglés) ó su valor RGB. Aunque es más sencillo utilizar el nombre del color, sucede que no todos los nombres están estandarizados y se aconseja utilizar el valor RGB.
background-color	Para definir el color de fondo de un elemento. Valor: es un nombre de color ó su valor RGB, ó la palabra "transparent".

Elemento	Descripción y valores
background-image	<p>Para situar una ó más imágenes de fondo para un elemento.</p> <p>Valor es un nombre de archivo de imagen y se referencia de forma absoluta ó relativa.</p>
background-repeat	<p>Permite repetir el fondo del documento, y si es así, cómo hay que hacerlo: horizontalmente ó verticalmente.</p> <p>Valores posibles: repeat-x, repeat-y, no-repeat. Si aparecen dos valores, el primero se aplica al eje X (repetición horizontal) y el segundo al eje Y (repetición vertical).</p>
background-position	<p>Para definir la posición inicial de la imagen de fondo.</p> <p>Valores posibles: porcentaje, tamaño, [top, center, bottom], [left, center, right].</p>
background-attachment	<p>Permite definir si la posición de la imagen de fondo será fija dentro de la pantalla ó se desplazará con su bloque contenedor.</p> <p>Valores posibles: scroll ó fixed.</p>
background	<p>Es un atajo para definir los valores individuales del fondo en una única regla CSS.</p> <p>Valores posibles: background-color, background-image, background-repeat, background-attachment, background-position.</p>

Propiedades de fuente

—

La siguiente tabla enumera las principales propiedades de formato de fuente en CSS.

Elemento	Descripción
font-size	Para indicar el tamaño del tipo de letra (fuente) del elemento. El valor puede ser el tamaño absoluto, relativo, distancia ó porcentaje.
font-family	Indica el tipo de letra o fuente del elemento. El valor es el nombre de la familia fuente.
font-weight	Para definir el grosor del trazo. Valores posibles: normal, bold, bolder, lighter y valores numéricos 100, 200, 300, 400, 500, 600, 700, 800 o 900.
font-style	Para elegir la inclinación de la fuente. Valores posibles: normal, italic, oblique (se inclina sin cambiar de forma).
font-variant	Para elegir entre fuente normal ó mayúsculas pequeñas. Valores posibles: normal, small-caps.
line-height	Para definir el espaciado entre líneas de un bloque de texto. Valores posibles: en unidad de longitud, porcentaje ó em.
font	Permite definir simultáneamente las propiedades relacionadas con el tipo de letra que figuran en la tabla. El orden es importante, aunque no es necesario que aparezcan todas ellas: font-style, font-variant, font-weight, font-size/line-height, font family. Deben ir separadas por espacios. Obligatorias son font-size y font family.

Propiedades de texto

a tabla siguiente enumera las principales propiedades que se pueden aplicar a un texto.

Elemento	Descripción
text-decoration	Para indicar si el texto es normal, subrayado, sobrerrayado ó tachado. Valores posibles: none, underline, overline y line-through.
text-align	Para definir la alineación horizontal en el texto. Valores posibles: left, right, center, justify.
vertical-align	Para definir la alineación vertical en el texto. Valores posibles: baseline, sub, super, top, text-top, middle, bottom, text-bottom ó un porcentaje.
text-indent	Para definir la tabulación ó sangría del texto. Valores posibles: una longitud ó porcentaje.
text-transform	Para cambiar el texto a mayúsculas ó minúsculas. Valores posibles: none, capitalize, uppercase, lowercase.
word-spacing	Para definir el espaciado entre las palabras de un texto. Valores posibles: tamaño positivo ó negativo, se establecen como una longitud y se añaden al espaciado normal.
letter-spacing	Para definir el espaciado entre las letras de un texto. Valores posibles: tamaño positivo o negativo, se establecen como una longitud y se añaden al espaciado normal.

Un ejemplo:

```

h1{
  color:blue;
  text-decoration:underline;
}

p{
  color:black;
  font-size:12px;
  text-align:center;
  text-transform:uppercase;
  word-spacing:20px;
}

ul{

```

```
text-align:left;  
text-indent:20px;  
font-weight:bold;  
font-style:italic;  
font-size:10px;  
color:red;}
```

Propiedades de caja

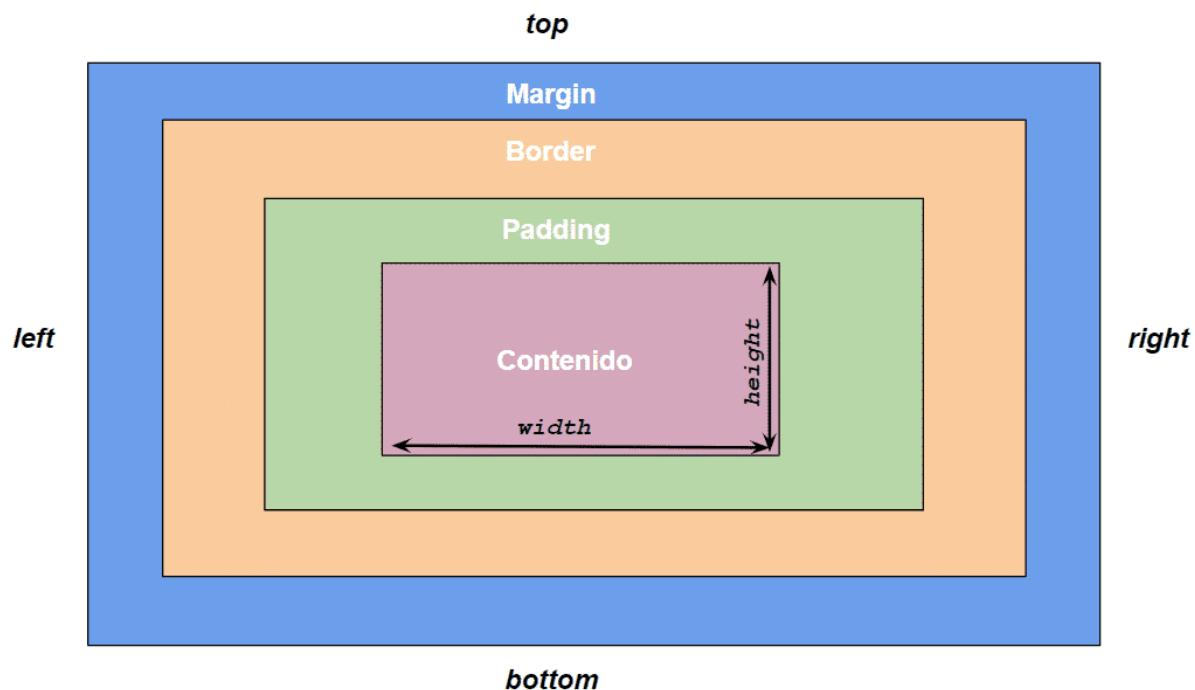
De forma general, el diseño de las hojas de estilo CSS se basan en el modelo de caja. De esta manera, cada caja ocupa un espacio en la página web y todo elemento de la página web está contenido en una caja.

Elemento	Descripción
margin-top margin-right margin-bottom margin-left	Indica el tamaño del margen superior, derecho, inferior e izquierdo. Por ejemplo: <pre>p { margin-top: 100px; margin-bottom: 100px; margin-right: 150px; margin-left: 80px; }</pre>
margin	Establece los márgenes de una vez pero en este orden: superior, derecho, inferior e izquierdo. Por ejemplo: <pre>p { margin: 25px 50px 75px 100px; }</pre>
padding-top padding-right padding-bottom padding-left	Indica el espacio entre el borde superior, derecho, inferior e izquierdo y el contenido. Por ejemplo: <pre>div { padding-top: 50px; padding-right: 30px; padding-bottom: 50px; }</pre>

Elemento	Descripción
	<pre data-bbox="600 264 861 340">padding-left: 80px; }</pre>
padding	<p>Establece el espacio entre los bordes y el contenido de una sola vez pero en este orden: superior, derecho, inferior e izquierdo.</p> <p>Por ejemplo:</p> <pre data-bbox="600 544 1008 656">div { padding: 25px 50px 75px 100px; }</pre>
border-top-color border-right-color border-bottom-color border-left-color	<p>Establece el color del borde superior, derecho, inferior e izquierdo.</p> <p>Por ejemplo:</p> <pre data-bbox="600 834 975 1051">div { border-right-color: red; border-bottom-color: green; border-top-color: blue; border-left-color: yellow; }</pre>
border-color	<p>Establece el espacio entre los bordes y el contenido de una sola vez pero en este orden: superior, derecho, inferior e izquierdo.</p> <p>Por ejemplo:</p> <pre data-bbox="600 1290 1033 1389">p { border-color: red green blue yellow; }</pre>
border-style	<p>Establece el estilo del borde.</p> <p>Los valores pueden ser: none=ninguno, dotted=punteado (no funciona siempre), solid=sólido, double=doble borde, los valores groove, ridge, inset y outset son bordes con varios efectos 3D. Por ejemplo:</p> <pre data-bbox="600 1670 1033 1833">p { border-style: solid; border-color: red green blue yellow; }</pre>

Elemento	Descripción
border-width	Establece el tamaño de los bordes del elemento al que lo aplicamos en el orden: superior, derecho, inferior, izquierdo.
width height	Establece el ancho y el alto del contenido del elemento.

La figura siguiente muestra la estructura de la caja:



Propiedades de clasificación

Estas propiedades influyen en la forma de representar elementos de clasificación HTML, como son las listas.

La siguiente tabla describe las principales propiedades de clasificación y sus valores posibles.

Propiedad	Descripción y valores
display	Indica si el elemento es de bloque, línea, lista ó ninguno de ellos. Valores posibles: block, inline, list-item o none.
white-space	Indica el modo de tratar los espacios en blanco. Valores posibles: pre (se representan todos los espacios), nowrap (la línea no se cortará al llegar al extremo de la página.), normal (agrupando espacios seguidos como uno solo).
list-style-type	Establece la viñeta que acompaña a un elemento de lista. Se aplica a los elementos que tengan definido su atributo display como list-item. Valores posibles: disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none.
list-style-image	Establece la imagen que se usará como marca de elemento de lista. Se aplica a los elementos que tengan definido su atributo display como list-item. El valor que toma es el path del archivo de la imagen.
list-style-position	Indica si la marca de lista aparece dentro del cuadro al que pertenece o aparece fuera del cuadro. Se aplica a los elementos que tengan definido su atributo display como list-item. Valores posibles: outside o inside.
list-style	Establece las propiedades de una lista de una vez pero en este orden: list-style-type, list-style-image y list-style-position.

Posicionamiento

Los navegadores crean una caja para representar cada elemento de la página HTML y la posicionan de forma automática en ella, pero desde CSS se puede modificar dicha posición.

Existen tres formas de posicionar una caja:

- Posicionamiento normal o estático: Es el utilizado por defecto en los navegadores. Los elementos se muestran en la pantalla en el mismo orden en que se encuentran en el código fuente. Hay que conocer si el elemento es de bloque o en línea, sus propiedades width, height y su contenido.
- Posicionamiento flotante: Los elementos se muestran en la pantalla en el mismo orden en que se encuentran en el código fuente, pero pueden estar desplazados a derecha o izquierda de la posición en la que inicialmente estaba. Utiliza la propiedad **float** con valores none, right ó left.
- Posicionamiento absoluto. Se utiliza para establecer de forma absoluta ó exacta la posición en la que se muestra la caja de un elemento. Los elementos se pueden mostrar en pantalla en orden diferente al del código fuente. Utiliza la propiedad **position** con los valores posibles static, relative, fixed ó absolute.

Definición y uso de clases

Cuando se quiere aplicar estilos a un solo elemento de la página HTML se puede utilizar el atributo **class** de HTML sobre ese elemento y así enlazar directamente con la regla CSS que se le debe aplicar.

La clase define un estilo que, en principio, no está asociado a ninguna etiqueta HTML, pero que es susceptible de asociar a etiquetas concretas.

El atributo class se define mediante una lista de elementos separados por espacio, y uno de esos elementos debe coincidir exactamente con el nombre de clase dado en el selector.

En la declaración del selector CSS se incluye el nombre de clase precedido de un punto (.). Así, el selector de identificador aplica el estilo al elemento HTML cuyo identificador sea ese nombre.

Por ejemplo, se define así una clase para el color rojo:

```
.clase_rojo{color:red}
```

Si se quiere que el título "h1" se muestre en rojo, se asocia en el archivo HTML el elemento h1 a dicha clase de la forma:

```
<h1 class="clase_rojo">Cabecera h1 en rojo</h1>
```

Si solo se quiere utilizar esta clase con el elemento HTML de párrafo, en la declaración de la clase hay que añadir el selector p, y queda:

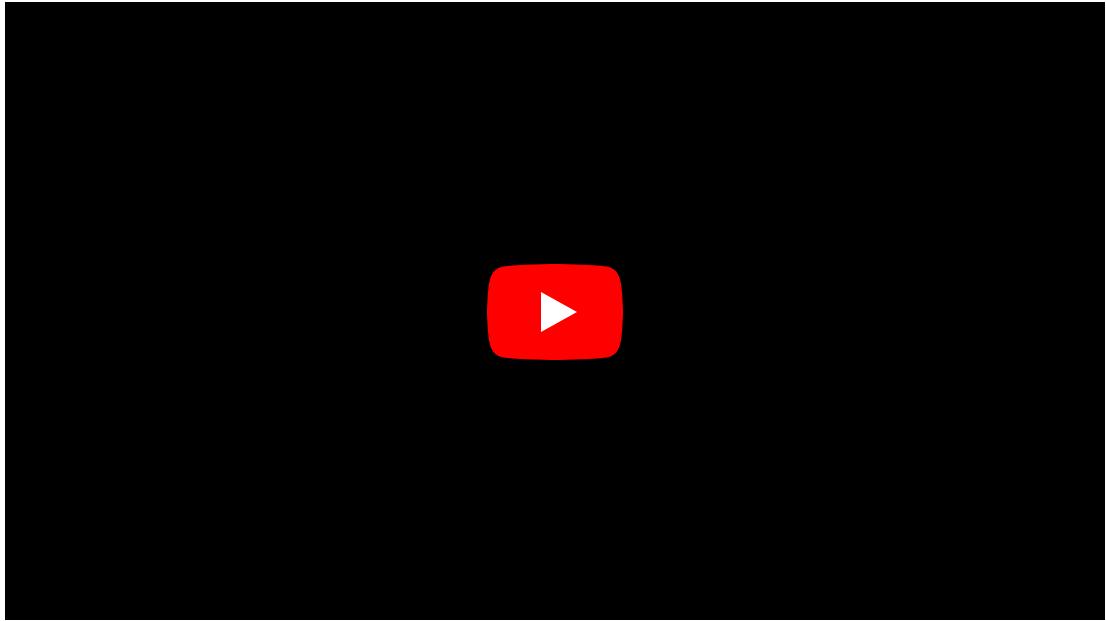
```
p.clase_rojo{color:red}
```

Por ejemplo, en la siguiente página web se ve el uso de clases:

```
<!DOCTYPE html>
<html>
```

```
<head>
    <title>Uso de clases con CSS interno</title>
    <style type="text/css">
        .clase_azul{color:blue}
        p.clase_rojo{color:red; font-weight:bold; }
    </style>
</head>
<body>
    <h1 class="clase_azul">Este encabezado está en azul.</h1>
    <p>Este párrafo no tiene estilo.</p>
    <p class="clase_azul">Este párrafo está en azul.</p>
    <p class="clase_rojo">Aqui el texto es rojo y en negrita.</p>
    <h3 class="clase_rojo">Pero este texto aparece sin estilo.</h3>
</body>
</html>
```





Autoevaluación

Pregunta

01/03

Toda página HTML5:

- Debe tener código HTML5, CSS y JavaScript.
- Debe tener código HTML5 y CSS.
- Debe tener solamente código HTML5.
- Debe tener un **prólogo** y un **ejemplar**.

Pregunta

02/03

El prólogo de un documento HTML5 está delimitado por las marcas <html> y </html>.

Verdadero

Falso

Pregunta

03/03

Una regla en CSS consta de:

- Un selector
- Una llave de apertura {
- Una declaración
- Una llave de cierre }
- Una serie de atributos