

## Unidad 2

---



# Lenguajes para la visualización de información

Lenguaje de Marcas y  
Sistemas de Gestión de  
la Información



# Índice



## 2.1. Herramientas de diseño web

- 2.1.1. KompoZer
- 2.1.2. Adobe Dreamweaver CC
- 2.1.3. Bluefish
- 2.1.4. HTML Online Editor

## 2.2. Identificación de etiquetas y atributos de HTML

- 2.2.1. Versiones de HTML
- 2.2.2. Estructura de un documento HTML
- 2.2.3. Declaración de tipo de documento
- 2.2.4. Etiquetas y atributos HTML
- 2.2.5. HTML5
- 2.2.6. Las nuevas etiquetas HTML5

## 2.3. Hojas de estilo CSS

- 2.3.1. Versiones de CSS
- 2.3.2. Sintaxis básica
- 2.3.3. Formas de aplicar estilos a un documento HTML
- 2.3.4. CSS3
- 2.3.5. Las nuevas propiedades de CSS3

## 2.4. XHTML

- 2.4.1. Versiones de XHTML
- 2.4.2. Diferencias sintácticas y estructurales con HTML
- 2.4.3. Ventajas de XHTML sobre HTML



## Introducción

Cuando se usan los navegadores para consultar las distintas webs, se aprecia una serie de páginas que contienen elementos como enlaces, texto, imágenes, vídeos, etc. Si se accede al código fuente de dicha página, se aprecia que el HTML (hypertext markup language) es el componente básico de la web, junto con otras tecnologías entre las que destacan CSS, AJAX, JavaScript, etc.

HTML es un lenguaje de marcado que se usa para la creación de páginas web y tuvo su origen en 1991. Hoy en día, se encuentra disponible el primer documento que se publicó.

Para poder crear cualquier documento, es necesario disponer de un editor de texto plano. Existen herramientas de escritorio gratuitas y comerciales que facilitan este trabajo. Si no desea instalarse ni usarse ninguna herramienta de escritorio, existen herramientas online que permiten crear y editar este tipo de documentos.

De HTML, se estudiará su estructura y sintaxis, así como sus elementos principales (listas, tablas, formato de texto, párrafos, cabeceras, formularios, etc.).

Elaborar una web sin un diseño atractivo es algo impensable. En este punto, es cuando entran en juego las hojas de estilo en cascada (CSS), que describen la manera de visualizar una página web por pantalla.

## Al finalizar esta unidad

- + Nos habremos introducido en el uso de los lenguajes de marcas en entorno web, identificando los relacionados con la web y sus diferentes versiones.
- + Analizaremos la estructura de un documento HTML e identificaremos las secciones que lo componen.
- + Describiremos la funcionalidad de las principales Etiquetas y atributo.
- + Aprenderemos las ventajas que aporta la utilización de hojas de estilo.
- + Identificaremos las semejanzas y diferencias semánticas y estructurales entre los lenguajes HTML y XHTML.

# 2.1.

## Herramientas de diseño web

En este apartado, se dará un repaso a las herramientas de diseño web más utilizadas por los desarrolladores.

### 2.1.1. KompoZer

Software para el diseño web de código abierto y de fácil uso que permite la administración de un sitio FTP, así como personalizar la barra de herramientas. Trae integrado un editor de CSS y un validador HTML. Disponible en varios idiomas, permite la descarga para sistemas operativos Windows, Linux, OS/2 y MacOSX.

### 2.1.2. Adobe Dreamweaver CC

Software comercial específico para diseñar y desarrollar sitios web basados en estándares con una interfaz moderna.

Entre sus características, destacan las siguientes: permite la previsualización en varios dispositivos; arquitectura 64 bits; compatible con PHP 7, JavaScript, GIT y CEE, y trae un espacio de trabajo para desarrolladores, que es compatible con procesadores de CSS como SASS, LESS y SCSS.

Fue creado inicialmente por macromedia y la empresa que lo proporciona actualmente es Adobe Systems.

### 2.1.3. Bluefish

Bluefish es un editor web open source multiplataforma con licencia GNU. Aunque las primeras versiones estaban plagadas de fallos y su funcionalidad era muy justa, las versiones actuales lo hacen una alternativa fiable a las plataformas de pago.

### 2.1.4. HTML Online Editor

Programa de editor HTML WYSIWYG gratuito que permite editar código fuente en línea sin descargar ninguna aplicación. Permite trabajar con documentos HTML, CSS y JS.

El funcionamiento es muy intuitivo. En el panel derecho, el usuario va introduciendo las líneas de código HTML y, de forma concurrente, el resultado aparecerá en el panel izquierdo. También puede trabajarse sobre el documento de forma visual y el código HTML asociado aparecerá en el panel derecho.

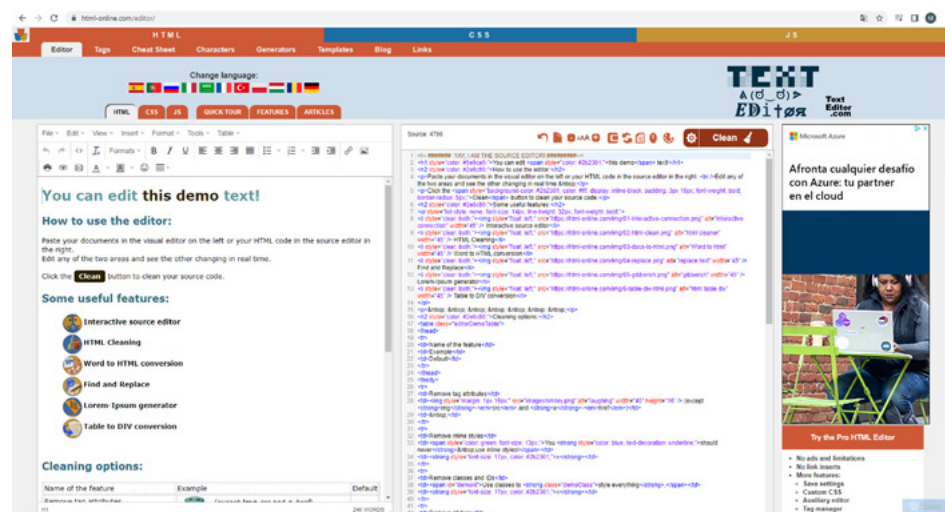


Imagen 1. Interfaz de HTML Online Editor.



# 2.2.

## Identificación de etiquetas y atributos de HTML

En este apartado del capítulo, se describe cuál es la estructura del lenguaje HTML y sus etiquetas, aspecto básico en la programación de páginas web. HTML es un lenguaje de marcado que va a ser la base del módulo Lenguajes de Marcas y Sistemas de Gestión de Información.

### 2.2.1. Versiones de HTML

Las versiones de HTML están definidas por la comunidad internacional W3C (World Wide Web Consortium). HTML nace en el año 1991, desarrollado por el científico Tim Berners-Lee (conocido como el padre de la web) y era denominado HTML Tags.

- > **HTML 2.0.** En 1995, el Internet Engineering Task Force (IETF) creó el primer estándar oficial, y, a partir del año siguiente, el W3C se encargó de publicar los estándares de HTML.
- > **HTML 3.2.** A principios de 1997, aparece esta versión, publicada por el W3C (incluyendo applets de Java, etc.).
- > **HTML 4.0.** El 24 de abril de 1998 se publica esta versión, que presenta como novedades: tablas complejas, mejoras en los formularios, hojas de estilo, añadir script y mejora de accesibilidad.
- > **HTML 4.01.** A finales de 1999, surge HTML 4.01 como actualización y revisión de HTML 4.0, aunque no aporta novedades relevantes.
- > **HTML 5.** Publicada a finales del 2014, establece nuevos elementos, algunos similares a los anteriores, pero con un significado semántico.

### 2.2.2. Estructura de un documento HTML

La estructura principal de una página web está compuesta por una serie de elementos dentro de un documento de texto plano que dan un formato al contenido que lo forma. Dicho contenido es interpretado por el navegador, que muestra el resultado final.

#### Estructura

Los navegadores también permiten ir inspeccionando el código conforme aparece el resultado. La estructura principal de una página web es la siguiente:

La etiqueta primera de un documento es, que define el inicio del fichero, por lo que es la última que tiene que escribirse. Después, se encuentra la etiqueta, donde se especifica la cabecera del documento y que puede anidar dentro otras como:

```
<html>
  <head>
    <title>Lenguaje de Marcas</title>
  </head>
  <body>
    <!-- información -->
  </body>
</html>
```

Imagen 2. Estructura principal de una página web.





- > **<title>**. Título que se muestra en la barra de título del explorador.
- > **<meta>**. Sirve para el uso de metadatos y posicionar la web en los buscadores.
- > **<style>**. Se utiliza en la definición de estilos.
- > **<link>**. Vincula las hojas de estilo como recurso externo.

Por último, se aprecia la cabecera o cuerpo, que es donde se define el contenido de una página web visible en los navegadores.

En el ejemplo anterior (estructura principal de una página web), se aprecia un comentario, ientre las etiquetas `<!--` y `-->`. El uso de comentarios en este tipo de documentos es muy importante, ya que ayuda a organizar el código y a ocultar texto que ya no se usa.

## Elementos HTML

La estructura básica de HTML se denomina elemento y está compuesto por dos propiedades: atributos y contenido. Por lo general, tiene una etiqueta de inicio (que puede contener un atributo con su valor) y otra de cierre.

Existen algunas etiquetas sin contenidos o vacías que no necesitan de etiqueta de cierre como, por ejemplo, `<br>`.

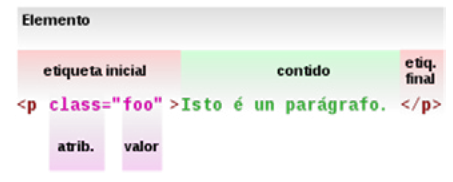


Imagen 3. Estructura de un elemento HTML.

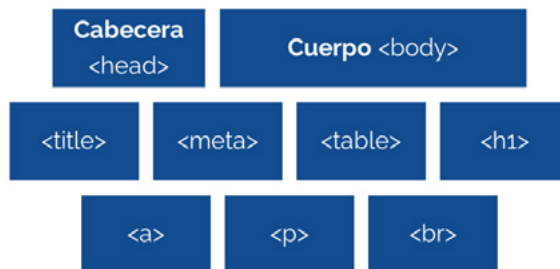


Imagen 4. Algunas de las etiquetas más usadas en HTML.

### 2.2.3. Declaración de tipo de documento

La declaración de tipo de documento (DTD) añade reglas sintácticas que permite definir el documento con sus reglas y atributos, permitiendo de esta manera comprobar si es válido. Entre la información que puede añadirse, destaca la versión HTML.

Debe declararse al principio del documento.

Por ejemplo, para HTML 4.01, existen tres DTD diferentes que incluyen los siguientes campos:

- > **Estricto**. Solo pueden usarse etiquetas HTML 4.01 y no permite marcado de presentación.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

Imagen 4. Declaración DTD Estricto.



- > **Transicional.** Permite todos los elementos HTML, incluyendo los antiguos, desaconsejados o en desuso.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd ">
```

Imagen 5. Declaración DTD Transicional.

- > **Marcos o frames.** Es similar al anterior, pero usa elementos para frames. Se ha quedado obsoleto.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd ">
```

Imagen 6. Declaración DTD de marcos.

Si se analiza cada uno de los campos de una declaración, se observa que consisten en:

- > **!DOCTYPE:** se usa esta cabecera para la declaración de tipo. Es la manera que tiene el navegador de saber la versión especificada.
- > **HTML:** informa de cuál es el elemento raíz del documento.
- > **PUBLIC:** presenta dos posibles valores para este campo, PUBLIC o SYSTEM, donde la primera de ellas indica que la declaración de tipo de documento está disponible de manera pública.
- > **"-//W3C//DTD HTML 4.01 Transitional//EN":** es el identificador público formal, que informa de la organización y contiene información diversa separada por los caracteres de la barra doble (//).
  - » **—:** presenta dos posibles valores (— es una organización no registrada por la ISO, mientras que + es una organización registrada por la ISO).
  - » **W3C:** es la organización responsable.
  - » **DTD HTML 4.01 Transitional:** indica el tipo de documento que está declarándose
  - » **EN:** indica el idioma de la DTD.
- > **"http://www.w3.org/TR/html4/loose.dtd ">:** indica el URL de la versión del DTD.



## 2.2.4. Etiquetas y atributos HTML

Los atributos son elementos compuestos por dos partes claramente diferenciadas (el nombre del atributo y su valor correspondiente) y se añaden en la etiqueta de inicio de un elemento HTML.

La manera de asignar un valor a un atributo es mediante el signo igual (=). El valor puede ponerse con comillas dobles, simples o sin comillas en HTML.

### Atributo bgcolor

Atributo para signar color usado en algunas etiquetas. La manera de asignar el color es muy diversa: en hexadecimal, mediante el nombre del color y en RGB.

- > **Hexadecimal:** el valor es "#xyyzz", siendo ese grupo de letras 3 agrupaciones numéricas con valores hexadecimales (0 1 2345 67 8 9ABCDE F). Los valores de baja intensidad se representan con 00, donde FF son los de mayor intensidad.

El color negro se representa con el código #000000, mientras que el blanco es #FFFFFF.

- » xx es la intensidad de rojo.
- » yy es la intensidad de verde.
- » zz es la intensidad de azul.
- > **Por nombre:** red, yellow, blue, brown, silver, gray, black, white, etc.
- > **RGB:** son colores aditivos o primarios y están basados en la combinación de píxeles de colores rojo (RED), verde (GREEN) y azul (BLUE).

- » Los colores se codifican con un número.
- » Los valores para cada color van desde 0 a 255, por lo tanto, hay 256 valores posibles para cada valor.
- » Con la unión de los tres colores en su máximo valor, se obtiene el color blanco y, con la ausencia de todos, el negro (todos a cero). El blanco será (255,255,255) y el negro será (0,0,0).
- » Permite representar 256 x 256 X 256 combinaciones diferentes de colores, con lo que hay 16 millones de colores.
- » Debe tenerse en cuenta que puede que no todos los navegadores admitan esta codificación de colores.

Si se mezclan los tres colores, se obtienen las siguientes combinaciones de colores secundarios:

- » Verde + azul = cian.
- » Rojo + azul = magenta.
- » Rojo + verde = amarillo.
- » Rojo + azul + verde = blanco.

```
<html>
  <head>
    <title>LM</title>
  </head>
  <body bgcolor="#FF0022">
    colores
  </body>
</html>
```

Imagen 7. Ejemplo de colores en HTML.

```
<html>
  <head>
    <title>LM</title>
  </head>
  <body bgcolor="blue">
    colores
  </body>
</html>
```

Imagen 8. Ejemplo de colores por nombre en HTML.

```
<html>
  <head>
    <title>LM</title>
  </head>
  <body bgcolor="rgb(0,0,255)">
    colores
  </body>
</html>
```

Imagen 9. Ejemplo de colores con estructura RGB en HTML.





### Etiquetas de formato de texto, línea horizontal y salto de línea

El formato dentro de un documento es fundamental y más si se habla de ver dicho documento en un navegador web. Para ello, existe una serie de etiquetas HTML que permiten formatear el texto a gusto del usuario, dotando al documento de un diseño más atractivo y legible.

- > Etiqueta **<b>**: pone el texto contenido en ella en **negrita**.
- > Etiqueta **<u>**: se usa para **subrayar** el texto.
- > Etiqueta **<i>**: permite poner en **cursiva** un texto.
- > Etiqueta **<s>**: se emplea **para tachar** una palabra o texto

Las etiquetas de línea horizontal y salto de línea se emplean mucho en los documentos creados en HTML. La de salto de línea permite algunos atributos respecto al grosor y el color.

- > Etiqueta **<br>**: salto de línea.
- > Etiqueta **<hr>**: línea horizontal (horizontal rules) que permite los atributos align para la alineación de la línea horizontal, tomando los valores "left" o "right", y width, que permite modificar el ancho.

```
<html>
<head>
  <title></title>
</head>
<body>
  <b>Ejemplos</b>
  <hr 1>
  <u>Lenguaje de marcas subrayado</u> <br 1>
  <b>Lenguaje de marcas negrita</b> <br 1>
  <s>Lenguaje de marcas tachado</s> <br 1>
  <i>Lenguaje de marcas cursiva</i><br />
  <!--ahora probamos con linea horizontal y salto de linea-->
  <hr width="50%" align = "left" 1>
    Ejemplo XML
  <br>
  <br>
  <br>
    Ejemplo HTML
</body>
</html>
```

Imagen 10. Ejemplo de etiquetas HTML

### Etiquetas para crear índices y subíndices:

- > Etiqueta **<sup>**: para el **superíndice**.
- > Etiqueta **<sub>**: para el **subíndice**.
- > Etiqueta **<p>**: esta etiqueta se usa para crear párrafos, siendo la de cierre opcional siempre y cuando no se use para alineación del texto. La alineación se realiza con el atributo align cuyos valores pueden ser "left", "center", "justify", "right", aunque existe otra serie de atributos genéricos menos importantes. La etiqueta de apertura actúa como salto de línea, similar a la etiqueta.
- > Etiqueta **<big>**: para aumentar texto.

```
<html>
<head>
  <title>LM</title>
</head>
<body>
  H<sub>2</sub></sub>0
  <br>
  <hr>
  FNMT<sub>1
</body>
</html>
```

Imagen 11. Ejemplo de etiquetas para índices.



> Etiqueta **<small>**: para disminuir texto.

```
<html>
<head>
  <title>LM</title>
</head>
<body>
  <p>Ejemplo de párrafos</p>
  <hr 1>
  <p>
  <p>
  <p align="center">Lenguaje de marcas.</p>
  <p>
  <p align="right"> Lenguaje de marcas.</p>
  <br>
  <p>lenguaje de marcas tiene <big>más de 20 alumnos matriculados</big></p>
  <br>
  <p>lenguaje de marcas tiene <small>menos de 20 alumnos matriculados</small> </p>
</body>
</html>
```

Imagen 12. Otro ejemplo de etiquetas en HTML.

> Etiqueta **<font>**: se usa para variar el tamaño, el color y la fuente de un texto. Además de los atributos genéricos, se destacan los siguientes:

- » **size**: para el tamaño de la fuente y con un valor que va del 1 al 7, donde 3 es el valor por defecto.
- » **face**: es el que especifica la fuente del texto y pueden colocarse varios nombres separados por coma.
- » **color**: especifica el color del texto.

```
<html>
<head>
  <title>LM</title>
</head>
<body>
  <p>Ejemplo de fuentes</p>
  <hr>
  <font size="6" color="blue" face="Verdana">alumnos LM</font>
</body>
</html>
```

Imagen 13. Ejemplo de etiquetas de fuentes.

## Etiquetas de encabezamientos

Estas etiquetas suelen usarse para posicionamiento web y añadir los títulos de las páginas web "recomendando que sean cortos debido al uso de palabras claves". Esta etiqueta tiene 6 posibles valores que pueden sustituirse por la letra x, siendo 1 el de mayor tamaño y 6 el de menor <hx> </hx>.

Suelen usarse las etiquetas <h1> y <h2> para el título principal de la página y los otros tamaños menores para subtítulos.

## Etiquetas de listas

Existen distintos tipos de listas que pueden definirse en HTML, pero entre ellas las más importantes son las siguientes:

- > Etiqueta **<ol>**: listas numéricas u ordenadas. Esta es la etiqueta que se usa para este tipo de listas, anidando dentro de ellas los elementos o líneas mediante la etiqueta <li>. Entre los atributos, destacan:
  - » **type**: se usa para especificar el estilo de numeración (1,a,i,I,A). Si se omite este atributo, aparecerá como ordenada numéricamente.
  - » **start**: se usa para indicar el primer número de la lista. Si no se indica, se entiende que empezará por el número 1.

```
<html>
<head>
  <title></title>
</head>
<body>
  <h1>Lenguaje de marcas</h1>
  <h2>Lenguaje de marcas</h2>
  <h3>Lenguaje de marcas</h3>
  <h4>Lenguaje de marcas</h4>
  <h5>Lenguaje de marcas</h5>
  <h6>Lenguaje de marcas</h6>
</body>
</html>
```

Imagen 14. Ejemplos de encabezamientos en HTML.

```
<html>
<head>
  <title>LM</title>
</head>
<body>
  <ol type="A">
    <li>capitulo 1</li>
    <li>capitulo 2</li>
    <li>capitulo 3</li>
    <li>capitulo 4</li>
  </ol>
  <br>
  <hr>
  <ol>
    <li>capitulo primero</li>
    <li>capitulo segundo</li>
    <li>capitulo tercero</li>
    <li>capitulo cuarto</li>
  </ol>
</body>
</html>
```

Imagen 15. Ejemplo de etiquetas de listas ordenadas.



- > **Etiqueta <ul>**: listas desordenadas. Al igual que las listas ordenadas, usa la etiqueta <li> para cada uno de sus elementos. El atributo que ha de destacarse es el siguiente:

- » **type**: informa del tipo de viñeta que ha de usarse ("disc", "square" o "circle").

- > **Etiqueta <dl>**: Listas de definición. Dentro de esta, se usan las etiquetas:

- » **<dt>**: para cada elemento

- » **<dd>**: para realizar una definición de cada elemento

```
<html>
  <head>
    <title>LM</title>
  </head>
  <body>
    <p>listas definición</p>
    <hr>
    <dl>
      <dt>HTML</dt>
      <dd>HyperText Markup Language</dd>
      <dt>CSS</dt>
      <dd>Cascading Stylesheets</dd>
    </dl>
  </body>
</html>
```

Imagen 17. Ejemplo de listas de definición en HTML.

```
<html>
  <head>
    <title>LM</title>
  </head>
  <body>
    <p>listas desordenadas</p>
    <hr>
    <ul type="square">
      <li>capitulo 1</li>
      <li>capitulo 2</li>
      <li>capitulo 3</li>
      <li>capitulo 4</li>
    </ul>
  </body>
</html>
```

Imagen 16. Ejemplo de etiquetas de listas desordenadas.

## Etiquetas de enlaces

Hoy en día, no se concibe una web sin hipervínculos a otras páginas o a otros apartados de esta, a la dirección de correo o a documentos. Pueden clasificarse en tres tipos: enlaces absolutos (a páginas de otras páginas web), enlaces relativos (a páginas situadas dentro del servidor local) y enlaces dentro de la misma página.

- > **Etiqueta <a>**. Anchor (ancla) tiene una de apertura y otra de cierre. Entre dichas etiquetas, puede insertarse un elemento como texto, imagen u otro objeto.

Entre los atributos que posee, destacan:

- » **href (hypertext reference)**: el valor que se le asigna suele ser una URL.
- » **Name**: se usa para definir una ubicación dentro del documento.
- » **Target**: se usa para indicar dónde desplegar la URL. Los valores posibles son:
  - + **\_blank**: se usa para abrir el contenido en una nueva ventana o pestaña.
  - + **\_self** es el valor por defecto. Se abre la URL en el mismo lugar donde se ha pulsado el enlace.
  - + **\_parent**: la URL se carga en el contexto padre o marco asociado.
  - + **\_top**: la URL se carga en el contexto más alto y ocupa toda la ventana.



## 1. Enlaces absolutos

Son los que se usan para poner una URL completa y deben incluir el protocolo `http://` acompañado del dominio. Por lo general, las URL pertenecen a otros servidores.

MEJOR VER EN:

```
<html>
  <head>
    <title>LM</title>
  </head>
  <body>
    <p>enlaces</p>
    <hr>
    <br>
    <a href="https://www.google.com" >ir a Google</a>
    <a href="https://www.google.com" target="_blank">ir a Google, abriendo la URL en otra ventana</a>
  </body>
</html>
```

Imagen 18. Ejemplo de enlaces absolutos en HTML.

## 2. Enlaces relativos

Se usan para acceder a URL alojadas en el mismo servidor, ya sea dentro o fuera del directorio donde se encuentra el enlace. La manera de actuar cuando quiere accederse a otra página dentro del mismo directorio es poner el nombre de la página junto a la extensión como valor del atributo `href`.

Si quiere accederse a una URL que está dentro de un subdirectorio, se tendrá que indicar el nombre de este junto a la URL.

Otro aspecto que puede darse es estar situado en un subdirectorio y querer acceder a una página del directorio padre. En ese caso, hay que poner dos puntos seguidos `..` por cada directorio que quiera subirse.

En el ejemplo siguiente, se muestran dos enlaces: uno situado en el mismo directorio, que accede a la página `autores.htm`, y otro situado dentro del directorio contenido, que accede al tema 1.

```
<html>
  <head>
    <title>LM</title>
  </head>
  <body>
    <p>enlaces</p>
    <hr>
    <a href = "autores.htm">ver los autores del libro LM</a>
    <br>
    <a href="contenido/tema 1.htm">acceder al tema 1</a>
  </body>
</html>
```

Imagen 19. Ejemplo de enlaces relativos en HTML.



### 3. Enlaces dentro de la misma página

Al disponer de páginas con mucho contenido, puede ser interesante crear un pequeño índice que acceda al contenido de ese apartado. En este punto, se verá la forma de actuar para poder crear enlaces dentro de la misma página.

Lo primero es crear los marcadores que serán las zonas de la página a las que pretende accederse. Para ello, se usa el atributo `name` de la etiqueta `<a>` con el valor del marcador que se desee.

El siguiente paso es añadir el enlace que lleve a un marcador. Para ello, dentro del atributo `href`, se pondrá el carácter almohadilla (`#`) seguido del nombre del marcador.

A continuación, se plasma una página con información sobre tres cultivos de plantas subtropicales. Se supone que los comentarios se sustituirán por abundante información sobre cada uno de ellos. En la parte superior, se ha creado un pequeño índice que accederá a cada una de las marcas creadas dentro del documento.

```
<html>
  <head>
    <title>LM</title>
  </head>
  <body>
    <p>enlaces</p>
    <hr>
    <a href="#Longan">ver el cultivo del longan</a><br>
    <a href="#Litchis">ver el cultivo del litchi</a> <br>
    <a href="#Carambole">ver el cultivo de la carambola</a><br>
    <h1>
      <a name="Longan"> cultivo del Longan </a>
    </h1>
    <!-- texto sobre el cultivo del longan-->
    <h1>
      <a name="Litchis">cultivo del litchi </a>
    </h1>
    <!-- texto sobre el cultivo del litchi-->
    <h1>
      <a name="Carambole">cultivo de la carambola </a>
    </h1>
    <!-- texto sobre el cultivo de la carambola-->
  </body>
</html>
```

Esto sería el Índice y donde se sitúan los enlaces

Esto serían los resultados del índice y a los que redirecciona el índice

Imagen 20. Ejemplo de enlaces dentro de la misma página.

### Etiquetas de imágenes

La etiqueta `<img>` se usa para añadir imágenes en una página y no tiene etiqueta de cierre. Por lo general, la imagen puede situarse en la misma carpeta donde se encuentra el documento HTML o situar todas las imágenes en una carpeta concreta y acceder a ella mediante rutas absolutas o relativas.

Las extensiones de las imágenes que pueden usarse son `.jpg`, `.bmp`, `.png`, `.gif`, etc. Por defecto, siempre aparecen en la parte superior izquierda de la página web.

Los atributos más importantes son los siguientes:

- > **src**: se añade el nombre de la imagen.
- > **width**: se establece el ancho de la imagen.
- > **height**: establece el alto de la imagen.
- > **alt**: se usa para dar una breve descripción de la imagen.



- > **align:** para la alineación de la imagen y puede tomar los valores:
  - » **top:** se sitúa en la parte superior del texto.
  - » **middle:** se sitúa centrada respecto al texto.
  - » **bottom:** se alineará con la parte inferior del texto.
  - » **left:** se alinea a la izquierda.
  - » **right:** se alinea a la derecha.
- > **border:** se le asigna un borde con valor numérico.

En este ejemplo se añaden tres imágenes de distinto tamaño:

```
<html>
  <head>
    <title>LM</title>
  </head>
  <body>
    <p>imágenes</p>
    <hr>
    
    
    
  </body>
</html>
```

Imagen 21. Ejemplo de etiquetas de imágenes.

A continuación, se muestra el uso de los atributos para el borde y la alineación:

```
<html>
  <head>
    <title>LM</title>
  </head>
  <body>
    <p>imágenes</p>
    <hr>
    
  </body>
</html>
```

Imagen 22. Ejemplo de bordes y alineación.

## Etiquetas de tablas

- > **Etiqueta <table>**: el uso de las tablas en páginas web es fundamental, pues ayuda a colocar de manera estructurada objetos dentro de ellas
- > **Etiqueta <caption>**: para añadir un título a la tabla.
- > **Etiqueta <th>**: celda de cabecera en la primera fila.
- > **Etiqueta <tr>**: va dentro de la etiqueta <table> y se usa para crear filas.
- > **Etiqueta <td>**: se sitúa anidada dentro de una etiqueta <tr> y se usa para asignar columnas o celdas dentro de una fila.
- > **Atributos:**
  - » **border:** se añade un borde a la tabla
  - » **cellpadding:** para separar el borde del contenido. El valor que se le asigna es en píxeles.
  - » **bgcolor:** puede añadirse a tablas, filas o columnas.





### > Atributos <td>:

- » **colspan =x**: combina columnas, siendo x el número de columnas por incluir.
- » **rowspan =x**: combina filas, siendo x el número de filas por incluir.
- » **Align**: alineación del contenido en horizontal con los valores ("left", "right" y "center").
- » **Valign**: alineación vertical del contenido con valores ("bottom", "top" y "middle"), siendo esta última la que se usa por defecto.

Ahora se muestra con el atributo cellpadding para separar el contenido del borde:

```
<html>
<head>
  <title>LM</title>
</head>
<body>
  <p>Tablas</p>
  <hr>
  <table border="5" height="250" width="250">
    <tr>
      <td valign="bottom">a</td>
      <td valign="top">b</td>
      <td valign="middle">c</td>
    </tr>
    <tr>
      <td align="right">1</td>
      <td align="center">2</td>
      <td>3</td>
    </tr>
    <tr>
      <td align="left">x</td>
      <td>y</td>
      <td align="right">z</td>
    </tr>
  </table>
</body>
</html>
```

Imagen 24. Ejemplo de tablas en HTML.

```
<html>
<head>
  <title>LM</title>
</head>
<body>
  <p>Tablas</p>
  <hr>
  <table>
    <tr>
      <td>a</td>
      <td>b</td>
      <td>c</td>
    </tr>
    <tr>
      <td>1</td>
      <td>2</td>
      <td>3</td>
    </tr>
    <tr>
      <td>x</td>
      <td>y</td>
      <td>z</td>
    </tr>
  </table>
  <!--con borde y combinando celdas -->
  <table border>
    <tr>
      <td colspan="3">abc</td>
    </tr>
    <tr>
      <td rowspan="2">1x</td>
      <td>2</td>
      <td>3</td>
    </tr>
    <tr>
      <td>y</td>
      <td>z</td>
    </tr>
  </table>
</body>
</html>
```

Imagen 23. Ejemplo de tablas en HTML.

## Formularios

Son uno de los elementos principales que se introducen en un documento HTML. Es la manera que tienen los usuarios de interactuar con la página web mediante una serie de elementos (botones, campos de texto, casillas de verificación, etc.), y su misión es la de obtener la información que los usuarios proporcionan a la hora de utilizar algunos de sus elementos.

Aunque, normalmente, al emplear formularios se trabaja en una arquitectura cliente servidor, donde los datos obtenidos a través del formulario se envían a un servidor para ser procesados y devolver en algunos casos la información tratada, puede que se usen de manera local, aunque es lo menos habitual.

Entre los formularios que pueden crearse, destacan:

- > Registro de usuarios.
- > Consultas.
- > Libro de visitas.
- > Pasarela de pago.
- > Motor de búsqueda.
- > Recordar contraseña.
- > Identificación de usuarios.



## 1. Estructura de los formularios

La estructura está compuesta por el elemento `<form>`, que actúa como contenedor de distintos controles que permiten interactuar con el usuario.

```
<html>
  <head>
    <title>LM</title>
  </head>
  <body>
    <form>
      <!-- controles de formulario-->
    </form>
  </body>
</html>
```

Imagen 25. Estructura de formularios en HTML.

- > **Elemento `<form>`:** tiene una serie de atributos que se detallan en el siguiente cuadro.
- > **Elemento `<input>`:** se usa para la creación de distintos tipos de controles que interactúan con el usuario. Dependiendo del valor del atributo `type`, aparecerá un control u otro.

Atributos	Descripción
<b>Action</b>	Indica la acción o URI del programa que procesa la información.
<b>Method</b>	Determina el método HTTP usado por el navegador y tiene dos posibles valores: <ul style="list-style-type: none"> <li>+ <b>Post:</b> los datos se incluyen en el cuerpo del formulario.</li> <li>+ <b>Get:</b> los datos se adjuntan en la URI después del carácter separador.</li> </ul>
<b>Enctype</b>	= "TEXT/PLAIN" indica que los datos se envían sin codificación.
<b>Target</b>	Es un apalabra clave que informa de dónde se mostrarán las respuestas una vez procesadas los datos del formulario. Algunos valores: <code>_blank</code> , <code>_parent</code> , <code>_top</code> , etc.

Cuadro 2. Atributos del elemento `<form>`.

## 2. Controles de texto

Permiten introducir información a través de unos campos de texto de una o varias líneas,

Los que se estudiarán en este apartado son:

- > **Texto de una sola línea:** aplicando el valor "text" al atributo `type` del elemento `<input>`, muestra un campo de texto que permite introducir una sola línea.
- > **Texto de múltiples líneas:** de características similares al anterior, pero permite añadir más líneas. Puede ser interesante usarlo cuando se pide una descripción, opinión o un campo que vaya a ocupar más caracteres de los esperados. Para añadir un control de este tipo, se usa el elemento `<textarea>`, junto al número de filas y columnas que deben aparecer en el control.



- > **Texto oculto:** aplicando el valor "password" al atributo type del elemento `<input>`, muestra un campo que sirve para ocultar información y de utilidad para añadir la contraseña.
- > **Subir archivos:** puede considerarse de esta categoría y se usa como examinador para subir archivos. Aplicando el valor "file" al atributo type del elemento `<input>`, muestra un campo que sirve para subir documentos al servidor.

```
<html>
  <head>
    <title>LM</title>
  </head>
  <body>
    <form action="">
      <p>
        Introduce tu nombre de usuario:
        <input type = "text" name = "nombre" size = "45" maxlength = "40" />
      </p>
      <br>
      <p>
        Introduce la contraseña:
        <input type="password" name="clave" size="45" maxlength="15" />
      </p>
      <br>
      <p>
        introduce un comentario:
        <textarea name="mensaje" rows="5" cols="50"></textarea>
      </p>
      <br>
      <p>
        subir curriculum
        <input type="file" name="fichero" size="60" />
      </p>
    </form>
  </body>
</html>
```

Imagen 26. Ejemplo de formulario en HTML.

### 3. Controles de selección, verificación y listas desplegables

Dependiendo del control que se use, podrá seleccionarse una o varias de las opciones que se presentan:

- > **Botón de opciones:** aplicando el valor "radio" al atributo type del elemento `<input>`, muestra un campo que sirve para seleccionar una de las opciones que aparecen en el control. Necesitan ser agrupados para que funcionen correctamente añadiendo el mismo valor en el atributo name. En el atributo value, se añade la opción seleccionada, que debe ser diferente en cada grupo. Mediante CHECKED, se selecciona uno de ellos.
- > **Control de selección o verificación:** aplicando el valor "checkbox" al atributo type del elemento `<input>`, muestra un campo que sirve para seleccionar una o varias opciones que aparecen en el control y, mediante CHECKED, se activa por defecto. El atributo value es opcional.
- > **Combo o listas desplegables:** es similar a los estudiados con anterioridad, pero con la peculiaridad de que, para ver las distintas opciones, hay que pulsar sobre el control y seleccionar solo una de las opciones que aparecen. Se usa el elemento `<select>` anidando cada valor que ha de elegirse dentro del elemento `<option>`.



```

<html>
  <head>
    <title>LM</title>
  </head>
  <body>
    <form action="">
      <p>
        <b>ejemplo de botón de opciones</b>
        <br>
        Seleccione un producto...<br>
        <input type="radio" name="productos" value="aguacaten">
        El aguacate de la Axarguia<br>
        <input type="radio" name="productos" value="platanon">
        El platano de Canarias<br>
        <input type="radio" name="productos" value="chirimollon">
        La chirimolla de Motril<br>
        <input type="radio" name="productos" value="longann">
        el longan del sur de China <br>
      </p>
      <br>
      <p>
        <b>Control de selección o verificación</b><br>
        Seleccione un producto favorito...<br>
        <input type="checkbox" name="a1" value = "aguacate">El Aguacate
        <br>
        <input type="checkbox" name="a2" value="platano n "> El platano<br>
        <input type="checkbox" name="a3" value="chirimollo"> El chirimollo<br>
        <input type="checkbox" name="a4" value = "longan" checked>El longan
      </p>
      <br>
      <p>
        <b>ejemplos de listas</b>
        <br>
        Seleccione una opción...<br>
        <select name="Color">
          <option>aguacate</option>
          <option>platano</option>
          <option>chirimollo</option>
          <option>Longan</option>
        </select>
      </p>
    </form>
  </body>
</html>

```

Imagen 27. Ejemplos de controles de selección.

#### 4. Controles de botones

Gracias a este tipo de controles, el usuario puede interactuar con el servidor, ya que, al presionarlo, se ejecuta un evento o acción. En este apartado, se verán los controles de botones más usados:

- > **Botón de envío:** es uno de los controles que no suele faltar a la hora de crear un formulario. Aplicando el valor "submit" al atributo type del elemento <input>, muestra un botón que, al pulsarlo, envía el formulario. Otro atributo que hay que cumplimentar es value, cuyo valor será el texto del botón.
- > **Botón de reinicio o borrado:** aplicando el valor "reset" al atributo type del elemento <input>, muestra un botón que, al pulsarlo, "limpia o restaura a valor inicial" los valores de los campos que hay en el formulario.



## 2.2.5. HTML5

El objetivo del consorcio W3C con la quinta versión de este lenguaje es modernizar las páginas web añadiendo nuevos elementos y eliminando aquellos que resultan obsoletos.

La revisión cuarta de HTML no podía, entre otras cosas, validar ciertos tipos de datos en formularios sin utilizar JavaScript, tampoco permitía arrastrar objetos y soltar, no tenía etiquetas para la web semántica o web 3.0, etc.

Básicamente, las mejoras de HTML5 se pueden agrupar en cuatro grupos:

- > Elementos multimedia, como <audio> y <video>.
- > Elementos semánticos, como <header>, <footer>, etc.
- > Nuevos atributos en los elementos de los formularios como email, number, date, time, range o calendar.
- > Elementos gráficos, como <canvas>.

A continuación, se muestra el aspecto de una sencilla página web en HTML5:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Editorial Sintesis</title>
  </head>
  <body>
    <h1>HTML5</h1>
    <p>Introducción a HTML5.</p>
  </body>
</html>
```

Imagen 28. Página web sencilla en HTML.

En HTML4 tendría el siguiente aspecto:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd ">
```

Imagen 29. HTML4.

Como se puede observar, la definición del tipo de página (Doc-type) es mucho más abreviado en HTML5.

También se puede observar en la siguiente línea, donde se especifica la codificación del documento "UTF-8":

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

Imagen 30. Codificación del documento en HTML.

Como en el caso anterior, la definición de la codificación se simplifica, puesto que en HTML4 sería la siguiente:

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

Imagen 31. Antigua codificación.



### 2.2.6. Las nuevas etiquetas HTML5

A continuación, se enumeran algunas de las nuevas etiquetas que añade HTML5:

- > **<article>**. Permite definir un artículo (contenido independiente) dentro de un documento.
- > **<aside>**. Permite definir contenido fuera del contenido de la página.
- > **<audio>**. Permite reproducir audio en la página web.
- > **<canvas>**. Lienzo en el que se pueden dibujar gráficos mientras que se está ejecutando la página utilizando JavaScript.
- > **<datalist>**. Especifica una lista de opciones predefinidas para los elementos de entrada de información.
- > **<details>**. Define detalles adicionales que el usuario puede ver u ocultar.
- > **<dialog>**. Define un cuadro o ventana de diálogo.
- > **<embed>**. Define un contenedor para una aplicación externa no HTML.
- > **<figure>**. Define contenido autocontenido.
- > **<footer>**. Permite incorporar un pie a un documento (o sección).
- > **<header>**. Permite incorporar una cabecera a un documento (o sección).
- > **<mark>**. Permite marcar o destacar texto en el documento.
- > **<meter>**. Define una medida dentro de un rango conocido.
- > **<nav>**. Define enlaces de navegación.
- > **<output>**. Define el resultado de un cálculo.
- > **<progress>**. Permite representar el progreso de una tarea.
- > **<section>**. Define una sección dentro de un documento.
- > **<source>**. Define recursos para elementos multimedia, como <video> y <audio>.
- > **<time>**. Define una fecha u hora.
- > **<video>**. Permite visualizar un vídeo.

A continuación, se mostrarán algunos ejemplos de utilización de las etiquetas HTML5 más frecuentes.

#### Elementos semánticos

Como se ha explicado anteriormente, HTML5 ofrece diferentes elementos semánticos para definir las partes de una página web.





## Formularios

Los formularios en HTML5 han sufrido una importante mejora, permitiendo validaciones sin tener que utilizar JavaScript, especificar listas predefinidas de valores de entrada, definir campos calculados, etc. A continuación, se muestran algunos ejemplos de nuevas etiquetas en HTML5.

### 1. Ejemplo de utilización de la etiqueta <output>

La etiqueta <output> en los formularios permite calcular valores dentro de estos y almacenar el resultado en otro campo. En el siguiente ejemplo se utilizan los campos z y x para almacenar el valor del radio y el valor de la longitud de la circunferencia.

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Longitud de la circunferencia</h1>
    <h3>Mueva el indicador para conocer cómo varia la longitud de to circunferencia con respecto al radio.</h3>
    <form oninput = "x.value=parseFloat(a.value * 2 * 3.1416);z.value=parseFloat(a.value)">
      Longitud del radio:0 <input type="range" id="a" name="a" value="0" > 100
      <br><br>
      Valor del radio: <output name="z" for="a"> </output>
      <br><br>
      Longitud de la circunferencia: <output name="x" for="a"> </output>
    </form>
  </body>
</html>
```

Imagen 32. Ejemplo de etiqueta <output>.

Se puede observar que en el campo z de tipo output se almacena el valor del radio y en el campo x, también de tipo output, se almacenará la longitud de la circunferencia ya calculada dependiendo de la longitud del radio. Los cálculos se realizarán dentro del parámetro oninput de la etiqueta <form>.

Si se ejecuta el ejemplo anterior, se observará que los valores del radio y de la longitud de la circunferencia irán variando, dependiendo de la posición en que se encuentre el calibrador, que podrá desplazarse entre los valores 0 y 100.

### 2. Ejemplo de utilización de la etiqueta <datalist>

El tipo de elemento datalist era una demanda de hace tiempo de los desarrolladores web, puesto que en muchos formularios hay que ofrecer al usuario una lista de valores predefinidos para que se elija finalmente uno de ellos. Antes se hacía con JavaScript, pero ahora, con HTML5, es posible, de una forma más sencilla, definir los valores predefinidos de entrada.

```
<!DOCTYPE html>
<html>
  <body>
    <form>
      <input list="frutas" name="frutas">
      <datalist id="frutas">
        <option value="Mango">
        <option value="Aguacate">
        <option value="Tamarillo">
        <option value="Pifia">
        <option value="Litchi">
        <option value="Rambután">
        <option value="Maracuyá">
      </datalist>
    </form>
  </body>
</html>
```

Imagen 33. Ejemplo de etiqueta <datalist>.



Si se ejecuta el ejemplo anterior, se observará que en el campo frutas se ofrecen una serie de valores predeterminados que podrán ser utilizados por el usuario. La carga de valores ya no es necesaria realizarla mediante JavaScript, sino que se define directamente en las etiquetas HTML.

## Elementos multimedia

Dentro de las mejoras que ofrece HTML5 están los elementos multimedia, como pueden ser las etiquetas <audio> y <video>. Antes había que utilizar un reproductor externo, con lo cual no existía una normalización y aspecto o utilización estándar. Ahora, estas nuevas etiquetas permiten a los desarrolladores incluir audio, vídeo y otros elementos multimedia directamente en sus páginas web.

A continuación, se muestra un ejemplo de utilización de la etiqueta <audio>:

```
<!DOCTYPE html>
<html>
  <body>
    <audio src="http://icarosproject.com/audio/mundodigital.mp3" controls="true" loop="true"autoplay="true"></audio>
  </body>
</html>
```

Imagen 34. Ejemplo de etiqueta <audio>.

Como se puede observar, en la etiqueta <audio> se han predefinido como verdaderos (true) varios atributos, como son:

- > **Controls.** Para poder incluir en la página la manera de parar la reproducción del sonido si no quiere ser escuchado por el usuario.
- > **Loop.** El audio comenzará a reproducirse de nuevo cuando termine.
- > **Autoplay.** El audio comienza a reproducirse una vez que se carga la página.

## Elementos gráficos

HTML5 ofrece la posibilidad de utilizar elementos gráficos mediante la nueva etiqueta <canvas>. Aunque esta etiqueta solamente proporciona un lienzo (sin borde, salvo que se especifique, y sin contenido) para poder dibujar sobre él mediante scripts (JavaScript), es posible dibujar gráficos como círculos, líneas, etc.

Un ejemplo de definición de un lienzo 300 x 300 en una página web sería el siguiente:

```
<canvas id="miLienzo" width="300" height="300"></canvas>
```

Imagen 35. Ejemplo de etiqueta <canvas>.

En el ejemplo anterior se define un lienzo de anchura (width) 300 y altura (height) 300 con el identificador (id) miLienzo.



En el caso de que se desee un lienzo delimitado por un borde, se puede definir dentro del atributo style el grosor, el color y el estilo del borde. Un ejemplo sería el siguiente:

```
<canvas id="miLienzo" width = "300" height = "300" style="border:3px dotted #000000;">
</canvas>
```

Imagen 36. Ejemplo etiqueta <canvas>.

En el ejemplo anterior se define un lienzo cuyo borde tiene de anchura 3 píxeles (3px), punteado (dotted) en vez de ser una línea normal (solid) y de color negro (#000000).

Este código HTML de lienzo insertado dentro de una página web tendría el siguiente aspecto:

```
<!DOCTYPE html>
<html>
  <body>
    <canvas id="miLienzo" width = "300" height="300" style = "border:3px dotted #000000;" > </canvas>
  </body>
</html>
```

Imagen 37. Ejemplo de etiqueta <canvas>.

A continuación, se muestra un ejemplo de cómo se puede escribir mediante JavaScript un texto dentro del lienzo anteriormente definido:

```
<!DOCTYPE html>
<html>
  <body>
    <canvas id="miLienzo" width = "300" height="300" style = "border:3px dotted #000000;" > </canvas>
    <script>
      var x = document.getElementById("miLienzo");
      var elem = x.getContext("2d");
      elem.font = "35px Arial";
      elem.fillText("Editorial Síntesis",15,150);
    </script>
  </body>
</html>
```

Imagen 38. Ejemplo de etiqueta <canvas>.

Como se puede observar, en la página web hay un script delimitado mediante etiquetas <script>, en el cual se referencia al lienzo con identificador "miLienzo" y dentro de él se crea un elemento de tipo texto con una fuente Arial de 35 píxeles. El texto se colocará en la posición 15,150 (ancho, alto) el lienzo y contendrá el literal "Editorial Síntesis".

Igualmente, aunque no es objeto de este libro que el lector aprenda JavaScript, en el código anterior se puede apreciar la versatilidad y posibilidades de este nuevo elemento HTML5.



# 2.3.

## Hojas de estilo CSS

En este apartado, se estudiarán los aspectos fundamentales para poder conocer y trabajar las hojas de estilo en cascada (cascading style sheets) o CSS, un lenguaje creado por W3C que añade diferentes estilos a documentos HTML o XHTML, dotándolos de formato.

Estas hojas de estilo se crearon para poder separar el contenido de un documento de su aspecto, lo que permite tener documentos mejor estructurados y limpios. Para separar el contenido de la presentación, se usa una serie de reglas sobre elementos que hay que declarar previamente, ya sea en la misma página o en un documento externo.

Las hojas de estilo pueden aplicarse a un documento entero, a una parte o, simplemente, a una etiqueta concreta, ya que pueden definirse varios estilos para una misma etiqueta. Permiten modificar la presentación de los elementos de un documento HTML sin tocar su código. Ahorran tiempo a la hora de definir estilos y ofrecen un abanico extenso de herramientas para el formato, más amplio que HTML.

Para poder comprender mejor el funcionamiento, puede observarse el ejemplo siguiente, donde se crean dos documentos independientes.

Primero, se aprecia la página web con código HTML:

```
<html>
  <head>
    <title>LM</title>
  </head>
  <body>
    <h1>
      Principales cultivos subtropicales</h1>
    <ul>
      <li>Longan</li>
      <li>Aguacate</li>
      <li>Chirimollo</li>
      <li>Litchi</li>
    </ul>
  </body>
</html>
```

Imagen 39. Página web en HTML sin CSS.

La visualización en un navegador muestra una cabecera y una lista desordenada con elementos subtropicales, sin ningún tipo de estilo o formato.

Al documento HTML del ejemplo, se le añade en la cabecera la siguiente línea, que vincula el fichero CSS creado:

```
<head>
  <title>LM</title>
  <link rel="stylesheet" type="text/css" href="hoja_estilo.css"/>
</head>
```

Imagen 40. Introducción de CSS en la cabecera.

Para aplicar estilos, se crea el siguiente fichero CSS, donde al cuerpo se le proporciona un color de fondo azul; al elemento h1, tipo de fuente verdana y color blanco, subrayando los elementos de la lista.

Fichero de hoja de estilos (hoja\_estilo.css) sería el siguiente:

```
body{background-color:Blue;}
h1{font-family:Verdana; color:White;}
li{text-decoration:underline}
```

Imagen 41. Hoja de estilos.



### 2.3.1. Versiones de CSS

El W3C es el encargado de mejorar cada una de las versiones que van saliendo al mercado, dotándolas de más funcionalidad y corrigiendo algunos problemas que pudieran tener versiones antiguas.

- > **CSS 1:** a finales de 1996, se publica la primera versión que permite describir formato, alineación y atributos de texto, así como las propiedades de las fuentes (tamaño, color, tipo, etc.). Actualmente, no se usa.
- > **CSS 2:** incluye casi la totalidad de CSS 1. A mediados de 1998, se publica la segunda recomendación oficial de W3C. Añade como mejora para las capas el tipo de posicionamiento y nivel.
- > **CSS 2.1:** elimina errores y funcionalidades en desuso de versiones anteriores.
- > **CSS 3:** usa módulos que añaden nueva funcionalidad a CSS 2, preservando de esa manera la compatibilidad con versiones anteriores. Surge a finales del 2011.

### 2.3.2. Sintaxis básica

En este apartado, se verán los aspectos más importantes que han de tenerse en cuenta para poder trabajar con hojas de estilo en cascada.

#### Reglas CSS

Las reglas son los pilares fundamentales a la hora de crear estilos a un documento. En este apartado, se describirá cada uno de sus elementos tratando un ejemplo concreto.

- > **Regla.** Es una declaración sobre un estilo de uno o varios elementos. Está compuesta por dos partes diferenciadas: un selector y una declaración.
- > **Selector.** Pueden ser uno o varios los elementos a los que se aplica la regla CSS. Dependiendo de la declaración que se añada, afectará a uno o a varios elementos del documento creado. En este ejemplo, el selector es h2 y a todos los elementos h2 del documento HTML o XHTML les afectará la declaración realizada. Para definir el estilo de un selector, se escribe la etiqueta seguida de la lista de declaraciones encerradas entre llaves.

Un selector puede tener varias declaraciones separadas por el carácter punto y coma (;).

- > **Declaración.** Está compuesta por dos partes diferenciadas, propiedad y valor, separadas por el carácter dos puntos (:), es donde se especifica el estilo del selector o elemento y puede contener varias propiedades en la misma declaración.

Así, la declaración es color:#0099FE.

Propiedad		Valor	
h2	{	color	: #0099FF; }
Selector		Declaración	

Imagen 42. Regla CSS.



- > **Propiedad.** Permite modificar el aspecto de una característica de un selector. Ejemplos de propiedades pueden aplicarse sobre: color, fuente, márgenes, texto, clasificación, etc.
- > **Valor.** Indica la asignación que se hace a la propiedad. Por ejemplo, a la propiedad color, pueden aplicársele valores de la paleta de Windows (negro, blanco, azul, etc.).

## Selectores

Tal y como se ha mencionado con anterioridad, cualquier elemento HTML puede ser un selector.

- > De tipo
- > Universal
- > Descendiente
- > De clase
- > Por ID
- > Hijo
- > Adyacente

### 1. Selector de tipo

Se aplica a los elementos especificados de una página. Hay que añadir el nombre del elemento seguido de una declaración.

```
body{
    background-color:Blue;
}
```

Imagen 44. Selector de tipo.

### 2. Selector universal

Utiliza el carácter asterisco (\*) y se aplica la regla a todos los elementos de la página web. No suele usarse con frecuencia.

```
*{
    declaración
}
```

Imagen 45. Selector universal.

### 3. Selector descendiente

Se usa cuando un elemento es descendiente de otro, es decir, se encuentra dentro de otro (entre la etiqueta de apertura y cierre).

No hace falta que sea descendiente directo y no afectará al resto de elementos de la página siempre y cuando no descendan del especificado.

Tal y como se aprecia a continuación, se ha creado un selector descendiente (h3 i) que afecta a los elementos i que se encuentren dentro de un elemento de cabecera h3 aplicándoles el formato: color de texto rojo, tipo de fuente Verdana y texto subrayado.

```
<html>
  <head>
    <title>LM</title>
    <style type="text/css">
      h3 i { color: red;
              font-family:Verdana;
              text-decoration: underline;
            }
    </style>
  </head>
  <body>
    <h3> Cultivo del <b>Litchi</b> en la zona de la <i>Axarquía</i></h3>
    <h3> >Cultivo del <b><i>longan</i></b> en Canarias</h3>
    <h3> >Cultivo de <b>la carambola</b> en Málaga</h3>
    <h3> >Cultivo del <i>canisté</i> en Canarias</h3>
    <h4> >Cultivo de la <i>guanábana</i> en Canarias</h4>
  </body>
</html>
```

Imagen 46. Selector descendiente.





#### 4. Selector de clase

Se usa para aplicar estilos a un elemento en concreto del documento y puede definirse dentro de la etiqueta `<style>`, dentro del documento, así como en un fichero externo.

El nombre que se asigna a un selector de clase puede ser el que se desee, aunque suele darse uno que identifique a la regla que quiere crearse y que empiece por el carácter punto (.). Existen dos maneras de definir un selector de clase;

- > `NombreSelector{ declaración } .`
- > `Elemento.NombreSelector{ declaración}.`

Para poder usar el selector en un elemento, se usa el atributo `class` de las etiquetas HTML a las que quiere aplicarse el estilo. La nueva regla creada podrá aplicarse a todas las etiquetas HTML del documento que use el atributo `class` seguido del valor, que será el nombre del selector o clase sin el punto.

```
<etiqueta class="selector de clase">
```

Imagen 47. Estructura de un selector de clase.

Para poder ver el funcionamiento de un selector de clase, se muestra el ejemplo siguiente, donde se declaran dos selectores de clase (`colorRojo` y `Color_fondo_azul`). El primero de ellos pondrá el texto de color rojo y tipo de fuente verdana. El segundo cambiará el fondo del elemento a color azul. Estos selectores solo afectarán a los elementos del documento que añadan el atributo `class`.

```
<html>
<head>
  <title>LM</title>
  <style type="text/css">
    .colorRojo
    {
      color: red;
      font-family: Verdana;
    }
    .Color_fondo_azul
    {
      background-color: Blue;
    }
  </style>
</head>
<body>
  <h3 class="colorRojo">Cultivo del Litchi.</h3>
  <h3>Cultivo del longan en Canarias.</h3>
  <h3>Cultivo de la <b class="Color_fondo_azul">carambola.</b> </h3>
  <h3>Cultivo del <i class="colorRojo">canisté</i>.</h3>
  <h4 class="Color_fondo_azul"> Cultivo de la guanábana.</h4>
</body>
</html>
```

Imagen 48. Ejemplo de selectores de clase.

En este apartado se han visto dos maneras de definir un selector. La segunda (`elemento.NombreSelector{ declaración}`) se interpreta como los elementos de tipo X que contengan el atributo `class` con el valor igual al nombre del selector.

En el ejemplo siguiente, el selector (`b.colorRojo`) afectará a los elementos `<b>` que contengan el atributo `class` con valor "colorRojo".



## 5. Selector por ID

Similar a los selectores de clase, pero siendo la almohadilla (#) el carácter que precede al selector en vez del punto. Se usa para aplicar un estilo a un elemento concreto del documento mediante el valor del atributo id de los elementos, que es único.

En el ejemplo, el selector (#colorRojo) afecta a la primera cabecera <h3>, ya que el atributo id tiene el valor del nombre del selector sin el carácter almohadilla.

```
<html>
  <head>
    <title>LM</title>
    <style type="text/css">
      #colorRojo
      {
        color: red;
      }
    </style>
  </head>
  <body>
    <h3 id="colorRojo">Cultivo del Litchi.</h3>
    <h3>Cultivo del longan en Canarias.</h3>
    <h3>Cultivo de la <b>carambola.</b></h3>
    <h3> Cultivo del canisté.</h3>
    <h4>Cultivo de la guanábana.</h4>
  </body>
</html>
```

Imagen 49. Ejemplo de selector por ID.

## 6. Selector hijo

Similar al descendiente, pero, esta vez, solo afecta a hijos directos de los elementos. Para ello, se usa el signo mayor que (>).

En el siguiente ejemplo, se presenta un selector descendiente (h3 i) y un selector descendiente (h3 i). El primero de ellos pondrá de color rojo el texto de las etiquetas que sean hijas directas de la cabecera h3, mientras que, en el segundo, al ser adyacente, no hace falta que sean hijas directas de h3, aplicando un estilo al texto (subrayado y aumentado de tamaño).

```
<html>
  <head>
    <title>LM</title>
    <style type="text/css">
      h3 > i {
        color: red;
      }
      h3 i {
        text-decoration: underline;
        font-size: xx-large;
      }
    </style>
  </head>
  <body>
    <h3> Cultivo del <b>Litchi</b> en la zona de la <i>Axarquía</i></h3>
    <h3> Cultivo del <b><i>longan</i></b> en Canarias</h3>
    <h3> Cultivo de <b>la carambola</b> en Málaga</h3>
    <h3> Cultivo del <i>canisté</i> en Canarias</h3>
    <h4> Cultivo de la <i>guanábana</i> en Canarias</h4>
  </body>
</html>
```

Imagen 50. Selector hijo.



## 7. Selector adyacente

Se usa para elementos que están seguidos de otros y son hermanos (el elemento padre de ambos es el mismo). El signo (+) se emplea para la declaración entre los elementos.

En el siguiente ejemplo, el selector adyacente es (h1 + h3), indica que se aplicará el estilo de color de texto rojo a los elementos de cabecera h3 que sean hermanos de h1 y aparece justo después de él. Hay que destacar que h1 y h3 son hermanos, ya que el padre de ambos es el elemento <body>.

```
<html>
  <head>
    <title>LM</title>
    <style type="text/css">
      h1 + h3 {
        color: red;
      }
    </style>
  </head>
  <body>
    <h1> Cultivo del <b>Litchi</b> en la zona de la <i>Axarquía</i></h1>
    <h3> Cultivo del <b><i>longan</i></b> en Málaga</h3>
    <h3> Cultivo del <b>platano</b> en Canarias</h3>
    <h1> Cultivo de <b>la carambola</b> en Málaga</h1>
    <h2> Cultivo del <i>caniste</i> en Canarias</h2>
    <h3> Cultivo de la <i>guanábana<a> en Canarias</h3>
  </body>
</html>
```

Imagen 51. Selector adyacente.

## Propiedades y valores

Un selector puede contener una o varias propiedades. En los siguientes cuadros se describen las relacionadas con el texto, el tipo de letra, el posicionamiento, etc., detallando algunos valores de sobre ellas junto a su descripción.

Propiedad	Valores	Descripción
Text-decoration	None, blink, line-through, overline, underline	Decora el texto
Color	Los especificados (RGB o nombre color)	Color del texto
Text-align	Center, justify, left, right	Alineación del texto
Direction	Ltr, rtl	Dirección del texto
Text-transform	None, capitalize, lowercase, uppercase	Convierte mayúsculas, minúsculas, etc

Cuadro 3. Propiedades de texto.



Propiedad	Valores	Descripción
<b>Font-size</b>	xx-small, x-small, small, medium, large, x-large, xx-large	Tamaño
<b>Font-family</b>		Tipo de letras (fuente)
<b>Font-style</b>	Normal, italic, oblique	Estilo fuente

Cuadro 4. Propiedades de tipo de letra.

Propiedad	Valores	Descripción
<b>Background-color</b>	Transparent, color	Color de fondo
<b>Background-image</b>	None, uri	Imagen de fondo

Cuadro 5. Propiedades de fondos.

Propiedad	Valores	Descripción
<b>Float</b>	None, left, right	Posicionamiento flotante
<b>Display</b>	None, block,	Tipo de caja
<b>Position</b>	Absolute, fixed, relative, static	Modo de posicionamiento
<b>z-index</b>	Auto, número - entero	Apilamiento

Cuadro 6. Propiedades de posicionamiento.

Propiedad	Valores	Descripción
<b>height</b>	Auto, distancia, porcentaje	Altura
<b>width</b>	Auto, distancia, porcentaje	Anchura

Cuadro 7. Propiedades de tamaño.

## Comentarios, agrupamiento y herencia

### 1. Comentarios

Los comentarios son fundamentales en todo lenguaje de programación que proporcionan información de utilidad a la hora de estructurar o indicar cualquier apreciación sobre el contenido. Pueden ocupar una o varias líneas, dependiendo de lo que se quiera añadir. En el caso de CSS, al igual que los comentarios en HTML, no son visibles por el navegador. Todo comentario va entre los caracteres de inicio y fin:

- > Los caracteres de inicio del comentario son la barra y el asterisco (/).
- > Los caracteres de fin de comentario son el asterisco y la barra (\*).

```
<style type="text/css">
  h1 + h3 {
    color: red;
    /*pone de color rojo el texto*/
  }
</style>
</head>
```

Imagen 52. Ejemplo de comentarios.



## 2. Agrupamientos

Cuando pretende aplicarse la misma declaración a distintos selectores, la manera de hacerlo es agrupando los selectores separados por el carácter coma (,).

Las reglas de estilo de un documento son las tres que se aprecian cuyos selectores (h1, h2 y h3) ponen de color rojo el texto del elemento.

Existe una manera de agrupar las tres reglas en una, con su declaración correspondiente.

Consiste en poner cada selector separado por el carácter coma (,) seguido de la declaración, que, en este caso, es idéntica en las tres reglas.

```
<style type="text/css">
  h1{color: red;}
  h2{color: red;}
  h3{color: red;}
</style>
```

Imagen 53. Ejemplo de agrupamientos.

```
h1,h2,h3{
  color: red;
}
```

Imagen 54. Ejemplo de agrupamiento.

## 3. Herencia

Al tener un documento HTML una estructura de árbol, los sectores anidados heredan las declaraciones de los padres. Estos se ven perfectamente con la etiqueta <body>, ya que, si se el aplica un estilo, afectará a todos los elementos del documento, debido a que es el padre del resto de etiquetas que se añadan.

En el caso de que el usuario modifique un elemento concreto mediante un selector de clase, ID u otros, prevalecerá dicho estilo en ese elemento.

A continuación, se muestra un ejemplo:

```
<html>
  <head>
    <title>LM</title>
    <style type="text/css">
      h1{color: red;}
      body{color:Blue;}
    </style>
  </head>
  <body>
    <h1> Cultivo del <b>Litchi</b> en la zona de la <i>Axarguia</i></h1>
    <h3> Cultivo del <b><i>longan</i></b> en Málaga</h3>
    <h3> Cultivo del <b>platano</b> en Canarias</h3>
    <h1> Cultivo de <b>la carambola</b> en Málaga</h1>
    <h2> Cultivo del <i>canisté</i> en Canarias</h2>
    <h3> Cultivo de la <i>guanábana</i> en Canarias</h3>
  </body>
</html>
```

Imagen 55. Ejemplo de herencia.

## Unidades de medida

Gracias a CSS, se consigue mayor precisión a la hora de ubicar en una zona del documento un elemento concreto y definir distancias y visibilidad de los elementos, contando con un mayor número de unidades de medida absolutas y relativas que se establecen con un valor entero o decimal que ayudan a definir los márgenes, anchura, etc., de los distintos elementos.

Puntos	Pulgadas	Centímetros	Píxeles	Milímetros
Pt	In	Cm	Px	mm

Cuadro 8. Principales unidades de medida absolutas.

Si el texto del cuerpo de la página tiene un tamaño de letra de 8 puntos, se mostraría así:

```
body{
  font-size: 8pt;
}
```

Imagen 56. Ejemplo de unidades absolutas.



### 2.3.3. Formas de aplicar estilos a un documento HTML

En este apartado, van a explicarse las distintas maneras existentes de aplicar estilos a un documento HTML.

#### Estilos aplicados en un documento externo

Se creará un documento de hojas de estilo, cuya extensión será .css, donde se declararán las distintas reglas de la página web. Para poder vincular el archivo externo .css al documento HTML, se usa la etiqueta <link> con una serie de atributos.

Este procedimiento es mucho más elegante, ya que los distintos estilos irán añadiéndose al fichero CSS que se haya creado, ahorrando tiempo y líneas HTML y dejando un código mucho más entendible, evitando tener que definir estilos de manera individual a todas las etiquetas que se tienen en HTML.

El fichero CSS que tiene que crearse será de texto plano, donde se añadirán las reglas con sintaxis CSS y se enlazará con el documento HTML mediante la etiqueta <link>, que se añade dentro de la etiqueta <head>, y una serie de atributos que se detallan a continuación.

La etiqueta <link> corresponde al tipo de etiquetas vacía al no tener contenido. Se usa para relacionar un recurso externo con el documento HTML, aunque se emplea para incluir el fichero de hojas de estilo externo, puede tener más usos.

Entre sus atributos, destacan:

- > **type:** informa del tipo de contenido del fichero por enlazar. Para el uso de hojas de estilo, el valor que ha de añadirse es "text/css", que informa de que el archivo es de texto y su sintaxis es CSS.
- > **href:** URL o ruta donde se localiza el nombre del documento CSS. Es la fuente del fichero que quiere enlazarse, que puede tener una ruta absoluta o relativa.
- > **rel:** informa de la relación que existe entre el documento externo y el actual. Para el caso particular de hojas de estilo, su valor es "stylesheet", que indica que el fichero por enlazar es una hoja de estilos que aplicará el formato al documento HTML.

#### Etiqueta <link>:

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

Imagen 57. Etiqueta <link>.





### Documento HTML:

```
<html>
  <head>
    <title>LM</title>
    <link rel="stylesheet" type="text/css" hrefhoja_estilo.css" 1>
  </head>
  <body>
    <h1>Estilos aplicados a una tabla</h1>
    <table border="1">
      <tr>
        <td>floración</td>
      </tr>
      <tr>
        <td>fruto</td>
      </tr>
    </table>
  </body>
</html>
```

Imagen 58. Documento HTML con etiqueta <link>.

### Fichero de hoja de estilos (miestilo.css):

```
td {
  background-color:Green;
  font-family:Arial Black;
}
```

Imagen 59. Hoja de estilos.

### Estilos aplicados en el mismo documento

El resultado que se obtuvo en el apartado anterior puede lograrse, pero esta vez unificando todo en un único documento, distinguiendo claramente la zona de aplicación de estilos que serán efectivos para los elementos de la página web que se hayan declarado.

Para ello, se usa la etiqueta <style> dentro de la etiqueta <head>. El contenido de la etiqueta <style> estará compuesto por una serie de reglas CSS que afectarán a todo el documento.

Cualquier cambio de formato que quiera hacerse sobre uno de los elementos de la página web tendrá que realizarse en dicho apartado.

El documento HTML que se observa lleva insertados los estilos en la cabecera sin necesidad de crear un documento externo para enlazarlos. Cuando se trabaja con pocas reglas, el documento queda elegante y legible, pero si, peor el contrario el contenido de la etiqueta <style> es muy extenso, es recomendable ponerlo en un documento externo.



## Estilos aplicados a las etiquetas de un documento

Es conveniente usarlos cuando pretenden tenerse dos elementos iguales, pero hay que aplicarles un formato distinto para cada uno.

Un ejemplo podría ser tener dos etiquetas <h3> y que la primera se muestre con un color de texto gris y la segunda, azul. Por otro lado, a cada una de las celdas de la tabla que se muestra, se le pondrá un color de fondo diferente. Esto no sería posible tal y como se ha visto en los apartados anteriores. Para ello, se usa el atributo style en cada una de las etiquetas que quiera ponerse.

```
<html>
  <head>
    <title>LM</title>
  </head>
  <body>
    <h3 style="color: Gray;">
      aplica estilo a la etiqueta h1 y pone el texto de color gris
    </h3>
    <h3 style="color: blue;">
      aplica estilo a la etiqueta h1 y pone el texto de color azul
    </h3>
    <table border="1">
      <tr>
        <td style="background-color:Yellow;">
          floración
        </td>
      </tr>
      <tr>
        <td style="background-color:Olive;">
          fruto
        </td>
      </tr>
    </table>
  </body>
</html>
```

Imagen 60. Ejemplo de aplicación de estilos en etiquetas.

## 2.3.4. CSS3

En este apartado se detallan las nuevas características y propiedades de la versión 3 de CSS, que es un lenguaje que dota de nuevos estilos y apariencia a los elementos de un documento como HTML, XML, etc. Permite, entre otros aspectos, añadir sombras de cajas, transiciones, poner esquinas redondeadas, usar gradientes lineales y radiales, insertar imágenes en los bordes, opacidad y transparencia, etc., ahorrando bastante tiempo de trabajo y mejorando el rendimiento.

CSS3 presenta nuevos selectores de atributos, como son:

- > **[atributo^="valor"]** → selecciona los elementos que tengan un atributo que empiece por el valor introducido
- > **[atributo\$= "valor"]** → selecciona los elementos que tengan un atributo que termine por el valor introducido.
- > **[atributos\*="valor"]** → selecciona los elementos que tengan un atributo que contengan el valor introducido.

Además, define pseudoclases (:nth-child(),:empty, :not, :invalid, :optional, etc.).



Igualmente, CSS3 presenta una serie de nuevas propiedades que permiten mayor control sobre los elementos entre las que destacan las siguientes:

- > `background`
- > `border-image`
- > `border-radius`
- > `box-shadow`
- > `box-sizing`
- > `DataURI`
- > `FlexBox`
- > `Generated Content`
- > `hsla()`
- > `Media Queries`
- > `Multiple backgrounds`
- > `opacity`
- > `overflow-scrolling`
- > `pointer-events`
- > `rgba()`
- > `text-shadow`
- > `CSS Reflections`
- > `CSS Transitions`
- > `CSS 2D Transforms`
- > `CSS 3D Transforms`
- > `CSS Animations`
- > `CSS Columns`
- > `CSS Gradients`

### 2.3.5. Las nuevas propiedades de CSS3

En este apartado se detallarán las propiedades más importantes de CSS3 junto a algunos ejemplos para una mayor comprensión.

#### Opacidad o transparencia

Se ha añadido un nuevo canal llamado Alpha al modo RGB, que toma como valores posibles los del rango [0.1], donde el valor 0 se interpreta como completamente transparente y el 1 corresponde a completamente opaco.

#### Tono, saturación y luminosidad

Los valores HSL y HSLA son otra manera de definir los colores mediante tono, saturación y luminosidad o brillo. La aplicación es similar al RGBA con los siguientes rangos de valores:

- > **Tono:** [0..360].
- > **Saturación:** valor porcentual.
- > **Luminosidad:** valor porcentual.

La sintaxis es la siguiente:

- > `hsl(tono, saturación y luminosidad)`
- > `hsla(tono, saturación y luminosidad, canal Alpha).`



## Gradientes

Ayuda a crear gradientes de varios colores de manera lineal o radial. Para la propiedad `background-image` se aportan nuevos valores de gradientes, como son:

- > **`linear-gradient()`**: degradado de manera lineal con al menos dos colores de arriba hacia abajo.
- > **`radial-gradient()`**: degradado radial con al menos dos colores del centro a los extremos.
- > **`repeating-linear-gradient()`**: repite un gradiente lineal.
- > **`repeating-radial-gradient()`**: repite un gradiente radial.

## Esquinas redondeadas

La propiedad `border-radius` se usa cuando se pretende redondear a un elemento (`table`, `div`, `span`...) alguna o todas las esquinas que presenta, permitiendo el redondeo individualizado. Sepuede usar de varias maneras diferentes.

- > **Un valor**: el dato introducido es aplicado a las cuatro esquinas del elemento por igual. Por ejemplo, `border-radius: 10px`.
- > **Dos valores**: los dos valores introducidos se aplican de la siguiente manera: el primero de ellos afectan a las esquinas inferior derecha y superior izquierda. El segundo, a las dos esquinas restantes, superior derecha e inferior izquierda. Por ejemplo, `border-radius: 10px 45px`.
- > **Tres valores**: el primer valor afecta a la esquina superior izquierda. El segundo, a las esquinas superior derecha e inferior izquierda. El último valor se aplica a la esquina inferior derecha. Por ejemplo, `border-radius: 10px 45px 18px`.
- > **Cuatro valores**: el primero de los valores es aplicable a la esquina superior derecha, el segundo, a la superior izquierda, el tercero, a la inferior derecha y el último, a la inferior izquierda. Por ejemplo, `border-radius: 10px 45px 18px 25px`.

También cabe la posibilidad de indicar el valor de cada esquina con cada una de las siguientes propiedades:

- > **`border-bottom-left-radius`**
- > **`border-bottom-right-radius`**
- > **`border-top-left-radius`**
- > **`border-top-right-radius`**



## Sombras

Con la propiedad `box-shadow` se crea una sombra sobre un elemento (div, table, etc.) con efecto de profundidad. La sintaxis es la siguiente:

**Box-shadow:** `none | h-offset v-offset | color | inset ;`

Donde:

- > **none** es el valor por defecto y no muestra sombreado.
- > **h-offset** es valor requerido. Se introduce el desplazamiento horizontal de la sombra donde los valores positivos representan el lado derecho, y el negativo el lado izquierdo del elemento.
- > **v-offset** es valor requerido. Se introduce el desplazamiento vertical de la sombra donde los valores positivos colocan la sombra debajo, y los negativos sobre el elemento.
- > **blur** es valor opcional. Es el radio de desenfoque. Para un desenfoque mayor, introducir un número grande.
- > **spread** es valor opcional. Es el radio de propagación. Los valores positivos aumentan la sombra, y los negativos la disminuyen.
- > **color** es valor opcional. Define el color de la sombra.
- > **inset** es valor opcional. La sombra pasa del exterior al interior.

## Transiciones

Se usa para animar las propiedades CSS de manera fácil, siendo una propiedad abreviada de las propiedades (transition-property, transition-duration, transition-timing-function, transition-delay).

La sintaxis sería la siguiente:

**Transition:** `property duration timing-function delay`

Donde:

- > **transition-property:** nombre propiedad o propiedades CSS separadas por coma.
  - » Ejemplo: `transition-property: height, width.`
- > **transition-duration:** duración del efecto en segundos o milisegundos.
- > **transition-timing-function:** velocidad del efecto, siendo algunos de sus valores:
  - » **linear:** misma velocidad.
  - » **ease:** valor por defecto.
  - » **ease-in:** inicio lento.
  - » **ease-out:** final de transición lenta.
  - » **ease-in-out:** comienzo y final de transición lenta.
- > **transition-delay:** retardo del inicio del efecto en segundos o milisegundos.



## Opacidad

Se usa la propiedad `opacity`, que permite establecer la opacidad o transparencia a un elemento, siendo los valores permitidos los del rango [0..1] donde 0 es transparente y 1 opaco.

## Columnas

La propiedad `columns` se usa para realizar columnas y es abreviada de estas dos propiedades:

- > `column-width`: ancho de columna.
- > `column-count`: número de columnas.

La sintaxis es la siguiente:

**Columns:** `auto` | `column-width` `column-count`

Otras propiedades relacionadas con columnas son:

- > `column-fill`: se especifica cómo se rellenan las columnas.
- > `column-gap`: espacio entre columnas.
- > `column-rule-color`: reglas de color entre columnas.
- > `column-rule-style`: reglas de estilo entre columnas.

## Animaciones

La propiedad `animation` permite realizar animaciones sobre elementos y es una propiedad abreviada de las siguientes:

- |  |   |
|--|---|
| > <code>animation-name</code>            | > <code>animation-timingfunction</code> |
| > <code>animation-iteration-count</code> | > <code>animation-fill-mode</code>      |
| > <code>animation-duration</code>        | > <code>animation-delay</code>          |
| > <code>animation-direction</code>       | > <code>animation-play-state</code>     |

## Tamaño de caja

La propiedad `box-sizing` define el cálculo del alto y ancho de un elemento. Su sintaxis es:

**Box-sizing:** `content-box` | `border-box`

- > `content-box` es el valor por defecto.
- > `border-box` son las propiedades de ancho y alto. Incluyen borde, contenido y relleno.

## Transformaciones

Gracias a la propiedad `transform` se puede aplicar una transformación (mover, escalar, rotar, etc.). Algunas de esas funciones son:

- |                               |   |
|-------------------------------|---|
| > <code>scale(x,y)</code>     | > <code>rotate 3 d (x , y, z , ángulo)</code> |
| > <code>scale3d(x,y,z)</code> | > <code>skewX(ángulo)</code>                  |
| > <code>rotate(ángulo)</code> |   |





## 2.4.

### XHTML

Se considera **XHTML**, sigla de **extensible hypertext markup language** (lenguaje de marcado de hipertexto extensible), como la adaptación de HTML en XML, pues mantiene las características de HTML e incluye normas XML.

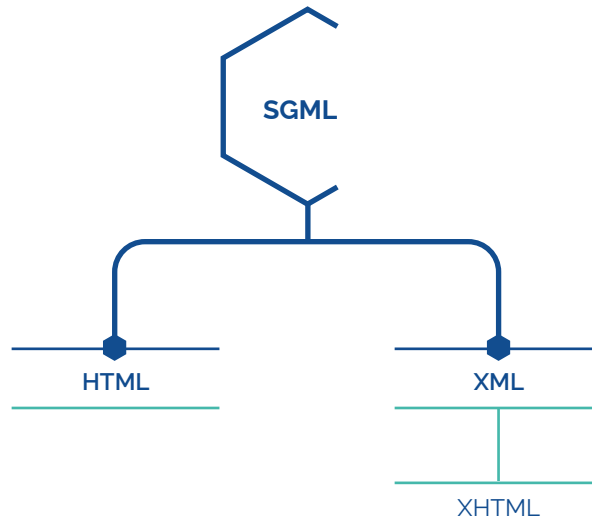


Diagrama 1. Jerarquía de los lenguajes de marcas.

#### 2.4.1. Versiones de XHTML

En la actualidad, existen varias versiones de XHTML:

- > **XHTML 1.0.** Surge en el año 2000, es la primera de todas y se considera similar a HTML 4.0 reformulado, cuya finalidad es que, gracias a ella, se use como lenguaje de contenidos. Esta versión presenta tres definiciones distintas para DTD (definición del tipo de documento) y es compatible con algunas versiones de HTML.
- > **XHTML 1.1.** Esta versión no es compatible con versiones anteriores de HTML. Se considera una versión basada en módulos que sirven de base para futuros documentos extendidos, quitando características que estaban en desuso. A continuación, se muestra un ejemplo:

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1 // EN " "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>LM</title>
  </head>
  <body>
    </body>
</html>
  
```

Imagen 66. Ejemplo de XHTML 1.1.



Módulo de lista	Dl, dt, dd, ol, ul, li
Módulo de formularios	Button, fieldset, form, input, label, legend, select, optgroup, option, textarea
Módulo de estructura	Body, head, html, title
Módulo de tablas	Caption, col, colgroup, table, tbody, td, tfoot, th, thead, tr
Módulo de imagen	Img

Cuadro 9. Módulos de XHTML 1.1.

- > **XHTML Básico.** Su objetivo es usarla por dispositivos de información que empleen acceso web como (teléfonos móviles, sistemas de navegación de coches, televisiones, relojes inteligentes, lectores de libros digitales, etc.) Presenta una versión más simplificada que las anteriores, que incluye un grupo reducido de módulos.

Módulo básico de formulario	Form, input, label, select, option, textarea
Módulo base	Base
Módulo de enlace	Link
Módulo de hipertexto	A
Módulo básico de tablas	Caption, table, td, th, tr

Cuadro 10. Módulos de XHTML Básico.

### 2.4.2. Diferencias sintácticas y estructurales con HTML

A la hora de crear documentos XHTML, no existe una gran diferencia con HTML, pero, para que esté bien formado, es necesario que cumpla una serie de reglas sintácticas heredadas.

XML es sensible a las mayúsculas y minúsculas, por lo que es necesario añadir los elementos y atributos en minúsculas.

Ya sean valores de texto, números o alfanuméricos, deberán ir entrecomillados.

En HTML, la gran mayoría de etiquetas tienen una de apertura y otra de cierre, aunque existen algunas, como las de comienzo de nueva línea o de imagen, en las que no es necesario su cierre.

A la hora de anidar etiquetas, hay que tener en cuenta que no deben solaparse (tienen que cerrarse conforme a cómo se abren).



Aunque los atributos minimizados no son los más usados, el nombre debe coincidir con su valor. Todos los atributos tienen que llevar un valor.

Respecto a la estructura, no varía mucho a la de un documento HTML y su mayor diferencia es la declaración del DTD para validar el documento, que es obligatoria. El elemento raíz tiene que ser la etiqueta <html>, que tendrá como atributos xmlns, que indica el espacio nominal XHTML (<http://www.w3.org/1999/xhtml>).

Al heredar XHTML de XML, comparte parte de su sintaxis y normas, por ello se utiliza DTD, sigla de document type definition o (definición del tipo de documento). Dentro de un DTD, se encuentra una serie de reglas que deben cumplirse.

```
<Declaración DTD>
<html ...>
  <head>
    <title>LM</title>
  </head>
  <body>
    <!-- contenido -->
  </body>
</html>
```

Imagen 67. Declaración DTD.

En la versión 1.0 de XHTML, existen tres posibilidades para la declaración del DTD (estricto, transitorio y frameset), incluyendo alguna de las tres fuera de la etiqueta <html> mediante la declaración DOCTYPE, pues esta es la primera que hay que añadir a la hora de crear un nuevo documento.

La declaración estricta se usa en el caso de no emplear elementos de estilo, pues es la transición la que sí los usa. En el caso de usar marcos dentro del documento, se usará la declaración frameset.

Tipo DTD	Declaración
Estricto	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" <a href="http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd</a> >
Transitorio	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" <a href="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd</a> >
Frameset	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" <a href="http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd</a> >

Cuadro 11. Declaración del DTD.



En el ejemplo siguiente se aprecia un documento XHTML versión 1.0 con una declaración de transición, que permite elementos en desuso.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
  <head>
    <title>LM</title>
  </head>
  <body>
  </body>
</html>
```

Imagen 68. Ejemplo de documento XHTML 1.0.

Aunque no es obligatorio, algunos autores usan como primera línea una declaración CML, necesaria para la codificación de caracteres:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Imagen 69. Declaración XML.

### 2.4.3. Ventajas de XHTML sobre HTML

XHTML presenta una serie de ventajas respecto a HTML, entre las que destacan:

- > A los documentos XHTML pueden agregárseles distintas aplicaciones (scripts y applets).
- > Pueden validarse con herramientas XML.
- > Compatible con los navegadores antiguos.
- > Pueden presentar mayor rendimiento.
- > Formato abierto.
- > Facilidad de mantener y editar.



 [www.universae.com](http://www.universae.com)

