

LMSGI

_02

Utilización de lenguajes de marcas en
entornos web

1.- Del HTML al XHTML: evolución y versiones.

HTML es el lenguaje utilizado para crear la mayor parte de las páginas web. Es un estándar reconocido en todos los navegadores, por lo tanto, todos ellos visualizan una página HTML de forma muy similar independientemente del sistema operativo sobre el que se ejecutan.

La evolución de sus versiones, desde su creación hasta la creación del XHTML podemos verla a continuación:

El origen de HTML fue un sistema de [hipertexto](#) para compartir documentos electrónicos en 1980. La primera propuesta oficial para convertir HTML en un estándar se realizó en 1993. Aunque ninguna de las dos propuestas de estándar que se hicieron (HTML y HTML+) consiguieron convertirse en estándar oficial.

- **HTML 2.0** fue la primera versión oficial de HTML el [IETF](#) publicó el estándar en septiembre de 1995.
- **HTML 3.2** se publicó el 14 de Enero de 1997 por el W3C. Incorpora los [applets de Java](#) y texto alrededor de las imágenes.
- **HTML 4.0** se publicó el 24 de Abril de 1998. Entre las novedades que presenta se encuentran las hojas de estilos CSS y la posibilidad de incluir pequeños programas en las páginas web.
- **HTML 4.01** es la última especificación oficial de HTML se publicó el 24 de diciembre de 1999. Es una actualización de la versión anterior. En ese momento el W3C detuvo la actividad de estandarización de HTML hasta marzo de 2007, momento en que se retoma debido a la fuerza de las empresas que forman el grupo [WHATWG](#) y a la publicación de los borradores de **HTML 5.0**, que será la siguiente versión de este lenguaje.

Tras la publicación del estándar HTML 4.01 se detecta su incompatibilidad con herramientas basadas en XML. Para evitar estos problemas se crea lenguaje XHTML que combina la sintaxis de HTML 4.0 con la de XML.

- **XHTML 1.0** fue la primera versión, se publicó el 26 de Enero de 2000. Es una adaptación de HTML 4.01 al lenguaje XML, por lo que mantiene sus características, y añade algunas restricciones y elementos de XML.
- La versión **XHTML 1.1** ya ha sido publicada en forma de borrador y pretende modularizar XHTML.
- El borrador de **XHTML 2.0 ya ha sido publicado**, que presenta grandes novedades respecto de las anteriores versiones

2.- Estructura de un documento HTML.

Caso práctico. Félix pregunta a Juan si existen grandes diferencias entre el XML y el HTML. Juan le explica que ambos lenguajes tienen origen en otro lenguaje que es el SGML y que sus diferencias son, principalmente, funcionales ya que la estructura del documento es semejante.

La estructura de una página HTML ha de ser coherente con la que hemos visto en el tema anterior para cualquier documento XML. Por ello tendrá un prólogo y un ejemplar.

Prólogo: Todo documento HTML ha de tener una declaración del tipo de documento donde se le indica al navegador el tipo de documento que se va a iniciar y la versión de HTML utilizada para la codificación del mismo y, además, le permite interpretarlo correctamente.

Para la versión HTML 4.0, hay tres prólogos distintos que definen tres tipos de documentos HTML:

- *HTML 4.0 Strict.* Es la DTD utilizada por defecto con HTML 4.0. En estos documentos no se permite el uso de los elementos declarados deprecated en otras versiones o Recomendaciones HTML. La declaración del tipo de documento correspondiente es:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 //EN" "http://www.w3.org/TR/REC-html40/strict.dtd">
```

- *HTML 4.0 Transitional.* Permite el uso de todos los elementos que permite el HTML 4.0 Strict, además de los elementos deprecated. La declaración del tipo de documento correspondiente es:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">
```

- *HTML 4.0 Frameset.* Es una variante de HTML 4.0 Transitional para documentos que usan [frames](#). En estos documentos el elemento body hay que reemplazarlo por un elemento frameset. La declaración del tipo de documento correspondiente es:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
"http://www.w3.org/TR/REC-html40/frameset.dtd">
```

- **Ejemplar:** En un documento HTML está delimitado por las etiquetas <html> y </html>. El ejemplar puede, a su vez dividirse en dos partes:
 - *La cabecera*, delimitada por las etiquetas <head> y </head>. Contiene la información sobre el título de la página, el autor, palabras clave, etc. Dentro de esta sección es obligatorio definir el título del documento, para ello se usan las etiquetas <title> </title>. Esta información no se presentará en la ventana del navegador, salvo el título que aparecerá en la barra de título de la parte superior.
 - *El cuerpo*, contiene la información que se va a presentar en la pantalla. Está limitado por las etiquetas <body> y </body>, salvo en los documentos de tipo HTML 4.0 Frameset donde éstas se sustituyen por <frameset> y </frameset>.

3.- Identificación de etiquetas y atributos de HTML.

Caso práctico

María, tras escuchar que XML y HTML tienen el mismo origen, se interesa por si utilizan las mismas etiquetas.

Juan le contesta que en HTML las etiquetas y los atributos están definidos previamente, mientras que en XML los define el programador.

Un documento HTML está formado por etiquetas y atributos.

Al igual que en XML las etiquetas pueden ser de apertura, <etiqueta>, o de cierre, </etiqueta>. Una de las diferencias con XML es que la cantidad de etiquetas de HTML está limitada a aquellas que están definidas por el lenguaje.

Aunque HTML define una gran cantidad de etiquetas, estas no son suficientes para crear páginas complejas ya que la definición completa de ciertos elementos, como las imágenes y los enlaces, requiere información adicional. Como no es posible crear una etiqueta por cada elemento diferente, se añade la información

adicional a las etiquetas mediante los atributos dando lugar a los elementos.

Para cada uno de los atributos hay definido un conjunto de valores que se le puede asignar, si el valor de un atributo no es válido, el navegador le ignora.

Cada una de las etiquetas HTML define los atributos que puede utilizar, aunque algunos de ellos son comunes a muchas etiquetas.

3.1.- Clasificación de los atributos comunes según su funcionalidad.

- **Atributos básicos:** Se pueden usar en casi todas las etiquetas HTML.

Atributo	Descripción
name = "texto"	Permite asignar el nombre "texto" a un objeto HTML
title = "texto"	Asigna un título a un elemento HTML, mejorando así la accesibilidad. Dicho título es mostrado por los navegadores cuando el usuario pasa el ratón por encima del elemento. Es especialmente útil con los elementos: a, link, img, object, abbr y acronym
id = "texto"	Permite identificar al elemento HTML sobre el que se aplica de forma única mediante el identificador "texto". Sólo es útil cuando se trabaja con CSS y con Javascript. No pueden empezar por números y sólo puede contener letras, números, guiones medios y/o guiones bajos.
style = "texto"	Permite aplicar al elemento HTML el estilo "texto" directamente.
class = "texto"	Permite aplicar al elemento HTML el estilo "texto" definido en las CSS. No pueden empezar por números y sólo puede contener letras, números, guiones medios y/o guiones bajos.

- **Atributos para internacionalización:** Los utilizan las páginas que muestran sus contenidos en varios idiomas o aquellas que quieren indicar de forma explícita el idioma de sus contenidos

Atributo	Descripción																
dir	Indica la dirección del texto por lo que sólo puede tomar dos valores: ltr (left to right) de izquierda a derecha. Es el valor por defecto. rtl (right to left) de derecha a izquierda.																
lang = "codigo"	<p>Especifica el idioma del elemento mediante un código predefinido. Los posibles valores de este atributo se encuentran en el documento RFC 1766, algunos de los valores posibles son:</p> <table><tr><th>Código</th><th>Idioma</th><th>Código</th><th>Idioma</th></tr><tr><td>en</td><td>Inglés (Gran Bretaña)</td><td>es</td><td>Español</td></tr><tr><td>en-US</td><td>Inglés americano</td><td>fr</td><td>Francés</td></tr><tr><td>ja</td><td>Japones</td><td>fr-CA</td><td>Francés de Canada</td></tr></table>	Código	Idioma	Código	Idioma	en	Inglés (Gran Bretaña)	es	Español	en-US	Inglés americano	fr	Francés	ja	Japones	fr-CA	Francés de Canada
Código	Idioma	Código	Idioma														
en	Inglés (Gran Bretaña)	es	Español														
en-US	Inglés americano	fr	Francés														
ja	Japones	fr-CA	Francés de Canada														
xml:lang = "codigo"	Especifica el idioma del elemento mediante un código definido según la recomendación RFC 1766.																

- En las páginas XHTML, el atributo xml:lang tiene más prioridad que lang y es obligatorio incluirlo siempre que se incluye el atributo lang.

- **Atributos de eventos y atributos para los elementos que pueden obtener el foco:** Sólo se utilizan en las páginas web dinámicas creadas con JavaScript. Como no es nuestro objetivo no lo vamos a contemplar.

3.2.- Elementos HTML.

Un elemento HTML está formado por:

- Una etiqueta de apertura.
- Cero o más atributos.
- Texto encerrado por la etiqueta. Es opcional, no todas las etiquetas pueden encerrar texto.
- Una etiqueta de cierre.

Según el modo en que ocupan el espacio disponible en la página los elementos pueden ser de dos tipos:

- **Elementos en línea.** Sólo ocupan el espacio necesario para mostrar sus contenidos. Su contenido puede ser texto u otros elementos en línea.
- **Elementos de bloque.** Los elementos de bloque siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea, aunque sus contenidos no lleguen hasta allí. Su contenido puede ser texto, elementos en línea u otros elementos de bloque.

Existen elementos cuyo comportamiento puede ser en línea o de bloque según las circunstancias.

3.2.1.- Elementos de la estructura básica del documento.

La estructura básica de un documento viene determinada por las siguientes etiquetas:

Elemento	Descripción
<i>html</i>	Documento HTML.
<i>head</i>	Cabecera del documento.
<i>body</i>	Cuerpo del documento. Permite definir formatos que se aplican a los elementos de la página de manera global, como son el color del fondo del texto, los márgenes, el color de los enlaces, ...

3.2.2.- Elementos de la sección de cabecera.

- **Elementos contenedores:**

Elemento	Descripción
<i>title</i>	Título del documento.
<i>script</i>	Script incrustado. Su contenido ha de ir situado entre las marcas de comentarios ya que no ha de ser interpretado.
<i>style</i>	Estilo aplicado al documento utilizando CSS. Su contenido ha de ir situado entre las marcas de comentarios ya que no ha de ser interpretado.

• Elementos no contenedores:

Elemento	Descripción
<i>base</i>	URI base del documento
<i>isindex</i>	Prompt de entrada de datos.
<i>link</i>	Enlaces a documentos externos de librerías
<i>meta</i>	Información que agiliza la búsqueda del documento en buscadores.

3.2.3.- Elementos que dan formato al texto de un párrafo.

Los distintos elementos que podemos utilizar para dar formato a nuestro texto son:

Elemento	Descripción
<i>p</i>	Delimita los párrafos
<i>h_i</i>	Encabezado de nivel i, donde i es un número entero entre 1 y 6, ambos inclusive. El tamaño de la letra del encabezado es mayor cuanto menor sea el valor de i. No deben de usarse estas etiquetas para formatear texto. Sólo estarán bien usadas para designar títulos de párrafos.
<i>b</i>	Indica que el texto que está en ese elemento se le pondrá en negrita.
<i>i</i>	Indica que el texto que está en ese elemento se le pondrá en itálica ó cursiva.
<i>u</i>	Indica que el texto que está en ese elemento se le pondrá subrayado.
<i>sup</i>	Indica que el texto que está en ese elemento es un supraíndice.
<i>sub</i>	Indica que el texto que está en ese elemento es un subíndice.
<i>strong</i>	Indica que el texto que está en ese elemento estará resaltado. Habitualmente los navegadores resaltan el texto poniéndolo en negrita aunque podría haber algún navegador que resaltase el texto poniéndolo en cursiva y en naranja.

3.2.4.- Elementos de listas.

Hay tres tipos de listas: ordenadas, desordenadas y listas de definición.

Elemento	Descripción
<i>ul</i>	Delimita los elementos que forman una lista desordenada
<i>ol</i>	Delimita los elementos que forman una lista ordenada
<i>li</i>	Indica cada uno de los elementos de una lista
<i>dl</i>	Delimita los elementos que forman una lista de definición
<i>dt</i>	Cada uno de los términos que se definen de una lista de definición.
<i>dd</i>	Cada una de las definiciones de una lista de definición.

3.2.5.- Elementos de tablas.

Los elementos para definir una tabla son los siguientes:

Elemento	Descripción
<i>table</i>	Delimita el contenido de una tabla.
<i>tr</i>	Delimita cada una de las líneas de la tabla.
<i>td</i>	Delimita el contenido de cada celda de la tabla.
<i>colgroup</i>	Permite agrupar columnas.
<i>tbody</i>	Permite agrupar líneas de la tabla.
<i>thead</i>	Define la línea cabecera de la tabla.
<i>th</i>	Delimita cada una de las celdas de la cabecera
<i>tfoot</i>	

3.2.6.- Elementos de formularios.

Lo elementos que puede contener un formulario son los siguientes:

Elemento	Descripción
<i>form</i>	Delimita el contenido del formulario .
<i>input</i>	Caja de texto para texto corto. Dependiendo del valor que tome el atributo type de este elemento podemos estar ante un texto sin más, un campo de texto donde al escribir no se visualice el contenido si no que escriba asteriscos, un botón de radio que el usuario podrá elegir, una opción que el usuario podrá activar, botones, ...
<i>textarea</i>	Caja de texto para texto largo.
<i>select</i>	Crea un menú desplegable que permite elegir de una lista de opciones que contiene el elemento.
<i>option</i>	Delimita cada una de las opciones de un menú desplegable que le contiene.
<i>button</i>	Permite definir un botón. Su principal ventaja frente a los botones hechos con input es que este elemento permite introducir en el botón cualquier otro elemento de HTML, x ejemplo imágenes.
<i>fieldset</i>	Permite agrupar elementos de un un formulario.
<i>legend</i>	Permite poner un título al fieldset.
<i>label</i>	Etiqueta de un campo del formulario.

3.2.7.- Elementos de frames.

Los elementos utilizados para trabajar con frames son:

Elemento	Descripción
<i>frameset</i>	Define la partición de la ventana del navegador en marcos. Sólo puede partirse en filas o en columnas. Para partir la ventana del navegador en filas y columnas hay que anidar frames.
<i>Frame</i>	Define un marco que contiene información.

--	--

3.2.8.- Otros elementos.

Otros elementos que podemos utilizar son:

Elemento	Descripción
<i>a</i>	Permite definir un enlace a una página web, un archivo o una dirección de correo.
<i>img</i>	Permite insertar una imagen en una página web. Es obligatorio utilizar el atributo src para determinar el path del fichero de imagen que queremos insertar.
<i>abbr</i>	Indica una forma abreviada.
<i>acronym</i>	Indica un acrónimo.
<i>blockquote</i>	Contiene un bloque de texto con sangría.
<i>q</i>	Contiene una cita por lo que el navegador le añade las marcas de citación
<i>br</i>	Inserta una línea en blanco. No tiene etiqueta de cierre, sólo se abre.

4.- XHTML frente a HTML.

Caso práctico

Puesto que **Juan** ha aceptado realizar la página web se plantea el hacerla en HTML o en XHTML consulta con **Marina**, trabajadora de su empresa informática.

Marina opina que, desde un punto de vista formal, no hay diferencias sustanciales entre utilizar uno u otro lenguaje, y siguiendo la evolución lógica le parece que sería más apropiado utilizar XHTML y añade que este lenguaje tiene la ventaja de ser compatible con navegadores antiguos.

Juan opina que es una buena opción, pero que algunos navegadores, a pesar de ser compatibles con el lenguaje no interpretan los formatos.

El lenguaje XHTML es muy similar al lenguaje HTML. De hecho, no es más que una adaptación de HTML al lenguaje XML, el estándar XHTML 1.0 sólo añade pequeñas mejoras y modificaciones menores al estándar HTML 4.01, por lo que este último está prácticamente incluido en el primero, lo que hace que pasar del HTML 4.01 Strict a XHTML no requiere casi ningún cambio.

El lenguaje HTML tiene una sintaxis muy permisiva, por lo que es posible escribir sus etiquetas y atributos de muchas formas diferentes. Las etiquetas, por ejemplo, podían escribirse en mayúsculas, en minúsculas e incluso combinando mayúsculas y minúsculas. El valor de los atributos de las etiquetas se pueden indicar con o sin comillas. Además, el orden en el que se abrían y cerraban las etiquetas no era importante.

La flexibilidad de HTML da lugar a páginas con un código desordenado, difícil de mantener y muy poco profesional.

XHTML soluciona estos problemas añadiendo ciertas normas en la forma de escribir las etiquetas y atributos.

4.1.- XHTML: diferencias sintácticas y estructurales con HTML.

El esquema básico del documento, para considerarse conforme a la especificación deberá cumplir las siguientes condiciones:

- *El elemento raíz del documento debe ser <html>.*
- *El elemento raíz del documento debe indicar el espacio nominal XHTML usando el atributo xmlns. El espacio nominal para XHTML es <http://www.w3.org/1999/xhtml>*
- *Debe haber una declaración DOCTYPE en el prólogo del documento. El identificador público incluido en la declaración DOCTYPE debe hacer referencia a alguna de las tres DTD definidas por el W3C usando el Identificador Formal Público correspondiente:*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Restricciones básicas que introduce XHTML respecto a HTML en la sintaxis de sus etiquetas:

- Las etiquetas se tienen que cerrar en orden inverso al que se abren, nunca pueden solaparse.
- Los nombres de las etiquetas y atributos siempre se escriben en minúsculas.
- El valor de los atributos, incluso los numéricos, siempre se encierra entre comillas.
- Los atributos en los que el nombre coincide con su valor, no puede darse el valor por entendido, es decir, no se pueden comprimir. Este tipo de atributos no son muy habituales.
- Todas las etiquetas deben cerrarse siempre. XHTML permite que en lugar de abrir y cerrar de forma consecutiva la etiqueta (

) se puede utilizar la sintaxis
 para indicar que es una etiqueta vacía que se abre y se cierra en ese mismo punto.

Otras restricciones:

Además de las cinco restricciones básicas, XHTML incluye otros cambios más avanzados respecto a HTML, entre ellas:

- Antes de acceder al valor de un atributo, se eliminan todos los espacios en blanco que se encuentran antes y después del valor. Además, se eliminan todos los espacios en blanco sobrantes dentro del valor de un atributo.
- El código JavaScript debe encerrarse entre unas etiquetas especiales (<![CDATA[y]]>) para evitar que el navegador interprete de forma errónea caracteres como & y <.
- Las páginas XHTML deben prescindir del atributo name en su lugar, siempre debe utilizarse el atributo id.
- XHTML es necesario separar el formato del contenido. Los párrafos deben separarse consistentemente y las cabeceras h1-h6 sólo deben usarse para destacar los diferentes apartados. Es recomendable dar el formato a los datos por medio del uso de las CSS.

4.2.- Ventajas e inconvenientes de XHTML sobre HTML.

- **Ventajas:**
 - Compatibilidad parcial con navegadores antiguos: la información se visualiza, aunque sin formato.
 - Un mismo documento puede adoptar diseños radicalmente distintos en diferentes apartados.
 - Sencillez a la hora de editar y mantener el código.
 - Es compatible con los estándares que está desarrollando el W3C como recomendación para futuros agentes de usuario o navegadores.
 - Los documentos escritos conforme a XHTML 1.0 presentan mejor rendimiento en las actuales herramientas web que aquellos escritos conforme a HTML.
 - La separación de los contenidos y su presentación hace que los documentos XHTML se adapten mejor a las diferentes plataformas: pantallas de ordenador, pantallas de dispositivos móviles, ...
 - Como es XML se pueden utilizar fácilmente herramientas creadas para procesar documentos XML genéricos (editores, XSLT, etc.).
- **Inconvenientes:**
 - Algunos navegadores antiguos no son totalmente compatibles con los estándares, lo que hace que las páginas no siempre se muestren correctamente. Esto cada vez es menos problemático ya que estos navegadores van cayendo en desuso.
 - Muchas herramientas de diseño web aún no generan código XHTML correcto.

5.- Herramientas de diseño web.

Caso práctico Juan muestra a **María** y **Félix** una primera versión de la página web.

Tras verla **Félix** tiene la curiosidad de saber si para codificarla hay que utilizar algún software característico o basta con usar un editor de texto plano, como en el caso de XML.

Juan le cuenta que puede bastar el bloc de notas, pero que existen varios editores que facilitan la tarea.

Actualmente hay herramientas que permiten diseñar un sitio web sin necesidad de saber programar HTML. Las más populares:

- **Macromedia Dreamweaver.**
- **Microsoft Front Page.**
- **Adobe Go live.**
- **NetObjects Fusion.**

- **Amaya** es una herramienta de libre distribución creada por el W3C que permite visualizar y editar páginas HTML y XHTML con hojas de estilo CSS, expresiones [MathML](#) y dibujos [SVG](#); además de documentos XML. Está disponible para plataformas Windows, GNU/Linux, Mac OS X, entre otras. La última versión soporta HTML 4.01, XHTML 1.0, XHTML Basic, XHTML 1.1, HTTP 1.1, MathML 2.0, muchas características CSS 2, e incluye soporte para gráficos SVG.

Además de las anteriores, para generar páginas web, es conveniente tener algunas de las siguientes herramientas:

- **Software de diseño**
 - *Macromedia Flash* para hacer animaciones, banners o sitios enteros con esta tecnología.
 - *Macromedia Fireworks* o *Adobe Illustrator* para diseñar botones, logos, imágenes, etc.
 - *Adobe Photoshop* o *Gimp*, para retocar fotografías y trabajar con imágenes.
- **Recursos: diseño web**
 - *My Fonts*, es un sitio web que nos vende fuentes que pueden ser utilizadas para la web.
 - *Color Voodoo*, tiene interesante información sobre el uso de los colores y su influencia en la web.
 - *moreCrayons*, una paleta de colores web seguros (Pues no todos los colores se ven iguales en distintos sistemas operativos).
 - *Yafra Color*, permite crear la gama a partir del color que se elija como principal además funciona como conversor entre los modelos de color [RGB](#) y [HSV](#).
 - *ColorJack*, permite crear una gama a partir de un color y además calcula equivalencias de colores entre RGB, HSV y [CMYK](#).
 - *Kuler*, es una aplicación online de Adobe Labs en la que se puede elegir una combinación de colores y compartirla con otros usuarios, que pueden votar sus favoritas. Requiere registro.

6.- Hojas de estilo o CSS.

Caso práctico_ María muestra su conformidad con la estructura y contenidos de la futura web corporativa que **Juan** ha realizado, aunque le gustaría probar otras gamas de colores y tipos de fuente.

Félix plantea que, quizás, resulte un trabajo demasiado laborioso ya que supondría modificar todos los ficheros de la web.

Juan, sonriendo, comenta que de hecho no es exactamente así. En realidad los datos están separados de sus formatos.

CSS (Cascading Style Sheets) permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo.

Antes del uso de CSS, los diseñadores de páginas web debían definir el aspecto de cada elemento dentro de las etiquetas HTML de la página. El principal problema de esta forma de definir el aspecto de los elementos es que habría que definir el formato de cada uno de los elementos que formen la página, lo cual

hace que sea muy difícil de actualizar.

CSS permite separar los contenidos de la página y su aspecto. Para ello se define en una zona reservada el formato de cada uno de los elementos de la web. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a ella en las que aparezca ese elemento. Las hojas de estilo están compuestas por una o más reglas de estilo aplicadas a un documento HTML o XML.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, cabecera, texto destacado, etc. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el formato de cada elemento.

CSS obliga a crear [documentos semánticos HTML/XHTML](#), mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Las hojas de estilos aparecieron poco después que el lenguaje de etiquetas SGML, alrededor del año 1970. Desde la creación de SGML, se observó la necesidad de definir un mecanismo que permitiera aplicar estilos a los documentos electrónicos. La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaban la creación de documentos que tuvieran igual apariencia en distintos navegadores.

El organismo W3C propuso la creación de un lenguaje de hojas de estilos específico para el lenguaje HTML.

En 1995, el W3C añadió a su grupo de trabajo de HTML el desarrollo y estandarización de CSS.

CSS 1, se publicó en 1996, es la primera recomendación oficial.

CSS 2, publicada en 1998, es la segunda recomendación oficial.

CSS 3, continúa en desarrollo desde 1998.

Actualmente se utiliza la versión CSS 2.1, actualizada por última vez el 19 de julio de 2007.

El diseño web siempre está limitado por las posibilidades de los navegadores que utilizan los usuarios para acceder a sus páginas. Por este motivo es imprescindible conocer el soporte de CSS en cada uno de los navegadores más utilizados del mercado.

6.1.- Soporte de CSS en los navegadores.

El soporte de CSS de un navegador viene determinado por el motor del mismo ya que es éste el encargado de interpretar el CSS.

La siguiente tabla muestra el soporte de **CSS 1**, **CSS 2.1** y **CSS 3** de los cinco navegadores más utilizados en la actualidad:

Navegador	Motor	CSS 1	CSS 2.1	CSS 3
Safari	WebKit	Completo	Casi completo	Parcial
Opera	Presto	Completo	Casi completo	Parcial
Firefox	Gecko	Completo	Casi completo	Parcial
Google Chrome	WebKit	Completo	Casi completo	Parcial
Internet Explorer	Trident	Completo, desde la versión 6.0	Completo, desde la versión 8.0	Prácticamente nulo

6.2.- Cómo incluir CSS en un documento HTML o XHTML.

Existen tres opciones para incluir CSS en un documento HTML o XHTML:

1. Definir CSS en un archivo externo.

En este caso, todos los estilos CSS se incluyen en uno, o varios, archivos de texto plano, cuya extensión es .css, que las páginas HTML enlazan mediante el elemento <link> de la cabecera del fichero HTML.

Puesto que una página web puede tener asociados varios ficheros CSS es recomendable agrupar estos últimos en un directorio.

El navegador descarga los archivos CSS externos, además de la página web asociada a ellos, y aplica los estilos a los contenidos de la página antes de mostrar sus contenidos.

Esta es la forma de incluir CSS en las páginas HTML más utilizada. La principal ventaja es que se puede incluir un mismo archivo CSS en multitud de páginas HTML, por lo que se garantiza la aplicación homogénea de los mismos estilos a todas las páginas que forman un sitio web.

Además, el mantenimiento del sitio web se simplifica al máximo, ya que el cambio en un solo archivo CSS permite variar de forma instantánea los estilos de todas las páginas HTML asociadas.

Puede hacerse de dos modos diferentes:

1. *Mediante enlaces.*
2. *Importando el fichero CSS.*

2. Incluir CSS en el documento HTML.

Este método se emplea cuando se definen pocos estilos o cuando se quieren incluir estilos específicos en una determinada página HTML que completen los estilos globales de todas las páginas del sitio web.

Tiene el inconveniente de que para modificar los estilos definidos, es necesario modificar todas las páginas que incluyen el estilo que se va a cambiar.

3. Incluir CSS en los elementos HTML.

El último método para incluir estilos CSS en documentos HTML es el peor y el menos utilizado, ya que para modificar un formato hay que cambiar todos los elementos que estén asociados a él.

Solamente se utiliza en determinadas situaciones en las que se debe incluir un estilo muy específico para un solo elemento concreto.

6.2.1.- Definir CSS en un archivo externo enlazado.

Para realizar una página web usando un archivo CSS externo, se deben seguir los tres pasos siguientes:

- Se crea un archivo de texto plano con las definiciones de los formatos.
- Dicho archivo de texto se guarda con extensión .css
- Se enlaza el archivo CSS externo mediante la etiqueta <link> en la cabecera de la página web.

El elemento <link> puede tener definidos cuatro atributos cuando se enlaza un archivo CSS:

- rel, indica el tipo de relación que tiene el archivo enlazado y la página HTML. Para los archivos CSS, siempre se utiliza el valor stylesheet
- type, indica el tipo de recurso enlazado. Para los archivos CSS su valor siempre es text/css
- href, indica la URL del archivo CSS que contiene los estilos. Puede ser relativa o absoluta y puede referenciar a un recurso interno o externo al sitio web.
- media, indica el medio en el que se van a aplicar los estilos del archivo CSS.

6.2.2.- Definir CSS en un archivo externo importado.

Se puede obtener el mismo resultado anterior utilizando el elemento <style> en lugar de <link>.

En este caso, se usa una regla de tipo @import seguida de una cadena de texto encerrada con comillas simples o dobles que se corresponde con la URL del archivo CSS, o de url() conteniendo dicha cadena entre los paréntesis. Las siguientes reglas @import son equivalentes para un fichero formatos.css que está en el directorio css.:

```
@import 'css/formatos.css';
@import "css/ formatos.css"
@import url('css/ formatos.css');
@import url("css/ formatos.css");
```

6.2.3.- Definir CSS en el documento HTML.

En este caso los formatos de los elementos se definen en la cabecera del documento HTML, dentro del elemento <style>.

6.3.- Sintaxis de las reglas de estilo.

Cada uno de los estilos que componen una hoja de estilos CSS se denomina regla. Cada regla se forma por:

- Selector: indica el elemento o elementos HTML a los que se aplica la regla CSS
- Llave de apertura, {
- Declaración: especifica los estilos que se aplican a los elementos.
 - Propiedad: permite modificar el aspecto de un atributo del elemento.
 - Valor: indica el nuevo valor del atributo modificado en el elemento.
- Llave de cierre, }.

Ejemplo: p{ color : blue; }

En este caso el selector es "p", la declaración es: "color : blue" y, dentro de ésta, podemos diferenciar la propiedad "color" y el valor "blue".

Un archivo CSS puede contener infinitas reglas CSS, cada regla puede contener varios selectores y cada declaración puede estar formada por diferentes declaraciones.

4.- Atributos principales.

En los siguientes subapartados vamos a ver los atributos principales que se usan en CSS como son:

- Atributos de color y fondo.
- Atributos de fuente.
- Atributos de texto.
- Atributos de caja.
- Atributos de clasificación.

6.4.1.- Atributos de color y fondo.

Los atributos de color y fondo son los que enumeramos a continuación:

Elemento	Descripción
<i>color</i>	Indica el color del texto. Lo admiten casi todas las etiquetas de HTML. El valor de este atributo es un color, con su nombre o su valor RGB.
<i>background-color</i>	Indica el color de fondo del elemento. El valor de este atributo es un color, con su nombre o su valor RGB.
<i>background-image</i>	Permite colocar una imagen de fondo del elemento. El valor que toma es el nombre de la imagen con su camino relativo o absoluto
<i>background-repeat</i>	Indica si ha de repetirse la imagen de fondo y, en ese caso, si debe ser horizontal o verticalmente. Los valores que puede tomar son: repeat-x, repeat-y o no-repeat.
<i>background-attachment</i>	Especifica si la imagen ha de permanecer fija o realizar un scroll. Los valores que pueden tomar son: scroll o fixed.
<i>background-position</i>	Es una medida, porcentaje o el posicionamiento vertical u horizontal con los valores establecidos que sirve para posicionar una imagen. Los valores que puede tomar son: porcentaje, tamaño, o [top, center, bottom] [left, center, right]
<i>background</i>	Establece en un solo paso cualquiera de las propiedades de background anteriores. Los valores que puede tomar son: background-color, background-image, background-repeat, background-attachment, background-position.

Dado que no todos los nombres de colores son admitidos en el estándar, es aconsejable utilizar el valor RGB.

6.4.2.- Atributos de fuente.

En este apartado vamos a ver los distintos atributos que podemos utilizar referentes a las fuentes de nuestro documento y que son:

Elemento	Descripción
<i>font-size</i>	Indica el tamaño de la fuente. Puede ser un tamaño absoluto, relativo o en porcentaje. Toma valores de unidades de CSS
<i>font-family</i>	Establece la familia a la que pertenece la fuente. Si el nombre de una fuente tiene espacios se utilizan comillas para que se entienda bien. El valor es el nombre de la familia fuente.
<i>font-weight</i>	Define el grosor de los caracteres. Los valores que puede tomar son: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800 o 900
<i>font-style</i>	Determina si la fuente es normal o cursiva. El estilo oblique es similar al cursiva. Los valores posibles son: normal, italic, oblique.
<i>font-variant</i>	Determina si la fuente es normal o mayúsculas pequeñas. Los valores que puede tomar son: normal o small-caps

<i>line-height</i>	El alto de una línea y por tanto, el espaciado entre líneas. Es una de esas características que no se podían modificar utilizando HTML.
<i>font</i>	Permite establecer todas las propiedades anteriores en el orden que se indica a continuación: font-style, font-variant, font-weight, font-size[line-height], font family. Los valores han de estar separados por espacios. No es obligatorio el uso de todos los valores.

6.4.3.- Atributos de texto.

En el apartado anterior vimos los atributos relacionados con las fuentes y en este vamos a ver los relacionados con el texto en sí y son los siguientes:

Elemento	Descripción
<i>text-decoration</i>	Establece si el texto está subrayado, sobrerayado o tachado. los valores que puede tomar son: none, underline, overline, line-through o blink
<i>text-align</i>	Indica la alineación del texto. Aunque las hojas de estilo permiten el justificado de texto no funciona en todos los sistemas. Los valores que puede tomar son: left, right, center o justify
<i>text-indent</i>	Determina la tabulación del texto. Los valores que toma son una longitud, en unidades CSS, o un porcentaje de la establecida.
<i>text-transform</i>	Nos permite transformar el texto, haciendo que tenga la primera letra en mayúsculas de todas las palabras, todo en mayúsculas o minúsculas. Los valores que puede tomar son: capitalize, uppercase, lowercase o none
<i>word-spacing</i>	Determina el espaciado entre las palabras. Los valores que puede tomar es un tamaño.
<i>letter-spacing</i>	Determina el espaciado entre letras. Los valores que puede tomar es un tamaño.
<i>vertical-align</i>	Establece la alineación vertical del texto. Sus valores posibles son: baseline, sub, super, top, text-top, middle, bottom, text-bottom o un porcentaje.
<i>line-height</i>	Altura de la línea. Puede establecerse mediante un tamaño o un porcentaje

6.4.4.- Atributos de caja.

Ahora vamos a ver otros atributos muy importantes y que utilizaremos muy a menudo. tributos de caja:

Elemento	Descripción
<i>margin-left</i>	Indica el tamaño del margen izquierdo. Puede usarse una longitud, en unidades CSS, o un porcentaje.
<i>margin-rigth</i>	Indica el tamaño del margen derecho. Puede usarse una longitud, en unidades CSS, o un porcentaje.
<i>margin-top</i>	Indica el tamaño del margen superior. Puede usarse una longitud, en unidades CSS, o un porcentaje.
<i>margin-bottom</i>	Indica el tamaño del margen inferior. Puede usarse una longitud, en unidades CSS, o un porcentaje.
<i>margin</i>	Permite establecer los márgenes de una vez. Hay que seguir el orden: superior, derecho, inferior e izquierdo.
<i>padding-left</i>	Indica el espacio izquierdo entre el borde y el contenido. Puede usarse una longitud, en unidades CSS, o un porcentaje.
<i>padding -rigth</i>	Indica el espacio derecho entre el borde y el contenido. Puede usarse una longitud, en unidades CSS, o un porcentaje.
<i>padding -top</i>	Indica el espacio superior entre el borde y el contenido. Puede usarse una longitud, en unidades CSS, o un porcentaje.
<i>padding -bottom</i>	Indica el espacio inferior entre el borde y el contenido. Puede usarse una longitud, en unidades CSS, o un porcentaje.
<i>padding</i>	Establece el espacio entre los bordes y el contenido de una sola vez. Hay que respetar el orden superior, derecho, inferior e izquierdo.
<i>border-left-color</i>	Establece el color del borde izquierdo del elemento. Su valor es un color RGB o el nombre del color.
<i>border-rigth-color</i>	Establece el color del borde derecho del elemento. Su valor es un color RGB o el

	nombre del color.
<i>border-top-color</i>	Establece el color del borde superior del elemento. Su valor es un color RGB o el nombre del color.
<i>border-bottom-color</i>	Establece el color del borde inferior del elemento. Su valor es un color RGB o el nombre del color.
<i>border-color</i>	Establece el color de los bordes del elemento de una sola vez. Hay que seguir el orden superior, derecho, inferior e izquierdo. Su valor es un color RGB o el nombre del color.
<i>border-style</i>	Establece el estilo del borde, los valores significan: none=ningun borde, dotted=punteado (no funciona siempre), solid=solido, double=doble borde, los valores groove, ridge, inset y outset son bordes con varios efectos 3D.
<i>border-left-width</i>	Grosor del borde izquierdo. Sus valores posibles son: thin, médium, thick o un tamaño.
<i>border-right-width</i>	Grosor del borde derecho. Sus valores posibles son: thin, médium, thick o un tamaño.
<i>border-top-width</i>	Grosor del borde superior. Sus valores posibles son: thin, médium, thick o un tamaño.
<i>border-bottom-width</i>	Grosor del borde inferior. Sus valores posibles son: thin, médium, thick o un tamaño.
<i>border-width</i>	Establece el tamaño de los bordes del elemento al que lo aplicamos. Hay que seguir el orden superior, derecho, inferior, izquierdo.
<i>width</i>	Establece el ancho del contenido del elemento. El valor es un porcentaje o un tamaño.
<i>height</i>	Establece la altura del contenido del elemento. El valor es un porcentaje o un tamaño.
<i>float</i>	Sirve para alinear un elemento a la izquierda o la derecha haciendo que el texto se agrupe alrededor de dicho elemento. Toma los valores none, left o right
<i>clear</i>	Establece si un elemento tiene a su altura imágenes u otros elementos alineados a la derecha o la izquierda. Sus valores posibles son: none, left, right o both.

6.4.5.- Atributos de clasificación.

En este apartado vamos a ver otros atributos que hemos etiquetado como atributos de clasificación y que son los siguientes:

Elemento	Descripción
<i>display</i>	Determina si el elemento es de bloque, línea, lista o ninguno de ellos. Los valores que puede tomar son: block, inline, list-item o none.
<i>white-space</i>	Indica el modo en que se ha de gestionar los espacios en blanco que hay en el elemento, es decir, si se mantienen todos los existentes tal y como estén en el documento o si se anulan a uno las secuencias de blancos, es el valor por defecto y el de la opción normal. Valores que puede tomar son: pre, nowrap, normal.
<i>list-style-type</i>	Indica cual es el símbolo que se utiliza como marcador en las listas. Valores que puede tomar son: disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none.
<i>list-style-image</i>	Permite utilizar el uso de una imagen como marcador en una lista. El valor que toma es la ruta del fichero imagen
<i>list-style-position</i>	Determinan la posición del marcador en una lista. Puede tomar los valores: outside o inside.
<i>list-style</i>	Permite establecer de una única vez todas las características de una lista. Hay que seguir el orden siguiente: list-style-type, list-style-position y list-style-image.

6.5.- CSS de posicionamiento.

Es un añadido a CSS que permite determinar el modo en que se ha de colocar un determinado elemento. Las propiedades definidas en CSS-P son las siguientes:

Elemento	Descripción
----------	-------------

<i>clip</i>	Permite seleccionar una zona. Los valores que puede tomar son: shape o auto.
<i>height</i>	Permite establecer la altura de un elemento. Los valores que puede tomar son: auto o un tamaño.
<i>width</i>	Permite establecer la anchura de un elemento. Los valores que puede tomar son: auto o un tamaño o porcentaje.
<i>visibility</i>	Indica si el elemento sobre el que actúa será visible o no. Los valores que puede tomar son:
<i>left</i>	Indica la posición del lado izquierdo del elemento. Los valores que puede tomar son: auto o un tamaño o porcentaje.
<i>top</i>	Indica la posición del lado superior del elemento. Los valores que puede tomar son: auto o un tamaño o porcentaje.
<i>overflow</i>	Indica si el elemento será visible o no en caso de superar los límites del contenedor. Los valores que pueden tomar son: visible, hidden, scroll o auto.
<i>position</i>	Determinan si el posicionamiento de un elemento es absoluto, relativo o estático. Los valores que puede tomar son: absolute, relative o static.
<i>z-index</i>	Define la posición del elemento en el tercer eje de coordenadas, permitiendo superponer unos elementos sobre otros como si fueran capas.

6.6.- Unidades de tamaño.

Las distintas unidades que podemos utilizar para indicar tamaños son las siguientes:

- **Relativas**
 - *Element (em)*: Expresa el tamaño relativo al tamaño de la fuente utilizada.
 - *X-height (ex)*: Expresa el tamaño relativo al de la letra "x".
 - *Pixel (px)*: Expresa el tamaño relativo a la resolución del monitor.
- **Absolutas**
 - *Milímetros (mm)*.
 - *Centímetros (cm)*: Cada centímetro son 10 mm.
 - *Pulgadas (in)*: Cada pulgada equivale a 2,54 cm.
 - *Puntos (pt)*: Cada punto son 1/72 in.
 - *Picas (pc)*: Cada pica son 12 pt.

6.7.- Definición y uso de clases.

Cuando las reglas de estilos se asocian a un documento HTML utilizando un fichero externo o incluyéndolas en el contenido de la etiqueta STYLE en la cabecera del documento, pueden definirse estilos y asociarlos a determinados elementos del documento.

Para definir una clase hay que usar la sintaxis siguiente:

.clase_azul{color:blue}

Para asociar un elemento HTML a una clase habrá que usar el atributo CLASS al usar dicho elemento en el documento HTML del siguiente modo:

<h3 class="clase_azul">El encabezado de tercer nivel es ahora azul</h3>

Para restringir la clase a un determinado elemento basta poner el elemento delante del punto al definir la regla. Por ejemplo, para restringir el uso de la clase a párrafos tendremos:

p.clase_azul{color:blue}