



**Y**

**CSS3**

INTRODUCCION

Le damos la bienvenida a este **CURSO DE HTML Y CSS**. El curso que acaba de comenzar en nuestro centro está realizado según las directrices pedagógicas que más se adecuan a la enseñanza de la informática. Le damos las gracias por elegirnos.

La evolución que ha seguido Internet y las posibilidades que ofrecen las nuevas versiones de los navegadores ofrecen unas oportunidades de diseño de páginas web más funcionales y potentes por lo que se hace necesario conocer los estándares de programación más actuales.

La profunda renovación que ha supuesto la evolución de **HTML** hace imprescindible conocer las aportaciones de las nuevas versiones para adaptar los diseños de las páginas web a los formatos más modernos.

Por otro lado, las hojas de estilo **CSS**, imprescindibles en las últimas especificaciones de **HTML**, suponen un avance verdadero en lo que respecta al diseño, las representaciones gráficas y la apariencia de las páginas web con nuevas aportaciones que les dan un formato más actualizado.

Por todo esto, y aunque no sabemos sus expectativas a la hora de comenzar este curso, estamos seguros de que al final se sentirá satisfecho al comprobar que es capaz de crear páginas web de carácter profesional con tan sólo **HTML** y hojas de estilo **CSS**.

**FRAY LUIS CENTRO ACADÉMICO** pretende conseguir con este **CURSO DE HTML Y CSS** los siguientes objetivos:

OBJETIVOS DEL CURSO DE HTML Y HOJAS DE ESTILO CSS.

* Conocer las etiquetas y sus funciones del lenguaje **HTML**.
* Conocer el funcionamiento de las hojas de estilo **CSS** y sus parámetros.
* Ser capaces de crear una página web desde cero utilizando las últimas especificaciones de **HTML** y **CSS**.
* Aplicar **HTML** y **CSS** conjuntamente para dar formato profesional a una página web basada en estándares.
* Crear páginas web de carácter profesional siguiendo las directrices y reglas del World Wide Web Consortium, (W3C), y validadas por él, garantizando por lo tanto la calidad del desarrollo y diseño de la página web.
* Aprender a crear páginas web más o menos sofisticadas, más interesantes, más atractivas y más útiles.

METODOLOGIA

El **CURSO DE HTML Y CSS** que tiene entre sus manos está realizado según las directrices pedagógicas que más se adecuan a la enseñanza de la informática y recoge la experiencia de más de 25 años de enseñanza de nuestro centro. Efectivamente; la metodología que seguimos en nuestros cursos de informática presentan las siguientes características pedagógicas:

* **ENSEÑANZA PERSONALIZADA**:

Cada alumno puede seguir el ritmo de aprendizaje que considere más oportuno, adecuándose a su propia capacidad de concentración, de comprensión, de rapidez lectora, de fijación de conceptos, del nivel motivacional, del cansancio, etc.

Por tanto, la velocidad en el avance dependerá de cada alumno dado que el mismo se producirá con la realización correcta de todos los ejercicios propuestos.

Es por esto que cada uno de nuestros profesores atiende personalmente a cada alumno, adecuándose a las peculiaridades particulares y profesionales de cada alumno.

* **ENSEÑANZA EMINENTEMENTE PRACTICA**:

Cada uno de nuestros alumnos dispone de un ordenador personal de alta gama, conectado en red, impresoras láser a color y escáner, para realizar las actividades que se le van proponiendo.

El alumno, desde la primera hora de clase, está trabajando y experimentando con **HTLM Y CSS**.

* **DIFICULTAD CRECIENTE**:

La explicación avanza en pequeños pasos de dificultad creciente.

La comprensión conseguida se confirma mediante la ejercitación en el propio ordenador, permitiendo aprender desde el error.

* **ENSEÑANZA INTERACTIVA**:

El aprendizaje exige constante participación activa del alumno, dado que se necesita la ATENCIÓN-CONCENTRACIÓN del mismo para poder avanzar en los contenidos, ya que las diferentes cuestiones sólo se pueden resolver si se presta suficiente atención a los contenidos que se proponen.

* **ENSEÑANZA MOTIVADORA**:

La enseñanza personalizada, eminentemente práctica, interactuando constantemente con el ordenador, además de la comprobación inmediata del éxito en las tareas que se proponen, ejerce como animador hacia el esfuerzo siguiente.

Teniendo en cuenta estas características pedagógicas de nuestro método, nos permitimos hacerle las siguientes:

Recomendaciones pedagógicas para un aprendizaje eficaz

* El orden en que se han editado los temas de cada bloque tiene una importancia pedagógica, al menos para un primer estudio de los mismos. La razón es que normalmente en cada tema se utilizan conceptos que se apoyan en contenidos explicados en los temas precedentes.
* Cada unidad didáctica, cada tema, guarda una estructura lógica interna. Por tanto, es recomendable estudiarla en el orden que se propone, puesto que los contenidos que se van explicando fundamentan los que a continuación aparecerán y viceversa.
* Es **muy importante realizar correctamente todos y cada uno de los ejercicios propuestos**. Nuestra experiencia confirma que los alumnos que se esfuerzan en realizar bien los ejercicios, comprenden y retienen con mayor facilidad los contenidos didácticos.
* El manual del curso está expresamente realizado pensando en usted y en sus ganas de aprender. **Léalo detenidamente**. En él está explicado, muy resumidamente, todo lo necesario para realizar los ejercicios que se le proponen. A veces, es conveniente realizar una especie de resumen-esquema que le ayudará a recordar con rapidez lo que ya ha estudiado.
* **Si tiene alguna duda, no pase adelante**. Resuélvala antes de continuar. Pregunte a su profesor. ¡Para eso está!
* El buen resultado del curso no guarda relación directa con la velocidad a la que son recorridos los temas. No se debe pretender abarcar demasiado espacio en poco tiempo. Cada contenido requiere su tiempo para asentarse convenientemente. Es tontería avanzar sin aprenderse bien los contenidos. Más adelante tendrá que volver a repasar y esto le hará perder tiempo.
* **¡MUY IMPORTANTE!** Para obtener el certificado o diploma de estudios expedido por el centro, correspondiente a este curso, será necesario demostrar que ha adquirido los conocimientos necesarios mediante una prueba o Control de Aprendizaje personal así como presentar realizados correctamente todos los ejercicios del temario.

Estamos seguros de que siguiendo estas recomendaciones dominará de una manera sencilla y cómoda todos los contenidos de éste **CURSO DE HTML Y CSS** que ahora comienza.

CAPÍTULO 1: CONCEPTOS PREVIOS

INTRODUCCIÓN

Este curso está pensado para todas aquellas personas que deseen desarrollar, crear y publicar páginas web de carácter profesional, más o menos sofisticadas, consiguiendo sitios web más interesantes, atractivos y útiles, utilizando para ello, la última tecnología del lenguaje **HTML** y algunas utilidades de apoyo que forman lo que podríamos llamar la “familia **HTML**”.

A la hora de elaborar este manual hemos procurado, como en el resto de los cursos de Fray Luis Centro Académico, fomentar la participación del alumno con numerosos ejemplos y ejercicios prácticos, pues consideramos que es la mejor manera de aprender cualquier utilidad informática.

Para ello, durante todo este curso y como tema principal vamos a crear una página Web sobre el turismo en España y en Valladolid, aunque también deberá crear otras páginas en los ejercicios de cada lección.

Es muy importante que siga paso a paso las instrucciones de este manual y que realice todos los ejercicios correctamente puesto que, cada lección y cada ejercicio, está basado en los anteriores.

Elementos necesarios para crear una página web.

Las páginas Web se crean utilizando un lenguaje llamado **HTML** complementado con unas hojas de estilo **CSS**.

Una página **HTML** y sus correspondientes estilos **CSS**, se crean en un editor de texto y luego se carga en un navegador. Por lo tanto los medios imprescindibles para crear páginas Web son esos:

**•Un editor de texto**: Vale cualquier editor de texto, nosotros utilizaremos el bloc de notas incluido gratuitamente en Windows. Como norma general no se pueden crear archivos **HTML** en procesadores de texto aunque algunos, como Word, ofrecen la posibilidad de guardar los ficheros como "Sólo texto" para que no introduzca órdenes de formateo, que pueden provocar errores al cargarlo en el navegador.

También existen programas que facilitan la escritura de código **HTML** y **CSS**, entre otros: Notepad++, Sublime Text, Home Site, UltraEdit, Arachnophilia, Quanta, BlueGriffon, Aloha Editor, Rendera, etc, etc. (Aunque los editores de **HTML** faciliten la tarea de escribir código **HTML**, recomendamos que para empezar y sobretodo aprender, utilice el bloc de notas. Cuando tenga más experiencia podrá utilizar el editor que más le convenza).

**•Un navegador del Web**: Este permite visualizar la página web y comprobar que todos los elementos están situados correctamente y que cumplen su función. En la actualidad existen los siguientes navegadores que entienden las últimas especificaciones de **HTML**: IE, Firefox, Chrome, Safari y Opera, todos ellos en sus últimas versiones. Nosotros utilizaremos preferentemente Firefox.

HISTORIA DEL HTML

El **HTML** (Hyper Text Markup Language), es el lenguaje utilizado por los navegadores para mostrar las páginas web a los usuarios. Este lenguaje nos permite utilizar textos, sonidos e imágenes para combinarlos y mostrarlos en la web a nuestro gusto. Además nos permite la introducción de referencias a otras páginas por medio de los enlaces hipertexto.

El **HTML** se creó con objetivos divulgativos. Al no preverse el enorme desarrollo que llegaría a tener internet con sus propiedades multimedia y sus diversos usos, el **HTML** no se creó con la suficiente planificación por lo que se han tenido que ir realizando modificaciones y mejoras a lo largo del tiempo.

El W3C, (World Wide Web Consortium), es el organismo que ha ido realizando esas modificaciones y mejoras. Es quien indica las especificaciones que tiene que tener el lenguaje **HTML** para que los navegadores puedan entenderlo y los desarrolladores de páginas web sepan a qué instrucciones atenerse.

En 1998, el W3C finalizó la especificación **HTML 4.0** que es la especificación en la que se basan la gran mayoría de las páginas web que podemos ver hoy en día.

El W3C siguió realizando estándares y creo un lenguaje llamado **XML**, con reglas más estrictas que el **HTML**, y recomendando que se dejara de utilizar este, ya que así se desarrollarían webs más consistentes y con menos errores.

Con el mismo objetivo, en el año 2000, el W3C reformuló el **HTML** para adaptarlo a las reglas más estrictas de **XML**. Nació así el **XHTML** que no fue muy aceptado por los desarrolladores de webs que siguieron programando en **HTML 4.0.**

El desarrollo de la versión 2.0 del **XHTML** que pretendía ser un punto de no retorno en el desarrollo de páginas web, conllevó que no fuera compatible con todo lo que existía hasta la fecha incluyendo al **HTML**.

Esta ausencia de compatibilidad y diversos problemas de tipo técnico, supuso un gran descontento entre los fabricantes de los navegadores, que en 2004 crearon un grupo disidente del W3C llamado WHATWG, (Web Hypertext Application Technology Working Group), formado por Firefox, Opera y Safari y a los que se les unió Google Chrome. Su objetivo consistió en crear un lenguaje **HTML** mejor, con nuevas aplicaciones para la web, pero manteniendo la compatibilidad con los navegadores existentes.

Durante unos años, W3C y WHATWG, trabajaron en paralelo hasta que en 2006 decidieron unir sus fuerzas y trabajar conjuntamente abandonándose el **XHTML 2.0** y naciendo la especificación **HTML5** que, a fecha de elaboración de este curso, aún no se ha terminado de fijar completamente. (Se ha anunciado que estará totalmente terminada para el 2022).

Por último, a la vez que la elaboración del estándar **HTML5**, se está reformulando este para adaptarlo a las reglas del **XML**, con el nombre de **XHTML5**.

Toda esta intensa historia tiene unas consecuencias importantes para los desarrolladores de páginas web y para este curso:

* Gracias a que el W3C tomó el liderazgo del desarrollo de **HTML5** tenemos la garantía de que se trata de un estándar gratuito que no depende de una o varias empresas.
* El hecho de que algunas empresas fabricantes de navegadores no participaran en un primer momento en el grupo WHATWG hace que no todos los navegadores actuales soporten todas las posibilidades de **HTML5** lo que habrá que tener en cuenta para que las páginas web que desarrollemos se visualicen en todos los navegadores lo mejor posible. Siempre que sea posible haremos referencia a la compatibilidad de los distintos navegadores con las últimas utilidades implementadas en **HTML5**.
* El **HTML5** no es un estándar terminado por lo que habrá nuevas posibilidades que aún no están implementadas y características actuales que en un futuro podrán desaparecer.
* Todas las características que estudiaremos en este curso están actualizadas a **HTML5** habiéndose eliminado aquellas de **HTML 4.0** que no se conservan en esta nueva implementación.
* A lo largo de este curso intentaremos utilizar la sintaxis estricta para asegurar la compatibilidad con **XHTML**.

HOJAS DE ESTILO CSS

Hasta **HTML 4.0**, la especificación de este lenguaje incluía órdenes destinadas al contenido de una página web así como al formato aplicado a dicho contenido. En **HTML5** se separa el contenido de la presentación dejando este último en manos de las **HOJAS DE ESTILO CSS**, (Cascading Style Sheets).

Así que las **HOJAS DE ESTILO CSS**, aunque no forman parte de **HTML5**, se convierten en elementos inseparables del **HTML**, y ya no es posible abordar su estudio por separado. Es por ello que a lo largo de este curso iremos alternando partes de **HTML** propiamente dicho y partes de **CSS**.

Las **hojas de Estilo** **CSS** es un lenguaje en evolución que también se identifica mediante números de versión, y la última es **CSS3**. Como **CSS3** ha ido evolucionando en paralelo a **HTML5**, muchos de sus componentes suelen considerarse como **HTML5**, pero es importante conocer que son diferentes, aunque sí están íntimamente ligados, como ya hemos comentado e iremos comprobando.

Resumiendo, las **HOJAS DE ESTILO CSS** son órdenes de formato, color y fuente del texto, color de fondo, bordes, etc, etc, que el lenguaje **HTML** y el navegador entiende y que permiten a los desarrolladores el control total sobre el estilo y el formato de las páginas web.

El desarrollo de las **HOJAS DE ESTILO CSS** se hizo necesario para dar más potencia al lenguaje **HTML** a la hora de aplicar formatos al documento y para facilitar la lectura y el depurado del código **HTML** que se hacía muy pesado al tener mezclado las órdenes de contenido, formateo y estilo de la página web.

Por último, tenemos que nombrar al **DHTML**, es posible que en alguna ocasión oiga hablar de él o del **HTML dinámico** como también se le conoce. Es preciso que conozca que no es otro lenguaje de creación de páginas web diferente. Más bien se trata de un concepto de programación web que engloba varias herramientas como son: **HTML**, **CSS** y lenguajes de Script como **JavaScript**, **VisualBasic Script**, **PHP** y otros que son objetos de otros cursos diferentes. (Pregunte a su profesor por ellos).

CAPÍTULO 2 - COMENZANDO A TRABAJAR

Ha llegado el momento de empezar a trabajar con **HTML**. Para familiarizarnos con él vamos a abrir una página Web ya creada que servirá de base para todo el curso.

NUESTRA PRIMERA PÁGINA

El nombre de la página Web que vamos a abrir es **turismo.html** (Lo encontrará en su carpeta de prácticas).

***Es el momento de realizar la práctica 2.1.***

LAS ETIQUETAS

El uso de las etiquetas es el principio esencial del lenguaje **HTML**. Hay dos tipos de etiquetas: de inicio y de cierre. En las etiquetas de inicio simplemente se escribe la orden, **<***orden***>**, mientras que en las etiquetas de cierre se escribe la orden precedida de una barra, **</***orden***>.** Por regla general, a toda etiqueta de apertura le corresponde una etiqueta de cierre. Pero también puede haber etiquetas sin cierre.

La porción de texto situada entre las etiquetas de inicio y de cierre se ve afectada por la orden de las etiquetas. Por ejemplo, la etiqueta **<h1>** y su correspondiente **</h1>** determinan que el texto situado entre ellas, “**Turismo en España**”, es un título de nivel 1. Al conjunto de etiqueta de apertura, etiqueta de cierre y texto u objeto incluido entre ellas, le denominamos **ELEMENTO**.

Un ejemplo de etiqueta sin cierre, también llamadas etiquetas únicas o vacías, es **<img>** cuyo significado veremos un poco más adelante.

A un **ELEMENTO** se le pueden aplicar varias etiquetas. Diremos entonces que las etiquetas están anidadas o enlazadas.

***Realice la práctica 2.2.***

PARTES DE UN DOCUMENTO HTML MÍNIMO

Un documento **HTML** tiene que tener como mínimo las siguientes partes:

**•EL DOCTYPE**: Indica al navegador a que versión de **HTML** se ajusta el documento.

Todo documento **HTML** debe comenzar, en su primera línea, por un doctype. En **HTML5** es muy sencillo:

**<!DOCTYPE html>**

El doctype le dice al navegador qué tipo de código fuente, **HTML** y **XHTML**, debe interpretar y representar. Lo usará para representar la página según los estándares de W3C por lo que se deberá visualizar de manera idéntica en todos los navegadores.

El doctype no es obligatorio, pero si es una buena práctica. En su ausencia, al no saber a qué reglas atenerse, el navegador se acoge a unas reglas genéricas propias de cada navegador, por lo que puede suceder, que el mismo código, se vea diferente en los distintos navegadores.

Por si acaso se encuentra con él en alguna página web que desee investigar su código, el doctype de **HTML 4.01** es:

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

Y en **XHTML 1.0**:

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

Por último el doctype no debe ser considerado como una etiqueta **HTML**, sino como una declaración del lenguaje que estamos utilizando, por lo que no es necesario utilizar una etiqueta de cierre.

**•LA ETIQUETA <html>**: Indica al navegador que se trata de un documento **HTML**. Es el elemento raíz del documento. Se declara justo debajo de la declaración del doctype.

***Es el momento de realizar la práctica 2.3.***

Note que la etiqueta **<html>** puede incluir, como en este caso, el atributo **lang=”es”** que le dice al navegador que el documento que va a procesar está en español. Además, está información es muy valiosa para los motores de búsqueda cómo Google. Veremos más sobre los atributos de las etiquetas en un apartado posterior.

**•EL ENCABEZADO**: comprendido entre las etiquetas **<head>** y **</head>**, que se colocan justo después de la etiqueta **<html>** y contiene el encabezado del documento con una serie de información relativa al propio documento. El encabezado no se muestra en pantalla pero es de gran utilidad para el navegador, para los motores de búsqueda y para el desarrollador. Entre la información que irá en el encabezado del documento **HTML** está:

* El título del documento entre las etiquetas **<title> … </title>**, que explicamos en el siguiente apartado.
* Las declaraciones de las hojas de estilo **CSS**.
* Funciones o vínculo a archivos **JavaScript**.
* Información dirigida a los motores de búsqueda para mejorar el posicionamiento de la página.
* Información dirigida a los navegadores.
* Meta-etiquetas que indican la codificación, la descripción de la página, las palabras clave asociadas a la página, el nombre del autor, menciones de copyright, etc, etc.

Dentro del encabezado debemos destacar dos elementos que no deben faltar:

**A.- El Título del documento**.

Como ya hemos visto la etiqueta **<title>Turismo en España</title>** asigna un título a la página que se visualiza en la barra de título de la ventana del navegador y/o en las pestañas abiertas por el mismo. Igualmente es el título con que se guarda la página cuando la agregamos a favoritos.

Tiene especial importancia en lo que respecta al posicionamiento en buscadores ya que es en lo primero que se fijan en las búsquedas e indexaciones, por lo que habrá que procurar que el título de la página sea atractivo y sintético reflejando la temática de nuestro sitio web.

No se puede tener más de un elemento **<title>** en un documento **HTML**.

**B.- La codificación del documento**:

Indica al navegador el juego de caracteres, (charset), a utilizar. Ejemplos de codificación son: ASCII, Windows-1252, iso-, etc.

Habitualmente se utilizan los juegos de caracteres utf-8, iso-8859-1 e iso-8859-15.

Forma parte de las buenas prácticas del diseño de páginas web definir explícitamente la codificación del documento.

En **HTML5** se utiliza la etiqueta: **<meta charset=“utf-8” />** para indicar el juego de caracteres que vamos a utilizar en nuestra página web.

***Realice ahora la práctica 2.4.***

**•EL CUERPO**: comprendido entre las etiquetas <**body**> y <**/body**> y que es donde se sitúa el contenido de la página web que verá el usuario en su navegador.

Concluyendo: En cualquier código de cualquier página Web creada con este lenguaje, la primera etiqueta, a parte del **doctype**, será <**html**> y la última <**/html**>.

El texto escrito en el encabezado no se visualiza cuando se abre la página pero no por eso deja de tener importancia ya que el texto del encabezado situado además entre las etiquetas de título (<**title**> y <**/title**>) es el nombre que adoptará nuestra página cuando se guarde en la carpeta de favoritos y por el que lo encontrarán los buscadores. Por esta razón el título debe ser breve y descriptivo.

Dentro del cuerpo de la página está todo lo que queremos que aparezca en pantalla, texto, imágenes, videos, etc, etc…

LOS TÍTULOS

El texto normal se escribe simplemente entre las etiquetas <**body**> y <**/body**>. Pero observe que si queremos que el texto tenga una título deberemos escribir ese título entre las etiquetas <**h1**> y <**/h1**>, <**h2**> y <**/h2**>, etc. (Hasta el número seis). El número es indicador del tamaño, así el número uno corresponde al título de mayor tamaño o importancia y el seis al de menor tamaño.

Observe que no debe utilizar estas etiquetas de título para variar el tamaño de los caracteres, para eso utilizaremos las hojas de estilo **CSS**.

En nuestra página el título "**Turismo en España**" es el principal y por eso se encuentra entre las etiquetas <**h1**> y <**/h1**>. El título "**Introducción**" por ser de menor importancia se encuentra entre las etiquetas <**h2**> y <**/h2**>.

Tenga en cuenta que las etiquetas de título:

* Por defecto escriben el texto en negrita.
* Por defecto incluyen una línea vacía entre el contenido de la etiqueta y el resto del documento.
* Si desea conseguir caracteres más grandes o más pequeños que los que **HTML** proporciona por defecto, habrá que utilizar hojas de estilo **CSS**.
* Las que aparecen al principio de la página web son tenidas muy en cuenta en los motores de búsqueda e influyen en la clasificación y recomendación de los sitios web.

LOS PÁRRAFOS

El texto normal de la página, en el bloc de notas, está escrito en líneas tabuladas pero observe que éstas no se corresponden con las de la página cargada en el navegador porque los saltos de línea y las tabulaciones no son identificados en el lenguaje **HTML**. Aunque, si observa bien, los saltos de línea que aparecen en el archivo de código HTML se sustituyen por un espacio en la visualización de la página web en el navegador. ¡Compruébelo!

Si se quiere comenzar un nuevo párrafo separado por una línea, dicho párrafo deberá estar precedido por la etiqueta <**p**> y su correspondiente de cierre.

Si lo que se desea es insertar un salto de línea utilizaremos la etiqueta <**br**> que no tiene una correspondiente de cierre (De ahí que utilicemos la etiqueta de la siguiente manera <**br /**> que explicaremos más adelante.)

Al utilizar la etiqueta <**p**>, por defecto, cada párrafo está precedido y seguido del espacio correspondiente a un salto de línea para marcar la separación con el contenido anterior y posterior. Este espacio puede modificarse mediante las hojas de estilo **CSS**.

Si se repite la etiqueta <**p**>, una detrás de otra, no se insertan líneas en blanco. Podría utilizar para ello la etiqueta <**br /**> pero recomendamos, también para este caso, hacerlo mediante las hojas de estilo **CSS**.

***Es hora de prácticar. Realice la práctica 2.5.***

ATRIBUTOS DE ETIQUETA

En ocasiones es preciso completar una etiqueta con instrucciones más precisas. Para ello se utilizan los atributos de etiqueta. El atributo se inserta en la etiqueta, entre la palabra de instrucción y el signo “>” final.

La sintaxis completa de una etiqueta con un atributo es:

**<etiqueta atributo=”valor”> … </etiqueta>**

El atributo contiene siempre un valor, que se indica mediante el signo igual (=) seguido del valor escrito entre comillas. El respeto estricto a la sintaxis exige que no exista espacio antes o después del signo igual.

En nuestra primera página web hemos incluido un atributo en la etiqueta <**html**>:

**<html lang="es">**

para indicar que el idioma utilizado en nuestra página web es el español.

Es posible utilizar varios atributos simultáneos, separados por espacio, en una misma etiqueta:

**<etiqueta atributo1=”valor” atributo2=”valor” > … </etiqueta>**

USO CORRECTO DE LAS ETIQUETAS

Antes de meternos en más profundidades conviene apuntar algunas reglas simples que será preciso respetar a la hora de escribir etiquetas en **HTML5**:

* En **HTML5** las etiquetas no distinguen entre mayúsculas y minúsculas, así que se pueden escribir de las dos maneras. Hay desarrolladores que utilizan la escritura de las etiquetas en mayúscula para distinguirlas del resto del documento, pero el uso generalizado es escribirlas en minúsculas.
* Por regla general se exige que toda etiqueta abierta se cierre.
* Las etiquetas deben enlazarse correctamente. Cuando se aplican varias etiquetas a un elemento, el orden de estas es esencial. La primera etiqueta de cierre debe corresponder con la última etiqueta de apertura que no haya sido cerrada como se ve en el siguiente ejemplo:
  + Correcto: **<a><b><c> contenido <c><b><a>**
  + Incorrecto: **<a><b><c> contenido <a><c><b>**
* Los valores de los atributos deben escribirse siempre entre comillas.

SINTÁXIS XHTML

Cómo comentamos al principio de este curso las páginas escritas en **XHTML** se pueden ver también en los navegadores actuales ya que las etiquetas y atributos de **XHTML** y **HTML** siguen siendo las mismas, pero la sintaxis del código **XHTML** es más estricta.

Las diferencias más importantes entre **XHTML** y **HTML** son estas:

* En **XHTML**, todas las etiquetas deben escribirse en minúsculas.
* **XHTML** exige que todas las etiquetas de apertura lleven su correspondiente etiqueta de cierre. Todas las etiquetas **XHTML** deben tener cierre, aun en el caso de etiquetas especiales que técnicamente no necesitan etiquetas de apertura y cierre. De ahí que, en la etiqueta <**br**>, que hemos utilizado para insertar un salto de línea, hayamos utilizado una sintaxis especial de autocierre para hacer nuestra página web compatible con **XHTML**. Recuerde que una etiqueta que se cierra ella misma tiene este aspecto (con un espacio y la barra inclinada antes del signo ">"): <**br /**>
* **XHTML** exige que el anidamiento de etiquetas (su inclusión unas dentro de otras), siga unas reglas estrictas.

Puesto que **HTML5** admite etiquetas **XHTML**, como ya hemos comentado, en este curso, utilizaremos código ajustado a **XHTML** para asegurar la compatibilidad hacia atrás, ayudar a que nuestros diseños sean compatibles con los navegadores y dispositivos móviles actuales, y también porque así es más fácil la resolución de errores cuando aparecen.

INSERCIÓN DE SEPARADORES

Otra etiqueta utilizada en nuestra página web es <**hr /**> que no necesita etiqueta de cierre porque sólo sirve para separar el contenido, o indicar un cambio de tema, en una página **HTML**. Añade un separador horizontal tan ancho como la pantalla y embutido en el fondo (no abuse de los separadores en sus páginas porque, en exceso, dan un aspecto fraccionado a la pantalla).

***Para terminar de dominar este tema realice ahora la ejercitación del capítulo 2.***

CAPÍTULO 3 – EMPEZANDO A DAR FORMATO

VALIDACIÓN DEL CÓDIGO HTML

Como ya hemos comentado, W3C es una organización sin ánimo de lucro cuyo objetivo consiste en orientar la evolución de la Web. Ofrece directrices y reglas para especificaciones como **HTML** y **XHTML**. Es posible comprobar la validad del código de nuestras páginas web utilizando el servicio online gratuito de validación del W3C.

***Para comprobar como se valida el código realice ahora la práctica 3.1***

Observe que la validación nos dice si el código está bien, si faltan etiquetas o no están bien escritas, pero no si la página es bonita.

El proceso de validación puede ser útil para resolver errores, ya que nos facilita la detección de cualquier problema de sintaxis. Tener un código conforme es uno de los pasos que nos garantizará que nuestras páginas web se van a ver en el mayor número posible de navegadores y dispositivos.

La validación de páginas del W3C es el aspecto más tangible de un diseño web conforme a estándares, pero hay también otras ventajas importantes al crear páginas bien estructuradas:

* Menos código: el uso de **HTML** y **CSS** nos permite crear páginas similares con menos líneas de código, es decir, menos trabajo a la hora de desarrollarlas y más rápida visualización en el navegador.
* Mantenimiento más sencillo: tener menos código supone un mantenimiento más sencillo, y esto beneficia tanto al que escribe el código como a cualquier otra persona que después deba encargarse de su mantenimiento y revisión.
* Accesibilidad: los documentos creados de forma semántica, es decir, que utilizan la etiqueta **HTML** más adecuada a su función en cada momento, pueden facilitar la lectura y navegación a personas con discapacidad visual y el visitante de la web encuentra la información más fácilmente.
* Optimización para motores de búsqueda: Las páginas web que incluyen secciones claras e identificadas de forma lógica, tanto a nivel de código como de contenidos, facilitan la labor a los motores de búsqueda a la hora de indexarlas y clasificarlas.
* Compatibilidad: los sitios web que independizan la estructura con respecto al estilo se adaptan mucho más fácilmente a dispositivos móviles y formatos de pantalla variados. **CSS** nos permite aplicar hojas de estilo alternativas que optimizan la presentación en pantalla dependiendo del dispositivo utilizado.
* Demostración de su propia capacidad y profesionalidad a la hora de realizar un código de calidad, conforme a los estándares de **HTML5**.

NEGRITA Y CURSIVA

Los dos métodos fundamentales para resaltar texto son los caracteres en negrita y en cursiva. Para que una parte del texto aparezca en negrita hay que situar ese texto entre las etiquetas **<b>** y **</b>** (Bold).

Para que el texto aparezca en letra cursiva hay que situarlo entre las etiquetas **<i>** y **</i>** (Italic).

También disponemos de la etiqueta <**strong**> … <**/strong**> que escribe texto en negrita en los navegadores. Esta etiqueta se diferencia de <**b**> en que si bien el efecto visual es el mismo, la interpretación es muy distinta en los programas de sintetización de voz que leen las páginas web para personas con deficiencias visuales, consiguiendo una entonación de voz distinta o un nivel sonoro más elevado.

Por otro lado, tenemos que hablar también de la etiqueta <**em**> … <**/em**>, (énfasis), que también escribe el texto en cursiva en los navegadores pero se diferencia de la etiqueta <**i**> en que tiene por objeto poner de relieve el texto que la contiene.

***Veálo en funcionamiento estas etiquetas realizando la práctica 3.3***

SUBINDICES Y SUPERINDICES

En ocasiones es necesario escribir algunos caracteres como exponentes o subíndices, por ejemplo para fórmulas matemáticas.

La etiqueta <**sup**> … <**/sup**> escribe el contenido en exponente y la etiqueta <**sub**> … <**/sub**> en subíndice.

TEXTO TACHADO

La etiqueta <**del**> … </**del**>, (delete), permite marcar un elemento de texto como tachado. Un ejemplo podría ser la actualización de un precio en una web comercial.

TEXTO PREFORMATEADO

La etiqueta <**pre**> … <**/pre**> permite visualizar el texto tal y como se ha codificado en el editor de texto. Los espacios, tabulaciones y saltos de línea se respetan tal cual en la pantalla y el texto se visualizará con un tipo de letra de paso fijo, normalmente Courier.

Como podrá haber observado la etiqueta <**pre**> se puede utilizar para mostrar texto con un formato inusual, o algún tipo de código informático.

DIRECCIÓN DEL TEXTO

Ahora una curiosidad. Algunos sistemas de escritura, como el árabe, se escriben de derecha a izquierda. Para ello podemos utilizar la etiqueta <**bdo**> … <**/bdo**>.

Esta etiqueta tiene dos atributos:

* dir=”ltr” (left to rigth) para el sentido de escritura de izquierda a derecha.
* dir=”rtl” (right to left) para el sentido de derecha a izquierda.

CITAS

La etiqueta <**blockquote**> y <**/blockquote**> se utiliza para destacar una cita textual dentro de un texto general. ¡Veámoslo!

Observe que la cita se visualiza con un salto de línea antes y otro después, igual que con los párrafos. El texto que contienen estas etiquetas aparece con márgenes a ambos lados de manera que destaca sobre el resto de la página.

Existe también la etiqueta <**q**> … </**q**> para las citas pero esta visualiza el texto entre comillas, no muestra espacio antes ni después, ni tampoco los márgenes a ambos lados.

***Es el momento de realizar la práctica 3.4***

COMENTARIOS NO VISIBLES

Puede que alguna vez necesite poner un comentario o nota en su programación de código **HTML** y no quiere que se visualice en su página Web. Para ello utilizaremos las etiqueta **<!--** que se cierra con **-- >**.

Por ejemplo: <!-- Este texto no se verá en mi página Web -->

Todo lo que escribamos entre estas etiquetas no se visualizará en la pantalla del navegador pero si en el archivo de código fuente. Los comentarios pueden ocupar varias líneas y pueden ir en cualquier parte de la página web, tanto en el <**head**> como en el <**body**>.

El uso de comentarios que documenten el código, es una de las buenas prácticas de todo buen diseñador de páginas web.

***Para terminar de dominar este tema realice ahora la ejercitación del capítulo 3.***

CAPÍTULO 4 – EMPEZANDO CON HOJAS

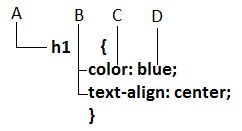
DE ESTILO CSS

Como ya hemos comentado, con **HTML5**, se establece una separación entre el contenido de la web y su formato o presentación. **HTML** se encarga solamente de la estructura del documento, y las hojas de estilo **CSS** se encargan de todo el aspecto relacionado con el formato visual.

En el capítulo anterior hemos utilizado algunas etiquetas para dar formato como cursiva o negrita. Efectivamente las hemos utilizado para iniciarnos en el aprendizaje del **HTML**, pero lo normal será utilizar hojas de estilo **CSS** y no esas etiquetas.

Las hojas de estilo pues, abarcan todo aquello relacionado con la presentación del documento **HTML** como el color, la alineación, el tamaño, los bordes, los fondos, etc, etc… Veamos qué son y cómo utilizarlas.

LA SINTAXIS DE CSS

 En la siguiente imagen puede observar una regla de **CSS**:

En ella podemos distinguir las siguientes partes:

A.- **Selector**. Se trata de los elementos o etiquetas del código **HTML** que se verán afectados por las órdenes de las hojas de estilo. En este caso todos los títulos de nivel 1 (**h1**). Tenga en cuenta que los selectores en una regla **CSS** se escriben simplemente con el nombre de la etiqueta sin los signos **<** y **>**.

Es posible aplicar el mismo efecto de estilo a distintos selectores. En este caso, cada selector implicado se escribe separado por una coma. Por ejemplo, la siguiente regla de estilo afectaría a los selectores h1, h2 y h3:

**h1, h2, h3 {**

**Color: blue;**

**Text-aling: center;**

**}**

B.- **Declaración**. Se refiere al conjunto propiedad-valor. Por ejemplo la declaración “**color: blue**” podríamos interpretarla como “poner en color azul”. En esta regla hay dos declaraciones. Es decir, que una regla de estilo puede incluir varias declaraciones para un mismo selector. Todas las declaraciones terminan con un punto y coma obligatoriamente.

C.- **Propiedad**. Indica el elemento de presentación que se ve implicado; por ejemplo “**color**” para el color, “**text-align**” para la alineación… La propiedad está separada de su valor mediante dos puntos. Para facilitar la legibilidad del código es buena costumbre dejar un espacio entre los dos puntos y el valor.

D.- **Valor**. Especifica una palabra clave, un porcentaje o un tamaño dependiendo de la propiedad a la que se refiera.

Toda la regla de estilo está contenida entre dos llaves **{** y **}**. Si nos olvidamos cualquiera de las dos llaves, la regla de estilo no será tenida en cuenta por el navegador.

Las hojas de estilo no distinguen entre mayúsculas o minúsculas aunque lo habitual es utilizar siempre minúsculas.

Concretando, la regla de estilo de la imagen le dice al navegador que todos los títulos de nivel 1 de la página web se escriban en color azul y con el texto centrado.

INCORPORACIÓN DE LA HOJA DE ESTILO A LA WEB

Existen 4 maneras diferentes de utilizar las hojas de estilo CSS en los archivos HTML:

* **Integradas en una etiqueta HTML.** En este caso se indicaría la declaración como si fuera un atributo más de la etiqueta. Por ejemplo:

**<h1 style=”color: blue;”>Título de nivel 1</h1>**

Aunque puede resultar útil en alguna ocasión, y en virtud del principio de separación entre contenido y presentación, lo habitual es evitar esta forma de utilizar los estilos.

* **Internas al documento HTML.** Las reglas de estilo se sitúan agrupadas en el encabezado del documento dentro las etiquetas <**head**> y <**/head**>. La sintaxis es la siguiente:

**<style type=”text/css”>**

**Las diferentes reglas de estilo…**

**</style>**

* **Externas al documento HTML.** La estudiaremos más adelante.
* **Importadas.** La estudiaremos más adelante.

***Aprenda a aplicar hojas de estilo realizando la práctica 4.1***

TIPOS DE SELECTORES CSS

Hasta ahora, en nuestras reglas de estilo, hemos utilizado selectores de etiqueta o selectores de tipo.

Se habrá dado cuenta que utilizando selectores de tipo, estamos limitados, puesto que se aplica el estilo a todas las apariciones de una etiqueta en el documento, pero no podemos aplicar un estilo a alguna cosa que no coincide exactamente con una etiqueta. Por ejemplo, a un trozo de texto, como aplicamos en su momento la negrita y cursiva. O asignar un color a una sola palabra. O asignar un formato sólo a una de las etiquetas y no a TODAS las que aparecen en el documento.

Para ello utilizaremos otros tipos de selectores:

A.- **Selectores de clase**. La declaración de clase se realiza escribiendo un punto seguido del nombre que queremos dar a la clase y seguido de la declaración de estilo escrita entre llaves, siguiendo este formato:

**.nombre\_de\_la\_clase { declaraciones de estilo }**

Por ejemplo:

**.azul {color: blue;}**

Para indicar que se aplique la clase se utiliza como si fuera un atributo de la etiqueta a la que queremos aplicar el estilo, de la siguiente manera:

**<etiqueta class=”nombre\_de\_la\_clase”>**

Hay que tener en cuenta que:

* El punto del selector de clase no se utiliza a la hora de llamar a la clase **class.**
* Una misma clase se puede invocar varias veces en un mismo documento **HTML**.
* Un nombre de clase puede estar formado por letras, cifras, el guión y el carácter de subrayado. El primer carácter no puede ser un número, un guión o un carácter de subrayado.

***Vamos a aplicarlo realizando la práctica 4.2***

B.- **Selectores de Identificador**. El selector **id**, también llamado identificador **id**, permite aplicar una hoja de estilo, de la misma forma que el selector **class**, aunque sólo se podrá utilizar una única vez en el documento **HTML**. La declaración de identificador se realiza escribiendo una almohadilla (**#**) seguida del nombre que queremos dar al identificador **id** y seguido de la declaración de estilo escrita entre llaves. Siguiendo este formato:

**#nombre\_del\_identificador { declaraciones de estilo }**

Por ejemplo:

**#azul {color: blue;}**

Esta selección sólo se podrá utilizar una única vez en el documento.

Para indicar que se aplique el identificador, se utiliza como si fuera un atributo de la etiqueta a la que queremos aplicar el estilo, de la siguiente manera:

**<etiqueta id=”nombre\_del\_identificador”>**

Hay que tener en cuenta que:

* Un identificador **id** no puede figurar más de una vez en el documento **HTML** en caso contrario, ¡el resto del código será incorrecto!
* Un documento puede contener varios identificadores **id** con nombres diferentes aunque sólo pueden utilizarse una única vez.
* Habitualmente, el identificador **id**, se utiliza para elementos **HTML** generales o estructurales que aprenderemos más adelante como <**section**>, <**article**>, etc, etc.
* Es posible mezclar declaraciones de estilo **class** e **id**.
* La almohadilla (**#**) del selector de identificador no se utiliza a la hora de llamar al **id**.
* Un nombre de identificador puede estar formado por letras, cifras, el guión y el carácter de subrayado. El primer carácter no puede ser un número, un guión o un carácter de subrayado.
* **Id** es más apropiado que **class** para usarlo en referencias con **JavaScript**.

Por último debemos de mencionar al asterisco (**\***) como **selector universal** que podemos utilizar para representar a cualquier tipo de elemento **HTML**.

COMENTARIOS EN CSS

Al igual que en **HTML**, en una hoja de estilo **CSS**, podemos y debemos insertar comentarios que documenten el código escrito para facilitar posteriores modificaciones y/o cambios.

En una hoja de estilo **CSS**, un comentario comienza con los caracteres de apertura barra oblicua y asterisco (**/\***) y termina con los caracteres asterisco y barra oblicua (**\*/**), siguiendo la siguiente forma:

**/\* Así se escribe un comentario en CSS \*/**

UNIDADES DE MEDIDA EN HOJAS DE ESTILO CSS

Las hojas de estilo **CSS** permiten utilizar numerosas unidades de medida de longitud distinguiendo entre:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UNIDADES ABSOLUTAS** | | | **UNIDADES RELATIVAS** | | |
| Especifican un tamaño fijo y están ancladas a alguna medida física. | | | Especifican un tamaño en relación a otra medida. | | |
| **UNIDAD** | **NOMBRE** | **EJEMPLO** | **UNIDAD** | **DESCRIPCIÓN** | **EJEMPLO** |
| pt | Punto. Utilizado en imprenta. | 48pt | em | Unidad relativa que se basa en el tamaño del tipo de letra del elemento padre. | 1.8em |
| pc | Pica | 4.5pc | rem | Unidad relativa que se basa en el tamaño del tipo de letra por defecto de la página. | 1.8rem |
| mm | Milímetro | 60mm | ex | Unidad relativa a la altura de la letra “x” minúscula en el elemento seleccionado. | 1.3ex |
| cm | Centímetro | 6cm | % | Porcentaje. También se basa en el tamaño de letra del elemento padre. | 80% |
| in | Pulgada (inch) | 0.1in | vw | Unidad relativa igual al 1% de la anchura de la ventana del navegador. | 8vw |
|  |  |  | vh | Unidad relativa igual al 1% de la altura de la ventana del navegador. | 3.0vh |
|  |  |  | vmin | Unidad relativa igual a la más pequeña de “vw” o “vh”. | 2vmin |
|  |  |  | vmax | Unidad relativa igual a la más grande de “vw” o “vh”. | 2vmax |
| px | Se trata de una medida absoluta. Un píxel es la parte más pequeña de una imagen. Algunos autores la consideran una medida relativa debido a que el tamaño del piexel depende de la resolución de pantalla. De hecho, hasta la especificación CSS2.1, se considerá una medida relativa. | | | | 220px |

Profundizaremos en algunas de estas unidades de medida a lo largo del curso.

***Para terminar de dominar este tema realice ahora la ejercitación del capítulo 4.***

CAPÍTULO 5 - LISTAS Y ENLACES EN HTML

CREACION DE LISTAS

Las listas son una forma muy eficaz de estructurar muchos contenidos. Podemos distinguir 4 tipos de listas:

A.- **Listas no numeradas o no ordenadas**. Se trata de una lista en la que cambiando el orden de los diferentes elementos no cambia el significado del documento. Los diferentes navegadores presentan los elementos de la lista con viñetas determinadas por ellos.

Las listas no numeradas se indican con las etiquetas <**ul**> … <**/ul**>. Cada elemento interior de la lista se indica con las etiquetas <**li**> … <**/li**>. La estructura de una lista no numerada es:

**<ul>**

**<li>Primer elemento de la lista</li>**

**<li>Segundo elemento de la lista</li>**

**<li>Tercer elemento de la lista</li>**

**<li>Cuarto elemento de la lista</li>**

**</ul>**

***Veamos cómo aplicarlas realizando la práctica 5.1***

B.- **Listas ordenadas**. Si la lista esta ordenada, es decir que lleva números por delante, habrá que poner <**ol**> antes de escribir la lista y <**/ol**> cuando se termine, y seguiremos poniendo <**li**> al empezar cada línea de la lista, así:

**<ol>**

**<li> primer elemento de la lista<\li>**

**<li> segundo elemento de la lista<\li>**

**<li> etc. </li>**

**</ol>**

Hay que tener en cuenta que:

* El atributo **start** permite comenzar la numeración en una cifra distinta al 1.
* El atributo **reversed** permite realizar una numeración descendente pero, por el momento, no está implementado en ningún navegador.
* El atributo **Type** permite remplazar las cifras por letras o números romanos aunque para eso lo lógico es utilizar las hojas de estilo CSS. La sintaxis es:

**<ol type=”1|a|A|i|I”>**

Siendo: **1** para números decimales (1, 2, 3, 4, …). Es lo que aparece por defecto.

**a** para listas ordenadas alfabéticamente en minúsculas (a, b, c, d, …).

**A** para listas ordenadas alfabéticamente en mayúsculas (A, B, C, …).

**i** para listas con números romanos en minúsculas (i, ii, iii, iv, …).

**I** para listas con números romanos en mayúsculas (I, II, III, IV…).

***Veamos cómo aplicar las listas ordenadas realizando la práctica 5.2***

C.- **Listas anidadas**. Podemos incluir listas dentro de otras listas con distintos niveles. Esto es lo que queríamos hacer en el caso de los hoteles de Benidorm. Por un lado queríamos una lista de nº de estrellas hoteleras y por otro una lista de hoteles para cada categoría de estrellas.

***¡Veamos cómo quedaría realizando la práctica 5.3!***

D.- **Listas de Definición**. Se trata de la típica lista de glosarios que hay al final de muchos libros. Esta lista presenta una lista de términos cada uno seguido de su definición o descripción. En este caso utilizaremos las etiquetas, <**dl**> antes de empezar la lista y <**/dl**> cuando la termine. Además para definir el elemento de la lista usaremos las etiquetas <**dt**> … <**/dt**>, y las etiquetas <**dd**> … <**/dd**>, para describir el elemento de la lista.

***Veamos un ejemplo realizando la práctica 5.4.***

LOS ENLACES. EL HIPERTEXTO.

Los enlaces son la esencia del lenguaje **HTML** y la web. Y es lo que le confiere la riqueza que se ha denominado: hipertexto. La etiqueta para los enlaces es <a> … </a>, cuya sintaxis básica es:

**<a href=”**destino del enlace**”>** texto del enlace **</a>**

El destino del enlace puede ser:

* Un lugar de la página en curso.
* Otra página del sitio Web.
* Algún lugar de otra página del sitio web.
* Una página de otro sitio existente en la web.
* Una dirección de correo electrónico.
* Un archivo para descargar.

Veamos cada uno de estos tipos de enlaces más detenidamente.

ENLACES DENTRO DE LA MISMA PÁGINA

Es normal ver en las páginas muy extensas un pequeño índice donde al hacer clic en una determinada opción lleva al usuario a una posición concreta o más avanzada de la página para ahorrar tiempo. Para ello se utiliza la técnica de las anclas o puntos de fijación.

Este enlace al interior de la página se realiza en dos tiempos:

La declaración del ancla: Sirve para definir un lugar en la página para poder hacer un enlace hacia ella. Por su puesto, el nombre del ancla no se visualiza en el navegador. La sintaxis de la declaración del ancla es:

**<a id=”**etiqueta o nombre del ancla**”> </a>**

El enlace hacia el ancla. Para crear el enlace, basta con escribir en el atributo **href** el nombre del ancla directamente precedido del signo almohadilla (**#**), de la siguiente forma:

**<a href=”#**nombre de etiqueta donde saltará**”>** mensaje que se desee **</a>**

***Realicemos un ejemplo con la práctica 5.5***

Observe que los enlaces se visualizan, por defecto, subrayados y en color azul para el enlace no visitado, en color púrpura, para el enlace visitado y en color rojo para el enlace activo.

El subrayado es una convención que se utiliza desde los comienzos del **HTML**. Para no desconcertar al navegante poco experimentado, seguiremos con esta norma, pero desde el punto de vista estético, una web que contenga muchos enlaces puede resultar fea. Veremos más adelante como quitar el subrayado usando hojas de estilo.

ENLACES CON OTRA PÁGINA DE NUESTRO SITIO WEB

Lo habitual es que un sitio web esté compuesto por varias páginas Web, una principal y otras secundarias que a su vez pueden estar relacionadas entre sí. En nuestro caso podemos considerar que estamos realizando un sitio web dedicado al turismo en España, con varias páginas diferentes y cuya página web principal es “**turismo.html**”.

Si el sito web es muy grande, puede suceder que, los diferentes archivos, se sitúen en varias carpetas para mantener el sitio bien estructurado. De ser así, tendríamos que tenerlo en cuenta para indicar correctamente la ruta donde se encuentra el archivo al que queremos enlazar.

Para enlazar nuestra página principal con otra situada en la misma carpeta de nuestro sitio web utilizaremos el código:

**<a href=”**nombre de la página**.**html**”>** mensaje que desee **</a>**

Siendo “**nombre de la página.html**” el nombre con el que se grabó el archivo de código de la página a donde deseamos ir.

Si la página con la que queremos enlazar no se encuentra en la misma carpeta habría que indicarle la ruta donde se encuentra de la siguiente manera:

**<a href=”**ruta decarpetas/nombre de la página**.**html**”>** mensaje que desee **</a>**

A la hora de realizar enlaces a otras páginas web es importante respetar las mayúsculas y minúsculas en los nombres de los archivos y carpetas así como tener en cuenta los espacios para buscar la compatibilidad con otros sistemas operativos diferentes a Windows.

ENLACE A UN LUGAR CONCRETO DE OTRA PÁGINA DEL SITIO WEB

También es posible realizar un enlace hacía un ancla situada en otra página. La sintaxis sería:

**<a href=”**ruta de carpetas**/**nombre de la página**.**html**#**ancla**”>**mensaje que desee**</a>**

Recuerde que previamente en esa página donde deseemos ir tendríamos que haber puesto la orden correspondiente al ancla de destino:

**<a id=”**ancla**”> </a>**

***Veamos el funcionamiento realizando la práctica 5.6:***

ENLACES A PÁGINAS DE CUALQUIER SITIO WEB

Si lo que queremos es enlazar con páginas Web de otros servidores de cualquier parte del mundo, lo único que deseamos saber es su dirección completa. La sintaxis:

**<a href=”**dirección completa de la página**”>** mensaje que desee **</a>**

Nótese que hay que incluir la dirección completa, incluido el texto del protocolo http://.

Por ejemplo si queremos enlazar con la web de nuestro centro la orden completa sería:

**<a href=”**http://www.academiafrayluis.es**”>Visite nuestra web** **</a>**

ENLACES A UNA DIRECCIÓN E-MAIL.

Podemos permitir a los visitantes de nuestro sitio web que contacten con nosotros a través del correo electrónico. Para ello realizaremos un enlace a una dirección de correo siguiendo el siguiente formato:

**<a href=”mailto:**dirección correo electrónico**”>** mensaje que desee **</a>**

***Veamos un ejemplo realizando la práctica 5.7***

Como ha podido comprobar, la activación del enlace hacia una dirección mail, abre una ventana de la aplicación de correo electrónico configurada por defecto en el puesto del visitante.

ENLACES PARA DESCARGAR UN ARCHIVO

Para descargar un archivo utilizamos la misma etiqueta <**a**> indicando la ruta y el nombre del archivo que se quiere descargar.

***Veámoslo con un ejemplo realizando la práctica 5.8.***

Si en el ordenador del visitante no hay ninguna aplicación instalada para leer archivo en formato .pdf, el navegador abre una ventana que invita a descargar el archivo.

ATRIBUTOS DE LA ETIQUETA DE ENLACES

La etiqueta <**a**> tiene varios atributos:

A.- **Href**. Indica la dirección del destino del enlace. Ya lo hemos estudiado ampliamente.

B.- **Hrflang**. Indica el idioma del documento destino si es diferente al del documento origen. Solo se usa si el atributo **href** está presente.

C.- **Ping**. Indica una lista de direcciones url que reciben una notificación cuando el usuario sigue el enlace. Es muy útil para realizar estadísticas de un sitio web.

D.- **Media**. Especifica que el contenido de la dirección url de destino está diseñado para dispositivos especiales, como el iPhone, el braille o sintetizadores de voz. Solo se usa si el atributo **href** está presente.

E.- **Rel**. Especifica la relación entre el documento de origen y el documento destino del enlace. Solo se usa si el atributo **href** está presente.

F. **Target**. Especifica donde abrir, en qué ventana, los documentos vinculados. Solo se usa si el atributo **href** está presente. Tiene las siguientes posibilidades:

* **Target=”\_self”**. La página de destino se abre en la misma ventana que la página de origen del enlace. Es lo que se hace por defecto.
* **Target=”\_top”**. La página de destino del enlace se abre en la misma ventana pero ocupará la totalidad de la ventana mostrada. (Con lo que sabemos hasta ahora de **HTML**, el resultado de este valor es exactamente el mismo que para la opción **\_self**).
* **Target=”\_blank”**. La página de destino del enlace se abre en una nueva instancia o en una nueva pestaña del navegador.

G.- **Type**. Indica al navegador el tipo MIME del archivo del enlace, por ejemplo si se trata de un archivo de sonido o una imagen. Solo se usa si el atributo **href** está presente.

DESCRIPCIÓN DEL ENLACE

Existen una serie de atributos, llamados globales, que se pueden utilizar prácticamente en todas las etiquetas de **HTML**. Uno de ellos es el atributo title que permite mostrar un texto explicativo cuando se pasa el puntero del ratón por encima de algún elemento de la web. Es una opción muy útil para las facilidades a los internautas invidentes.

La sintaxis es:

**<a href=”t-deport.htm” title=”Ir a la web de Turismo Deportivo”>Turismo Deportivo</a>**

***Veamos un ejemplo realizando la práctica 5.9.***

CAPÍTULO 6 – TIPOGRAFÍA, TEXTO

Y LISTAS EN CSS

USO DE FUENTES Y TIPOS DE LETRA EN EL DISEÑO WEB

Al diseñar páginas web, podemos dar formato al texto de una forma parecida a como se hace en un tratamiento de textos o un programa de diseño en cualquier ordenador. El problema es que cuando indicamos que se use una fuente, (tipo de letra), concreta, esa fuente ha de estar instalada en el ordenador o el dispositivo del usuario que visita nuestra página. Si el ordenador o dispositivo no dispone de esa fuente, el navegador la sustituye por otra.

Puesto que es imposible saber qué fuentes están instaladas en los ordenadores de los usuarios, y debido a la capacidad que tienen los navegadores para sustituir unos tipos de letra por otros, es posible que los resultados obtenidos en ciertos equipos no se correspondan con el diseño que hemos previsto.

Una opción es utilizar fuentes que sabemos que podemos encontrar en la mayoría de los ordenadores, pero son tan pocas las fuentes de las cuales podemos tener la certeza de que se encuentran en prácticamente todos los ordenadores, que los buenos diseñadores se sienten muy limitados.

Teniendo en cuenta las plataformas Mac y Windows y los ordenadores más antiguos, las fuentes que podemos utilizar con más confianza en la Web son:

|  |  |
| --- | --- |
| * Arial | * Verdana |
| * Georgia | * Times New Roman |
| * Courier | * Trebuchet |
| * Lucida | * Tahoma |
| * Impact |  |

Las fuentes Courier e Impact se usan poco porque, aunque estén disponibles de forma general, su apariencia es muy peculiar y limita mucho su uso de forma habitual.

Sin solucionar totalmente el problema, las Hojas de Estilo **CSS**, aportan mayor diversidad en la elección de la tipografía mediante el uso de una "pila" de fuentes, ("font stack" en inglés). Se trata de indicar al navegador una lista de varios tipos de letra, alternativos entre si, que puede utilizar para mostrar en pantalla.

La sintaxis del estilo es:

**font-family: nombre de la fuente o familia de la fuente;**

Por ejemplo:

**font-family: Arial, Helvética, Georgia, Sans-serif;**

En este caso, hemos indicado varias fuentes separadas por comas, para que el navegador utilice la primera fuente, Arial, si está instalada en el sistema del usuario. En caso contrario se pasa a la siguiente de la lista y así sucesivamente.

La última fuente es un tipo genérico, una familia de fuentes, la familia de fuentes Sans-serif, que indica al navegador que emplee cualquier fuente Sans-serif que encuentre.

Si no se encuentra en el sistema del usuario, ninguna de las fuentes indicadas, el navegador utilizará la fuente por defecto que es lo que pasa en los ejercicios que hemos realizado hasta ahora en este curso ya que no hemos indicado aún ninguna fuente.

Por defecto, los navegadores más habituales emplean la fuente Times New Roman, de 14 píxeles y color negro.

Si el nombre de la fuente incluye espacios, como “Courier New”, se debe escribir entre comillas.

En el nombre de las fuentes no es necesario distinguir entre mayúsculas y minúsculas.

Las familias de fuentes más habituales son:

* **Serif**. Tienen terminaciones al final de sus trazos y un espaciamiento proporcional entre las letras. Ej: Times New Roman, Bodoni, Georgia, Garamond…
* **Sans-Serif**. Tienen terminaciones planas, sin ensanchamiento y un espaciamiento proporcional entre las letras: Ej: Arial, Verdana, Helvética, Trebuchet…
* **Cursive**. Familia de fuentes que se muestran parecido a una escritura manuscrita. Ej: Script, Adobe Poética…
* **Fantasy**. Fuentes de fantasía o decorativas. Ej: Wingdings…
* **Monospace**. Todas las letras tienen las mismas dimensiones como si se tratara de una máquina de escribir manual. Habitualmente se utilizan para programar código informático por su facilidad de lectura en este ámbito. Ej: Courier, Courier New…

Como ya hemos comentado, al indicar una familia de fuentes, el navegador escogerá por sí mismo la fuente disponible en el ordenador del usuario que se adapte a los criterios del nombre genérico. Por ello es recomendable especificar un nombre de fuente genérico tras realizar una enumeración de nombres de fuente para que la visualización respete el tipo de fuente deseado por encima de la fuente por defecto del navegador.

***Pero… ¡pasemos a la acción realizando la práctica 6.1!***

ITALICA O CURSIVA

Habíamos utilizado la etiqueta <**i**> para resaltar algún texto en cursiva. Y habíamos comentado que aunque existía dicha etiqueta era mejor utilizar las hojas de estilo. Esta es la sintaxis para escribir texto en cursiva:

**font-style: italic;**

Otros valores para esta propiedad son:

* **oblique:** inclina hacia la derecha los caracteres. El resultado es parecido a la letra cursiva.
* **Normal:** retoma la escritura normal

NEGRITA

Esta propiedad permite escribir texto en negrita con muchas más posibilidades, aunque poco perceptibles, que las de la etiqueta <**b**>. La sintaxis es:

**font-weight: bold;**

Otros valores para esta propiedad son:

* **Bolder:** acentúa la negrita respecto al elemento padre.
* **Lighter:** disminuye la negrita respecto al elemento padre.
* **Un valor numérico entre 100 y 900.** Los valores disponibles son: 100, 200, 300, 400, 500, 600, 700, 800 y 900. El 500 se corresponde con el valor de letra normal. El 700 con la negrita y el 900 con la negrita más llamativa.
* **Normal:** retoma la escritura normal.

VERSALES

Esta propiedad permite visualizar el texto en mayúsculas versales, (pequeñas mayúsculas). La sintaxis es:

**Font-variant: small-caps**

Otro valor para esta propiedad es:

* **Normal:** retoma la escritura normal.

***Pero… ¡pasemos a la acción realizando la práctica 6.2!***

COMO NOMBRAR LOS COLORES EN HOJAS DE ESTILO CSS

En temas anteriores hemos utilizado los nombres de colores para utilizarlos en un estilo. Pero los colores en las hojas de estilo **CSS** se pueden indicar de varias maneras:

La notación hexadecimal clásica, como #FFCC00, es la notación más utilizada. Con esta notación podemos reproducir prácticamente cualquier color. En la notación hexadecimal, el código se divide en tres pares de dígitos de tal manera que si tenemos un código como #AABBCC:

* AA= Es un número en hexadecimal de dos dígitos donde se le indica cuanta cantidad de rojo se desea que tenga el color.

(Un número Hexadecimal tiene 16 valores posibles que son: 0 1 2 3 4 5 6 7 8 9 A B C D E F, es decir el 0 será el valor mínimo y la F será su valor máximo, pero al poder poner en éste caso dos dígitos hay más combinaciones donde el valor mínimo será el 00 y el máximo será FF).

* BB= Lo mismo que el anterior pero para la cantidad de verde.
* CC= Igual pero para el color azul.

Veamos cómo haríamos los colores más primarios utilizando la notación hexadecimal:

* #FF0000: Sería el color rojo ya que BB y CC están a 0 mientras que AA está a su valor máximo: FF.
* #00FF00: Sería el color verde por estar en este caso AA=00, BB=FF (su valor máximo) y CC=0.
* #0000FF: Sería el color Azul.

Como se puede suponer, con la combinación de todos los valores de AA, BB y CC podemos generar todos los colores que usted imagine. Para que tenga como referencia:

* #FFFFFF dará el color blanco.
* #000000 dará el color negro.
* #FFFF00 dará el color amarillo.

Si desea hacer tonalidades oscuras deberá de reducir el número de color a oscurecer dejando los otros dos sin tocar. Veamos, si tiene un verde #00FF00 y lo quiere poner más oscuro reduzca su valor por ejemplo a #009900 y observe, si aún lo quiere más oscuro pruebe a poner #004400.

Si lo que quiere es que ese color sea más suave o difuminado, (que no más claro), deberá probar a poner los otros dos colores con números más altos, (claros), que el que tengan en ese momento, pero con un número igual.

Si usamos la combinación #FF7070 obtendremos un rosa ya que como podemos observar el BB y el CC han aumentado su valor a 70 por lo que el rojo puro FF se va suavizando.

Ya sólo es cuestión de tiempo para buscar el color que desee y probar hasta conseguirlo, con un poco de práctica podrá fácilmente usar unos colores lo suficientemente atractivos para sus páginas WEB.

Aunque habitualmente utilizaremos la notación hexadecimal clásica, otras formas de asignar los colores en **CSS** son:

* Las palabras clave que ya conoce como blue, Green, yellow, etc, etc.
* La notación hexadecimal abreviada como #fd3. Esta notación permite ahorrar algunos caracteres. Cada cifra está implícitamente duplicada de tal manera que #fd3 se corresponde con la notación clásica #ffdd33. Como es lógico con esta notación abreviada nunca se podría indicar un color como #cfe4f5.
* La notación decimal, por ejemplo, rgb(0,0,255). El código rgb se codificaría con un número comprendido entre 0 y 255. Es el equivalente decimal a la notación hexadecimal.
* La notación en porcentaje, por ejemplo; rgb(25%, 50%, 0%). El valor 0% significa la ausencia del color y 100% su valor máximo.
* La notación RGBa de la siguiente manera: rgba(0,0,0,0). En este caso las tres primeras cantidades se corresponden con la notación RGB vista anteriormente y la cuarta cantidad, el último valor indica el grado de opacidad o de transparencia, definido entre 0 y 1.
* La notación HSL (Tonalidad, Saturación, Luminosidad). La notación HSL consiste en tres valores. El primero se expresa en grados, de 0º a 359º. Se corresponde con un color de la rueda cromática: rojo (0º), amarillo (60º), verde (120º), cian (180º), azul (240º) y magenta (300º). El segundo y tercer valor se expresan en porcentaje y describen respectivamente la saturación y la luminosidad. Por ejemplo: hsl(0, 100%, 50%) sería la forma de indicar el color rojo.
* La notación HSLa, que agrega un valor comprendido entre 0 y 1 para la transparencia o la opacidad. Por ejemplo, hsl(0, 100%, 50%, 0.5) indicaría un color rojo semitransparente.

***Pero vayamos a verlo en la práctica realizando la práctica 6.3.***

TAMAÑO DE LA LETRA

La propiedad **CSS** que controla el tamaño del texto es “**font-size**”.

Cómo hemos visto en páginas anteriores podemos indicar el valor de la propiedad “**font-size**” con varias unidades de medida y formas diferentes:

* **Tamaño absoluto**: se trata de una serie de palabras clave que indican tamaños predefinidos de letra. Los tamaños nominados escalan de acuerdo con las preferencias del usuario con respecto a la letra. Los valores posibles son: xx-small, x-small, small, medium, large, x-large y xx-large.
* **Longitud**: Es un número seguido de un indicador del sistema de medida (cm, mm, in, pt, o pc) o bien de unidades relativas (em, ex, o px).
* **Porcentaje**: Un entero seguido del signo de porcentaje (%). El valor es un tanto por ciento del tamaño de letra del objeto padre.
* **Tamaño relativo**: una serie de palabras clave que se interpretan como relativas al tamaño de letra del objeto padre. Los valores posibles son larger y smaller.

La elección de la unidad de medida para el tamaño de la letra en una página web es importante y no es tan fácil como parece. La principal dificultad en este sentido tiene que ver con la resolución de la pantalla y las distintas maneras en que, a lo largo del tiempo, los navegadores han resuelto la presentación de los textos. También tenemos que tener en cuenta los usuarios que navegan con teléfonos y dispositivos móviles.

Podemos indicar el tamaño de la letra en píxeles y puntos de la siguiente manera:

**p {**

**font-size: 12px;**

**}**

o

**p {**

**font-size: 12pt;**

**}**

Sin embargo, aunque nos parezca práctico al estar acostumbrados a su uso, por ejemplo en Word, no se recomienda utilizar estas medidas en el diseño de páginas web ya que:

* los puntos son una unidad de medida absoluta pensadas para la imprenta y que no tiene una equivalencia exacta en la pantalla.
* Los píxeles son la unidad de medida que se suele utilizar en los programas de diseño gráfico por ordenador. Los tamaños y resoluciones de pantalla se expresan en píxeles pero todos los navegadores ofrecen la posibilidad de cambiar el tamaño del texto lo que deja sin control al diseñador web.

Una forma de resolver el problema del cambio de tamaño de los textos que causa el uso de píxeles consiste en aplicar otras unidades de medida. Estableceremos un cambio de tamaño fiable combinando el porcentaje y la unidad "**em**".

***Veámoslo en la práctica realizando la práctica 6.4.***

SUBRAYADOS

Esta propiedad permite subrayar de diversas maneras el texto. Su sintaxis es:

**text-decoration: underline;**

y los valores posibles son:

* **underline**: subraya el texto por debajo.
* **overline**: subraya el texto por encima.
* **line-through**: traza una línea sobre el texto. Equivale al tachado.
* **none**: quita cualquier subrayado indicado anteriormente.

***Practiquemos con un ejemplo realizando la práctica 6.5.***

Sobre esta propiedad también debemos tener en cuenta:

* Se pueden definir varios valores, separados por un espacio, en una misma declaración. Ej: **text-decoration: underline overline**;
* El valor “**line-through**” es particularmente útil en webs comerciales para destacar una revisión de precios. Recuerde que la etiqueta <**del**> tiene el mismo efecto.
* El principal uso que podemos dar al valor “**none**” es aplicarlo a la etiqueta de enlace <**a**> para suprimir el subrayado que aparece por defecto.

TRANSFORMACIÓN

Esta propiedad muestra un texto en mayúsculas o minúsculas, independientemente de cómo esté escrito en el código fuente. Su sintaxis es:

**text-transform: capitalize;**

y los valores posibles son:

* **capitalize**: escribe la primera letra de cada palabra en mayúsculas.
* **uppercase**: escribe todas las letras en mayúsculas.
* **lowercase**: escribe todas las letras en minúsculas.
* **none**: deja las letras sin modificación alguna.

Esta propiedad es muy útil para dar formato a los datos destinados a una base de datos.

INDENTACIÓN

Esta propiedad realiza una sangría de primera línea. Su sintaxis es:

**text-indent: valor preciso (20px) o valor relativo (20%);**

ESPACIO ENTRE LETRAS

Esta propiedad varía el espacio entre caracteres. Si no se abusa de su uso, esta propiedad permite mejorar la legibilidad del texto. Su sintaxis es:

**letter-spacing: valor de longitud (8px) / normal;**

A tener en cuenta:

* Los valores relativos, por ejemplo un porcentaje, no están admitidos en esta propiedad.
* Un valor positivo agrega espacio entre las letras. Un valor negativo disminuye el espacio entre las letras.
* La alineación a ambos márgenes del texto exige al navegador separar el espacio entre letras. Por lógica, si indicamos un valor “**letter-spacing**” de **0,** impediremos la alineación del texto.

ESPACIO ENTRE PALABRAS

Esta propiedad varía el espacio entre palabras. Si no se abusa de su uso, esta propiedad permite mejorar la legibilidad del texto. Su sintaxis es:

**word-spacing: valor de longitud (8px) / normal;**

A tener en cuenta:

* Los valores relativos, por ejemplo un porcentaje, no están admitidos en esta propiedad.
* Un valor positivo agrega espacio entre las palabras. Un valor negativo disminuye el espacio entre las palabras.
* La alineación a ambos márgenes del texto exige al navegador separar el espacio entre letras. Por lógica, si indicamos un valor “**word-spacing**” de **0,** impediremos la alineación del texto.

***Veamos un ejemplo práctico de estas propiedades CSS realizando la práctica 6.6.***

INTERLINEADO

Esta propiedad modifica el interlineado del texto. Su sintaxis es:

**line-height: número / valor / porcentaje / normal;**

A tener en cuenta:

* Un número multiplica el interlineado por ese número. Por ejemplo 2 indica un interlineado doble.
* Un valor de longitud indica un interlineado sin tener en cuenta el tipo de fuente. Ej: 16px.
* Un porcentaje determina el interlineado en función del interlineado normal. Ej: 200% indica interlineado doble.
* Si indicamos un valor negativo se disminuirá el interlineado normal y se puede hacer ilegible el texto.
* El valor normal corresponde al interlineado por defecto.

ESPACIOS VACÍOS

Esta propiedad permite controlar los espacios vacíos. Su sintaxis es:

**white-space: pre / nowrap / normal;**

A tener en cuenta:

* El valor **pre** preserva los espacios múltiples y los saltos de línea.
* El valor **nowrap** impide los saltos de línea automáticos y muestra el texto de forma horizontal, sea cual sea el espacio disponible en el navegador.
* El valor **normal** deja que el navegador gestione a su criterio los saltos de línea y los espacios múltiples.

***Para ver el efecto de esta propiedad realice la práctica 6.7.***

ALINEACIÓN HORIZONTAL

Ya hemos utilizado anteriormente esta propiedad. Permite alinear el texto. Su sintaxis es:

**Text-align: left / right / center / justify;**

ALINEACIÓN VERTICAL

Esta propiedad indica la alineación vertical del texto y cualquier otro objeto. Su sintaxis es:

**vertical-align: baseline / sub / super / top / middle / bottom / valor / porcentaje;**

A tener en cuenta:

* Los valores **baseline, sup** y **super** indican la alineación vertical respecto a la línea de texto:
  + **baseline** alinea el texto a la parte inferior de la línea.
  + **sub** escribe el texto por debajo de la línea base.
  + **super** escribe el texto por encima de la línea base
* Los valores **top, middle** y **bottom** alinean el texto respecto al elemento padre.
* Un valor de longitud positivo alinea el texto por encima de la línea base. Un valor negativo lo hace por debajo. Ej: vertical-align: 1.7em;
* Un porcentaje positivo alinea el texto por encima de la línea base. Un porcentaje negativo lo hace por debajo. Ej: vertical-align: -10%;

DIRECCIÓN DEL TEXTO

Esta propiedad permite cambiar la dirección del texto. Su sintaxis es:

**direction: ltr (left to right) / rtl (right to left);**

AÑADIR SOMBRA AL TEXTO

Es posible añadir sobra al texto con la propiedad “text-shadow”. Su sintaxis es:

**text-shadow: x y z color;**

A tener en cuenta:

* **X** es el margen de sombra hacia la derecha.
* **Y** es el margen de la sombra hacia abajo.
* **Z** es la intensidad de degradado o de desenfocado. Es opcional y su valor por defecto 0.
* **Color** es el color de la sombra.
* Es posible utilizar valores negativos para los parámetros x e y. En este caso, la sombra se desplaza hacia la izquierda y hacía arriba

DECLARACIÓN ABREVIADA DE ESTILOS DE FUENTE

La declaración abreviada de estilos de fuente permite agrupar casi todas las propiedades de estilo de fuente en una única declaración siguiendo el siguiente ejemplo:

**font: italic bold small-caps 24pt/1.5 Arial, sans-serif;**

Que se corresponde con:

font-style: italic;

font-weight: bold;

font-variant: small-caps;

font-size: 24pt;

line-height: 1.5;

font-family: Arial, sans-serif;

A tener en cuenta:

* Se puede concluir que la ganancia de tiempo, al tener que teclear menos texto, es importante.
* Los distintos atributos deben informarse seguidos y separados por un espacio.
* No importa el orden de declaración de los distintos atributos. No es obligatorio definir todas las propiedades. Como es lógico, para aquellas propiedades no definidas, el navegador utilizará los valores por defecto.
* Como se puede observar la notación correspondiente al interlineado es algo particular: se indica el valor del tamaño de letra seguido de barra seguido del valor del interlineado.

TIPOS DE LETRA PERSONALIZADOS

Como hemos comentado, los tipos de letras utilizados en las páginas web dependen de las tipografías instaladas en el ordenador del internauta. Sin embargo con la propiedad “**@font-face**” incorporada en **CSS3** es posible importar y utilizar cualquier tipografía personalizada incluso aunque no esté instalada en el ordenador del visitante. De esta manera ahora es posible añadir una gran variedad de tipos de letra y tener la certeza de que nuestras páginas se podrán ver tal y como lo pretendemos.

Añadir tipos de letra personalizada se realiza en dos fases: primero tenemos que descargar el tipo de letra mediante **@font-face** y luego aplicarla como ya sabemos mediante la propiedad “**font-family**”. La sintaxis de **@font-face** es:

**@font-face {**

**font-family: “nombre\_de\_la\_fuente”;**

**src: url(’ruta\_del\_archivo\_de\_la\_fuente’);**

**}**

A tener en cuenta:

* **font-family: “nombre\_de\_la\_fuente”** indica el nombre que se va a utilizar en página web.
* **src: url(‘ruta\_del\_archivo\_de\_la\_fuente’)** indica la dirección completa, si es necesaria, y el nombre del archivo que contiene el tipo de letra.

***Veamos un ejemplo realizando la práctica 6.8.***

Por último comentar que existen tipografías proporcionadas por servicios de empresas como los de **Google Font Api** que están perfectamente adaptadas a la visualización de cualquier ordenador. Para aplicarlas a nuestros diseños web basta con seguir las instrucciones indicadas por la correspondiente empresa. (Si quiere conocer alguno de estos servicios busque en Google la palabra “**webfont**”).

AJUSTAR LA ALTURA DE LOS CARACTERES

A veces tipos de letra distintos pueden parecer de distinto tamaño aunque realmente hayamos indicado que sean iguales. Para solucionar este problema podemos aplicar la propiedad “Font-size-ajust” que permite obtener caracteres siempre del mismo tamaño, tanto para las mayúsculas como para las minúsculas, fijando un valor de coeficiente de aspecto. Su sintaxis es:

**Font-size-ajust: valor;**

Por ejemplo: **Font-size-ajust: 0.465;**

El valor de ajuste hay que definirlo por tanteo pero empieza a haber páginas web dedicadas a esta propiedad que nos pueden facilitar la tarea.

TIPOS DE MARCADORES PARA LISTAS

Esta propiedad permite cambiar la apariencia del marcador o estilo de numeración de una lista. Su sintaxis es:

**list-style-type: disc / circle / square / decimal / decimal-leading-zero / upper-roman / lower-roman / upper-alpha / lower-alpha;**

A tener en cuenta:

* El valor **disc** muestra un círculo relleno.
* El valor **circle** muestra un círculo vacío.
* El valor **square** muestra un cuadrado.
* El valor **decimal** muestra números decimales: 1, 2, 3, etc.
* El valor **decimal-leading-zero** muestra números decimales precedidos de 0: 01, 02, 03, etc.
* El valor **upper-roman** muestra números romanos en mayúscula: I, II, III, IV, etc.
* El valor **lower-roman** muestra números romanos en minúscula: i, ii, iii, iv, etc.
* El valor **upper-alpha** muestra letras en mayúscula: A, B, C, etc.
* El valor **lower-alpha** muestra un letras en minúscula: a, b, c, etc

POSICIÓN DE LAS LISTAS

Esta propiedad determina la posición de la segunda línea y siguientes de un elemento de lista respecto al marcador. Su sintaxis es:

**list-style-position: outside / inside;**

A tener en cuenta:

* El valor **outside** es el valor por defecto y coincide con la presentación normal de las listas.
* El valor **inside** presenta un desplazamiento solamente para la primera línea del elemento de la lista. Las demás líneas del elemento se alinean con el marcador.

INDICAR UNA IMAGEN COMO MARCADOR DE LISTAS

Esta propiedad permite utilizar una imagen como marcador de una lista. Su sintaxis es:

**list-style-image: url(ruta\_del\_archivo\_de\_imagen) / none;**

***Veamos como funcionan estas propiedades realizando la práctica 6.9.***

DECLARACIÓN ABREVIADA PARA ESTILOS DE LISTAS

La declaración abreviada de estilos de listas permite agrupar los valores correspondientes a las listas en una única declaración siguiendo el siguiente ejemplo:

**list-style: url(mimarcador.png) circle inside;**

A tener en cuenta:

* Los valores se reagrupan y separan por espacios.
* Ningún valor es obligatorio, y es posible definir un solo valor.
* Hemos puesto el valor “**circle**” para indicar que si el navegador no encuentra la imagen en la ruta especificada, muestre un marcador de tipo círculo.

***Para terminar de dominar este tema realice ahora la ejercitación del capítulo 6.***

CAPÍTULO 7 – USO DE IMÁGENES Y FONDOS

CON HTML y CSS

FORMATOS DE IMAGEN PARA LA WEB

Existen tres formatos de imagen más utilizados en la creación de páginas web. Debemos conocer sus características básicas para emplear el más adecuado a nuestras necesidades. Son:

* Formato GIF (Graphics Interchange Format). Fue el primero que se utilizó en la web.
  + Tiene un peso reducido.
  + Sólo permite un máximo de 256 colores.
  + Permite construir animaciones, lo que se conoce como GIFs animados.
  + No es bueno para fotos pero si para logotipos, botones, puntos, barras y elementos gráficos que no precisen de una paleta de colores muy extensa.
* Formato JPEG (Joint Photographic Expert Group). Es el formato más extendido en el mundo de la fotografía digital.
  + Permite hasta 16,7 millones de colores.
  + Una tasa de comprensión de entre el 15% y el 20% generará una imagen de calidad satisfactoria con un tamaño de archivo reducido.
  + Es un formato excelente para las fotografías aunque puede generar deformaciones en los degradados de color, las esquinas y los bordes.
* Formato PNG (Portable Network Graphic). Se trata de un formato concebido especialmente para uso de imágenes en páginas Web que trata de reunir todas las ventajas de GIF y JPEG. Los profesionales lo están adoptando mayoritariamente sobre todo para fotografías de pequeño tamaño.
  + Soporta 16,7 millones de colores.
  + Su tasa de comprensión de entre 5% y el 25% se realiza sin pérdida de calidad.
  + Permite realizar transparencias de hasta 256 colores.

Hay un nuevo formato de imagen, el WebP, desarrollado por Google y que seguramente gane importancia en los próximos años.

A la hora de utilizar imágenes en la creación de páginas web se recomienda disminuir el tamaño del archivo a una dimensión razonable para mejorar el tiempo de descarga.

INSERTAR UNA IMAGEN CON HTML

Para insertar una imagen en una página web utilizamos la etiqueta **HTML** <**img**>. Su formato es el siguiente:

**<img src=”ruta\_del\_archivo\_de\_imagen” />**

Tenga en cuenta que esta etiqueta no tiene su correspondiente de cierre. Recuerde que nosotros utilizamos la notación **XHTML** aunque no sea necesaria.

***Veamos un ejemplo realizando la práctica 7.1.***

La etiqueta <**img**> tiene el atributo: **alt=”texto explicativo”** que permite incluir una breve descripción de la imagen. Este atributo es muy útil para las personas que utilizan sistemas de lectura de pantalla para navegar por la Web, ya que permite convertir a voz el texto indicado como valor. También sirve para dar un contenido al espacio vacío que se genera cuando, por cualquier razón, la imagen no se puede representar (porque el enlace está roto o el archivo tiene un formato incorrecto, por ejemplo).

Además el contenido del atributo **alt** es usado en los motores de búsqueda para alimentar su base de datos de imágenes.

Otros atributos de la etiqueta <**img**> son “**height**” y “**width**”. Estos dos atributos indican al navegador las dimensiones, en píxeles, que tendrá la imagen en pantalla, ancho y alto respectivamente, que no necesariamente tienen que coincidir con las dimensiones originales.

La definición de estos atributos hace que el navegador reserve un lugar para la imagen antes de descargarla completamente, mientras continúa con la composición de la página y la visualización del texto de la misma. Aunque estos atributos son opcionales, están considerados como esenciales por parte de los diseñadores profesionales.

INSERTAR UN ENLACE SOBRE UNA IMAGEN CON HTML

Los enlaces desde una imagen se realizan simplemente insertando la imagen entre las etiquetas de enlace <**a**> … <**/a**>.

***Veamos un ejemplo realizando la práctica 7.2.***

APLICAR UN COLOR DE FONDO CON CSS

La propiedad **background-color** permite asignar un color de fondo a un elemento. Su sintaxis es:

**background-color: notación de color / transparent;**

A tener en cuenta:

* La notación de color se realiza como ya hemos comentado en el caso de la propiedad “**color**”.
* El valor **transparent** aplicado al fondo de la página muestra el valor por defecto del navegador.

***Compruébelo realizando la práctica 7.3.***

UTILIZAR UNA IMAGEN COMO FONDO CON CSS

La propiedad **background-image** permite asignar una imagen de fondo a un elemento. Su sintaxis es:

**background-image: url(ruta\_archivo\_imagen) / none;**

A tener en cuenta:

* El valor **none** no muestra ninguna imagen de fondo.
* Por defecto la imagen se coloca en la esquina superior izquierda del elemento y se repite tanto vertical como horizontalmente, es decir se muestra como un mosaico.

***Veamos un ejemplo realizando la práctica 7.4.***

Cómo ha podido comprobar nada impide definir un color de fondo y a la vez una imagen de fondo. Jugar con esta posibilidad nos permite crear efectos más o menos creativos. Además, el fondo de color se mostrará en el caso de que no se encuentre la imagen en la ruta indicada.

REPETICIÓN DE LA IMAGEN CON CSS

Con la propiedad **background-repeat** podemos elegir la repetición de la imagen de fondo en el caso de que no queramos que se muestre como un mosaico. Su sintaxis es:

**background-repeat: repeat / repeat-x / repeat-y / no-repeat;**

A tener en cuenta:

* Lógicamente para utilizar esta propiedad previamente tenemos que haber definido una imagen de fondo.
* El valor **repeat** repite la imagen horizontal y verticalmente consiguiéndose el mismo resultado de mosaico de la imagen.
* El valor **repeat-x** repite la imagen horizontalmente.
* El valor **repeat-y** repite la imagen verticalmente.
* El valor **no-repeat** muestra la imagen una sola vez, sin repetirla.

***Veamos un ejemplo realizando la práctica 7.5.***

POSICIONAMIENTO DE LA IMAGEN CON CSS

Hemos dejado la imagen anterior en la esquina superior izquierda de la pantalla, sería estupendo poder colocarla en cualquier parte. Esto es lo que hace la propiedad **background-position**. Su sintaxis es:

**background-position: longitudes en px / porcentajes / left center right top center bottom;**

Por ejemplo: **background-position: 80px 120px;**

**background-position: 30% 55%;**

**background-position: center top;**

A tener en cuenta:

* Lógicamente para utilizar esta propiedad previamente tenemos que haber definido una imagen de fondo.
* También es lógico que utilicemos esta propiedad cuando la imagen no está repetida.
* El primer valor indica la posición respecto al borde izquierdo y el segundo valor respecto al borde superior. Ambos valores tienen que estar separados por un espacio.
* Es posible asignar un único valor que será recogido como el valor horizontal. En este caso el segundo valor, o valor horizontal se definirá de forma automática al valor por defecto que es **center** o **50%**.
* Es posible combinar los valores en porcentaje y en longitud como: **background-position: 150px 40%;**
* Los valores **left**, **center** o **right** determinan la posición horizontal. Mientras que los valores **top**, **center** y **bottom** determinan la posición vertical.
* También se aceptan valores negativos.

FIJAR LA IMAGEN DE FONDO CON CSS

La propiedad **background-attachment** permite fijar una imagen de fondo para que no se mueva con el resto de la página. Su sintaxis es:

**background-attachment: scroll / fixed;**

A tener en cuenta:

* El valor **scroll** permite a la imagen de fondo deslizarse con el contenido de la página.
* El valor **fixed** deja la imagen de fondo fija mientras se desliza el contenido de la página.

***Veámos estas dos propiedades en la práctica realizando la práctica 7.6.***

DECLARACIÓN ABREVIADA DE PROPIEDADES DE FONDO EN CSS

Al igual en que en los estilos de tipo de letra, es posible declarar todas las propiedades correspondientes al fondo de la página web de manera abreviada en una sola línea. Así por ejemplo, la declaración:

**Background: url(mimarcador.npg) blue no-repeat right center fixed;**

Se corresponde con: **background-image: url(mimarcador.npg);**

**Background-color: blue;**

**Background-repeat: no-repeat;**

**Background-position: right center;**

FONDOS MÚLTIPLES CON CSS

Es posible visualizar varias imágenes de fondo en una declaración “**background**”. Un ejemplo sería:

**Background: url(imagen1.png) left top no-repeat,**

**url(imagen2.png) right top no-repeat,**

**url(imagen3.png) right bottom no-repeat;**

DEGRADADOS DE COLOR CON CSS

Aunque a la hora de escribir este manual los degradados de color no están totalmente definidos en las especificaciones de **HTML5** y **CSS3** ya casi todos los navegadores los contemplan. Para aplicar degradados en Firefox utilizaremos las siguientes declaraciones:

A.- Para un degradado lineal:

**Background: -moz-linear-gradient (punto\_de\_origen, color\_inicial, color\_final);**

A tener en cuenta:

* **Linear** indica un degradado lineal.
* **Punto\_de\_origen** se puede escoger entre **top**, **right**, **bottom** o **left** o un par de valores, separados por un espacio, elaborados a partir de **top**, **right**, **bottom**, o **left**. Por ejemplo: **bottom** **right**.
* **Color\_inicial** indica el color de inicio del degradado.
* **Color\_final** indica el color de fin del degradado.

B.- Para un degradado circular:

**Background: -moz-radial-gradient(posición, forma, color\_inicial, color\_final);**

A tener en cuenta:

* **Radial** indica un degradado circular.
* **Posición** es el punto central del gradiente radial. Es una palabra clave a escoger entre **top**, **right**, **bottom**, **left**, center o un par de valores, separados por un espacio, elaborados a partir de **top**, **right**, **bottom**, **left** o **center**. Por ejemplo: **center** **center**.
* **Forma**, la forma del gradiente radial, **circle** o **ellipse**.
* **Color\_inicial** indica el color de inicio del degradado.
* **Color\_final** indica el color de fin del degradado.

***Veámoslo realizando la práctica 7.7.***

***Para terminar de dominar este tema realice ahora la ejercitación del capítulo 7.***