

TEMA 2.

HTML. PÁGINAS WEB ESTÁTICAS. HTML (I)

Lenguajes de Marcas
CFGS DAW 1

Autor: Pascual Ligeró

Revisado por:

Fco. Javier Valero – franciscojavier.valero@ceedcv.es

2019/2020

Versión:190920.2225

Licencia



CC BY-NC-SA 3.0 ES Reconocimiento - NoComercial -

CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

NOTA: Esta es una obra derivada de la original realizada por Pascual Ligeró.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:



Importante



Atención



Interesante

ÍNDICE DE CONTENIDO

| | |
|--|-----------|
| 1. WORLD WIDE WEB..... | 4 |
| 1.1 Utilización del servicio World Wide Web..... | 4 |
| 1.2 ¿Qué es HTML?..... | 5 |
| 1.3 ¿Qué necesito para empezar?..... | 5 |
| 1.4 Mi primer documento HTML..... | 6 |
| 2. CARACTERÍSTICAS GENERALES DEL LENGUAJE HTML..... | 7 |
| 2.1 Marcas y atributos..... | 7 |
| 2.2 Formato..... | 7 |
| 2.3 Atributos..... | 8 |
| 3. ESTRUCTURA DE UN DOCUMENTO HTML..... | 9 |
| 3.1 Etiquetas principales de un documento HTML..... | 11 |
| 3.2 Los colores en HTML..... | 12 |
| 4. TEXTO EN HTML..... | 13 |
| 4.1 Párrafos..... | 15 |
| 4.2 Contenedores o Layers..... | 17 |
| 4.3 Títulos de sección..... | 17 |
| 4.4 Formato de las fuentes..... | 21 |
| 4.5 Etiqueta | 22 |
| 4.6 Espacios en blanco y nuevas líneas..... | 23 |
| 4.7 Nuevas líneas..... | 24 |
| 4.8 Espacios en blanco..... | 25 |
| 4.9 Texto preformateado..... | 26 |
| 4.10 Codificación de caracteres..... | 28 |
| 4.11 Comentarios..... | 30 |
| 5. Bibliografía..... | 31 |

UD02. HTML (I)

1. WORLD WIDE WEB

El servicio **World Wide Web** (La Telaraña Mundial), también conocido como **WWW** simplemente **Web**, es un Sistema de Información distribuido por Internet basado en la tecnología hipertexto/hipermedia, que proporciona una interface común a los distintos formatos de datos (texto, gráficos, vídeo, audio, etc.) y a los servicios de Internet existentes (FTP, news, telnet,...). Todo esto hace que el servicio Web sea el servicio de más utilizado en Internet.

Un **documento hipertexto**, es un texto en que cualquier palabra puede ser especificada como un enlace a otros documentos que contienen más información sobre dicha palabra, por lo que la lectura de un documento hipertexto no es secuencial o lineal, sino que se puede acceder a la información que nos interese desde otros conceptos relacionados (simplemente haciendo click con el ratón en la palabra relacionada), y de esta forma avanzar de documento en documento hasta encontrar la información deseada. Estas palabras que poseen enlaces a otros documentos están marcadas de alguna manera para poder diferenciarlas.

Un **documento hipermedia** es un hipertexto, pero que no incluye solo información textual sino también información multimedia, es decir, puede incluir gráficos, vídeo, y sonido.

A pesar de las diferencias de estos dos conceptos, a menudo se utiliza el término hipertexto para designar el significado de hipermedia.

Los **documentos Web** o también llamados **páginas Web** pueden estar localizados en diferentes sitios de Internet, estos sitios son llamados servidores Web. De manera que un documento WWW puede contener enlaces a otros documentos que se encuentran en el mismo servidor Web o en otros servidores Web, logrando así formar una telaraña mundial de información.

El lenguaje estandarizado para la creación de páginas Web es el **lenguaje HTML** (Hipertexto Markup Language, Lenguaje de Marcas Hipertexto). HTML es un lenguaje muy sencillo que permite describir documentos hipertexto. La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado).

1.1 Utilización del servicio World Wide Web.

Para utilizar el servicio Web se necesita una aplicación cliente capaz de entender o interpretar información HTML, a este tipo de aplicaciones se le conoce como **browsers o navegadores o visualizadores**. Mediante el navegador el usuario puede acceder a los documentos HTML y moverse de un documento a otro a través de sus vínculos o enlaces, este hecho de moverse con el navegador por las páginas WWW a través de sus enlaces se le conoce como Navegar por Internet.

El navegador sabe cómo acceder a cada recurso de Internet, sabe cómo acceder a un servidor de FTP anónimo, a un servidor de News, ..., y por supuesto cómo conectarse a los servidores Web. El mecanismo que utiliza el navegador para acceder a un recurso en cualquier lugar de Internet es el **URL** (Uniform Resource Locator, Localizador de Recursos Uniforme), comúnmente llamado dirección Internet.

Los URLs combinan el **protocolo** a utilizar para obtener el recurso: http (es el del Web), ftp, telnet, ...; junto con el nombre del **host** servidor, y el **path** completo del recurso (directorios y nombre de archivo). Los URLs constituyen en realidad los enlaces que permiten movernos de una página a otra, es decir Navegar por Internet, y que las podemos identificar dentro de una página WWW porque estas incluidos comúnmente como Texto en color subrayado, también puede ir incluidos dentro de una imagen, etc.

1.2 ¿Qué es HTML?

HTML es un lenguaje que se utiliza para la creación de páginas en la WWW. Por página entenderemos el documento que aparece en el visualizador o navegador.

HTML se compone de una serie de comandos, que son interpretados por el visualizador, o programa que utilizamos para navegar por el WWW. En última instancia es el visualizador el que ejecuta todas las órdenes contenidas en el código HTML, de forma que un visualizador puede estar capacitado para unas prestaciones, pero no para otras. Así, podremos especificar que una página tenga una imagen de fondo, o un texto parpadeando, pero si nuestro visualizador no está capacitado para esas funciones, no podremos verlas.

1.3 ¿Qué necesito para empezar?

Para comenzar solo son necesarios dos elementos:

- **Un editor de texto.** Vale cualquiera, siempre que no formatee el texto. Por ello, podemos utilizar los más sencillos como el bloc de notas o utilizar editores específicos para programadores como *bluefish* (editor único para HTML) o *notepad++*, *visual studio code* . Si se utilizan procesadores como el Word, se deben guardar los ficheros como 'solo Texto', para que no introduzca órdenes de formateo, que pueden provocar errores al cargarlo en el navegador.
 - **Un visualizador o navegador** del Web: Firefox, Chrome, Microsoft Explorer, Opera, etc. El editor de texto lo usaremos para ir escribiendo los documentos HTML, que será posteriormente interpretado por el navegador, con lo que iremos comprobando los cambios y añadidos que vayamos efectuando.
 - **Método de trabajo** : Con el editor de texto crearemos un fichero con el nombre que queramos (p. ej. ejemplo1), pero que debe tener necesariamente la extensión **.html**.

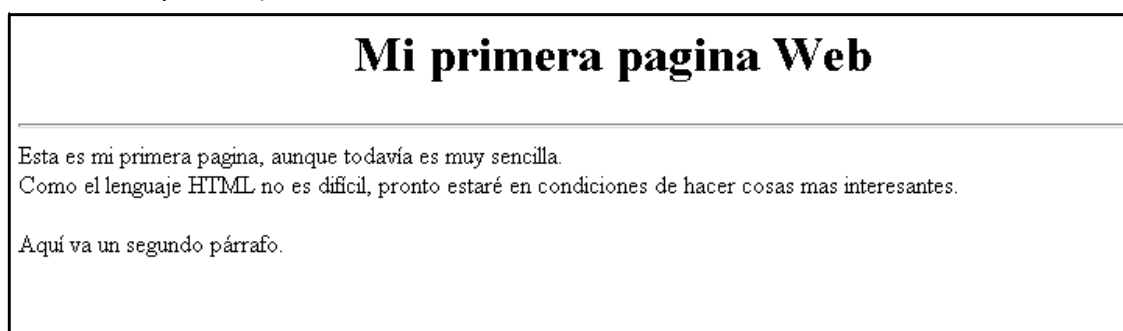
1.4 Mi primer documento HTML

Sigue las siguientes instrucciones para comenzar si es la primera vez que haces una página web.

1. Abra el editor, por ejemplo Bluefish, Notepad++, Visual Studio Code
2. Teclea lo siguiente:

```
<HTML>
<HEAD>
<TITLE>Ejemplo 1 - Mi primera pagina Web</TITLE>
</HEAD>
<BODY>
<CENTER><H1>Mi primera pagina Web</H1></CENTER>
<HR>
  Esta es mi primera pagina, aunque todavía es muy sencilla.
  <BR>Como el lenguaje HTML no es difícil, pronto estaré en
  condiciones de hacer cosas mas interesantes.
  <P> Aquí va un segundo párrafo.</P>
</BODY>
</HTML>
```

3. Graba este archivo con el nombre: **ejemplo1.html**
4. Visualiza la página en el navegador, bien utilizando la opción correspondiente en el editor o bien haciendo doble clic en el propio fichero creado. (En Bluefish, en la barra de herramientas el icono del planeta)



Las líneas en blanco y las indentaciones del texto se han puesto para mayor claridad, pero no son necesarias. De hecho, podría estar todo en una sola línea. Lo importante es el orden correcto de las etiquetas. Por cierto, una etiqueta puede estar anidada dentro de otra. Véase en el ejemplo cómo lo está la etiqueta `<CENTER>` dentro de la etiqueta

`<H1>`. Es muy importante, en estos casos, que las etiquetas de inicio y de cierre vayan en el orden correcto, pues de lo contrario se producirían errores.

2. CARACTERÍSTICAS GENERALES DEL LENGUAJE HTML

2.1 Marcas y atributos.

El lenguaje HTML se estructura utilizando marcas o etiquetas. La forma general de una marca es la de un comando HTML encerrado entre dos signos de menor y mayor como a continuación se muestra:

`<marca [atributos]>[</marca>]`

El mecanismo de funcionamiento de estas marcas es muy sencillo. Cuando el navegador encuentra el signo menor (<), examina todos los caracteres hasta que encuentra el final de la marca - el símbolo mayor (>). Entonces, interpreta el contenido de la marca, y aplica esa propiedad al texto que viene a continuación.

2.2 Formato

Aunque existen algunas excepciones, en general las etiquetas se indican por pares y se forman de la siguiente manera:

- **Etiqueta de apertura:** carácter <, seguido del nombre de la etiqueta (sin espacios en blanco) y terminado con el carácter >. Declaran las características o formatos del elemento.
- **Etiqueta de cierre:** carácter <, seguido del carácter /, seguido del nombre de la etiqueta (sin espacios en blanco) y terminado con el carácter >. Indican al navegador que el formato que afectaba al elemento desde la etiqueta de inicio, se acaba.

Así, la estructura típica de las etiquetas HTML es:

`<nombre_etiqueta> ... </nombre_etiqueta>`

Por lo tanto, gracias a la palabra o palabras reservadas que se encierran entre estos símbolos, será posible conocer las características o formatos que se le darán al elemento afectado. Lo ideal es que todo elemento de una página web esté suscrito a una etiqueta, evitando así que el navegador malinterprete el código.

No todas las etiquetas de HTML tienen su correspondiente etiqueta de fin.

Hay que diferenciar tres tipos:

- **Etiquetas de apertura y cierre:** Son las que se han explicado hasta ahora. La instrucción solo se aplica al elemento que encierran. (ejemplo : texto en cursiva `<i> . . . </i>`).

- **Etiquetas solo de apertura:** El efecto se producirá en un punto determinado del documento sin afectar a otros elementos. (ejemplo : salto de línea
)
- **Etiquetas de cierre opcional:** Pueden adoptar uno de los dos formatos vistos anteriormente aunque se aconseja cerrarlas para estructurar mejor el documento y dotarlo de mayor claridad.

Las etiquetas con apertura y cierre deben estar “balanceadas”. Si se abren en un orden determinado, se deben cerrar en el orden inverso para que las primeras engloben a las siguientes:

```
<etiqueta1>
  <etiqueta2>
    <etiqueta3>
      . . .
    </etiqueta3>
  </etiqueta2>
</etiqueta1>
```

Por otra parte, en HTML no diferencia entre mayúsculas y minúsculas a la hora de escribir e interpretar las etiquetas.

Con el fin de lograr una mayor uniformidad en el código, lo normal es que siempre se escriban de la misma forma. Las recomendaciones de W3C es ponerlas en minúsculas

2.3 Atributos

En muchas ocasiones, habrá una serie de parámetros que harán que los efectos que producen las etiquetas varíen, ya sea en colores, alineación, estilos, etc...

Estos parámetros son modificadores de la etiqueta genérica de inicio o apertura y se denominan **atributos**.

Con ellos se pueden definir diferentes posibilidades de la instrucción HTML y, generalmente, están formados por el nombre del atributo, que es una palabra reservada del lenguaje, el signo “igual” (=) y el valor que toma.

Una etiqueta genérica con atributos sería:

```
<etiqueta atributo1="valor1" atributo2="valor2"> . . . . </etiqueta>
```

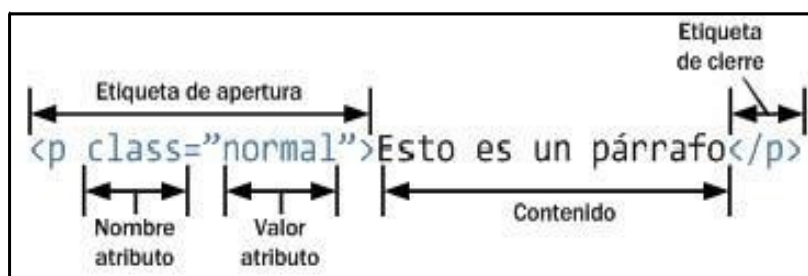
Aquí atributo1 y atributo2 son dos atributos distintos que toman los valores valor1 y valor2 respectivamente.

El orden en el que se incluyan los atributos es indiferente aunque hay que tener en cuenta dos puntos importantes:

Si el valor del atributo contiene dos o más palabras separadas por espacios, debe ir entre comillas para evitar que el navegador malinterprete el código.

Cada atributo solo puede tener un valor en cada etiqueta. En caso de que en una misma etiqueta, se pusiese el mismo atributo con distintos valores, el resultado sería imprevisible y dependería de la interpretación que diese a dicho código el navegador.

El siguiente esquema muestra un elemento HTML, formado por una etiqueta <p>, atributos y contenidos de texto:

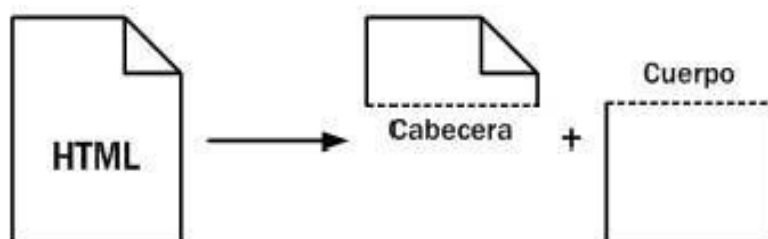


La estructura mostrada en el esquema anterior es un elemento HTML ya que comienza con una etiqueta de apertura (<p>), contiene cero o más atributos (class="normal"), dispone de un contenido de texto (Esto es un párrafo) y finaliza con una etiqueta de cierre (</p>).

3. ESTRUCTURA DE UN DOCUMENTO HTML

Las páginas HTML se dividen en dos partes: la cabecera y el cuerpo.

La cabecera incluye información sobre la propia página, como por ejemplo su título y su idioma. El cuerpo de la página incluye todos sus contenidos, como párrafos de texto e imágenes.



El **cuerpo** (**body** en inglés) contiene todo lo que el usuario ve en su pantalla y la cabecera (head en inglés) contiene todo lo que no se ve (con la única excepción del título de la página, que los navegadores muestran como título de sus ventanas).

A continuación se muestra el código HTML de una página web muy sencilla (**primer ejemplo**):

```
<html>

<head>
<title>El primer documento HTML</title>
</head>

<body>
```

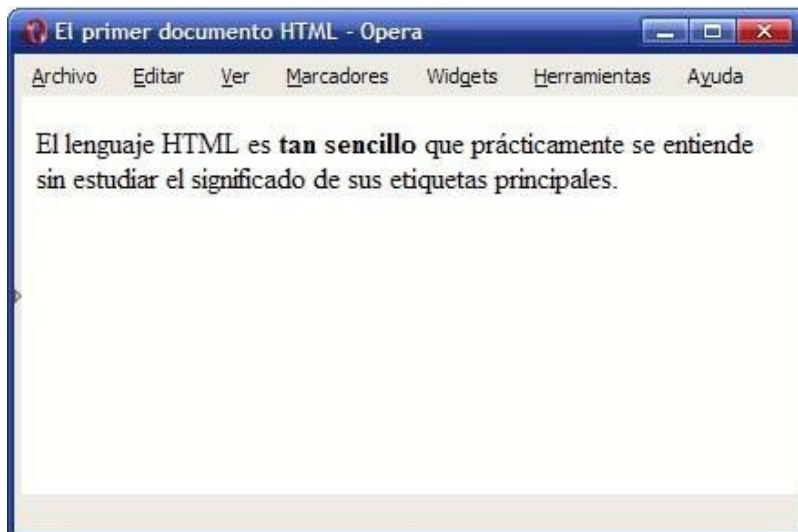
```
<p>El lenguaje HTML es  
<strong>tan sencillo</strong>  
que prácticamente se entiende sin estudiar el significado de sus etiquetas  
principales.</p>  
</body>  
  
</html>
```

Para probar este primer ejemplo, recuerda que debes hacer lo siguiente:

- Abre un editor de archivos de texto y crea un archivo nuevo
- Copia el código HTML mostrado anteriormente y pégalo tal cual en el archivo que has creado
- Guarda el archivo con el nombre que quieras, pero con la extensión .html

Para que el ejemplo anterior funcione correctamente, es imprescindible que utilices un editor de texto sin formato. Si tu sistema operativo es Windows no puedes utilizar un procesador de textos como **Word** o **LibreOffice**. Si utilizas sistemas operativos tipo Linux, puedes utilizar editores como **Gedit**, **Kedit**, **Kate** e incluso **Vi**, pero no utilices **KOffice** ni **Libre Office**.

Después de crear el archivo con el contenido HTML, ya se puede abrir con cualquier navegador para que se muestre con el siguiente aspecto:



Si ya estás viendo tu primera página HTML en el navegador chrome, prueba a pulsar **F12** y podrás ver el código HTML de la página que está cargada en el navegador. De hecho, puedes ver el código HTML de cualquier página de Internet mediante F12. Prueba a ver el código HTML de tu página preferida y verás cuantas etiquetas puede llegar a tener una página compleja.

| Etiqueta | Atributo | Valor | Significado |
|----------|------------|-------|--|
| body | bgcolor | Color | Define el color de fondo del documento. |
| | background | URL | Indica la dirección web de la imagen de fondo del documento. Si dicha imagen se encuentra en el propio servidor, se utilizará la dirección relativa (ruta del directorio). |
| | text | Color | Establece en el que aparecerá el texto del documento. |
| | link | Color | Establece el color en el que aparecerá el texto de los enlaces del documento. |
| | vlink | Color | Determina el color en el que aparecerá el texto de los enlaces visitados del documento. |

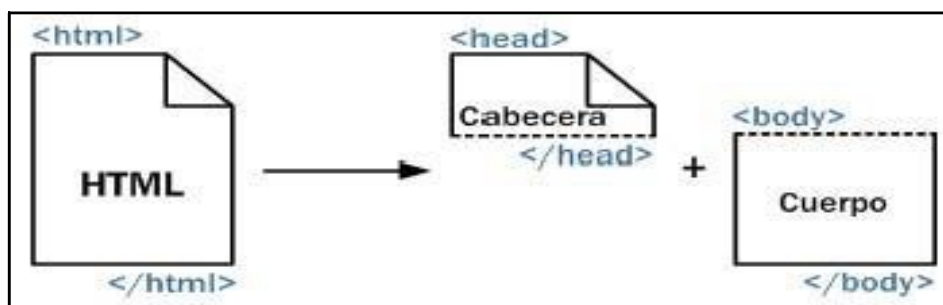
3.1 Etiquetas principales de un documento HTML.

<html> : indica el comienzo y el final de un documento HTML. **Ninguna etiqueta o contenido puede colocarse antes** . En el interior de la etiqueta <html> se definen la cabecera y el cuerpo del documento HTML y todo lo que se coloque fuera de la etiqueta de cierre </html> se ignora.

<head> : delimita la parte de la cabecera del documento. La cabecera contiene información sobre el propio documento HTML, como por ejemplo su título. **Los contenidos indicados en la cabecera no son visibles para el usuario, con la excepción de la etiqueta <title>, que se utiliza para indicar el título del documento y que los navegadores lo visualizan en la parte superior izquierda de la ventana del navegador** (si no te has fijado anteriormente, vuelve a abrir el primer ejemplo en cualquier navegador y observa dónde se muestra el título de la página).

<body> : delimita el cuerpo del documento HTML. El cuerpo encierra todos los contenidos que se muestran al usuario (párrafos de texto, imágenes, tablas). En general, el <body> de un documento contiene cientos de etiquetas HTML, mientras que el <head> no contiene más que unas pocas.

EJEMPLO DE ALGUNAS ETIQUETAS QUE PODEMOS ENCONTRAR EN BODY



La **etiqueta <body>**, a diferencia de <head>, admite una serie de atributos que son de carácter global para todo el documento.

Cuatro de estos **atributos (bgcolor, text, link y vlink)** modifican el color de algunos elementos básicos de documento como son el fondo de la página o el texto. Simplemente habrá que prestar atención al necesario contraste entre el fondo y los textos para garantizar la legibilidad del documento.

Por otra parte, el cambio de color de los enlaces no siempre tiene el comportamiento deseado en todos los navegadores por lo que es conveniente realizar una comprobación.

Más atención hay que prestar al **atributo background**, con el que se declara la imagen de fondo de la página. Su valor debe ser el nombre de una imagen y su extensión, si dicha imagen está en la misma carpeta que el documento HTML, o el de la URL donde se halla dicha imagen.

Esta imagen se mostrará por debajo del texto y los elementos que compongan el documento. Si el navegador no pudiera encontrar la imagen definida, mostraría simplemente el color de fondo establecido.

Hay que tener mucho cuidado que la imagen tenga el tamaño adecuado, ya que si es menor que la ventana del navegador del usuario, la imagen será reproducida como un mosaico por toda la ventana hasta rellenarla por completo. Si este no es el efecto deseado, lo mejor es utilizar una imagen que sea del tamaño concreto para el que se realice la página, indicando en ésta que se ha optimizado para una resolución determinada. Si el usuario utilizara una resolución de pantalla distinta, sabe que se arriesga a visualizar mal la página.

3.2 Los colores en HTML

En HTML, el color se debe expresar bien por su nombre en inglés, bien por su anotación hexadecimal.

Esta última resulta de calcular la proporción de rojo, verde y azul (los tres colores básicos en los sistemas RGB que utilizan los monitores) que componen el color final. El valor de cada proporción estará entre 0 y 255. Este valor hay que convertirlo a hexadecimal lo que resulta dos cifras en números con base 16 que van del 0 al 9 y de la A a la F. Una vez calculado el valor en hexadecimal de cada una de las componentes se expresará como #RGB, siendo R la componente del color rojo, G la componente del color verde y B la componente del color azul.

Un color que tuviera componente rojo=255, verde=0, azul=0, se expresaría como #FF0000

Con la utilidad Paint se puede obtener la componente roja, verde y azul de cualquier color. Para convertirlo a hexadecimal se puede utilizar la calculadora de Windows.

Normalmente suele ser suficiente con los colores especificados con su nombre en inglés. Estos son:

Nombres de colores y valores #RGB



Black = "#000000" (Negro)



Green = "#008000" (Verde)



Silver = "#C0C0C0" (Plateado)



Lime = "#00FF00" (Verde lima)

| | | | |
|---|------------------------------|---|---------------------------------|
|  | Gray = "#808080" (Gris) |  | Olive = "#808000" (Verde oliva) |
|  | White = "#FFFFFF" (Blanco) |  | Yellow = "#FFFF00" (Amarillo) |
|  | Maroon = "#800000" (Marrón) |  | Navy = "#000080" (Azul marino) |
|  | Red = "#FF0000" (Rojo) |  | Blue = "#0000FF" (Azul) |
|  | Purple = "#800080" (Púrpura) |  | Teal = "#008080" (Azul verdoso) |
|  | Fuchsia = "#FF00FF" (Fucsia) |  | Aqua = "#00FFFF" (Celeste) |

La mayoría de los editores HTML llevan incorporado una paleta de colores que permite a los usuarios una elección de los mismos meramente visual y su fácil conversión a sistema hexadecimal, por lo que de momento solo utilizaremos los nombres en inglés para especificar el color.

Prueba lo siguiente:

```
<html>
  <head>
    <title>color</title>
  </head>

  <body bgcolor="red">
    <p>Texto textoso
    </p>
  </body>
</html>
```

4. TEXTO EN HTML

La mayor parte del contenido de las páginas HTML habituales está formado por texto, llegando a ser más del 90% del código de la página. Por este motivo, es muy importante conocer los elementos y etiquetas que define HTML para el manejo del texto.

El lenguaje HTML incorpora al tratamiento del texto muchas de las ideas y normas establecidas en otros entornos de publicación de contenidos. De esta forma, HTML define etiquetas para estructurar el contenido en secciones y párrafos y define otras etiquetas para marcar elementos importantes dentro del texto.

La tarea inicial del editor de contenidos HTML consiste en estructurar el texto original definiendo sus párrafos, titulares y títulos de sección, como se muestra en la siguiente imagen:



El proceso de estructurar un texto simple consiste en indicar las diferentes zonas o secciones que componen el texto. De esta forma, los textos estructurados utilizan etiquetas para delimitar cada párrafo y títulos de sección para delimitar cada una de las secciones que forman el texto.

Una vez definida la estructura básica de los contenidos de la página, el siguiente paso consiste en marcar los diferentes elementos dentro del propio texto: definiciones, abreviaturas, textos importantes, textos modificados, citas a otras referencias, etc.



El anterior ejemplo muestra la transformación de un párrafo con un texto simple en un párrafo cuyo texto contiene elementos marcados de forma especial. Así, algunas palabras del texto se muestran en negrita porque se consideran importantes; otras palabras aparecen en cursiva, ya

que se han marcado como destacadas e incluso una frase aparece tabulada y entre comillas, indicando que es una cita textual de otro contenido.

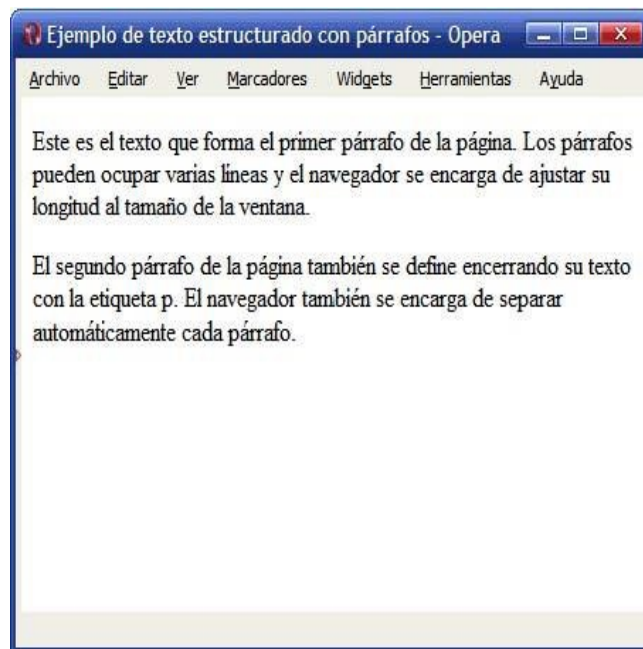
En las secciones siguientes se muestran las etiquetas que define HTML para estructurar y marcar el texto. Además, se hace una mención especial al tratamiento que hace HTML de los espacios en blanco y las nuevas líneas.

4.1 Párrafos

Una de las etiquetas más utilizadas de HTML es la etiqueta **<p>**, que permite definir los párrafos, **introduciendo un salto y deja una línea en blanco antes de continuar con el resto del documento**. Para delimitar el texto de un párrafo, se encierra ese texto con la etiqueta **<p>**, como muestra el siguiente ejemplo:

```
<html>
<head>
<title>Ejemplo de texto estructurado con párrafos</title>
</head>
<body>
<p>Este es el texto que forma el primer párrafo de la página. Los párrafos
pueden ocupar varias líneas y el navegador se encargade ajustar su longitud al
tamaño de la ventana.
</p>
<p>El segundo párrafo de la página también se define encerrando su texto
con la etiqueta p. El navegador también se encarga de separar
automáticamente cada párrafo.
</p>
</body>
</html>
```

El ejemplo anterior se visualiza de la siguiente manera en cualquier navegador:



Los párrafos creados con HTML son elementos de bloque, por lo que siempre ocupan toda la anchura de la ventana del navegador. Además, no tienen atributos específicos.

Con la etiqueta **
** podemos realizar un **salto de línea**, puede poner tantas como desee y realizará un salto de línea por cada una de ellas. **En esta etiqueta, al igual que sucede en otras tantas, no existe su cierre correspondiente, esto se debe a que un salto de línea no empieza y acaba más adelante sino que solo tiene presencia en un lugar puntual.**

Para justificar un párrafo utilizamos el **atributo align**. **Un atributo es un parámetro incluido en el interior de la etiqueta que ayuda a definir el funcionamiento de la etiqueta de una forma más personal.** Los atributos tienen sus valores indicados entre comillas (") pero si no los indicamos entre comillas también funcionará en la mayoría de los casos. Eso si, **en caso de que el valor del atributo contenga espacios las comillas son obligatorias.** De todas maneras, para evitar posibles errores es mejor acostumbrarse a ponerlas siempre.

El atributo **align** puede tener 3 valores:

1. **left**: texto alineado a la izquierda.
2. **right**: alineado a la derecha.
3. **center**: alineado al centro.

Para **centrar el texto** también podemos utilizar la etiqueta **<center>** seguida del texto que queremos centrar y posteriormente el cierre correspondiente, **</center>**.

Por ejemplo:

```
<html>
<head>
<title>Ejemplo01</title>
</head>

<body>
<p>Esto es un párrafo, se introduce un salto y deja una línea en
blanco
antes de continuar con el resto del documento.</p>
Vemos que ha dejado un espacio en blanco.<br><br> A través de la
etiqueta
<br>
hacemos un salto de carro.
<!-- Esto es un comentario. Los comentarios no son mostrados -->
<p><center>Texto centrado utilizando la etiqueta
center.</center></p>
<p align="left">Texto alineado a la izquierda.</p>
<p align="right">Texto alineado a la derecha.</p>
<p align="center">Texto centrado.</p>
</body>
</html>
```


4.2 Contenedores o Layers

Para cuando tengamos un texto largo y queramos que todos los párrafos estén justificados de alguna manera lo ideal es utilizar la etiqueta **<div>** denominada **contenedor**, o comúnmente “layer” (**capa**). Todo lo que esté incluido dentro de esta etiqueta será alineado tal y como hayamos especificado en el atributo **align**.

Con la versión 4.0 de HTML disponemos también de un nuevo valor para el atributo **align**, que es “**justify**”.

Este valor nos permite justificar el texto a ambos márgenes del navegador.

Por ejemplo:

```
<html>

<head>
<title>Ejemplo02</title>
</head>

<body>
<p>Para simplificar el código a la hora de justificar utilizamos la
etiqueta &lt;div&gt;.</p>

<div align="right">
  <p>centrado a la derecha</p>
  <p>insisto en que está centrado a la derecha</p>
</div>

<div align="center">
  <p>todo lo incluido dentro estará centrado</p>
  <p>insisto en que está centrado</p>
</div>

<div align="justify">
  <p>Con el valor "justify" hacemos un justificado a ambos
márgenes. A partir de HTML 4.0</p>
</div>

</body>
</html>
```

4.3 Títulos de sección

Las páginas HTML habituales suelen tener una estructura más compleja que la que se puede crear solamente mediante párrafos. De hecho, es habitual que las páginas se dividan en diferentes secciones jerárquicas.

Los títulos de sección se utilizan para delimitar el comienzo de cada sección de la página. HTML permite crear secciones de hasta seis niveles de importancia. De esta forma, aunque una página

puede definir cualquier número de secciones, solo puede incluir seis niveles jerárquicos.

Las etiquetas que definen los títulos de sección son **<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>** y **<h6>**. La etiqueta **<h1>** es la de mayor importancia y por tanto se utiliza para definir los titulares de la página. La importancia del resto de etiquetas es descendiente, de forma que la etiqueta **<h6>** es la que se utiliza para delimitar las secciones menos importantes de la página.

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

Las etiquetas **<h1>**, ..., **<h6>** definen títulos de sección, no secciones completas. Por este motivo, no es necesario encerrar los contenidos de una sección con su etiqueta correspondiente. Solamente se debe encerrar con las etiquetas **<h1>**, ..., **<h6>** los títulos de cada sección.

El siguiente ejemplo muestra el uso de las etiquetas de título de sección:

```
<head>
<title>Ejemplo03</title>
</head>

<body>
<p>Vamos a ver los 6 tipos de encabezados:<br>
<h1>Encabezado de nivel 1</h1>
<h2>Encabezado de nivel 2</h2>
<h3>Encabezado de nivel 3</h3>
<h4>Encabezado de nivel 4</h4>
<h5>Encabezado de nivel 5</h5>
<h6>Encabezado de nivel 6</h6>
</p>
</body>
</html>
```

Vamos a ver los 6 tipos de encabezados:

Encabezado de nivel 1

Encabezado de nivel 2

Encabezado de nivel 3

Encabezado de nivel 4

Encabezado de nivel 5

Encabezado de nivel 6

Otro ejemplo:

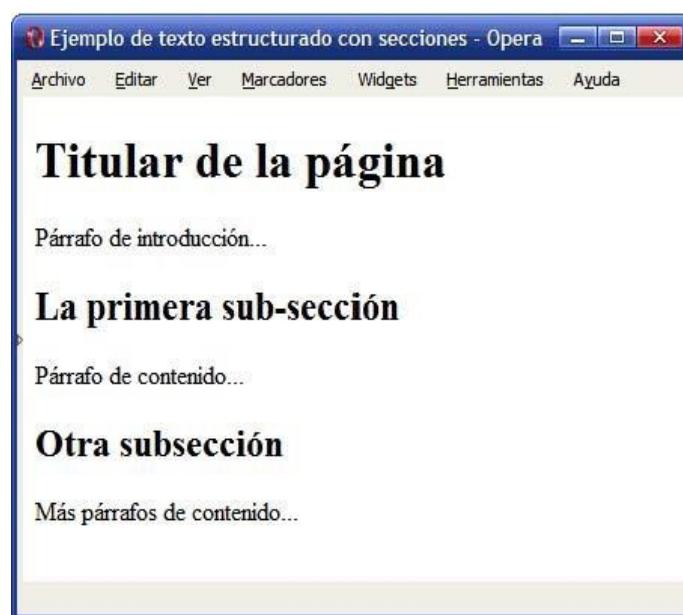
```
<html>
<head>
  <title>Ejemplo de texto estructurado con secciones</title>
</head>

<body>

  <h1>Titular de la página</h1>
  <p>Párrafo de introducción...</p>
  <h2>La primera sub-sección</h2>
  <p>Párrafo de contenido...</p>
  <h2>Otra subsección</h2>
  <p>Más párrafos de contenido...</p>
</body>
</html>
```

Los navegadores muestran el ejemplo anterior de la siguiente manera:

Los navegadores asignan de forma automáticamente el tamaño del título de cada sección en



función de su importancia. Así, los títulos de sección <h1> se muestran con el tamaño de letra más grande, ya que son el nivel jerárquico superior, mientras que los títulos de sección <h6> se visualizan con un tamaño de letra muy pequeño, adecuado para el nivel jerárquico de menor importancia.

Evidentemente, el aspecto que los navegadores aplican por defecto a los títulos de sección se puede modificar utilizando las hojas de estilos de CSS. La siguiente imagen muestra el tamaño por defecto con el que los navegadores muestran cada titular:



4.4 Formato de las fuentes

Una vez estructurado el texto en párrafos y secciones, el siguiente paso es el marcado de los elementos que componen el texto. Los textos habituales están formados por elementos como palabras en negrita o cursiva, etc.

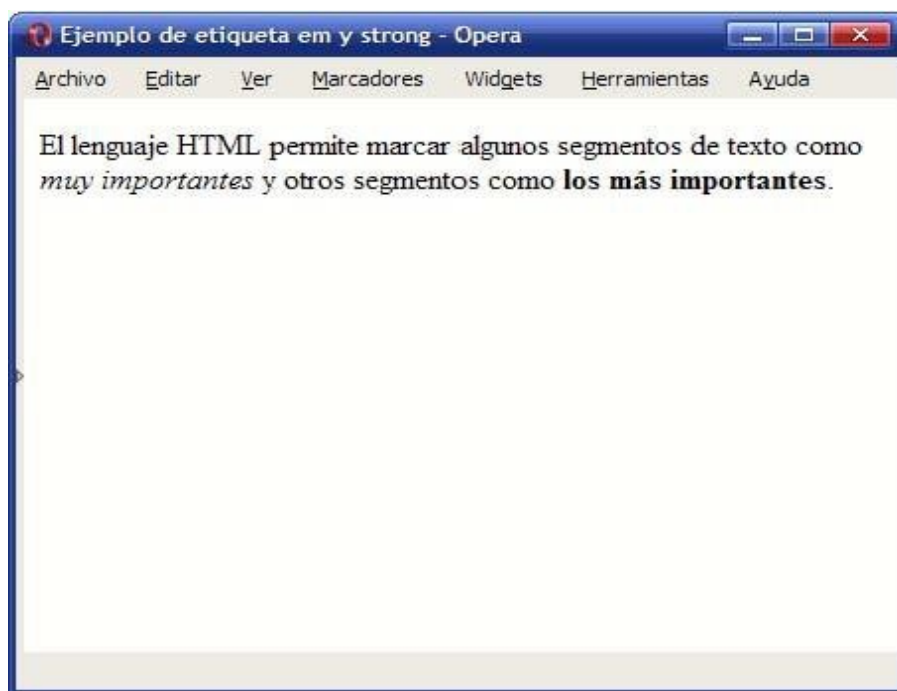
HTML proporciona varias etiquetas para **marcar** cada uno de los diferentes tipos de texto.

Entre las etiquetas más utilizadas para marcar texto se encuentran `` y ``.

La etiqueta `` marca un texto indicando que su importancia es mayor que la del resto del texto. La etiqueta `` indica que un determinado texto es de la mayor importancia dentro de la página.

```
<html>
<head>
<title>Ejemplo de etiqueta em y strong</title>
</head>
<body>
<p>El lenguaje HTML permite marcar algunos segmentos de texto como
<em>muy importantes</em>
y otros segmentos como
<strong>los más importantes</strong>.
</p>
</body>
</html>
```

Por defecto, los navegadores muestran los elementos `` en cursiva para hacer evidente su importancia y muestran los elementos `` en negrita, para indicar que son los más importantes:



El texto también se puede formatear cambiando la fuente, el tamaño, su color o resaltando ciertas palabras mediante el uso de negritas o cursivas.

Aunque desde HTML 4.0 se prefiere que el formato se aplique a través del lenguaje CSS, esto también es posible con las siguientes etiquetas.

Una de las etiquetas de formato más importantes es la etiqueta ****

Mediante la etiqueta **** es posible modificar la fuente por defecto (Times New Roman) y definir el tipo de fuente base. Para evitar problemas si el usuario no tiene dicha fuente instalada en su ordenador, se puede dar un listado de fuentes para que el navegador elija la primera que encuentre instalada:

Texto

Por otra parte, el **atributo size** permite definir el tamaño de la fuente.. En HTML, el tamaño de esta se define en referencia a una escala del 1 al 7, siendo 3 el valor por defecto. Es posible definir el tamaño de la fuente de manera absoluta (2) o de manera relativa (+2, -4, etc). En este último caso, tomará como referencia el valor por defecto (3) o el último indicado.

Además de la fuente de un bloque de texto, también se pueden resaltar palabras o un conjunto de ellas mediante el uso de negritas, cursivas o subrayado.

| Etiqueta | Efecto |
|----------------------------|-----------------------|
| | Negrita (bold) |
| <i></i> | Cursiva (Italia) |
| <u></u> | Subrayada (underline) |

Hay que tener en cuenta que este tipo de etiquetas puede usarse por separado o de forma conjunta. Si es necesario que un texto aparezca subrayado y en negrita, tan solo habrá que usar las dos etiquetas de manera balanceada:

<u>Texto</u>

4.5 Etiqueta ****

El estándar HTML incluye una etiqueta llamada **** que se emplea para marcar cualquier elemento que no se puede marcar con las otras etiquetas definidas.

La etiqueta **** se visualiza por defecto con el mismo aspecto que el texto normal. Por tanto es habitual utilizar esta etiqueta junto con los atributos **id** y **class** para modificar posteriormente su aspecto con CSS (las hojas de estilo CSS las veremos en los siguientes temas).

La etiqueta **** solo se puede utilizar para encerrar contenidos y etiquetas en línea.

4.6 Espacios en blanco y nuevas líneas

El aspecto más sorprendente del lenguaje HTML cuando se desarrollan los primeros documentos es el tratamiento especial de los **"espacios en blanco"** del texto. HTML considera **espacio en blanco** a los espacios en blanco, los tabuladores, los retornos de carro y el carácter de nueva línea (ENTER o Intro).

El siguiente ejemplo ilustra este comportamiento:

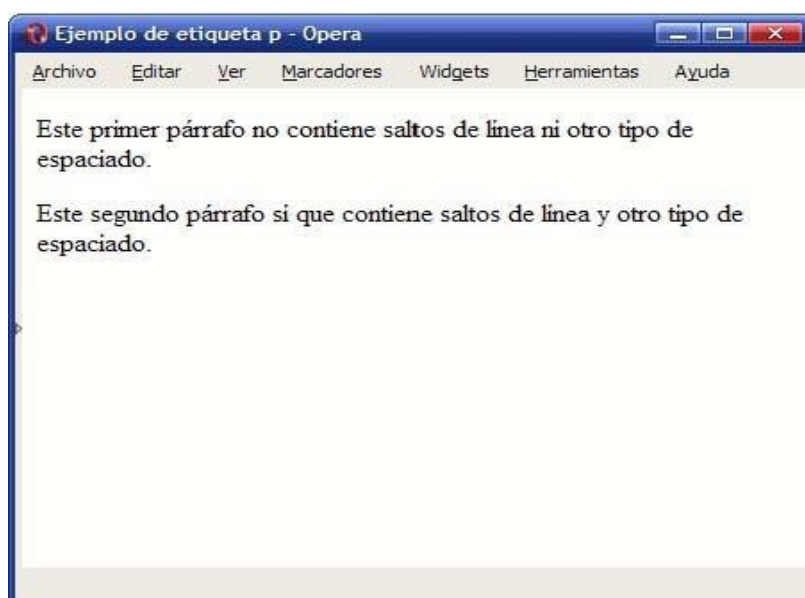
```
<html>

<head>
<title>Ejemplo de etiqueta p</title>
</head>

<body>
<p>Este primer párrafo no contiene saltos de línea ni otro tipo
de espaciado.</p>
<p>Este segundo párrafo sí que contiene saltos
de línea
y otro tipo de espaciado.</p>
</body>

</html>
```

El anterior código HTML se visualiza en cualquier navegador de la siguiente manera:



Los dos párrafos de la imagen anterior se ven idénticos, aunque el segundo párrafo incluye varios espacios en blanco y está escrito en varias líneas diferentes. La razón de este comportamiento es

que HTML ignora todos los espacios en blanco **sobrantes**, es decir, todos los espacios en blanco que no son el espacio en blanco que separa las palabras.

No obstante, HTML proporciona varias alternativas para poder incluir tantos espacios en blanco y tantas nuevas líneas como sean necesarias dentro del contenido textual de las páginas.

4.7 Nuevas líneas

Para incluir una nueva línea en un punto y forzar a que el texto que sigue se muestre en la línea inferior, se utiliza la etiqueta **
**. En cierta manera, insertar la etiqueta
 en un determinado punto del texto equivale a presionar la tecla ENTER(o Intro) en ese mismo punto.

La etiqueta
 es una de las pocas **etiquetas especiales** de HTML. La particularidad de
 es que es una etiqueta vacía, es decir, no encierra ningún texto.

Utilizando la etiqueta
 se puede rehacer el ejemplo anterior para que respete las líneas que forman el segundo párrafo:

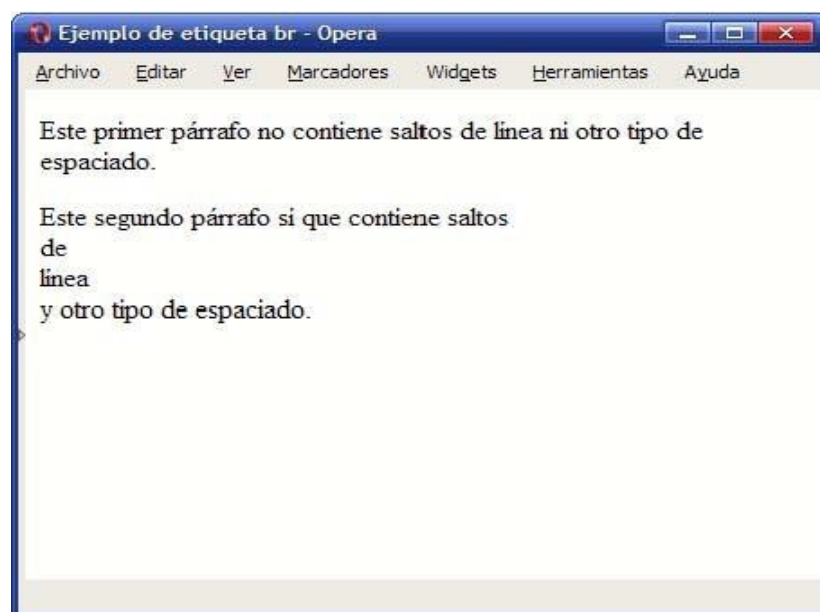
```
<html>

<head>
  <title>Ejemplo de etiqueta br</title>
</head>

<body>
  <p>Este primer párrafo no contiene saltos de línea ni otro tipo
de espaciado.</p>
  <p>Este segundo párrafo sí que contiene saltos
  <br> de <br> línea <br>y otro tipo de espaciado.</p>
</body>

</html>
```

El navegador ahora sí que muestra correctamente las nuevas líneas que se querían insertar:



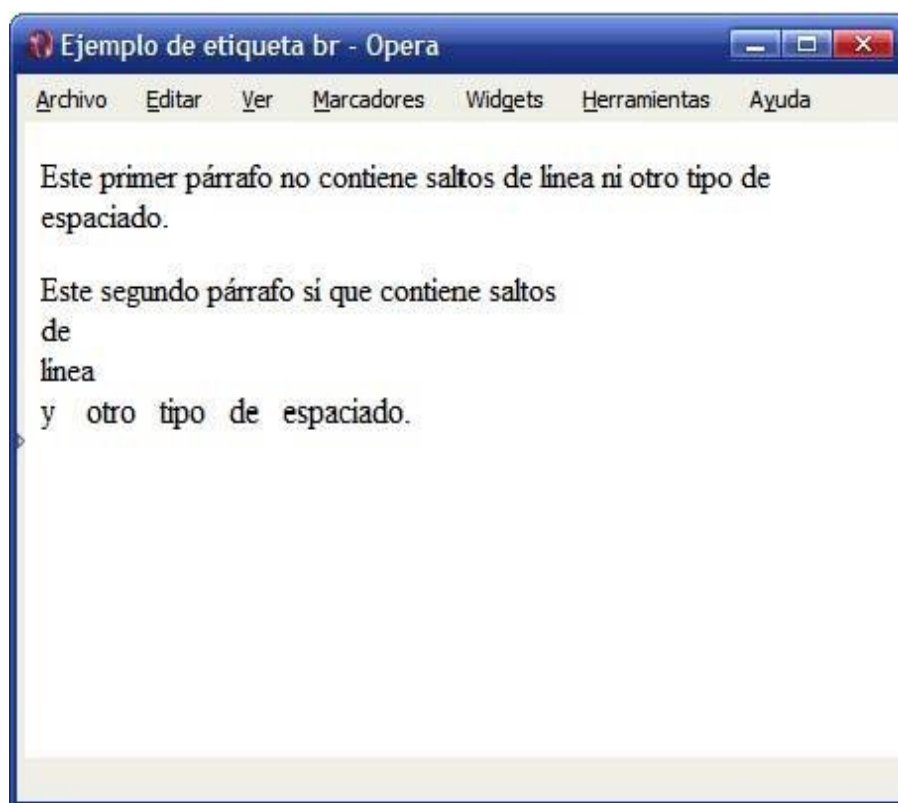
4.8 Espacios en blanco

La solución al problema de los espacios en blanco no es tan sencilla como el de las nuevas líneas. Para incluir espacios en blanco adicionales, se debe sustituir cada nuevo espacio en blanco por el texto ** **; (es importante incluir el símbolo & al principio y el símbolo ; al final).

Así, el código HTML del ejemplo anterior se debe rehacer para incluir los espacios en blanco adicionales:

[illegible]

Ahora el navegador sí que muestra correctamente los espacios en blanco (y las nuevas líneas) del segundo párrafo:



Cada texto ` ` solamente equivale a un espacio en blanco, por lo que se deben escribir tantos ` ` seguidos como espacios en blanco seguidos existan en el texto.

Más adelante se profundiza en el origen de ` ` y se comprenderá por qué es necesario incluir esa sucesión tan extraña de caracteres cada vez que se quiere incluir un espacio en blanco adicional.

4.9 Texto preformateado

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

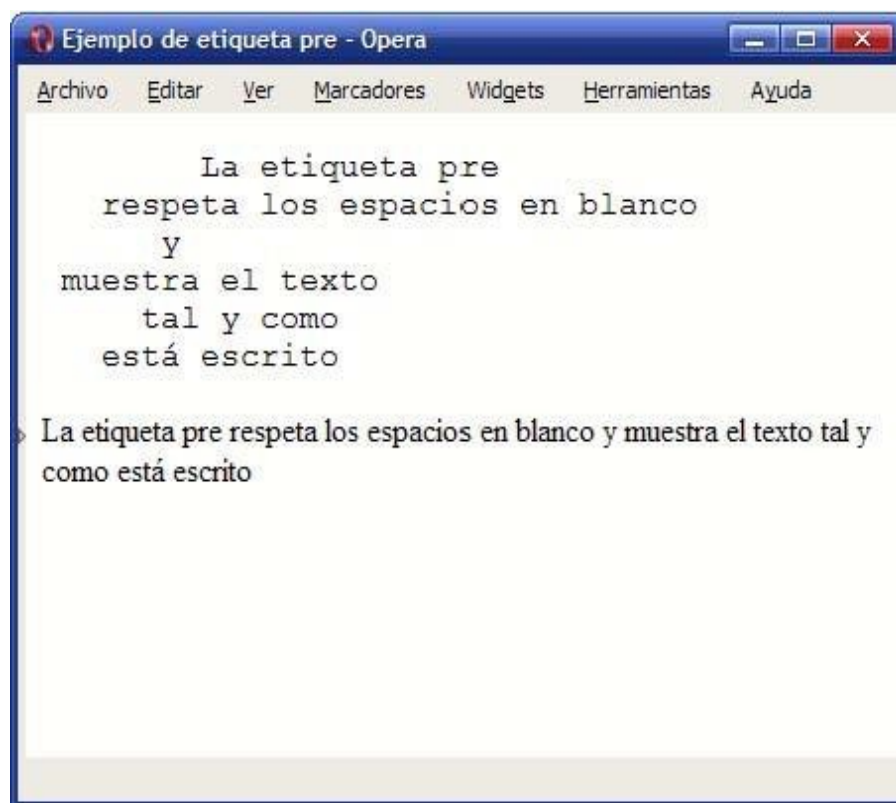
En estos casos, se puede utilizar la etiqueta `<pre>`, que muestra el texto tal y como se ha escrito, respetando todos los espacios en blanco y todas las nuevas líneas

El siguiente ejemplo muestra el uso de la etiqueta `<pre>`:

```
<html>

<head>
<title>Ejemplo de etiqueta pre</title>
</head>

<body>
<pre>La etiqueta pre
respeto los espacios en blanco y
muestra el texto tal y como
está escrito
</pre>
<p>
La etiqueta pre
respeto los espacios en blanco y
muestra el texto tal y como
está escrito</p>
</body>
</html>
```



El ejemplo anterior incluye el mismo texto (con espacios en blanco y varias líneas) dentro de una etiqueta `<pre>` y dentro de una etiqueta `<p>`. Las diferencias visuales en un navegador son muy evidentes:

El primer texto se ve en pantalla tal y como se ha escrito, respetando todos los espacios en blanco y todas las nuevas líneas. El segundo texto se ve como un párrafo normal, ya que HTML ha eliminado todos los espacios en blanco sobrantes. Los elementos `<pre>` son **especiales**, ya que los navegadores les aplican las siguientes reglas:

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

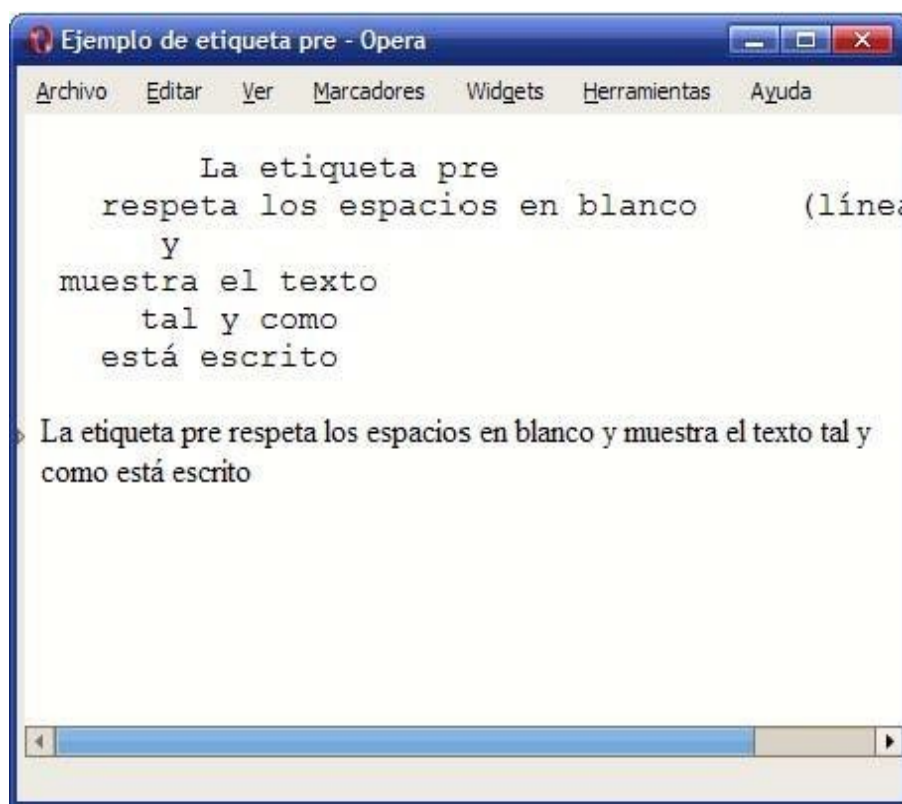
- Muestra el texto con un tipo de letra especial, denominado "**de ancho fijo**", ya que todas sus letras son de la misma anchura

- No se ajusta la longitud de las líneas (las líneas largas producen un **scroll** horizontal en la ventana del navegador)

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

Si en el ejemplo anterior se añade más texto al final de la segunda línea (para producir una línea

larga), el navegador muestra un **scroll** horizontal ya que el texto completo no cabe en el tamaño de la ventana y las líneas de los elementos `<pre>` nunca se ajustan.



4.10 Codificación de caracteres

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

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

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

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

De esta forma, si se considera el siguiente texto:

Los caracteres <, >, " y & pueden dar problemas con los textos en HTML.

Para mostrar correctamente el texto anterior en una página HTML, **se debe sustituir cada carácter especial por su entidad HTML**:

<p>Los caracteres <, >, " y & pueden dar problemas con los textos en HTML</p>

Por otra parte, los caracteres propios de los idiomas diferentes al inglés también pueden ser problemáticos.

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

| Entidad | Carácter | Descripción oficial |
|----------|----------|-----------------------------------|
| ñ | ñ | latin letter n with tilde |
| Ñ | Ñ | latin capital n letter with tilde |
| á | á | a acute |
| é | é | e acute |
| í | í | i acute |
| ó | ó | o acute |
| ú | ú | u acute |
| Á | Á | A acute |
| É | É | E acute |
| Í | Í | I acute |
| Ó | Ó | O acute |
| Ú | Ú | U acute |
| € | € | euro |

Pueden usarse con y sin ;

Así, la palabra párrafo se podría escribir de la siguiente manera:

```
<p>Este p&aacute;rrafo contiene caracteres acentuados y se
almacena en formato UTF-8</p>
```

Si se utilizan las entidades HTML en vez de los caracteres problemáticos, es indiferente pasar de una codificación de caracteres a otra diferente. En la Wikipedia se puede consultar la lista completa de las 252 [entidades HTML definidas](#)

4.11 Comentarios

Al igual que la mayoría de lenguajes de marcado, HTML permite incluir comentarios dentro de su código para añadir información que no se debe mostrar por pantalla.

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

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

La sintaxis de los comentarios es la siguiente:

```
Apertura del comentario: <!--
Contenido del comentario: (cualquier texto)
Cierre del comentario: -->
```

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

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

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

5. BIBLIOGRAFÍA

<https://www.um.es/atica/documentos/html.pdf>

<http://librosweb.es/libro/xhtml/>