

LENGUAJE HTML.

Conceptos básicos

Componentes de HTML

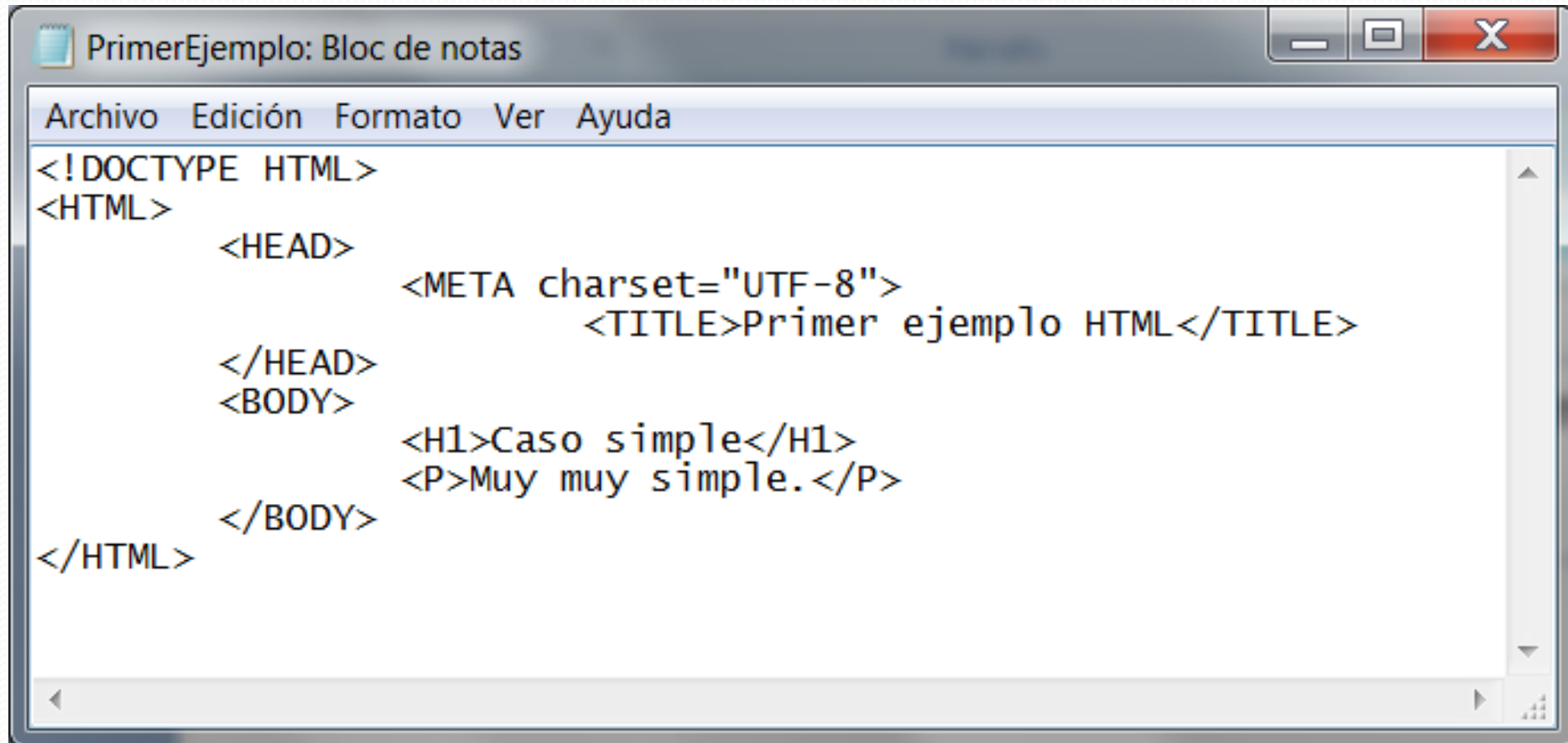
- HTML ha sido siempre un lenguaje bastante permisivo, en donde los navegadores suelen suplir los errores u olvidos de los programadores con decisiones propias, con el objetivo principal de que la página web se haga visible, pues para el navegador eso es lo fundamental, aunque no sea del modo esperado por el diseñador. Así se facilita su uso por parte de todo tipo de personas.
- Desde el punto de vista del programador web, HTML es un lenguaje simple y bastante estructurado en la forma. El código fuente suele ser fácil de seguir y de mantener; los principales problemas del codificador surgen cuando se dejan atributos sin codificar, y es el navegador quien se encarga de llenar esos vacíos. Esto se agrava si tenemos en cuenta que cada navegador, e incluso cada versión de éste, suele asignar sus propios valores por defecto, y que no siempre son coincidentes.

¿Qué sabemos de un documento HTML?

- Cada día, todos estamos viendo los resultados de la ejecución de código HTML al navegar por internet.
- A continuación, vamos a ver un ejemplo básico de una página HTML, primero su presentación en un navegador y después su código:



- El código que hay detrás de esta página es el siguiente:

A screenshot of a Notepad application window. The title bar reads 'PrimerEjemplo: Bloc de notas'. The menu bar includes 'Archivo', 'Edición', 'Formato', 'Ver', and 'Ayuda'. The text area contains the following HTML code:

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <META charset="UTF-8">
    <TITLE>Primer ejemplo HTML</TITLE>
  </HEAD>
  <BODY>
    <H1>Caso simple</H1>
    <P>Muy muy simple.</P>
  </BODY>
</HTML>
```

¿Cómo se generó esta página?

- Muy fácil, los pasos son los siguientes:
 1. En un bloc de notas o cualquier procesador de texto se introdujo el código correspondiente.
 2. Se guardó el archivo con extensión .htm.
 3. Se abrió el archivo con un navegador, en este caso Chrome.
- El navegador “interpreta” el código del archivo y genera el contenido visual en la pantalla. El modo de interpretar el código entre los distintos navegadores no es exactamente el mismo, aunque se va tendiendo a la estandarización, por lo que la visualización puede variar según el entorno utilizado para visualizar la página.

Introducción a HTML

- Los documentos HTML están compuestos por una estructura de elementos denominados etiquetas y por texto. Las etiquetas generalmente tienen una definición de apertura y otra de cierre, por ejemplo, `<BODY>` y `</BODY>`, pero algunas etiquetas de cierre, en ciertos casos, se pueden omitir

Concepto de etiquetas

- Las etiquetas pueden anidarse de manera que su contenido queda encerrado por etiquetas de nivel superior.
- Etiqueta HTML, elemento HTML o directiva de HTML son términos que significan lo mismo y que usaremos indistintamente.
- Nunca una etiqueta abierta dentro de la definición de una etiqueta de nivel superior quedará pendiente de cerrarse si se define el cierre de la etiqueta que la anida.

- Un ejemplo de lo que **no se permite** puede ser el siguiente:

```
<body>
```

```
    <h1>Ejemplo incorrecto</body>
```

```
</h1>
```

- La especificación del estándar define un amplio grupo de etiquetas que HTML puede interpretar, y qué atributos tiene cada tipo de etiqueta. Nos vamos a dedicar al estudio de las etiquetas más utilizadas.

Concepto de atributos de las etiquetas

- Podríamos definir los **atributos** como parámetros que nos permiten configurar el modo en que actúan las etiquetas.
- Los atributos, cuando son necesarios, siempre se incluyen en la etiqueta de apertura, nunca en la de cierre.
- Los atributos están compuestos por dos partes: *nombre* y *valor*, separados entre sí por el signo “=”. Si el valor se omite, por ser una cadena vacía, puede también omitirse el signo “=”.

<meta charset=UTF-8>

<input name=id1 disabled>

<!-- lo que es igual a ...>

< input name=id1 disabled="">

- Si el valor contiene espacios, comillas simples o dobles o algunos signos especiales (<, >, =), entonces deberán utilizarse comillas obligatoriamente.

- Resumiendo, las reglas de los atributos son las siguientes:
 1. Los atributos, cuando se utilizan, van después del nombre de la etiqueta de apertura, nunca en la de cierre.
 2. Para asignar un valor al atributo, el nombre del atributo debe ser seguido por el signo =.
 3. El valor del atributo sigue inmediatamente después del signo =.
 4. El valor del atributo se puede encerrar entre comillas simples o dobles.
 5. El atributo de tipo booleano es el que simplemente utiliza el atributo (sin signo = ni valor). Su presencia o no presencia es lo que lo define. En el ejemplo, *disabled* se considera un atributo booleano.

Estructura jerárquica del documento

- Cuando los navegadores interpretan el código HTML, lo convierten a un modelo de objeto DOM (Document Object Model), que es una representación en memoria del documento.
- La estructura del modelo utiliza un esquema en forma de árbol con distintos tipos de nodos y un nodo raíz.
- Si esquematizamos el código HTML del primer ejemplo, nos quedaría esta estructura, donde la etiqueta html es el elemento raíz del árbol DOM:



DOCTYPE

html

head

meta

title

texto

body


texto

h1

texto

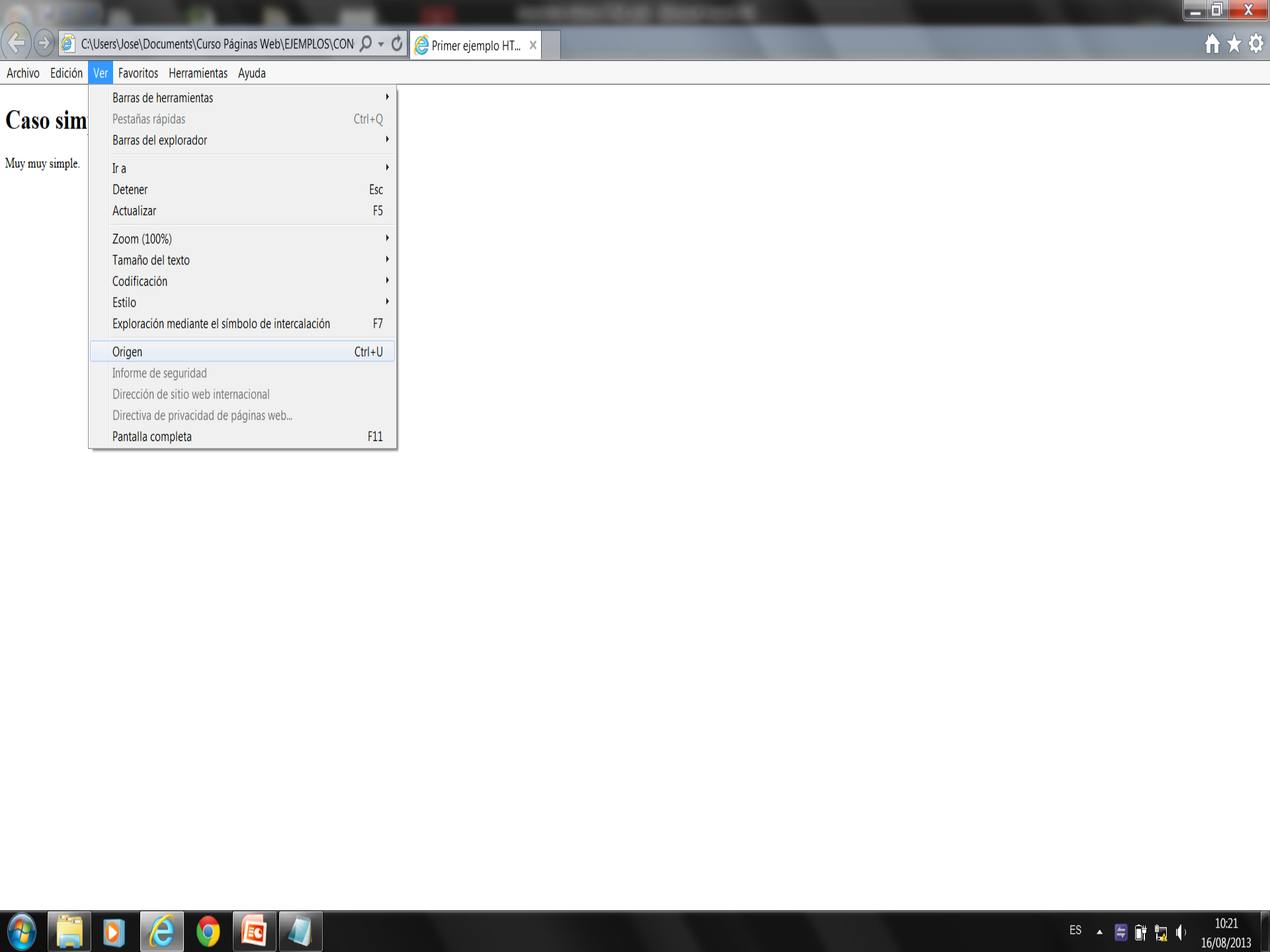
p

texto

- 
- Del elemento raíz dependen directamente dos nodos: head y body.
 - El elemento head contiene al elemento meta y al elemento title, que contiene, por su parte, un texto de título.
 - El elemento body contiene un elemento h1 y un elemento p.

Cómo se conoce el código fuente

- Los navegadores muestran la página web resultante de interpretar el archivo HTML recibido. Para saber qué es lo que hay detrás de esa página es necesario visualizar el código fuente.
- Todos los navegadores tienen la opción de mostrar el código fuente de la página que se está visualizando. En todos los navegadores es posible, pero en cada caso se hace de manera ligeramente diferente. En Internet Explorer, por ejemplo, se hace clic en el menú VER, y dentro de él en la opción Código Fuente, o la opción Origen, según la versión del navegador:



Caso sim

Muy muy simple.

- Barras de herramientas
- Pestañas rápidas
- Barras del explorador
- Ir a
- Detener
- Actualizar
- Zoom (100%)
- Tamaño del texto
- Codificación
- Estilo
- Exploración mediante el símbolo de intercalación
- Origen
- Informe de seguridad
- Dirección de sitio web internacional
- Directiva de privacidad de páginas web...
- Pantalla completa

El texto y los caracteres especiales

- En su origen, HTML era un lenguaje muy orientado a la gestión del texto, y así es como están formados la mayoría de los documentos, pero la evolución lo ha convertido en un lenguaje con capacidad para tratar distintos tipos de formatos de información: gráficos, sonidos y vídeos.
- El texto es una cadena de caracteres y HTML se encarga de representarlo en la salida impresa en la ventana del navegador.
- En el primer ejemplo, hemos usado un texto que no representaba problemas para el navegador, no había acentos, diéresis ni símbolos especiales, pero, como lenguaje gestor de contenidos de texto, HTML se encontró con un primer problema: los denominados caracteres especiales.
- Como siempre en informática, el inglés tiene un tratamiento preferencial y en los otros idiomas se piensa en un segundo término. Esto provoca que caracteres propios de otros lenguajes (letras acentuadas, ñ, etc...) para HTML sean especiales, y por lo tanto deben codificarse de modo especial. El código ASCII de 7 bits es muy limitado para las lenguas no inglesas.

Solución elegida: texto codificado

- La solución es el uso de caracteres especializados que incluso pueden utilizar estándares de 16 bits, lo cual da muchas más posibilidades.
- El modo de definir el juego de caracteres es mediante la etiqueta `<meta charset="...">` para indicar el estándar elegido, por ejemplo, UNICODE (UTF-8), como en la página de ejemplo.
- Para superar problemas puntuales de algún carácter especial, se puede utilizar la siguiente codificación:
 - `&#NRO`: siendo NRO un valor entero que indica el código del carácter. Por ejemplo, `ñ` es la letra ñ.
 - `OD`: siendo COD un valor hexadecimal. Por ejemplo, `ñ` es la letra ñ, pero en hexadecimal.
 - `á`: indica la á (a con acento agudo).
 - `ñ`: indica la ñ (n con tilde).
 - `ò`: indica la ò (o con acento grave).
- Hay gran cantidad de valores simbólicos: acute, grave, circ, tilde, uml, pound, reg...

Etiquetas

- El elemento básico de la codificación HTML es la etiqueta, que es el modo en que el desarrollador define las directivas para obtener los resultados. Las etiquetas tienen el siguiente formato:

<NombreEtiqueta atributos...> **</NombreEtiqueta>**

- Los signos < y > se utilizan como delimitadores de la etiqueta y una variante aceptada para indicar el final de una etiqueta es utilizar la siguiente codificación:

<NombreEtiqueta atributos... />

- En general, las etiquetas o directivas HTML actúan como contenedores de bloques de texto y que, además, pueden encerrar otras etiquetas.
- El inicio del bloque se define con el nombre de la directiva o etiqueta (también denominado *tag*) entre los signos <>, por ejemplo <P>.

- El final del bloque se define con el formato, en el caso de la etiqueta de párrafo `<P>`, `</P>`. Uno de los olvidos más comunes es no colocar el final del bloque y dejar así la directiva abierta. El resultado de este error dependerá en gran parte de la secuencia de etiquetas que viene después de la etiqueta olvidada y también de la acción del navegador.
- Un bloque de etiquetas puede encerrar de modo anidado otros bloques de otras etiquetas. El efecto de estos anidamientos es que se acumulan directivas sobre el contenido.
- Por ejemplo, este código que se muestra a continuación define un párrafo de texto normal:
`<p>Ésta es un línea de texto normal.</p>`
- Lo que aparecerá en el navegador con este estilo:
Ésta es un línea de texto normal.
- Si se quiere remarcar con negrita una parte del texto se utiliza la etiqueta ``:
`<p>Ésta es un línea de texto normal.</p>`
- Esto aparecerá en el navegador con el siguiente estilo:
Ésta es un **línea de texto** normal.

- `<P>` y `<p>` es exactamente lo mismo: para HTML, es indistinto codificar las etiquetas y los atributos en mayúsculas o en minúsculas. Simplemente, para dar mayor claridad al código, podríamos crearnos nuestra propia convención y utilizar mayúsculas para identificar más fácilmente las etiquetas, pero es totalmente opcional.
- El lenguaje XHTML tiende a ser muy estricto respecto a los cierres de los bloques, pero es una realidad que los desarrolladores siempre han sido muy olvidadizos con las etiquetas de cierre, y HTML 5 es más permisivo y tolerante con algunas de estas distracciones, siguiendo la actitud de los navegadores, que siempre lo han sido.

Atributos

- Una parte importante de las etiquetas son los atributos, que es la manera que tiene el lenguaje para definir las propiedades de la entidad (forma sinónima para indicar atributos de etiqueta o directiva).
- Los atributos permiten modificar la configuración predeterminada de la etiqueta y siempre se definen en la etiqueta de apertura (o inicio) y jamás en la de cierre. Por ejemplo, ésta sería una etiqueta `<A>` con su atributo *href*:
`Haga clic aquí para ir al índice.`
- En este caso, el atributo *href* nos permite definir la dirección de enlace. En HTML 5 ya no es obligatorio que el valor del atributo se codifique entre comillas, salvo que contenga caracteres especiales.

Atributos comunes a todas las etiquetas

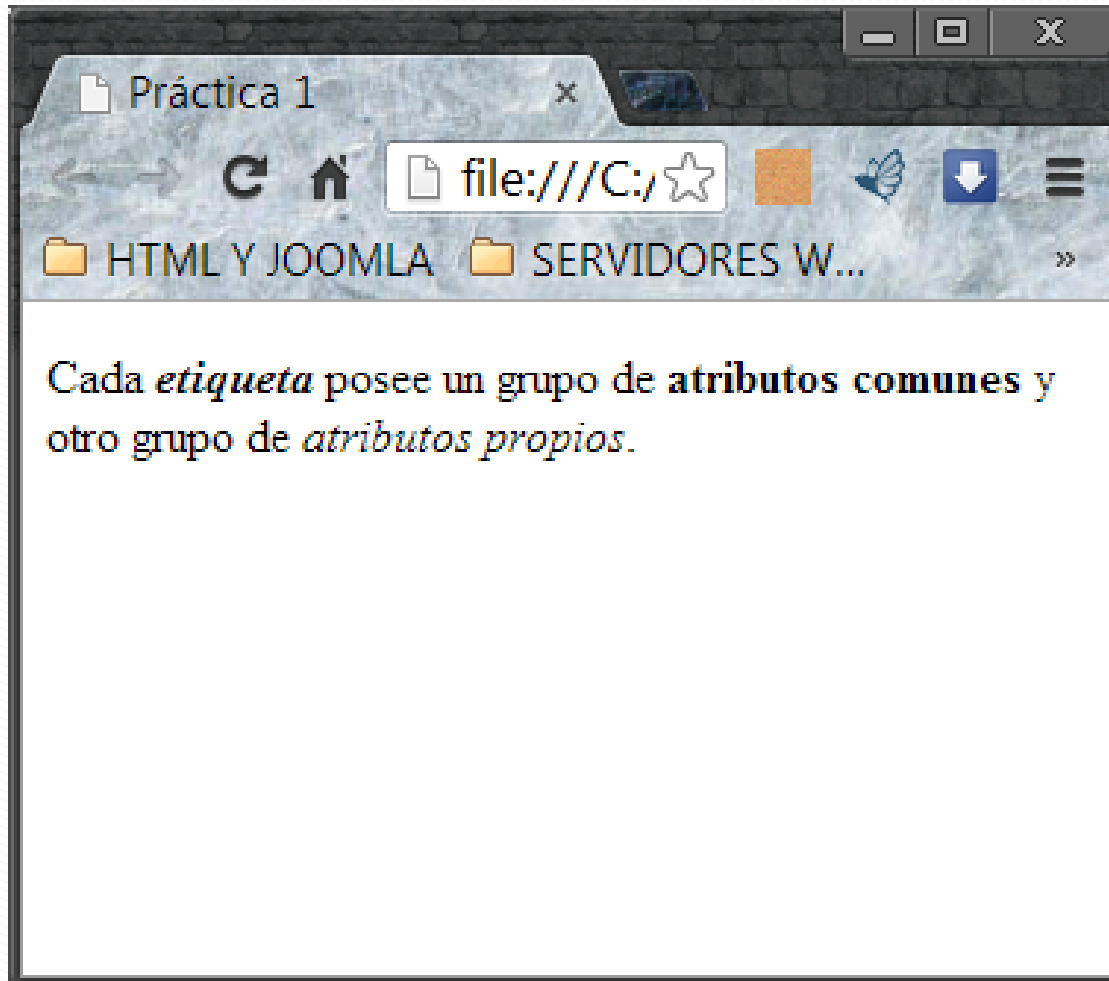
Cada tipo de etiqueta tiene sus propios atributos, pero existe un conjunto de atributos comunes a todas las etiquetas, de los cuales veremos algunos:

- **accesskey:** Permite definir un acceso directo de teclado para que el elemento reciba el foco. Su funcionamiento depende bastante del entorno (navegador, versión y sistema operativo). En Windows con IE la tecla funciona pulsándola junto con la tecla ALT.
- **class:** Permite definir una clase a la directiva, que luego se asocia a una clase CSS (Hoja de estilo en cascada).
- **contextmenu:** Permite asociar un menú de contexto al elemento. El menú debe estar definido en una etiqueta <MENU>.
- **contenteditable:** Es un valor booleano (true o false) que determina si el contenido delimitado por la etiqueta se puede o no modificar. El contenido debe ser seleccionable y el navegador debe suministrar un símbolo que marca la posición vigente de la edición.

- **dir**: Indica la direccionabilidad del elemento (ltr, left to right, izquierda a derecha, o, rtl, right to left, derecha a izquierda).
- **draggable**: Indica que el elemento puede ser arrastrado usando la API drag-and-drop.
- **hidden**: Este atributo le indica al navegador que el contenido de este elemento no debe mostrarse. Mediante código JavaScript podríamos hacer aparecer este contenido en un momento posterior.
- **id**: Indica el identificador único de un elemento. No puede contener espacios. Este valor se puede utilizar como referencia en scripts o en CSS.
- **lang**: Indica el idioma del elemento.
- **spellcheck**: Indica si se comprueba la gramática y ortografía (true) o no (false).
- **style**: Define el estilo aplicable a la etiqueta.
- **tabindex**: Indica el orden de tabulación (obtención del foco) del elemento o si el elemento no es enfocable.
- **title**: Define un texto que se utiliza como microayuda (tooltip) para el elemento.
- **eventos**: Una larga serie de eventos se pueden asociar a cualquier elemento HTML para ejecutar un script relacionado, por ejemplo, onclick, ondblclick, onmouseup, onmouseover, onfocus, onplay, onreset, etc...

- **Práctica:** Codificar este párrafo de texto:

Cada ***etiqueta*** posee un grupo de **atributos comunes** y otro grupo de *atributos propios*:



Comentarios

- HTML es un lenguaje que combina código de programación con contenido textual, y eso tiene una desventaja muy evidente: el archivo resultante suele ser bastante difícil de leer.
- En todos los lenguajes de programación existen modos de documentar el código para facilitar el mantenimiento, y HTML no es diferente en ese aspecto, por lo que también tiene la posibilidad de incluir comentarios (los navegadores ignoran totalmente los comentarios y no los visualizan en la página resultante).
- El modo de introducir comentarios es muy simple: hay que comenzar el comentario con los símbolos `<!--` y finalizarlos con los símbolos `-->`.
- Podemos incluir cualquier comentario en uno de los ejemplos que ya tenemos y al visualizarlo veremos como el navegador lo ignora a la hora de mostrar la página resultante.

Estructura básica de los documentos HTML

- Para ser considerado documento HTML, existían pocas reglas y por eso no había una estructura obligatoria. En ejemplos anteriores, hemos creado un documento HTML simplemente con la etiqueta <P>, e incluso podría no haber estado, y aún así el navegador hubiera interpretado el archivo .htm y lo hubiese presentado asumiendo una serie de valores no informados en el documento.
- Esta libertad tiene su parte negativa, ya que los navegadores, ante la falta de directivas, asumen valores que no siempre son los que necesitamos para presentar el contenido. El estándar HTML 5 es más estricto. En él, para que un documento se pueda considerar HTML 5 debe cumplir ciertos requisitos mínimos. En caso contrario, dado que los navegadores son compatibles hacia atrás, será interpretado como un documento HTML, pero no como HTML 5.

- Existen dos elementos básicos que no pueden faltar para que el documento sea tratado como documento HTML 5:
 - Un comentario inicial `<!DOCTYPE HTML>` que indica que el documento está escrito en HTML.
 - Una etiqueta `<TITLE></TITLE>`.
- El elemento DOCTYPE ya existía previamente, pero su codificación era más compleja. La etiqueta TITLE tiene la misma función de siempre, da el título a la ventana.

Estructura general

La estructura de un documento HTML la podemos definir en estas categorías:

- Secciones principales.
- Bloques de texto.
- Marcas de caracteres.
- Caracteres especiales.
- Elementos de organización.
- Enlace a otras páginas.
- Imágenes.
- Scripts.

Vamos a presentar brevemente cada una de estas categorías:

Secciones principales

- Todos los documentos HTML poseen en común tres etiquetas: `<HTML>`, `<HEAD>` y `<BODY>`, que delimitan las principales secciones del documento. Estos elementos usan siempre una etiqueta de cierre que indica hasta donde abarca el bloque, por ejemplo, `<HEAD>....</HEAD>`.

Bloques de texto

- Como sucede en los procesadores de texto, HTML incluye varias etiquetas para dar formato a los bloques de texto. Estos elementos utilizan siempre una etiqueta de cierre que indica hasta donde abarca el bloque, por ejemplo, `<PRE>....</PRE>`. Son las siguientes:
 - `<P>`: Para definir párrafos formateados.
 - `<H1>...<H6>`: Para definir encabezamientos.
 - `<BLOCKQUOTE>`: Para definir texto de un segundo nivel.
 - `<PRE>`: Para definir texto preformateado.
 - ``, ``, `<DL>`: Para definir listas sin numerar, ordenadas y definiciones.
 - `<DIV>`: Una división de documentos.

Marcas de caracteres

- El texto puede contener elementos inline (embebidos dentro del texto) para realizar marcas de caracteres. Estos elementos utilizan siempre una etiqueta de cierre que indica hasta donde abarca la marca, por ejemplo, Algunos de los elementos de este tipo son los siguientes:
 - : Negrita.
 - <I>: Cursiva.
 - <BIG>: Texto grande.
 - <SMALL>: Texto pequeño.
 - : Texto resaltado.
 - : Texto en negrita.

Caracteres especiales

- Algunos caracteres no pueden usarse directamente en el texto del documento, y para resolver estos problemas se utilizan entidades que representan estos caracteres. Una manera de reemplazar estos caracteres es mediante el uso del símbolo **&**, por ejemplo:
 - `<`: Se mostrará como `<`.
 - `>`: Se mostrará como `>`.
 - `&`: Se mostrará como `&`.
 - ` `: Indica un espacio que no debe producir rotura, para mantener juntas las palabras.
 - `&#xxx;`: Siendo xxx un número que indica el código ASCII del símbolo. Por ejemplo, `©` indica el símbolo de copyright.

Elementos de organización

- HTML incluye dos elementos para organizar la información en el documento:
 - `<TABLE>`: Para ordenar los datos en forma de filas y columnas. Dentro del grupo `<TABLE>` se utiliza una serie de etiquetas internas: `<TR>`, `<TD>`, `<TH>`, etc... Dentro de una celda de datos de una tabla puede haber información de cualquier tipo: una imagen, un vídeo, un formulario, una página... Estos elementos siempre utilizan una etiqueta de cierre que indica hasta donde llega el bloque, por ejemplo, `<TR>...</TR>`.
 - `<FORM>`: para crear un elemento que permite mostrar y obtener datos. Dentro de la etiqueta `<FORM>` se pueden utilizar varias etiquetas que representen cuadros de texto, botones, archivos, áreas de texto, casillas de verificación, botones de opción... Estos elementos siempre utilizan una etiqueta de cierre que indica hasta donde llega el bloque, por ejemplo, `<FORM>...</FORM>`.

Enlace a otras páginas (o internos)

- La base de la navegación web se realiza mediante los enlaces.
- En HTML, esto se implementa principalmente mediante la etiqueta <A>, la cual permite definir la referencia, que puede ser una ubicación interna del sitio, incluso dentro de la propia página desde donde se hace la referencia, o externa, en cualquier otro sitio web.

Imágenes

- En los documentos HTML podemos incluir imágenes y recursos multimedia en general. La etiqueta `` incluye un enlace a un archivo de imagen con información adicional para indicar cómo mostrarla en la interfaz.

Scripts

- HTML es básicamente un mecanismo para presentar contenido estático en la interfaz de usuario.
- La necesidad de gestionar la interactividad con el usuario se resuelve mediante el uso de código de programa (por ejemplo, JavaScript) que se ejecuta dentro del navegador cliente sin necesidad de utilizar una transmisión de datos hasta el servidor.
- La etiqueta <SCRIPT> permite pasar código al *agente de usuario* dentro del código HTML. Opcionalmente se puede utilizar un archivo externo.
- Un *agente de usuario* es cualquier aplicación informática que funciona como cliente en un protocolo de red; de modo genérico, es una aplicación que accede a la Word Wide Web. Por lo tanto, todos los navegadores web son agentes de usuario, y también lo son los robots de los buscadores y otros programas similares.

- El uso de la etiqueta `<HTML>` parecería redundante, dado que `DOCTYPE` ya lo indica, pero no debemos limitarnos al uso que vemos normalmente de este tipo de documentos, y sí pensar que el código HTML también se utiliza combinado con otros elementos, por ejemplo, dentro de un mensaje de correo electrónico, y esto hace necesario el uso de los delimitadores `<HTML></HTML>`.
- La cabecera queda definida dentro de las etiquetas `<HEAD>` y `</HEAD>`, y es allí donde se incluye información relativa al propio documento.
- Las etiquetas `<BODY>` y `</BODY>` encierran el contenido del cuerpo del documento.
- Toda esta estructura es prácticamente fija en todos los documentos HTML, tanto es así que casi todos los editores de HTML la insertan automáticamente cuando creamos un nuevo documento.

Etiqueta <HTML>

- Ya sabemos qué es lo que se encuentra en una estructura básica que encontraremos en un documento HTML; veamos ahora cómo se forma la sección principal del documento:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>El nombre de la ventana</title>
```

```
  </head>
```

```
  <body>
```

```
    Algo de texto
```

```
  </body>
```

```
</html>
```

Etiqueta <HEAD>

- Esta directiva o etiqueta delimita la sección de encabezamiento y encierra un bloque de información sobre el documento (metadatos), y suele contener el siguiente grupo de etiquetas:
 - <TITLE>
 - <BASE>
 - <LINK>
 - <META>
 - <ISINDEX>
- Por ejemplo, éste es un bloque <HEAD> que utiliza parte de estas etiquetas:



```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <BASE href="http://www.misitio.es/">
```

```
    <title>Mi primer HTML</title>
```

```
    <link REL="STYLESHEET" HREF="default.css">
```

```
    <script SRC="code01.js"></script>
```

```
    <meta NAME="APPLICATION-NAME" CONTENT="Ejemplo 01">
```

```
  </head>
```

```
  <body>
```

```
...
```


Etiqueta <TITLE>

- Esta etiqueta es casi obligatoria para que el documento sea considerado HTML pero, aunque parece una etiqueta inofensiva e inocua que sólo aporta el título de la ventana, es una directiva muy importante para que una página web sea bien posicionada en los buscadores.
- Los robots de análisis de páginas, es decir, los programas alimentadores de las bases de datos y de los índices de los buscadores, le asignan a esta etiqueta un valor muy alto. Por tanto, el título debe ser lo más adecuado posible para definir el contenido de la página.
- La etiqueta <TITLE> y el identificador de fin del bloque </TITLE> delimitan el contenido de texto que los navegadores colocan como título de la ventana. Se recomienda que el título no sea excesivamente largo para que se pueda visualizar de modo completo.

Etiqueta <BASE>

- Esta etiqueta nos permite especificar un directorio que actuará como base para las referencias relativas que utilizaremos en el documento HTML.
- Esta dirección, cuando se informa, reemplaza a la base que se toma por defecto, que es la dirección del propio documento cuando se compone una dirección informada de modo relativo (una dirección absoluta es lo opuesto, es toda la ruta de acceso completa para llegar al documento).
- Es normal que en un documento HTML tengamos que realizar referencias a direcciones URL en distintos elementos de la página, por ejemplo, para indicar el sitio desde donde se obtienen las imágenes.
- Los atributos específicos de este elemento son los siguientes:
 - **target:** Si se especifica debe contener un nombre de destino válido (ventana) que se utilizará como entorno de navegación de hiperenlaces y formularios del documento.
 - **href:** Dirección URL válida que se utiliza como dirección base.

- Volviendo al ejemplo anterior, supongamos que ésta es una página web que está en la siguiente dirección web:

`http://www.miblog.com/mihtml.htm`

- Y que tiene este contenido:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <base href="http://www.misitio.es/">
```

```
    ...
```

```
  </head>
```

```
  ...
```

- Si tenemos una imagen con la siguiente codificación de la etiqueta (es una etiqueta para definir archivos de imágenes):

```

```

```
...
```

```
</html>
```

- Esta imagen se buscará en el sitio web siguiente:
`http://www.misitio.es/imagenes/punto.jpg`
- En cambio, si no existiese la directiva <BASE>, se buscaría en la siguiente dirección:
`http://www.miblog.com/imagenes/punto.jpg`
- En un documento sólo debería haber una directiva <BASE> como máximo, ya que si hay más se ignoran todas salvo la primera. Además, debe aparecer antes de que se utilice cualquier referencia relativa, y sólo afecta al documento en donde aparece.
- Si estamos trabajando en local, y nuestras imágenes estuvieran, por ejemplo, en la ruta “C:/PruebasHTML/Imag/”, usaríamos la siguiente directiva <BASE>:
`<base href=“file://localhost/C:/Pruebas/imag/”>`
- Practica: crear un HTML con título “Mi práctica más simple” y con una directiva que permita que las imágenes se busquen en el sitio web “www.almacendeimagenes.com”.

Etiqueta <LINK>

- Permite establecer enlaces con otros recursos. Sus atributos fundamentales son href y rel, el primero debe tener una dirección URL válida, y el segundo indica el tipo de enlace. Si se omite alguno de estos dos atributos, la etiqueta no tendrá ningún efecto. Los principales atributos son:
 - **rel**: Indica el tipo de enlace, normalmente se usa para enlazar con hojas de estilo, por lo que su valor es stylesheet. Existen otros tipos de relaciones pero menos frecuentes: alternate (versiones alternativas), author (información del autor del documento), prev (documento de nivel superior), next (documento siguiente en el orden). El tipo de relación prefetch se utiliza para que el navegador use los tiempos ociosos de la conexión para realizar la precarga de documentos, para que estén disponibles en el momento en que el usuario los requiera sin tener que esperar el tiempo de descarga.
 - **href**: Permite especificar la dirección del documento enlazado. Este atributo es obligatorio, si se quiere que la etiqueta funcione realmente como enlace.
 - **hreflang**: El lenguaje del recurso.
 - **title**: Para asignar un título a la relación.
 - **media**: Indica el tipo de dispositivo más adecuado para presentar el recurso (pantalla, impresora, sintetizador e voz...).

- Para definir relaciones se utilizan directivas independientes, una por enlace:

<!-- Contenido -->

<link rel="Contents" href="miContenido.html"/>

<!-- Hoja de estilo -->

<link rel="StyleSheet" type="/text/css" href="miCSS.css"/>

<!-- Precarga de un archivo -->

<link rel="prefetch" href="milmagen.jpeg"/>

- La etiqueta LINK también se suele usar para indicar a los motores de búsqueda dónde pueden encontrar la versión alternativa, es decir, en otro idioma, del documento vigente:


<link rel="alternate" title="Versión inglesa" type="text/html"
href="http://www.miPagina.com/pag01.html" hreflang=en/>

Etiqueta <META>

- La etiqueta <META> es un contenedor de metainformación relacionada con el documento. El uso más habitual es para incluir información que queremos que sea utilizada por los robots de los buscadores. Se usa para metadatos que no se pueden expresar mediante las otras etiquetas del grupo: TITLE, BASE, LINK o STYLE.
- Cada etiqueta <META> debe tener siempre uno de estos tres atributos: name, charset o http-equiv.
- Los atributos más utilizados son los siguientes:
 - **name:** identifica el tipo de la información, por ejemplo, application-name, keywords, description, author, generator, robots.
`<meta name=keywords content="asistentes, ayudas, help">`
`<!-- Para que los robots no indexen el documento en los motores de búsqueda -->`
`<meta name=robots content="noindex, nofollow">`
 - **charset:** Determina el conjunto de caracteres del documento. Como máximo, puede haber una etiqueta <META charset=...>. Es decir, se define una única vez.
`<meta charset = "utf-8">`
 - **content:** Indica los metadatos. Su contenido depende del tipo de etiqueta <META>.
`<meta name="author" content="José Luis Briega">`

Etiqueta <STYLE>

- La etiqueta <STYLE> permite incluir información sobre el estilo dentro del propio documento. Puede tener estos atributos:
 - **type**: indica el lenguaje de estilo, y debe ser un tipo MIME válido. En caso de que no se informe, el atributo type se presupone “text/css”.
 - **media**: Indica a qué medio se aplica el estilo. Si no se indica, se presupone “all”, es decir, que se aplica a todos los medios del documento.
 - **title**: Define una hoja de estilo alternativa.

- 
- Con todo esto podemos ya crear un ejemplo muy simple utilizando la estructura formal explicada: en HTML 5, la etiqueta `<HTML>` sigue siendo la que encierra todo el contenido del archivo HTML.
 - En la cabecera `<HEAD>` se especifica un título y en el contenido, por ahora, nos limitaremos a incluir un párrafo corto de texto:

```
<!DOCTYPE html>
```

```
<!-- definimos un estilo para el cuerpo normal y otro para la etiqueta  
    <em></em>-->
```

```
<html>
```

```
    <head>
```

```
        <title>Un ejemplo de style</title>
```

```
        <style>
```

```
            body {color:blue; background: yellow;}
```

```
            em {font-weight: bold; color: red;}
```

```
        </style>
```

```
    </head>
```

```
    <body>
```

```
        <p>Pongamos un poco de <em>énfasis en esta parte</em> y  
sigamos con cuerpo normal.</p>
```

```
    </body>
```

```
</html>
```