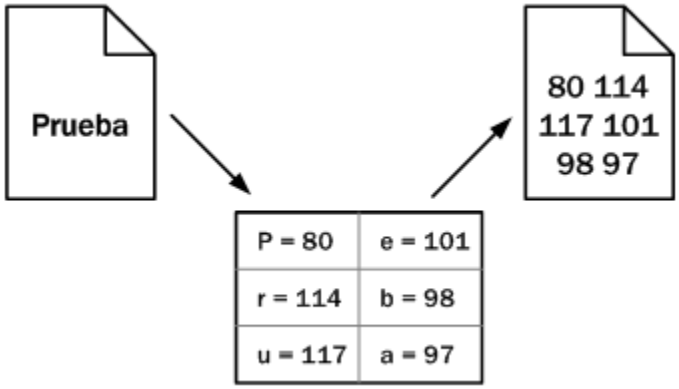
Características básicas

Lenguajes de etiquetas

Uno de los retos iniciales a los que se tuvo que enfrentar la informática fue el de cómo almacenar la información en los archivos digitales. Como los primeros archivos sólo contenían texto sin formato, la solución utilizada era muy sencilla: se codificaban las letras del alfabeto y se transformaban en números.

De esta forma, para almacenar un contenido de texto en un archivo electrónico, se utiliza una tabla de conversión que transforma cada carácter en un número. Una vez almacenada la secuencia de números, el contenido del archivo se puede recuperar realizando el proceso inverso.



Ejemplo sencillo de codificación de caracteres

El proceso de transformación de caracteres en secuencias de números se denomina **codificación de caracteres** y cada una de las tablas que se han definido para realizar la transformación se conocen con el nombre de **páginas de código**. Una de las codificaciones más conocidas (y una de las primeras que se publicaron) es la codificación ASCII. La importancia de las codificaciones en HTML se verá más adelante.

Una vez resuelto el problema de almacenar el texto simple, se presenta el reto de almacenar los contenidos de texto con formato. En otras palabras, ¿cómo se almacena un texto en negrita? ¿y un texto de color rojo? ¿y otro texto azul, en negrita y subrayado?

Utilizar una tabla de conversión similar a las que se utilizan para textos sin formato no es posible, ya que existen infinitos posibles estilos para aplicar al texto. Una solución técnicamente viable consiste en almacenar la información sobre el formato del texto en una zona especial reservada dentro del propio archivo. En esta zona se podría indicar dónde comienza y dónde termina cada formato.

No obstante, la solución que realmente se emplea para guardar la información con formato es mucho más sencilla: el archivo electrónico almacena tanto los contenidos como la información sobre el formato de esos contenidos. Si por ejemplo

se quiere dividir el texto en párrafos y se desea dar especial importancia a algunas palabras, se podría indicar de la siguiente manera:

```
<parrafo>
Contenido de texto con <importante>algunas palabras</importante> resaltadas de forma especial.
</parrafo>
```

El principio de un párrafo se indica mediante la palabra `<parrafo>` y el final de un párrafo se indica mediante la palabra `</parrafo>`. De la misma manera, para asignar más importancia a ciertas palabras del texto, se encierran entre `<importante>` y `</importante>`.

El proceso de indicar las diferentes partes que componen la información se denomina **marcar** (*markup* en inglés). Cada una de las palabras que se emplean para marcar el inicio y el final de una sección se denominan **etiquetas**.

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 `>`
- Etiqueta de cierre: carácter `<`, seguido del carácter `/`, seguido del nombre de la etiqueta (sin espacios en blanco) y terminado con el carácter `>`

Así, la estructura típica de las etiquetas HTML es:

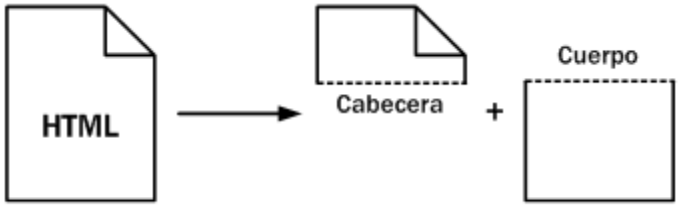
```
<nombre_etiqueta> ... </nombre_etiqueta>
```

HTML es un **lenguaje de etiquetas** (también llamado **lenguaje de marcado**) y las páginas web habituales están formadas por cientos o miles de pares de etiquetas. De hecho, las letras "ML" de la sigla HTML significan *"markup language"*, que es como se denominan en inglés a los *lenguajes de marcado*. Además de HTML, existen muchos otros lenguajes de etiquetas como XML, SGML, DocBook y MathML.

La principal ventaja de los lenguajes de etiquetas es que son muy sencillos de leer y escribir por parte de las personas y de los sistemas electrónicos. La principal desventaja es que pueden aumentar mucho el tamaño del documento, por lo que en general se utilizan etiquetas con nombres muy cortos.

El primer 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.



Esquema de las partes que forman un documento HTML

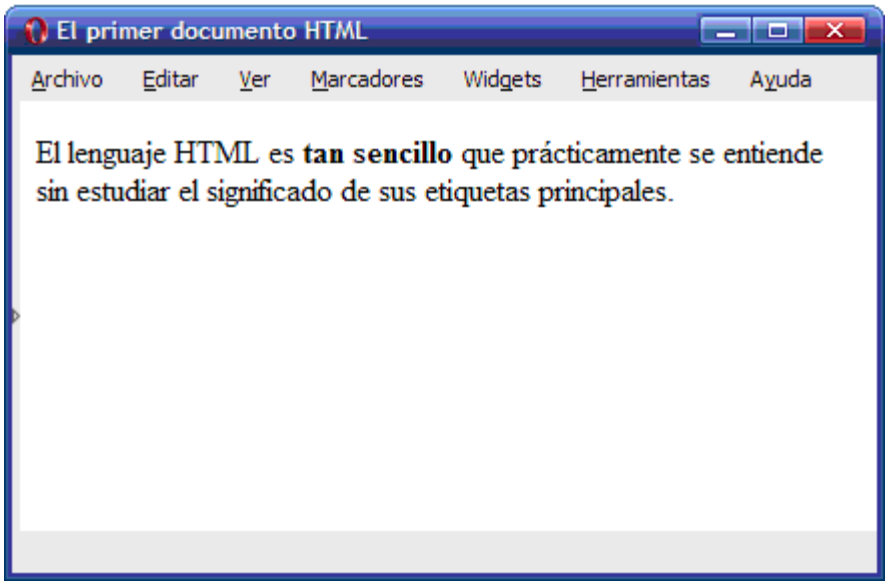
El cuerpo (llamado *body* en inglés) contiene todo lo que el usuario ve en su pantalla y la cabecera (llamada *head* en inglés) contiene todo lo que no se ve (con la única excepción del título de la página, que los navegadores muestran como título de sus ventanas).

A continuación se muestra el código HTML de una página web muy sencilla:

```
<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 que el ejemplo anterior funcione correctamente, es imprescindible que se utilice un editor de texto sin formato. Si el sistema operativo es Windows, se puede utilizar el *Bloc de notas*, *Wordpad*, *EmEditor*, *UltraEdit*, *Notepad++*, etc. pero no puedes utilizar un procesador de textos como *Word* o *Open Office*. Si utilizas sistemas operativos tipo Linux, puedes utilizar editores como *Gedit*, *Kedit*, *Kate* e incluso *Vi*, pero no utilices *KOffice* ni *Open Office*.

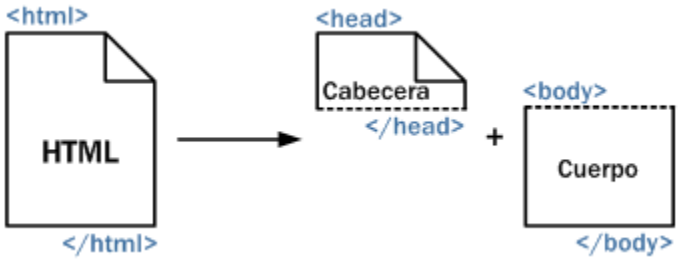
Después de crear el archivo con el contenido HTML, ya se puede abrir con cualquier navegador para que se muestre con el siguiente aspecto:



Aspecto que muestra el primer documento HTML en cualquier navegador

Volviendo al código HTML del primer ejemplo, es importante conocer las tres etiquetas principales de un documento HTML (`<html>`, `<head>`, `<body>`):

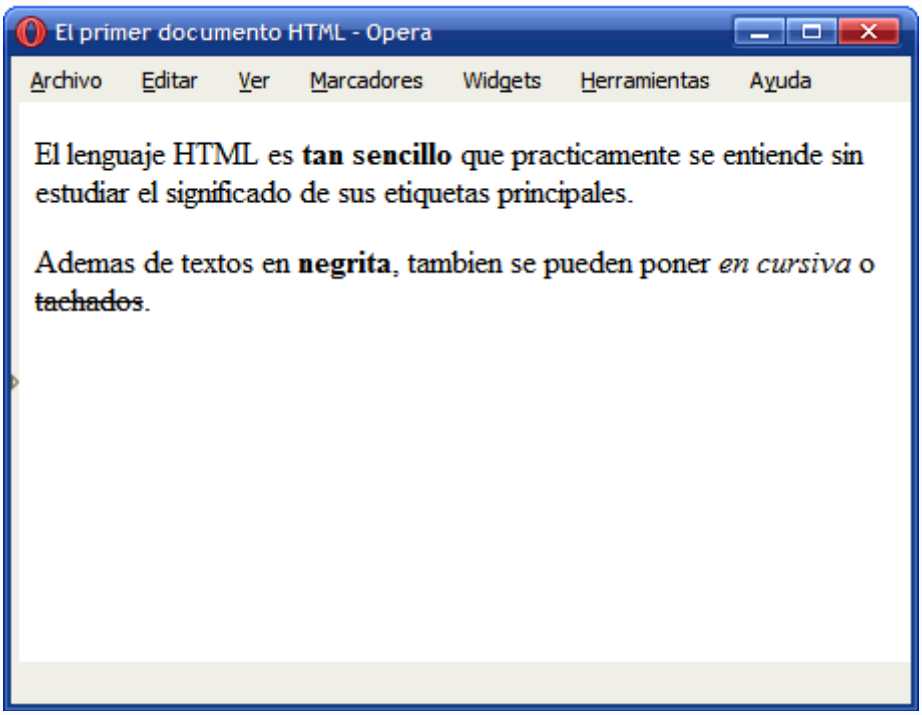
- `<html>`: indica el comienzo y el final de un documento HTML. Ninguna etiqueta o contenido puede colocarse antes o después de la etiqueta `<html>` (con una sola excepción que se verá más adelante). En el interior de la etiqueta `<html>` se definen la cabecera y el cuerpo del documento HTML y todo lo que se coloque fuera de la etiqueta `<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 y el idioma de la página. 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 .
- `<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.



Esquema de las etiquetas principales que contiene un documento HTML

Ejercicio 1 (Ejercicio1.html)

Determinar el código HTML correspondiente a la siguiente página:



Página HTML sencilla que resalta algunas partes del texto

Etiquetas y atributos

HTML define 91 etiquetas que los diseñadores pueden utilizar para *marcar* los diferentes elementos que componen una página:

`a`, `abbr`, `acronym`, `address`, `applet`, `area`, `b`, `base`, `basefont`, `bdo`, `big`, `blockquote`, `body`, `br`, `button`, `caption`, `center`, `cite`, `code`, `col`, `colgroup`, `dd`, `del`, `dfn`, `dir`, `div`, `dl`, `dt`, `em`, `fieldset`, `font`, `form`, `frame`, `frameset`, `h1`, `h2`, `h3`, `h4`, `h5`, `h6`, `head`, `hr`, `html`, `i`, `iframe`, `img`, `input`, `ins`, `isindex`, `kbd`, `label`, `legend`, `li`, `link`, `map`, `menu`, `meta`, `noframes`, `noscript`, `object`, `ol`, `optgroup`, `option`, `p`, `param`, `pre`, `q`, `s`, `samp`, `script`, `select`, `small`, `span`, `strike`, `strong`, `style`, `sub`, `sup`, `table`, `tbody`, `td`, `textarea`, `tfoot`, `th`, `thead`, `title`, `tr`, `tt`, `u`, `ul`, `var`.

De todas las etiquetas disponibles, las siguientes se consideran **obsoletas** y no se pueden utilizar: `applet`, `basefont`, `center`, `dir`, `font`, `isindex`, `menu`, `s`, `strike`, `u`.

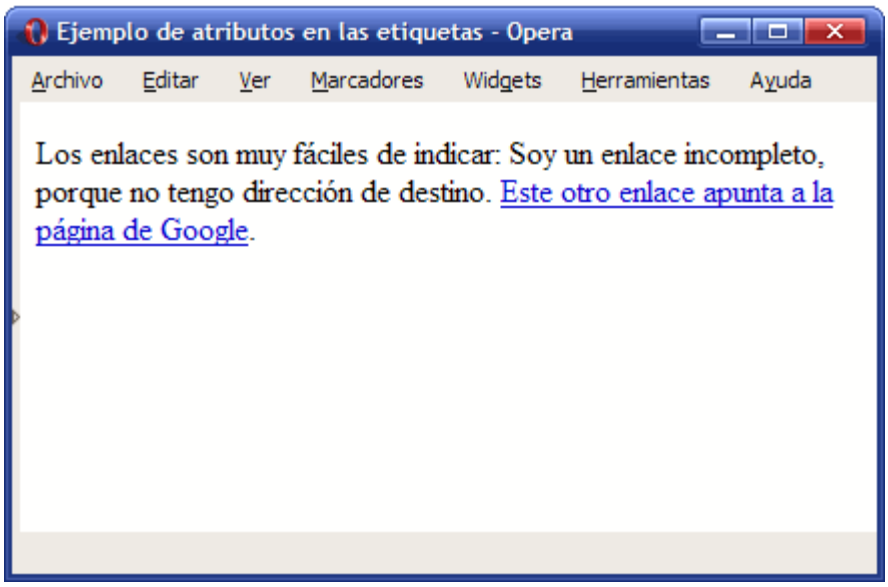
A pesar de que se trata de un número de etiquetas muy grande, no es suficiente para crear páginas complejas. Algunos elementos como las imágenes y los enlaces requieren cierta información adicional para estar completamente definidos.

La etiqueta `<a>` por ejemplo se emplea para incluir un enlace en una página. Utilizando sólo la etiqueta `<a>` no es posible establecer la dirección a la que apunta cada enlace. Como no es viable crear una etiqueta por cada enlace diferente, la

solución consiste en personalizar las etiquetas HTML mediante cierta información adicional llamada **atributos**.

De esta forma, se utiliza la misma etiqueta `<a>` para todos los enlaces de la página y se utilizan los atributos para indicar la dirección a la que apunta cada enlace.

```
<html>

<head>
<title>Ejemplo de atributos en las etiquetas</title>
</head>

<body>
<p>
  Los enlaces son muy fáciles de indicar:
  <a>Soy un enlace incompleto, porque no tengo dirección de destino</a>.
  <a href="http://www.google.com">Este otro enlace apunta a la página de Google</a>.
</p>
</body>

</html>
```



Los atributos permiten personalizar las etiquetas HTML

El primer enlace del ejemplo anterior no está completamente definido, ya que no apunta a ninguna dirección. El segundo enlace, utiliza la misma etiqueta `<a>`, pero añade información adicional mediante un atributo llamado `href`. Los atributos se incluyen dentro de la etiqueta de apertura. Por ahora no es importante comprender la etiqueta `<a>` ni el atributo `href`, ya que se explicarán con todo detalle más adelante.

No todos los atributos se pueden utilizar en todas las etiquetas. Por ello, cada etiqueta define su propia lista de atributos disponibles. Además, cada atributo también indica el tipo de valor que se le puede asignar. Si el valor de un atributo no es válido, el navegador ignora ese atributo.

Aunque cada una de las etiquetas HTML define sus propios atributos, algunos de los atributos son comunes a muchas o casi todas las etiquetas. De esta forma, es

habitual explicar por separado los atributos comunes de las etiquetas para no tener que volver a hacerlo cada vez que se explica una nueva etiqueta. Los atributos comunes se dividen en cuatro grupos según su funcionalidad:

1) **Atributos básicos**: se pueden utilizar prácticamente en todas las etiquetas HTML

Atributo	Descripción
`id = "texto"`	Establece un identificador único a cada elemento dentro de una página HTML
`class = "texto"`	Establece la clase CSS que se aplica a los estilos del elemento
`style = "texto"`	Establece de forma directa los estilos CSS de un elemento
`title = "texto"`	Establece el título a un elemento (mejora la accesibilidad y los navegadores lo muestran cuando el usuario pasa el ratón por encima del elemento)

La mayoría de páginas web actuales utilizan los atributos `id` y `class` de forma masiva. Sin embargo, estos atributos sólo son realmente útiles cuando se trabaja con CSS y con Javascript.

Respecto al valor de los atributos `id` y `class`, sólo pueden contener guiones medios (`-`), guiones bajos (_), letras y/o números, pero no pueden empezar por números. Además, los navegadores distinguen mayúsculas de minúsculas y no se recomienda utilizar letras como `ñ` y acentos, ya que no es seguro que funcionen correctamente en todas las versiones de todos los navegadores.

2) **Atributos para internacionalización**: los utilizan las páginas que muestran sus contenidos en varios idiomas o aquellas que quieren indicar de forma explítica el idioma de sus contenidos:

Atributo	Descripción
`lang = "codigo de idioma"`	Indica el idioma del elemento mediante un código predefinido

`xml:lang = "codigo de idioma"`	Indica el idioma del elemento mediante un código predefinido
`dir`	Indica la dirección del texto (útil para los idiomas que escriben de derecha a izquierda)

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

Como la palabra `internacionalización` es muy larga, se suele sustituir por la abreviatura `i18n` (el número `18` se refiere al número de letras que existen entre la letra `i` y la letra `n` de la palabra `internacionalización`).

3) **Atributos de eventos**: sólo se utilizan en las páginas web dinámicas creadas con JavaScript.

Atributo	**Descripción**
`onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup`	Permiten controlar los eventos producidos sobre cada elemento de la página

Cada vez que el usuario pulsa una tecla, mueve su ratón o pulsa cualquier botón del ratón, se produce un evento dentro del navegador. Utilizando JavaScript y los atributos anteriores, es posible responder de forma adecuada a cada evento.

4) **Atributos para los elementos que pueden obtener el foco**:

Cuando el usuario selecciona un elemento de la interfaz de una aplicación, se dice que *"el elemento tiene el foco del programa"*. Si por ejemplo un usuario pincha con su ratón sobre un cuadro de texto y comienza a escribir, ese cuadro de texto tiene el foco del programa, llamado *"focus"* en inglés. Si el usuario selecciona después otro elemento, el elemento original pierde el foco y el nuevo elemento es el que tiene el foco del programa.

Los elementos de las páginas web también pueden obtener el foco de la aplicación (en este caso, el foco del navegador) y HTML define algunos atributos específicos para controlar cómo se seleccionan los elementos.

Atributo	Descripción
`accesskey = "letra"`	Establece una tecla de acceso rápido a un elemento HTML
`tabindex = "numero"`	Establece la posición del elemento en el orden de tabulación de la página. Su valor debe estar comprendido entre `0` y `32.767`
`onfocus, onblur`	Controlan los eventos JavaScript que se ejecutan cuando el elemento obtiene o pierde el foco

Cuando se pulsa repetidamente la tecla del tabulador sobre una página web, el navegador selecciona de forma alternativa todos los elementos de la página que se pueden seleccionar (principalmente los enlaces y los elementos de formulario). El atributo `tabindex` permite alterar el orden en el que se seleccionan los elementos, por lo que es muy útil cuando se quiere controlar de forma precisa cómo se seleccionan los campos de un formulario complejo.

Por su parte, el atributo `accesskey` permite establecer una tecla para acceder de forma rápida a cualquier elemento. Aunque la tecla de acceso rápido se establece mediante HTML, la combinación de teclas necesarias para activar ese acceso rápido depende del navegador. En el navegador Internet Explorer se pulsa la tecla `ALT` + la tecla definida; en el navegador Firefox se pulsa `Alt` + `Shift` + la tecla definida; en el navegador Opera se pulsa `Shift` + `Esc` + la tecla definida; en el navegador Safari se pulsa `Ctrl` + la tecla definida.

En el resto de la documentación, se emplearán las palabras `"básicos"`, `"i18n"`, `"eventos"` y `"foco"` respectivamente para referirse a cada uno de los grupos de atributos comunes definidos anteriormente.

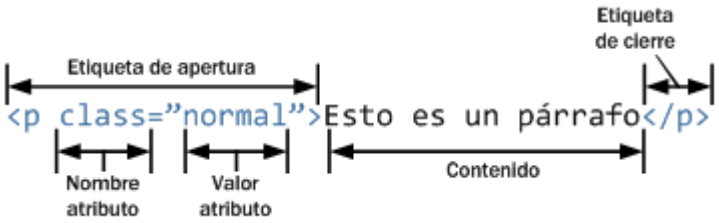
Elementos HTML

Además de etiquetas y atributos, HTML define el término **elemento** para referirse a las partes que componen los documentos HTML.

Aunque en ocasiones se habla de forma indistinta de "elementos" y "etiquetas", en realidad un elemento HTML es mucho más que una etiqueta, ya que está formado por:

- Una etiqueta de apertura.
- Cero o más atributos.
- Texto encerrado por la etiqueta.
- Una etiqueta de cierre.

El texto encerrado por la etiqueta es opcional, ya que algunas etiquetas de HTML no pueden encerrar ningún texto. El siguiente esquema muestra un elemento HTML, formado por una etiqueta `<p>`, atributos y contenidos de texto:



Esquema de las partes que componen un elemento HTML

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>`).

Por tanto, si una página web tiene dos párrafos de texto, la página contiene dos elementos y cuatro etiquetas (dos etiquetas `<p>` de apertura y dos etiquetas `</p>` de cierre). De todas formas, aunque estrictamente no son lo mismo, es habitual intercambiar las palabras "elemento" y "etiqueta".

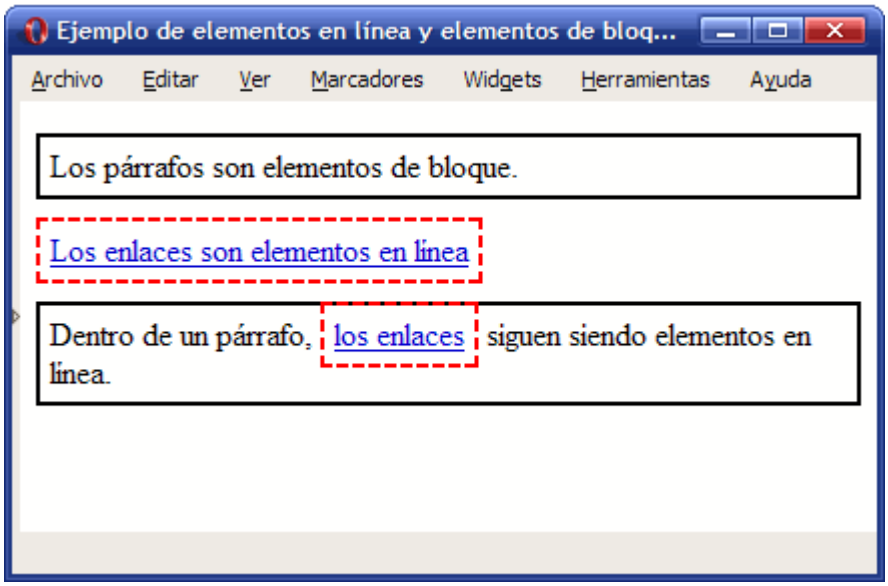
Por otra parte, el lenguaje HTML clasifica a todos los elementos en dos grupos: elementos **en línea** (*inline elements* en inglés) y elementos de **bloque** (*block elements* en inglés).

La principal diferencia entre los dos tipos de elementos es la forma en la que ocupan el espacio disponible en la página. Los elementos de bloque siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea, aunque sus contenidos no lleguen hasta el final de la línea. Por su parte, los elementos en línea sólo ocupan el espacio necesario para mostrar sus contenidos.

Si se considera el siguiente ejemplo:

```
<html>

<head>
  <title>Ejemplo de elementos en línea y elementos de bloque</title>
</head>

<body>
<p>Los párrafos son elementos de bloque.</p>
<a href="http://www.google.com">Los enlaces son elementos en línea</a>
<p>Dentro de un párrafo, <a href="http://www.google.com">los enlaces</a>
siguen siendo elementos en línea.</p>
</body>

</html>
```

La siguiente imagen muestra cómo visualizan los navegadores el código HTML anterior (mediante CSS se han añadido bordes que muestran el espacio ocupado por cada elemento):



Diferencias entre elementos en línea y elementos de bloque

El primer párrafo contiene un texto corto que sólo ocupa la mitad de la anchura de la ventana del navegador. No obstante, el espacio reservado por el navegador para el primer párrafo llega hasta el final de esa línea, por lo que resulta evidente que los elementos `<p>` son elementos de bloque.

Por otra parte, el primer enlace del ejemplo anterior también tiene un texto corto que ocupa solamente la mitad de la anchura de la ventana del navegador. En este caso, el navegador sólo reserva para el enlace el sitio necesario para mostrar sus contenidos. Si se añade otro enlace en esa misma línea, se mostraría a continuación del primer enlace. Por tanto, los elementos `<a>` son elementos en línea.

Por último, el segundo párrafo sigue ocupando todo el espacio disponible hasta el final de cada línea (por ser un elemento de bloque) y el enlace que se encuentra

dentro del párrafo sólo ocupa el sitio necesario para mostrar sus contenidos (por ser un elemento en línea).

La mayoría de elementos de bloque pueden contener en su interior elementos en línea y otros elementos de bloque. Los elementos en línea sólo pueden contener texto u otros elementos en línea. En otras palabras, un elemento de bloque no puede aparecer dentro de un elemento en línea. En cambio, un elemento en línea puede aparecer dentro de un elemento de bloque y dentro de otro elemento en línea.

Los elementos en línea definidos por HTML son: `a`, `abbr`, `acronym`, `b`, `basefont`, `bdo`, `big`, `br`, `cite`, `code`, `dfn`, `em`, `font`, `i`, `img`, `input`, `kbd`, `label`, `q`, `s`, `samp`, `select`, `small`, `span`, `strike`, `strong`, `sub`, `sup`, `textarea`, `tt`, `u`, `var`.

Los elementos de bloque definidos por HTML son: `address`, `blockquote`, `center`, `dir`, `div`, `dl`, `fieldset`, `form`, `h1`, `h2`, `h3`, `h4`, `h5`, `h6`, `hr`, `isindex`, `menu`, `noframes`, `nos-cript`, `ol`, `p`, `pre`, `table`, `ul`.

Los siguientes elementos también se considera que son de bloque: `dd`, `dt`, `frame-set`, `li`, `tbody`, `td`, `tfoot`, `th`, `thead`, `tr`.

Los siguientes elementos pueden ser en línea y de bloque según las circunstancias: `button`, `del`, `iframe`, `ins`, `map`, `object`, `script`.