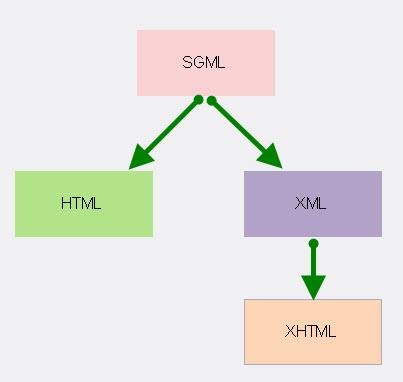
UNIDAD 2. MANEJO BÁSICO DE HTML

# MANEJO BÁSICO DE HTML

En este tema vamos a trabajar en los aspectos fundamentales de la construcción de un documento HTML5, que no es otra cosa que un archivo de texto, para lo cual debemos conocer las peculiaridades de este lenguaje de marcas como son su estructura, elementos etiquetas, etc. que tienen bastante en común con las anteriores versiones de HTML por lo que esta primera aproximación al diseño web nos va a llevar a familiarizarnos con los compontes elementales de cualquier versión HTML y sus reglas para la construcción de páginas web. Veremos la estructura de un documento HTML, lo que son los elementos HTML alrededor de los cuales gira toda creación HTML, y su componente más importante: las etiquetas o marcas y, por último, crearemos nuestras primeras páginas web.

# RELACIÓN DE HTML5 CON HTML4 y XML

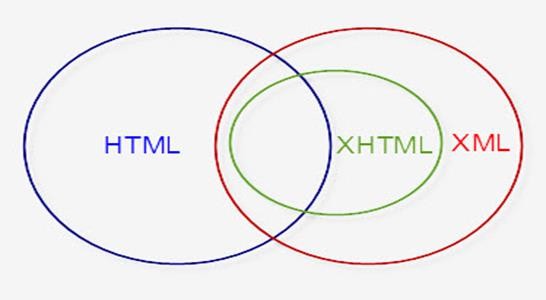
Tanto HTML5 como HTML4, XML y XHTML son lenguajes de marcas lo que implica el uso de etiquetas para darle un determinado aspecto a un texto o ejecutar una determinada acción. Hay que recordar que estos cuatro lenguajes de marcas tienen bastantes estructuras comunes lo que permite que, si se conoce cualquiera de ellos, se puede aprender la sintaxis de los demás sin gran esfuerzo, aunque difieran en el uso de unas normas de escritura más o menos estrictas.

El precedente de estos lenguajes de marcas es el estándar internacional SGML (Standard Generalized Markup Language) que define la estructura de diferentes tipos de documentos basándose en la relación lógica de sus partes.

XML y HTML son lenguajes muy diferentes. Su sintaxis es similar, aunque [cada](http://www.maestrosdelweb.com/editorial/xmlvshtml/) uno fue diseñado para cumplir distintas funciones.

El XML (eXtensive Markup Language) es un lenguaje que fue concebido para describir información. Su función principal es ayudarnos a organizar contenidos y eso hace que los documentos XML sean

portables [hacia](http://www.maestrosdelweb.com/editorial/xmlvshtml/) diferentes tipos de aplicaciones y que los navegadores los interpreten mejor.

Un dato importante es que al hablar de XML hablamos de documentos bien formados (well formed), es decir, aquellos que deben cumplir estrictamente tanto las normas sintácticas como semánticas, mientras que en HTML podemos ser desordenados y no tiene la rigidez de XML

HTML (HyperText Markup Language), por otro lado, ha sido concebido [para](http://www.maestrosdelweb.com/editorial/xmlvshtml/) mostrar información, determinar cómo actúa y qué hace. Su función radica en ayudarnos a darle formato a los diversos contenidos de una página.

XHTML es muy parecido a HTML ya que es una adaptación de

HTML al lenguaje XML lo que significa que XHTML tiene las mismas funcionalidades de HTML pero cumple las especificaciones más estrictas de XML. Su aparición intentaba paliar un gran problema del lenguaje HTML y es que los navegadores representaban de distinta forma la misma página html. XHTML, al ser más estricto, permite a los diferentes navegadores representar prácticamente de la misma manera una misma página html.

XHTML se considera descendiente más de XML que de HTML.

Resumiendo, el XML sirve para describir información y el HTML sirve para darle formato y presentarla a través de un navegador. O sea que el XML no ha sido nunca un reemplazo de HTML sino un complemento que sirve para manejar la información separada del formato.

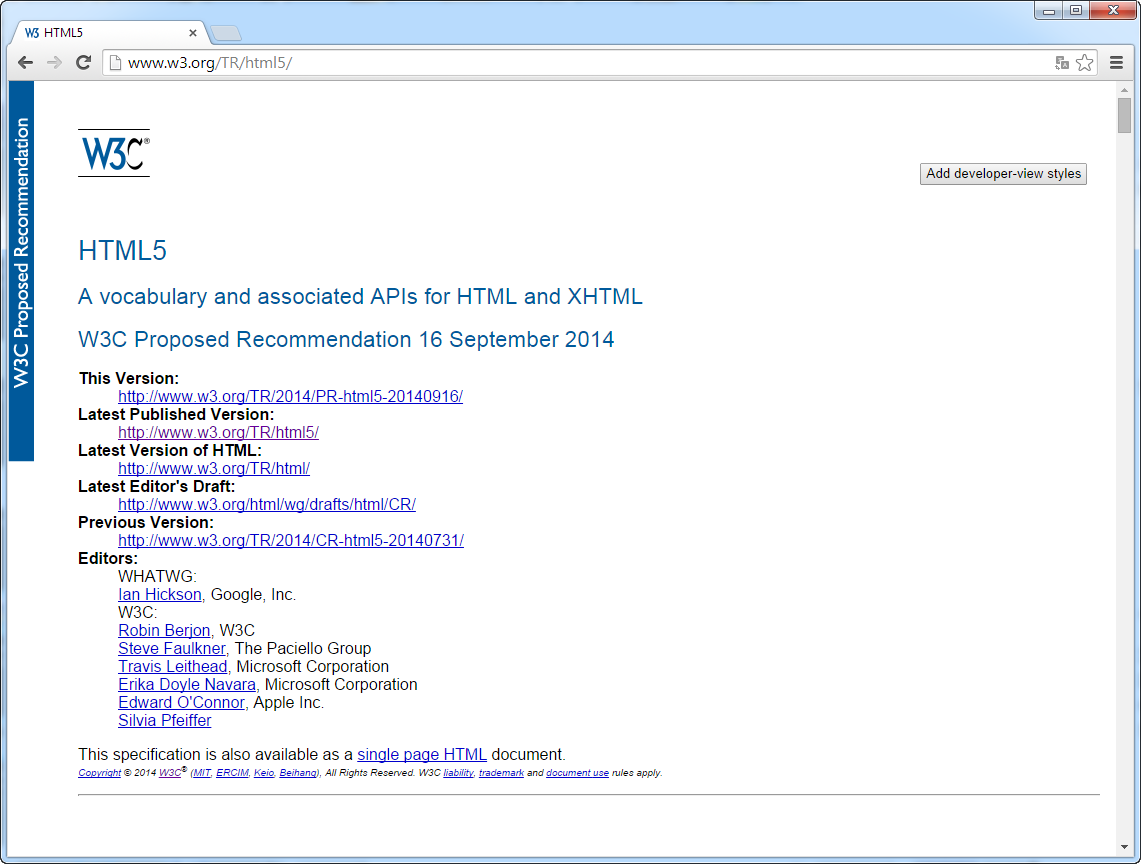
Para finalizar HTML5 es la última versión de HTML cuya sintaxis es compatible con HTML4 y con XHTML. Este incorpora elementos innovadores, emplea hojas de estilo en cascada (CSS) para los efectos estéticos de aspecto, no reconoce algunos de los elementos de HTML4 e implementa nuevas funcionalidades a través de APIs Javascript.

#### Especificaciones oficiales

El organismo [W3C](http://www.w3.org/) (*World Wide Web Consortium*) elabora las normas que deben seguir los usuarios de cualquiera de esos lenguajes anteriores para que no haya problemas a la hora de reconocer el tipo de documento que se ha creado.

Todos los lenguajes de marcas usados en Internet tienen unas especificaciones obligatorias de cumplir que están marcadas en documentos publicados por ese organismo.

En el caso de HTML5 podemos encontrar esa especificación en la dirección: <http://www.w3.org/TR/html5/>

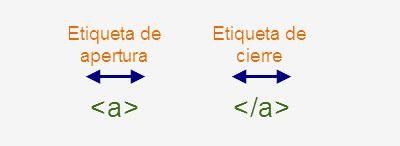


# ETIQUETAS

#### Definición

Las etiquetas son textos incluidos entre los símbolos *menor que* “<” y *mayor que* “>” que marcan los diferentes elementos que integran una página html y les aplican características concretas a estos. Aunque hay versiones de html que permiten usar tanto mayúsculas como minúsculas para escribir las etiquetas, es muy conveniente hacerlo solo en minúsculas.

Existe normalmente una etiqueta de inicio y otra de fin (hay excepciones en las que solo se necesita la etiqueta de inicio). En HTML la etiqueta de fin es la misma que la de inicio añadiéndole al principio una barra inclinada “/” y afecta a todo lo que haya entre ambas ya sea texto u otras etiquetas.



En esta imagen se usa como ejemplo la etiqueta <a> que sirve para insertar un enlace o hipervínculo en una página html.

*En todo lenguaje de marcas existe un número concreto de etiquetas con una sintaxis y semántica predefinida que desempeñan una función preestablecida de antemano*



*Una etiqueta en HTML se puede escribir en mayúsculas o minúsculas indistintamente, pero se recomienda escribirlas en minúsculas.*



HTML5 incorpora nuevas etiquetas y no soporta algunas de HTML 4.01

* Listado de etiquetas HTML5 (guías de referencia).
* Listado de etiquetas HTML 4.01 no soportadas en HTML5 (guías de referencia).

#### Atributos

Los atributos son aspectos que se pueden modificar del funcionamiento de una etiqueta y que se insertan en cualquier orden dentro de la etiqueta de inicio poniendo el nombre del atributo y su valor entre comillas (“ “) separados por un signo *igual* (=)



En la imagen **href** es el atributo particular que modifica la etiqueta **<a>** cuyo valor es [**http://www.mentor.mec.es**](http://www.mentor.mec.es/)y que indica a donde apunta el hipervínculo, por lo que, la etiqueta completa es la referencia a la página principal del proyecto Mentor.

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](http://librosweb.es/xhtml/capitulo_2/etiquetas_y_atributos.html) de valor que se le puede asignar.

Si el valor de un atributo no es válido, el navegador lo ignora



Aunque cada una de las etiquetas HTML define sus propios atributos, algunos de ellos son comunes a muchas o casi todas las etiquetas siendo su funcionalidad la misma independientemente de la etiqueta en la que se aplica. Los atributos comunes se dividen en tres grupos según su funcionalidad:

- **Atributos globales** se pueden utilizar prácticamente en todas las etiquetas HTML. Los más comunes son los siguientes:

|  |  |
| --- | --- |
| **Atributo** | **Descripción** |
| class | Especifica uno o más nombres de clase para un elemento (hojas de estilo CSS) |
| id | Identificativo del elemento. |
| style | Estilo que se aplica al elemento |
| title | Información extra sobre el elemento. |

- **Atributos particulares**: Son aquellos que solo afectan a una etiqueta en concreto.

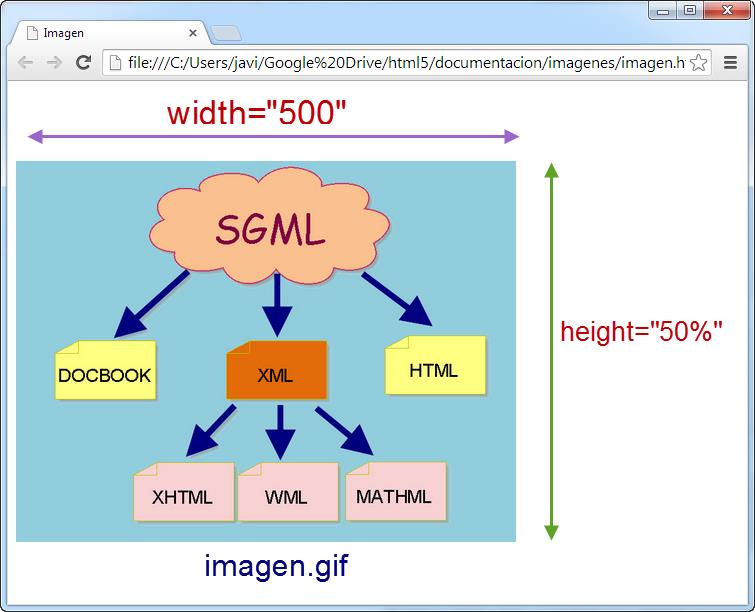
- **Atributos de eventos**: sólo se utilizan en las páginas web dinámicas creadas con JavaScript.

#### Ejemplo:

**<img src=”imagen.gif” width=”500” height=”50%”>**

La etiqueta img permite la inserción de una imagen en el archivo html. En este caso el atributo src indica que archivo se va a visualizar, el atributo width indica que la imagen tendrá 500 pixels de anchura y height indica que ocupa el 50% del alto de la pantalla.

En la siguiente imagen se ve cómo quedaría el archivo imagen.gif dentro de una página web que lo visualiza.



#### Elementos HTML

Los elementos HTML son los auténticos componentes de una página web ya que marcan cada una de las partes de los documentos HTML.

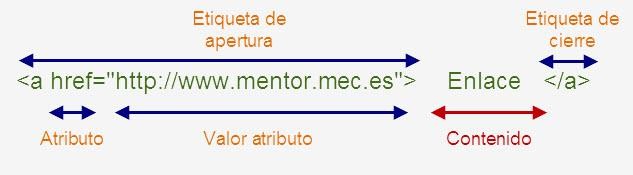
*La mayoría de los elementos HTML se pueden anidar, es decir, un elemento HTML puede contener a otro u otros*



Muchas veces se confunde el término “elemento” con el de “etiqueta” ya que se piensa que son dos cosas iguales pero un elemento html es más que una etiqueta ya que está formado normalmente por:

Un elemento HTML no es una etiqueta

* Una etiqueta de apertura.
* Cero o más atributos en la etiqueta de apertura.
* Texto u otros elementos que son formateados con arreglo a la función de la etiqueta.
* Una etiqueta de cierre.



En el ejemplo anterior el elemento html está formado de la etiqueta **<a>** (indica que es un hipervínculo o enlace), el atributo **href** (indica la dirección hacia donde nos dirigirá el enlace) cuyo valor es [**http://www.mentor.mec.es**](http://www.mentor.mec.es/)(dirección del enlace), el contenido cuyo valor es **Enlace** (texto al cual se aplica la etiqueta html) y una etiqueta de cierre **</a>** que es igual a la de apertura, pero con la barra **/.**

Algunos elementos no llevan etiqueta de cierre y son denominados elementos vacíos ya que no tienen contenido (un ejemplo es <br> que devuelve un retorno de carro).

Un elemento vacío se puede cerrar dentro de la propia etiqueta de apertura. Ejemplo <br />



*Un elemento HTML puede ser reconocido por un navegador, pero puede que alguno de sus atributos no lo sea*

Por otra parte, el lenguaje HTML clasifica a todos los elementos en tres grupos dependiendo de cómo ocupan el espacio en el documento. Estos grupos son de tipo bloque (block), de tipo en línea (inline) y los que no son ni en línea ni en bloque.

* + **Elementos de tipo bloque (block):** Los elementos del tipo bloque tienen las siguientes características:
    - Forman un bloque y se posiciona dentro del documento html de forma vertical finalizado con un retorno de carro que lo separa de los demás elementos.
    - Ocupan toda la anchura que le proporciona el elemento en el cual está contenido, aunque, usando hojas de estilos CSS, se puede aplicar una anchura fija.
    - Su altura va en función del contenido del elemento, aunque, al igual que pasa con la anchura, se puede utilizar hojas de estilos CSS para aplicar una altura fija.
    - Puede contener otros elementos de tipo inline y/o block

Ejemplos de elementos de tipo bloque:<p>, <h1>, <h2>, <h3>, <h4>, <h5>,

<h6>, <div>, <ul>, <ol>, <li>, <menú>, <dir>, <pre>, <hr>, <blockquote>,

<address>, <center>, <noframes>, <isindex>, <fieldset>, <table>, <form>

* + **Elementos en línea (inline):** Los elementos de tipo en línea tienen las siguientes características
    - Se ajustan horizontalmente con los elementos que estén junto a él.
    - La anchura se define en base al contenido.
    - En el caso de permita contener otros elementos, estos deben ser de tipo inline, es decir, no pueden contener elementos de tipo block.

Ejemplos de elementos de tipo en línea: <a>, <br>, <span>, <bdo>, <object>, <applet>,

<img>, <map>, <iframe>, <tt>, <i>, <b>, <big>, <small>, <u>, <s>, <strike>, <Font>,

<basefont>, <em>, <strong>, <dfn>, <code>, <q>, <sub>, <sup>, <samp>, <kbd>, <var>,

<cite >, <input>, <select>, <textarea>, <label>, <button>

* + **Elementos que no son ni en línea ni bloque:** Son los elementos que no forman parte del flujo de la información porque son elementos subordinados a otros elementos***.***

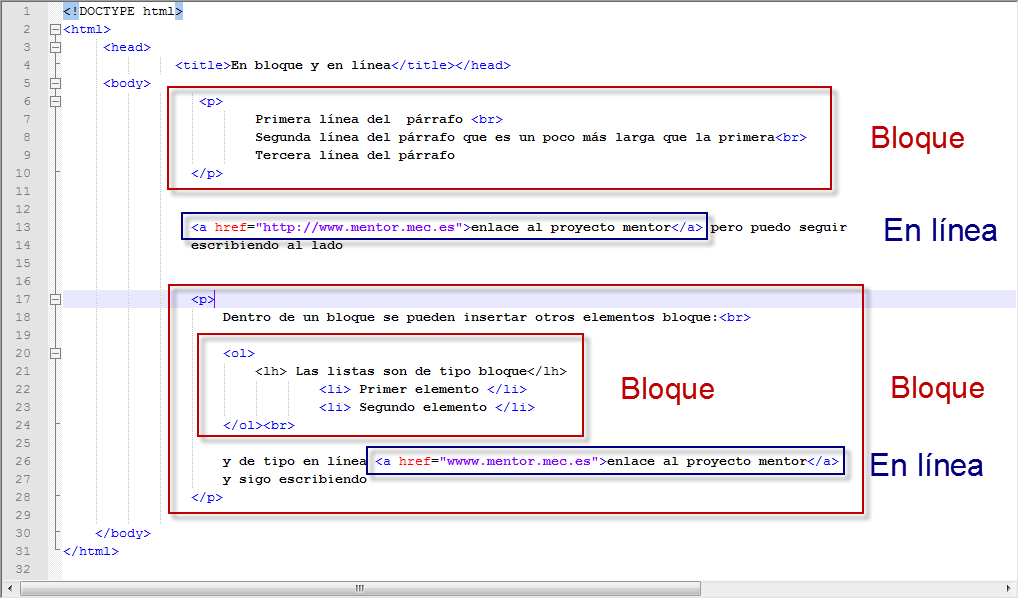
Ejemplos de Elementos subordinados son

* + - Subordinados a table: [caption](https://developer.mozilla.org/es/docs/HTML/Elemento/caption) [tr](https://developer.mozilla.org/es/docs/HTML/Elemento/tr) [th](https://developer.mozilla.org/es/docs/HTML/Elemento/th) [td](https://developer.mozilla.org/es/docs/HTML/Elemento/td) [thead](https://developer.mozilla.org/es/docs/HTML/Elemento/thead) [tbody](https://developer.mozilla.org/es/docs/HTML/Elemento/tbody) [tfoot](https://developer.mozilla.org/es/docs/HTML/Elemento/tfoot) [col](https://developer.mozilla.org/es/docs/HTML/Elemento/col) [colgroup](https://developer.mozilla.org/es/docs/HTML/Elemento/colgroup)
    - Subordinados de listas: [li](https://developer.mozilla.org/es/docs/HTML/Elemento/li) [dd](https://developer.mozilla.org/es/docs/HTML/Elemento/dd) [dl](https://developer.mozilla.org/es/docs/HTML/Elemento/dl)
    - Subordinados a select son: [optgroup](https://developer.mozilla.org/es/docs/HTML/Elemento/optgroup) [option](https://developer.mozilla.org/es/docs/HTML/Elemento/option)
    - subordinado a map: área
    - subordinado a object: param
    - subordinado a fieldset: legend

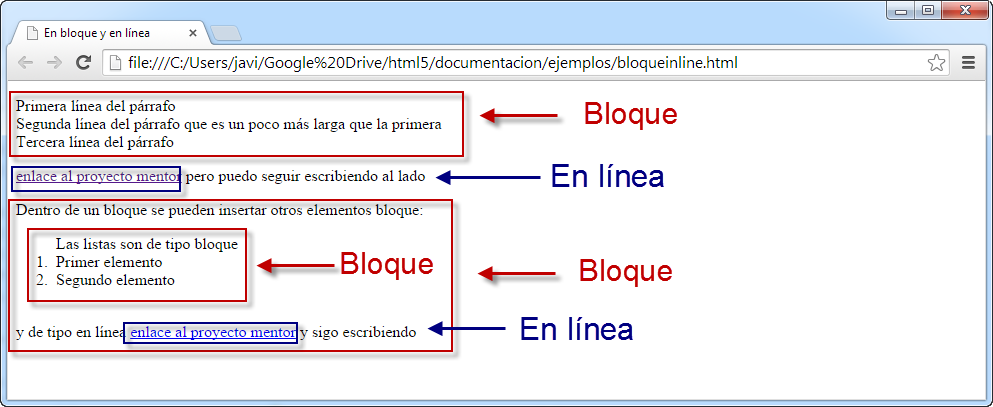
*Por medio de CSS se puede cambiar la forma de ver un elemento mediante las reglas* ***display:block*** *(visualizar el elemento en forma bloque) y* ***display:inline*** *(visualizar el elemento en forma en línea)*



En el siguiente ejemplo se crean dos elementos de tipo bloque (párrafos <p>) y elementos en línea (enlaces <a>) y como se pueden anidar unos dentro de otros.



Visualización del ejemplo anterior en el navegador Chrome



# ESTRUCTURA DE UN DOCUMENTO HTML

La estructura básica de una página web es la siguiente:

<html>

<head>

<title> Título de la página</title> Elementos de cabecera

.

.

</head>

<body>

Elementos y contenido de la página

.

.

</body>

</html>



#### Elemento <html>

La cabecera y el contenido de una página web está dentro del elemento

<html> que comienza con la etiqueta <html> y finaliza con </html>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <html> | | | | | | |
| Descripción | Representa la raíz de un documento html | | | | | | |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |

|  |  |
| --- | --- |
| Atributos | Globales |
| Atributos propios | manifest, xmlns |
| Diferencia entre html  4.01 y html5 | El atributo manifest se ha añadido en HTML5 |

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| manifest | URL | Localización del archivo que sirve de caché o memoria temporal de la página web |
| xmlns | <http://www.w3.org/1999/xhtml/> | Localización del espacio de nombres XML. Opcional en html5 |

  Nuevo en html5

Su principal atributo global es lang que indica el idioma del contenido de los elementos del documento HTML.

En los siguientes ejemplos se indica que el lenguaje del contenido por defecto de los elementos de la página web está escrito en español (España).

<html lang=”es”>

También se puede poner

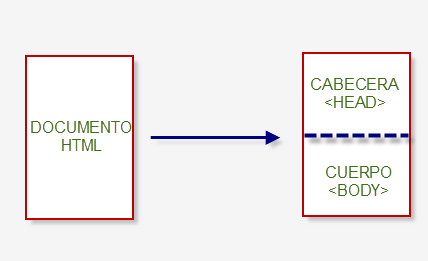
<html lang=”es-es”>

<html> contiene otros dos elementos que son:

* <head> </head>: Es la cabecera de la página que contiene definiciones generales que afectan a todo el documento. Esta información no forma parte del contenido de la página.

Todos sus elementos son opcionales y se utilizarán en casos muy determinados. El elemento <title>, título de la página, no es estrictamente obligatorio, pero siempre debe ponerse ya que permite poner un título en la parte superior de la ventana del navegador que de indicios del contenido de la página.

* <body> </body>: El elemento <body> define el cuerpo de un documento y contiene todos los elementos que se van a visualizar de la página web como textos, enlaces, imágenes, etc. En definitiva, <body> es el contenido del documento HTML.



La declaración HTML <!DOCTYPE>

Aunque la estructura anterior es perfectamente válida para cualquier versión de HTML o XML hay que reseñar que es muy importante indicarle al navegador el tipo de documento que va a utilizar (si es HTML4, HTML5, etc.) a través de la declaración <!DOCTYPE>. Esta declaración siempre va antes que la etiqueta de inicio de documento <html>.

En html4, xml y xhtml dentro de la DOCTYPE va reflejada la dirección documento DTD (Definición del tipo de documento) que recoge las normas y restricciones con las que se ha de visualizar el documento HTML. Esto, en cambio, se simplifica mucho en HTML5 ya que no tiene DTD asociado.

Navegadores que soportan esta declaración

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |

Declaraciones DOCTYPE según el tipo de documento

* HTML 4.01 Strict

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"

["http](http://www.w3.org/TR/html4/strict.dtd):[//www.w3.org/TR/html4/strict.dtd"](http://www.w3.org/TR/html4/strict.dtd)>

* HTML 4.0.1 Transitional

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

* HTML 4.01 Frameset

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"

["http](http://www.w3.org/TR/html4/frameset.dtd):[//www.w3.org/TR/html4/frameset.dtd"](http://www.w3.org/TR/html4/frameset.dtd)>

* XHTML 1.0 Strict

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

* XHTML 1.0 Transitional

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

* XHTML 1.0 Frameset

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

* XHTML 1.1

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"

["http](http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd):[//www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">](http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd)

* HTML5

<!DOCTYPE html>

Según la declaración anterior, la estructura básica de una página HTML5 en español es:

<!DOCTYPE html>

<html lang=”es”>

<head>

<title> Título de la página</title> Elementos de cabecera

.

.

</head>

<body>

Elementos y contenido de la página

.

.

</body>

</html>

Ejercicio nº.1: Una página html5 es un archivo texto como extensión .htm ó .html con la estructura que hemos visto anteriormente y compuesto de etiquetas y texto. Para crear una página web básica, solo nos hace falta un simple editor de textos y escribir tanto etiquetas como texto. Vamos a crear una página web con notepad++ llamada primera.html que visualice por pantalla el mensaje “Mi primer documento html” y tenga como título “Mi primer documento html”. Después de creada, ábrela con tu navegador favorito y observa el resultado

#### Cabecera de un documento html. <head>

Las páginas y documentos html incluyen más información de la que los usuarios ven en sus pantallas. Estos datos adicionales siempre están relacionados con la propia página, por lo que se denominan *metainformación* o *metadatos*. La metainformación siempre se incluye en la sección de la cabecera, es decir, dentro del elemento <head>

Aunque la metainformación más conocida y utilizada es el título de la propia página, se puede incluir mucha otra información útil para los navegadores y para los buscadores.

En <head> no se puede incluir contenido de la página web ya que no se visualiza con el navegador



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <head> | | | | | | |
| Descripción | Define la cabecera de un archivo html | | | | | | |
| Elemento | No es bloque ni inline | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales | | | | | | |
| Atributos propios |  | | | | | | |
| Diferencia  entre html | El atributo profile no lo soporta HTML5 | | | | | | |

|  |  |
| --- | --- |
| 4.01 y html5 |  |

Los elementos más importantes de <HEAD> son:

|  |  |
| --- | --- |
| **Elementos <head>** | **Descripción** |
| [<title>](http://www.w3schools.com/tags/tag_title.asp) | Define el título de un documento |
| [<base>](http://www.w3schools.com/tags/tag_base.asp) | Define una dirección por defecto o una página por defecto de todos los enlaces de una página |
| [<link>](http://www.w3schools.com/tags/tag_link.asp) | Define una relación entre un documento y un recurso externo |
| [<meta>](http://www.w3schools.com/tags/tag_meta.asp) | Información adicional de un documento html. |
| [<script>](http://www.w3schools.com/tags/tag_script.asp) | Define un script javascript local |
| [<style>](http://www.w3schools.com/tags/tag_style.asp) | Define los estilos de un documento html |



* **<title>** Especificará el título del documento html. Dentro de esta etiqueta no se podrá usar ninguna de las restantes etiquetas html.

*El título no forma parte del contenido del documento HTML y es obligatorio incluirlo.*

*Todos estos elementos soportan los atributos globales HTML.*



#### Ejemplo:

<head>

<title>Titulo de la página web</title>

</head>

El elemento <title>:

* + Se muestra en la barra de herramientas (toolbar) del navegador.
  + Proporciona el título de la página cuando se añade a los favoritos del navegador.
  + Muestra el título de la página en los resultados de los motores de búsqueda.
* **<base>** Especifica la dirección o marco destino para todas las direcciones relativas que se encuentren en la página.

La razón más importante para usar <base> es asegurar que cualquier URL relativa dentro del documento se resolverá en una dirección correcta, aun si el documento se mueve o renombra.

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| href | URL | Especifica la dirección base para todas las URL relativas del documento. |
| target | \_blank  \_parent | Especifica el lugar por defecto donde se visualizará un archivo o página enlazada. |
|  | \_self |  |
|  | \_top |  |
|  | Nombre marco |  |

#### Ejemplo:

<head>

<base href=”<http://www.prueba.es>”>

</head>

* **<link>** Define una relación entre un documento y otro. Su uso más importante es para la referencia a archivos de estilos.

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| href | URL | Especifica la localización del archivo al que se hace referencia |
| hreflang | Código idioma | Especifica el idioma del documento |

|  |  |  |
| --- | --- | --- |
|  |  | enlazado |
| media | Dispositivo | Especifica el dispositivo para el cual está optimizado el documento |
| rel | alternate archives author bookmark external first  help icon last license next  nofollow noreferrer pingback prefetch prev search sidebar stylesheet tag  up | Especifica la relación entre el propio documento y el enlazado. |
| sizes | Anchoxlargo | Especifica el tamaño en pixels de un icono. Solo se puede usar con rel=”icon”. Por ahora no lo soporta ningún navegador. |
| type | Tipo MIME | Especifica el tipo MIME del documento al que se hace referencia |

  Nuevo en HTML5

#### Ejemplo:

<head>

<link href=”estilos.css” rel=”stylesheet” type=”text/css”>

</head>

En este ejemplo se hace referencia al archivo estilos.css que una hoja estilos (rel=”stylesheet”) y que se reconoce con el tipo MIME “text/css”.

* **<meta>** Da información adicional sobre la página que puede ser empleada por los navegadores y por los buscadores.

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| charset | Mapa de caracteres | Especifica el mapa de caracteres de un documento html |
| content | Texto | Especifica el valor que toma el atributo *name* o *http-equiv* |
| http-equiv | content-type  default-style refresh | Especifica una cabecera HTTPde información o valor del atributo *content* |
| name | application-name author description generator keywords | Especifica el autor, nombre de aplicación, descripción, con qué paquete ha sido generado y palabras clave de un documento html. |

  Nuevo en HTML5

#### Ejemplo:

<head>

<meta name=”author” content=”María del Mar”>

<meta name=”description” content=”Página de astronomía”>

<meta name=”keywords” content=”estrellas, cielo, constelaciones, Luna”>

</head>

En este se indica que la autora de esta página es María del mar, que trata de astronomía y que los buscadores de información de Internet van a asociar dicha página con los términos “estrellas”, “cielo”, “constelaciones” y “Luna” siempre y cuando esta página esté alojada en un servidor web de Internet

*charset en HTML5 reemplaza a http-equiv a la hora definir el mapa de caracteres de una página web.*



#### Ejemplo HTML5:

*UTF-8 es el conjunto de caracteres más empleado en Internet*

<head>

<meta charset=”utf-8”>

</head>

**Ejemplo HTML 4..01:**

<head>

<meta http-equiv="content-type" content="text/html; charset=UTF-8">

</head>

* **<script>**Es usado para incluir un script o programa local javascript en el documento html.

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| async | async | Especifica que el script se ejecutará de forma asíncrona (solo scripts externos) |
| charset | Mapa de caracteres | Especifica el mapa de caracteres usado en un script externo |
| defer | defer | Especifica que el script se ejecuta al final del análisis del archivo. No soportado en el navegador Opera 12 ni anteriores versiones. |
| src | URL | Especifica la localización del script al que se hace referencia- |
| type | Tipo MIME | Especifica el tipo MIME del script al que se hace referencia |

  Nuevo en HTML5

#### Ejemplo:

<head>

<script>

document.getElementById("elemento").innerHTML = "Hola”;

</script>

</head>

En este ejemplo se escribe “Hola” dentro del elemento html con id=”elemento”

* **<noscript>**: Cuando se intenta ejecutar un script define un contenido alternativo para los navegadores que tengan deshabilitadas esta opción o no la soporten.

*<noscript> en HTML5 puede ir tanto en <body> como en <head> a diferencia de HTML4 que solo puede ir en <body>*



#### Ejemplo:

<script>

document.getElementById("elemento").innerHTML = "Hola”;

</script>

<noscript> Tu navegador no soporta JavaScript</noscript>

#### HTML5 y los caracteres especiales castellanos (ñ. á, ü, etc.).

Los caracteres o, más genéricamente, símbolos que nosotros escribimos en un documento no se pueden representar como tales dentro de las aplicaciones por lo que necesitamos algún tipo de conversión que permita transformar esos caracteres en símbolos que las aplicaciones puedan entender y manejar. Esta equivalencia se refleja en lo que denominamos **tabla de caracteres**

Para visualizar correctamente una página web, el navegador debe conocer la tabla de caracteres en la que está codificada dicha página y así realizar la conversión correcta.

En las primeras versiones de HTML se usa el código ASCII como tabla de caracteres por defecto, es decir, que si no se especificaba lo contrario el documento se interpretaba que estaba codificado en ASCII.

Más adelante Microsoft incorporó el código ANSI (Windows-1252) a sus sistemas operativos (en realidad ANSI es una extensión de ASCII que pretendía ampliarlo) y muchas aplicaciones lo tomaron como tabla de caracteres por defecto.

Con la aparición de HTML4.01, las aplicaciones utilizaban como tabla de caracteres por defecto ISO-8859-1 que también era una extensión de ASCII con muchas similitudes con ANSI aunque no tan amplia (muchos navegadores en sus versiones más antiguas interpretaban las páginas escritas en ISO-8859-1 como ANSI debido a esa similitud).

Pero tanto las anteriores codificaciones no llegaban a representar todos los caracteres internacionales que se pueden utilizar por lo que surgió el código UNICODE con la finalidad de paliar la carencia anterior. De aquí surgió la tabla de caracteres UTF-8 que es la que por defecto se utiliza en aplicaciones para HTML5.

Después de esta introducción debemos saber que la codificación de las páginas web (charset) depende de si:

* El editor en que se ha escrito el documento HTML usa por defecto UTF-8 o ISO- 8859-1. Si el archivo original estaba escrito en ISO-8859-1 y se edita en UTF-8, los caracteres específicos castellanos se verán mal codificados
* La configuración del servidor web en el caso de que esté alojada en su estructura la página web.
* Si se especifica el conjunto de caracteres en la etiqueta META, atributo charset, de la sección HEAD de la página web.

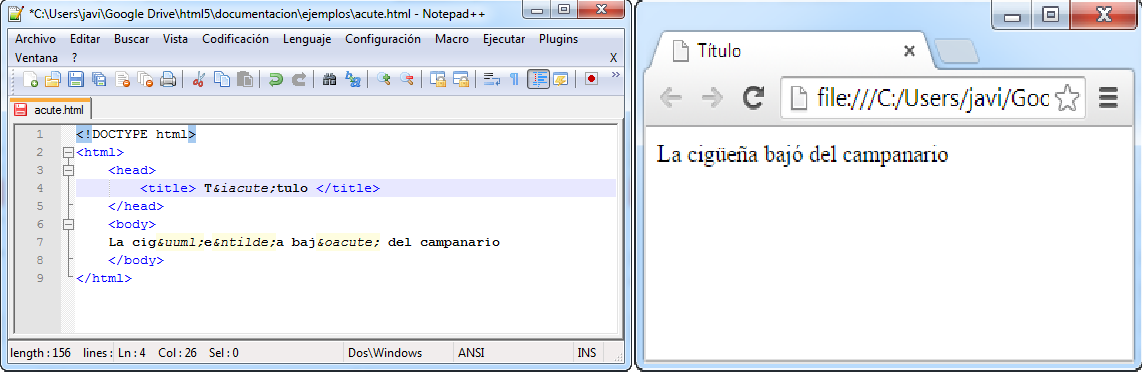
No obstante, y a modo de ejemplo, el estándar HTML representa de la siguiente manera algunos de los caracteres especiales del lenguaje castellano

:

|  |  |  |  |
| --- | --- | --- | --- |
| **Caráter** | Representación HTML | **Caráter** | Representación HTML |
| Á | &Aacute; | á | &aacute; |
| É | &Eacute; | é | &eacute; |
| Í | &Iacute; | í | &iacute; |
| Ó | &Oacute; | ó | &oacute; |
| Ú | &Uacute; | ú | &uacute; |
| À | &Agrave; | à | &agrave; |
| È | &Egrave; | è | &egrave; |
| Ö | &Ouml; | ö | &ouml; |
| Ü | &Uuml; | ü | &uuml; |
| Ñ | &Ntilde; | Ñ | &ntilde; |
| < | &lt; | > | &gt; |
| & | &amp; | “ | &quot; |

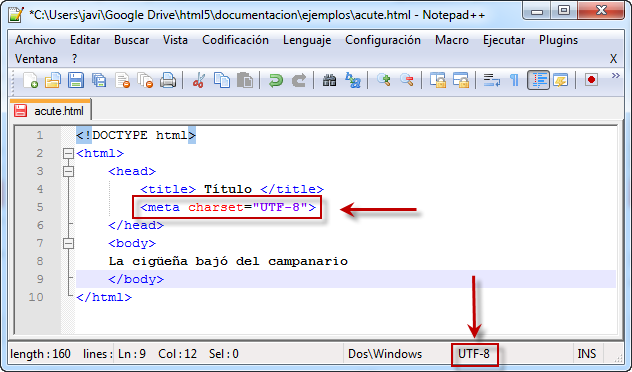
Utilizando las representaciones anteriores nunca habrá ningún problema a la hora de visualizar esos caracteres en un navegador independientemente de la codificación que se haya usado para hacer el documento.

Ejemplo:



No obstante, el mapa de caracteres UTF-8 evita los problemas a la hora de representar los caracteres especiales castellanos, aunque puede haberlos con algún carácter de lenguajes orientales.

En resumen, lo más aconsejable es crear nuestra página web con un editor que la codifique en UTF-8 y agregar la etiqueta <meta charset=”UTF-8”> a la sección <head> de dicha página.



#### Cuerpo de un documento html. <body>

El elemento <BODY> contiene tanto el texto como otros elementos o etiquetas que hacen que la información de una página web sea visible con un determinado aspecto a través de un navegador.

|  |  |
| --- | --- |
| Etiqueta | <body> |
| Descripción | Define el cuerpo de un documento html |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios |  | | | | | | |
| Diferencia entre html  4.01 y html5 | En html5 se suprimen todos los atributos que modifican el aspecto general de un documento HTML tales como background, bgcolor, etc. | | | | | | |

Tanto en html4 como en xhtml y otros lenguajes de marcas más antiguos el elemento

<BODY> permitía, a través de sus atributos, dar aspectos generales tanto al entorno como al contenido de un documento html como, por ejemplo, dar un color de fondo a la página (<BODY bgcolor=”#red”>).

En html5 se usan hojas de estilo CSS para aplicar cualquier modificación de aspecto de

<BODY> y los atributos que tenía esta etiqueta en html4 o xhtml desaparecen como tales.

**Ejercicio nº. 2:** Seguimos utilizando notepad++ para crear una página html que respete la estructura básica de una página html5 y contenga lo siguiente con el fin de practicar con las partes <head> y <body>:

* Título: Página de lengua castellana
* Autor: El creador de la misma
* Descripción: Una descripción adecuada a una página de castellano
* Palabras clave: Palabras relacionadas con la lengua castellana
* Código de caracteres: El que no nos de problemas a la hora de poner ñ o acentos.
* Como cuerpo de la página, o sea, contenido de la misma, tendrá solamente este texto:

Probando códigos \ ñ í Ú @ & á Ñ Ç ç

Guárdala como segunda.html y visualízala con Chrome y algún otro más navegador para ver sus diferencias y si los caracteres se reproducen fidedignamente.

# VALIDACIÓN DE UNA PÁGINA WEB

El proceso de validación es aquel que indica si un documento escrito en un determinado lenguaje cumple las especificaciones del mismo, es decir, si respeta sus normas.

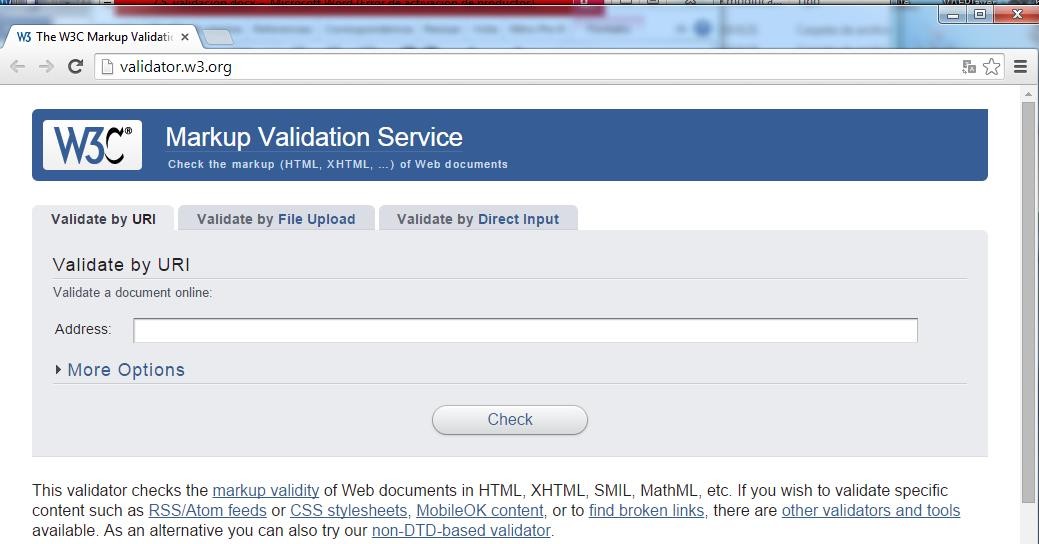
En el ámbito del diseño web la validación de una página no es obligatoria ya que un documento html puede ser reproducido correctamente por un navegador, aunque no cumpla las normas del lenguaje en el que está escrito, aunque siempre es mejor que lo haga.

En versiones anteriores de HTML a HTML5 y en XML y sus derivados (XTHTML) estas especificaciones vienen refljejadas en el DTD o Document Type Definition (Definición del Tipo de Documento). HTML5 no tiene DTD al no basarse en SGML.

*El organismo W3C se encarga de mantener los DTD tanto de los lenguajes HTML como XML*



El organismo W3C ofrece una herramienta pública a través de la cual se puede saber si una página web cumple las normas del lenguaje en la que está escrita y que es accesible a través de la dirección [http://validator.w3.org](http://validator.w3.org/)



Esta página nos permite validar páginas de tres maneras diferentes:

* **Validate by URI**: Esta opción solo sirve para páginas que están alojadas en algún servidor de Internet como, por ejemplo, <http://www.elcorreo.es> por lo que se escribe la dirección de Internet para validar dichas páginas.
* **Validate by File Upload**: Esta opción permite subir un documento html desde nuestro equipo y validarlo.
* **Validate by Direct Input**: Esta opción permite validar partes o páginas enteras escritas directamente en un editor. Esta opción es muy útil cuando se quiere saber determinados trozos de código son válidos para insertarlos en una página web.

Si el documento html llevado por cualquiera de los métodos anteriores al proceso de validación no lo supera se mostrará un informe completo de los errores que tuviera.

*Es muy conveniente que valides tus páginas web en esta página para saber si cumplen el estándar HTML5*



**Ejercicio nº. 3:** A través de la opción **Validate by File Upload** de la página [http://validator.w3.org,](http://validator.w3.org/) valida la página del ejercicio nº. 2 del tema anterior comprobando si es un documento HTML5 y, en el caso de que hubiera errores, intenta corregirlos (recuerda que los warnings no son errores).

**Ejercicio nº. 4**: Valida cualquiera de las páginas web que sueles visitar (puedes usar http://google.es) indicando:

* + Dirección de la página.
  + Mapa de caracteres
  + Tipo de documento
  + Tres errores, si los tuviera, intentando dar una solución a cada uno de ellos.

# COMENTARIOS

Un comentario es texto que se inserta en una página web para que, cuando se accede al código html de dicha página, explique una parte muy concreta del documento HTML.

*Los comentarios no aparecen cuando se visualiza la página web con un navegador, pero sirven de información sobre algún elemento o parte del documento.*



Para insertar un comentario primero se abre con <!.- - , se pone el texto del comentario y se cierra con - ->

*Agrandan el documento HTML con información que no se va a ver*

#### Ejemplo:

<!- - Esto es un comentario - ->

#### Ejemplo completo:

<!DOCTYPE html>

<html lang=”es-es”>

<head>

<meta charset="utf-8">

<!- - Debajo va el título de la página - ->

<title>Título</title>

</head>

<body>

<! - - Uso la etiqueta de párrafo - ->

<p>

Esto es un párrafo

</p>

<!- - Aquí acaba la página - - >

</body>

</html>

En el navegador devolvería lo siguiente:

Esto es un párrafo

Es decir, la información que va entre <p> y </p>.

# DIVISIONES, PÁRRAFOS Y AGRUPACIONES

A diferencia de la mayoría de los procesadores de texto, HTML utiliza etiquetas de división explícita (<div>), párrafo (<p>), salto de línea (<br>) y agrupación de varios elementos dentro de un bloque (<span>) para controlar la alineación y flujo del texto.

* <br>: Devuelve un retorno de carro o salto de línea. No tiene etiqueta de cierre, es decir, no existe </br> ya es una etiqueta vacía

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <br> | | | | | | |
| Descripción | Devuelve un retorno de carro | | | | | | |
| Elemento | En línea y etiqueta vacía | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y de eventos | | | | | | |
| Atributos propios | No tiene | | | | | | |
| Diferencia entre html  4.01 y html5 | No hay | | | | | | |

* <div>: Divide el documento en secciones distintas e independientes y se puede utilizar estrictamente como una herramienta organizativa, sin ningún tipo de formato asociado.



Ejemplo:

*Su aspecto se maneja con estilos CSS*

<!DOCTYPE html>

<html lang=”es-es”>

<head>

<meta charset="utf-8">

<title>Div y span</title>

</head>

<body>

<div style="border: 1px solid red;"> Cabecera del documento html

</div>

<br>

Texto entre la cabecera y el cuerpo<br>

<br>

<div style="border: 1px solid green;"> Cuerpo del documento html

</div>

<br>

Texto entre el cuerpo y el pie<br>

<br>

<div style="border: 1px solid blue;"> Pie del documento html

</div>

</body>

</html>



En este ejemplo se crean tres divisiones en un documento html (una para la cabecera, otra para el cuerpo del documento y otra para el pie del mismo) y entre cada división hay un texto.

Para el aspecto de la división se emplean estilos que permiten crear un borde de color de un determinado grosor.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <div> | | | | | | |
| Descripción | Define una división en una página html | | | | | | |
| Elemento | En bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y de eventos | | | | | | |
| Atributos propios | No tiene | | | | | | |
| Diferencia entre html  4.01 y html5 | HTML5 no soporta el atributo align | | | | | | |

* <p>: Señala el inicio de un párrafo que finaliza con </p>



Ejemplo:

*Su aspecto se maneja con estilos CSS*

<!DOCTYPE html>

<html lang=”es-es”>

<head>

<meta charset="UTF-8">

<title>Divisiones y párrafos</title></head>

<body>

<p style="color:yellow;background-color:grey ;">



Primera línea del párrafo <br> Segunda línea del párrafo <br>

</p>

</body>

</html>

En este ejemplo el párrafo definido tiene un color de fondo gris y un color de texto amarillo.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <p> | | | | | | |
| Descripción | Define un párrafo | | | | | | |
| Elemento | bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y de eventos | | | | | | |
| Atributos propios | No tiene | | | | | | |
| Diferencia entre html  4.01 y html5 | HTML5 no soporta el atributo align | | | | | | |

* <span>: Permite agrupar varios elementos en línea seguidos dentro de un mismo bloque para después darles formato con hojas de estilo CSS.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <span> | | | | | | |
| Descripción | Define elementos en línea dentro de un bloque | | | | | | |
| Elemento | En línea | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y de eventos | | | | | | |
| Atributos propios | Ninguno | | | | | | |
| Diferencia entre html  4.01 y html5 | Ninguna | | | | | | |

<span> y <div> tienen las mismas propiedades, pero la diferencia varia en la función que cumplen, ya que mientras la etiqueta span trabaja sólo como contenedor en línea, la etiqueta div trabaja como contenedor de bloque.

|  |
| --- |
|  |



*En HTML5 se usan las llamadas etiquetas semánticas que suplen las funciones de div y span*

#### Ejercicios

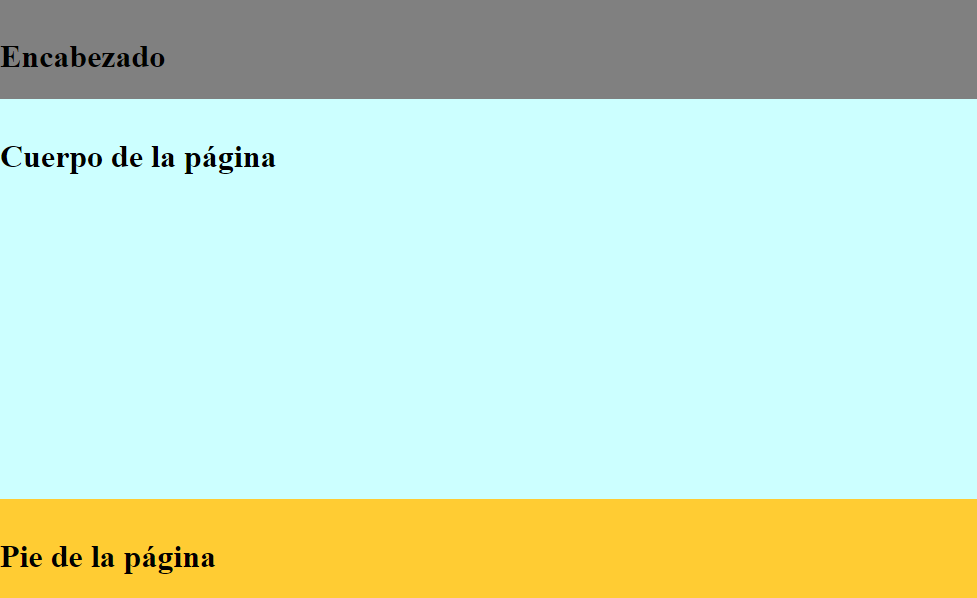
Para los ejercicios siguientes puedes usar el editor web BlueGriffon o Notepad++, pon un título coherente en cada página (recuerda que el título va en la sección <HEAD> por lo que no tienes que escribirlo como contenido en <BODY>) y después de guardar cada ejercicio ábrelo con, al menos, tres de tus navegadores para ver las diferencias que pueda haber al visualizarlos con un navegador u otro.

**Ejercicio nº. 5:** Como aperitivo a la estructura de una página html5, vamos a dividir nuestra página web en partes significativas. Fíjate en la imagen y con bluegriffon o notepad++ crea las divisiones con el elemento <div>. Aunque no hemos visto nada aún de estilos los vamos a utilizar para ver mejor el resultado. Aquí tienes unas pistas para hacer el ejercicio:

* + - * Antes de empezar y por si no conoces cómo funcionan los colores en HTML pásate por el apéndice de colores (https://htmlcolorcodes.com/es/)
      * Pon el título “Divisiones” a la página, usa español (España) para el lenguaje de los elementos de la página y mapa de caracteres UTF-8
      * Encabezado: <div style="background-color: grey; height: 80px;">

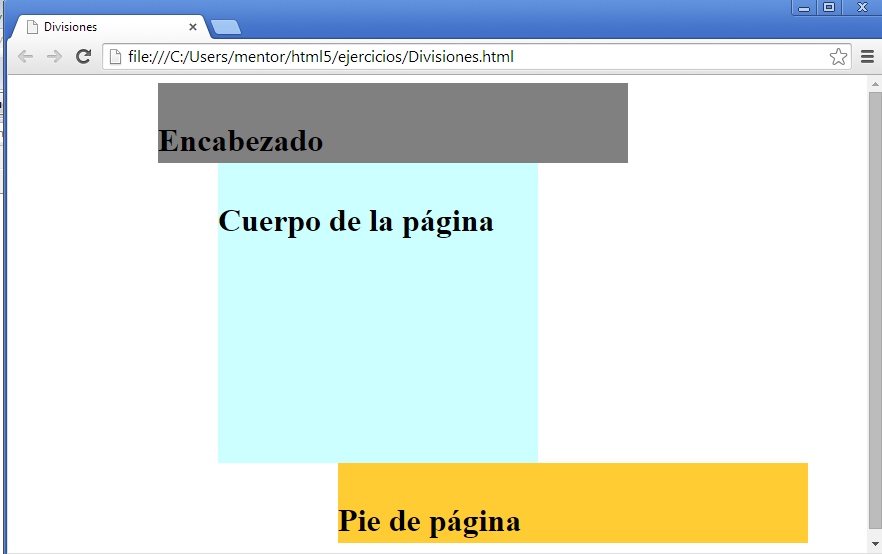
Lo que viene a decir que el color de fondo (background-color) es gris y el ancho (height) de <div> es 80 pixels.

* + - * Cuerpo de la página: <div style="background-color: #ccffff; height: 200px;">
      * Pie de página: <div style="height: 80px; background-color: #ffcc33;">
      * No tengas en cuenta el tamaño de la letra ya que simplemente está puesto así para que se vea mejor. Escribe el texto sin negrita, de tamaño normal y puede estar en cualquier parte de la división, es decir, puede estar pegado, por ejemplo, al borde superior de la misma.
      * Guárdalo como ejercicio3.html



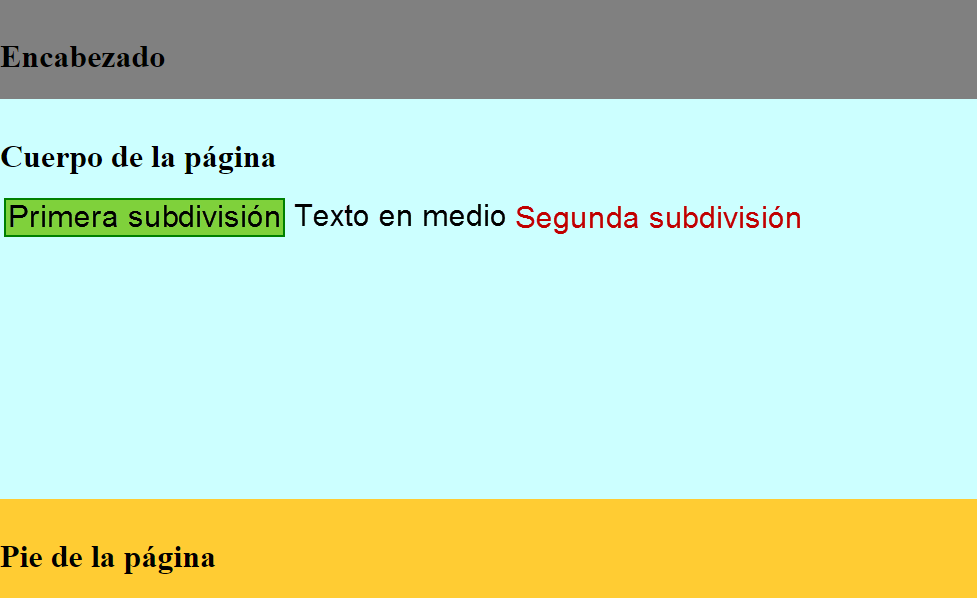
**Ejercicio nº. 6:** Vamos a trabajar con la anchura de las divisiones anteriores y su relación con el cuerpo del documento (<BODY>))

* Abre el ejercicio anterior con BlueGriffon o notepad++.
* Cambia el título por “Anchura de divisiones”
* Para los puntos siguientes vamos a usar otras opciones de estilos como son:
  + *margin-left: 150px;* que significa que el margen izquierdo está a 150 pixels del borde de su contenedor (en el caso de que fuera <BODY> lo sería del borde izquierdo de la ventana)
  + *width: 470px;* que indica que el elemento sobre el que se aplica tiene 470 pixels de ancho.
* Modifica la anchura de <BODY> dejando el su borde izquierdo a 150 pixels del borde la página y que tenga una anchura de 650 pixels. Observa como las divisiones que has creado se adaptan al ancho de <BODY> debido a que <BODY> es un elemento de tipo bloque lo que implica que el ancho de los elementos que hay dentro se ajustan al suyo.
* Modifica el ancho del encabezado y déjalo con 470 pixels de ancho.
* Modifica el cuerpo de la página dejando un margen a la izquierda de 60 pixels y con un ancho de 320 pixels
* Modifica el ancho del pie de página dejando un margen a la izquierda de 180 pixels.
* Guarda la página como ejercicio4.html
* Fíjate en la imagen ya que es lo que tienes que ver en el navegador cuando abras con él la página (no tengas en cuenta el tamaño y posición del texto).



**Ejercicio nº. 7:** Vamos a practicar con el elemento <span>. Este elemento es in-line y sirve para hacer subdivisiones normalmente dentro de una división. Abre el ejercicio “divisiones.html” con BlueGriffon o notepad++ y vamos a crear lo siguiente dentro de “Cuerpo de la página” y debajo del texto que pusimos:

* Cambia el título de la página por “Span”
* Un elemento <span> con el texto “Primera subdivisión” con color de fondo #33cc00
* El texto “Estoy en medio de dos <span>”
* Otro elemento <span> con el texto “Segunda subdivisión” de color rojo (red). Usa el estilo “color” para el texto de este elemento y aplícalo sobre <span> no sobre dicho texto.
* No tengas en cuenta el tamaño del texto ni su posición, pero si el color y la disposición del mismo con arreglo a la imagen de abajo.
* Guarda el archivo como ejercicio5.html



**Ejercicio nº. 8:** Vamos con los párrafos y seguimos avanzando un poco en el tema de aplicar estilos a los diferentes elementos.

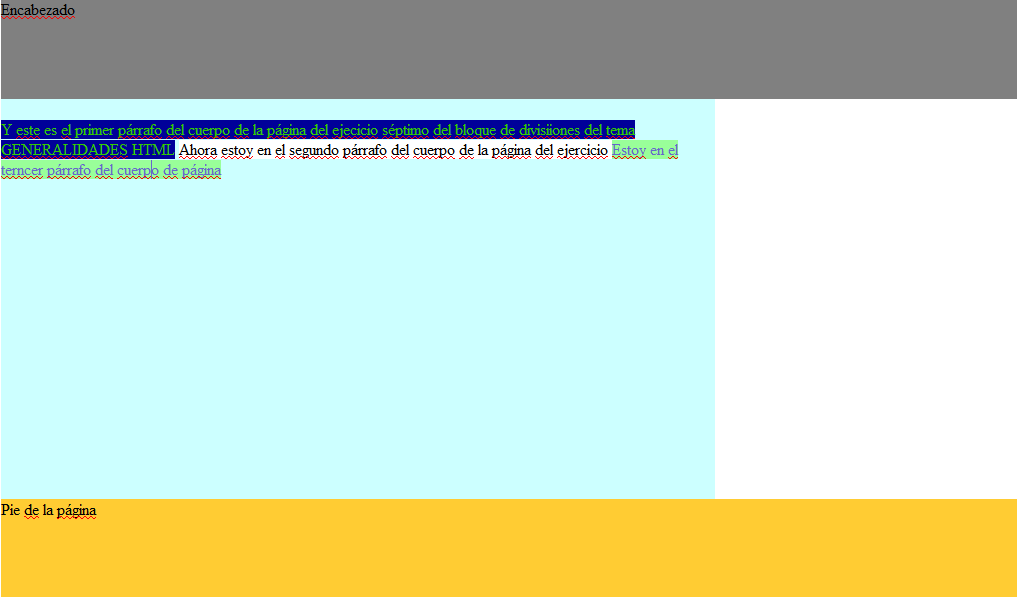
* Abre el ejercicio3.html.
* Cambia el título de la página por “Párrafos”
* En la primera división, “encabezado”, borra el texto que hay y crea un párrafo con el color de fondo #ff99ff y el texto “Párrafo de encabezado”.
* Ajusta el ancho de la segunda división o “cuerpo de la página” a 150 pixels
* En la división cuerpo de la página, borra el texto que hay y haz lo siguiente:
  + Inserta un retorno de carro (<br>)
  + Crea un párrafo con el texto “Estoy en el segundo párrafo del cuerpo de página”, con el fondo de color azul, el texto de color #ffff99 y centrado.
  + Crea otro párrafo con el texto “Tercer párrafo”, con el fondo de color #ffccff y ajustado el texto a la derecha.
* Guarda la página como ejercicio6.html.
* Fíjate en cómo se ajustan los párrafos a la anchura de las divisiones y que los dos párrafos de la segunda división o “cuerpo de la página” están uno debajo del otro debido a que <p> es un elemento de tipo bloque.



**Ejercicio nº.9:** Ahora vamos a ver cómo podemos poner dos párrafos uno al lado del otro, es decir, como cambiar un elemento de tipo bloque para que funcione como en línea o inline.

* Abre el archivo ejercicio3.html.
* Borra el texto de la división “Cuerpo de la página” y haz lo siguiente en esa división:
  + Déjale un ancho de 700px.
  + Inserta un retorno de carro (<br>)
  + Crea un párrafo que tenga de ancho 550px, un color de fondo #009999, un color de texto de #33cc00 y el siguiente texto “Y este es el primer párrafo del cuerpo de página del ejercicio séptimo del bloque de divisiones del tema GENERALIDADES HTML”.
  + Crea otro párrafo que tenga de ancho 335px, un color de fondo blanco, un color de texto negro y el siguiente texto “Ahora estoy en el segundo párrafo del cuerpo de página”
  + Y, por último, crea otro párrafo que ocupe 225px, un color de fondo de #99fff99, un color de texto de #6666cc
  + La primera imagen es la que te muestra cómo te deben quedar los párrafos en primera instancia.
  + Ahora cambia las propiedades de los párrafos para que queden uno junto a otro tal y como te muestra la segunda imagen.
  + Fíjate que ahora los párrafos ya no cumplen el ancho que les has puesto en los puntos anteriores debido a que ahora son elementos de tipo inline.
  + Guarda la página como ejercicio7.html.





# TABLAS

Una tabla es una colección de filas donde cada una ellas contiene un número determinado de celdas. Dentro de estas celdas se puede introducir texto, imágenes o cualquier otro elemento html incluidas otras tablas.

*Las tablas en HTML5 solo valen para presentar datos tabulados no para organizar los elementos de una página como se usaba con HTML4*



* **<table>** Define, empleando normalmente estilos CSS. los parámetros generales de la tabla que marcan su aspecto.

El contenido de la tabla está entre la etiqueta de apertura <table> y la de cierre </table>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <table> | | | | | | |
| Descripción | Define una tabla de datos | | | | | | |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios | sortable | | | | | | |
| Diferencia entre html  4.01 y html5 | Los atributos de aspecto como align, bgcolor, border, etc. están suprimidos en HTML5 | | | | | | |

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| sortable | sortable | Especifica que la tabla debería estar ordenada |

Dentro de <table> </table> se definen otros elementos que marcan el aspecto de la tabla.

* + <th>: Etiqueta que indica que la celda es de cabecera o título de una columna, fila, conjunto de filas o conjunto de columnas. El texto del título se pone en negrita y centrado por defecto.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <th> | | | | | | |
| Descripción | Define la cabecera de fila, grupo de filas, columna, grupo de columanas de una tabla | | | | | | |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios | scope | | | | | | |
| Diferencia entre html  4.01 y html5 | Los atributos de aspecto como align, bgcolor,, etc. están suprimidos en HTML5 | | | | | | |

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| abbr | texto | Descripción corta del contenido de la celda de cabecera |
| colspan | número | Especifica el número de columnas que se pueden fusionar en una celda de cabecera. Solo Firefox permite colspan=”0” |
| headers | identificador | Indica un identificador que relaciona celdas de cabecera entre si |
| rowspan | número | Especifica el número de filas que se pueden fusionar en una celda de cabecera |
| scope | col colgroup row  rowgroup | Indica si es cabecera de una columna, fila, conjunto de filas o conjunto de columnas |
| sorted | Reversed | Define el sentido del orden de una columna |

|  |  |  |
| --- | --- | --- |
|  | Number  Reversed number  Number reversed |  |

* + <tr>: Etiqueta que define una fila. La fila finaliza con el etiqueta </tr>.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <tr> | | | | | | |
| Descripción | Define una fila de una tabla | | | | | | |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios |  | | | | | | |
| Diferencia entre html  4.01 y html5 | Los atributos de aspecto como align, bgcolor,, etc. están suprimidos en html5 | | | | | | |

* + <td>: Etiqueta que define una celda. Su texto se pone normal y ya alineado a la izquierda por defecto. La celda finaliza con la etiqueta </td>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <td> | | | | | | |
| Descripción | Define una celda en una tabla | | | | | | |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |

|  |  |
| --- | --- |
| Atributos | Globales y eventos |
| Atributos propios |  |
| Diferencia entre html  4.01 y html5 | Los atributos de aspecto como align, bgcolor,, etc. están suprimidos en html5 |



Ejemplo de tabla simple:

*Una tabla tiene el mismo número de celdas en todas las filas aunque se pueden agrupar.*

<!DOCTYPE html>

<html lang=”es-es”>

<head>

<title>Col y colgroup</title>

<meta charset="UTF-8">

</head>

<body>

<table>

<tr>

<th> Titulo columna 1</th>

<th> Título columna 2</th>

<th> Título columna 3</th>

</tr>

<tr>

<td> fila 1 celda 1 </td>

<td> fila 1 celda 2 </td>

<td> fila1 celda 3 </td>

</tr>

<tr>

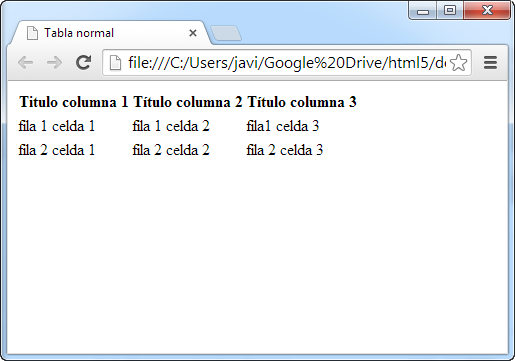
<td> fila 2 celda 1 </td>

<td> fila 2 celda 2 </td>

<td> fila 2 celda 3 </td>

</tr>

</table>



</body>

</html>

En este ejemplo la tabla sale sin borde por lo que no se distingue la separación entre las celdas. Hay que evitar atributos como border (que dibuja los bordes de las celdas) que están totalmente obsoletos y usar hojas de estilos para hacer este tipo de cosas.

* + <caption> Título de una tabla que debería dar una descripción breve del contenido de la tabla. Tiene que ir justo debajo de la etiqueta de apertura de la tabla, es decir, debajo de <table>
  + ​

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <caption> | | | | | | |
| Descripción | Define el título de una tabla | | | | | | |
| Elemento | En línea | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios |  | | | | | | |
| Diferencia entre html  4.01 y html5 | El atributo align está suprimido en HTML5 | | | | | | |



* <colgroup> Define un grupo de columnas de una tabla para darles un formato común, es decir, se pueden aplicar el mismo estilo a un conjunto de columnas. Va detrás justo de <caption> y delante de <thead>

*Solo puede haber una etiqueta <caption> por tabla*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <colgroup> | | | | | | |
| Descripción | Especifica un grupo de una o más columnas en una tabla | | | | | | |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios | span | | | | | | |
| Diferencia entre html  4.01 y html5 | Los atributos de aspecto como align, bgcolor,, etc. están suprimidos en html5 | | | | | | |

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| Span | número | Especifica el número de columnas contiguas que se pueden agrupar |

* <col> Especifica una columna en un tabla y permite dar estilos a una columna en concreto. Normalmente es componente de <colgroup> aunque se puede utilizar como etiqueta independiente.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <col> | | | | | | |
| Descripción | Especifica una columna en un tabla | | | | | | |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios | span | | | | | | |
| Diferencia entre html  4.01 y html5 | Los atributos de aspecto como align, bgcolor,, etc. están suprimidos en HTML5 | | | | | | |

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| Span | número | Especifica el número de columnas contiguas que se pueden agrupar |



*El atributo span tiene el mismo uso tanto para <colgroup> como para*

*<col>. Esto es debido a que <col> puede ser usado como elemento independiente, es decir, que no esté dentro de <colgroup>*

#### Ejemplo:

<!DOCTYPE html>

<html lang=”es-es”>

<head>

<title>Col y colgroup</title>

<meta charset="UTF-8">

</head>

<body>

<table>

<tr>

<th>Número</th>

<th>Nombre</th>

<th>Apellido</th>

<th>Salario</th>

</tr>

<tr>

<td>0001</td>

<td>Marta</td>

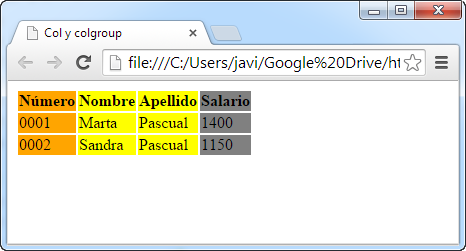
<colgroup>

<col style="background-color:orange">

<col span="2" style="background-color:yellow">

<col style="background-color:grey">

</colgroup>



</body>

</html>

<td>Pascual</td>

<td> 1400</td>

</tr>

<tr>

<td>0002</td>

<td>Sandra</td>

<td>Pascual</td>

<td> 1150</td>

</tr>

</table>

En este ejemplo se ve que se definen tres grupos de columnas dentro de <colgroup>. El primero formado por una columna a la que se aplica color naranja de fondo, el segundo formado por dos columnas (span=2) y al que se aplica el color amarillo de fondo y el tercer y último grupo formado por una columna a la que se aplica el color gris de fondo.



*<colgroup> y <col> se usan principalmente para aplicar estilos a una o varias columnas.*

*Si se aplica un color de fondo o cualquier otro estilo a una celda individual prevalece sobre lo que se defina con <col>*



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiquetas | <thead> <tbody> y <tfoot> | | | | | | |
| Descripción | <thead> define una o más filas de encabezado en una tabla, <tbody> define una o más filas como sección de una tabla y <tfoot> define una o más filas de pie de la tabla | | | | | | |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios |  | | | | | | |
| Diferencia entre html  4.01 y html5 | Ninguno de los atributos de HTML4 son soportados en HTML5 | | | | | | |



El uso de estas etiquetas permite que si la tabla tiene demasiadas filas que no pueden ser visualizadas a la vez y hay que irlas pasando, entonces la cabecera y el pie se quedan fijos y se ven siempre.

*<thead>, <tbody> y <tfoot> siempre deben ser usados juntos*

La etiqueta <thead> va detrás de los elementos de tabla como <caption>, <colgroup> o

<col> y antes de cualquier <tr>

*En una tabla solo hay una etiqueta <thead> y una <tfoot> aunque puede haber varias <tbody>*



#### Operaciones con tablas

*<tfoot> se usa antes que <tbody> por lo que el orden que debemos seguir para empelar estas etiquetas sería: <thead>, <tfoot> y <tbody>*

- Unión horizontal de celdas: Con el atributo colspan de la etiqueta <td> se pueden unir celdas contiguas de una misma fila.

Ejemplo:

<!DOCTYPE html>

<html lang=”es-es”>

<head>

<title>Fusión columnas</title>

<meta charset=”UTF-8”>

</head>

<body>

</body>

</html>

<table style="border:1px solid #000">

<tr>

<td style="border:1px solid #000"> fila 1 celda 1 </td>

<td style="border:1px solid #000"> fila 1 celda 2 </td>

<td style="border:1px solid #000"> fila 1 celda 3 </td>

</tr>

<tr>

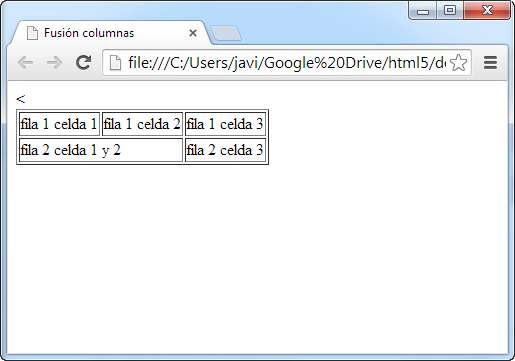
<td colspan=2 style="border:1px solid #000"> fila 2 celda 1 y 2

</td>

<td style="border:1px solid #000"> fila 2 celda 3 </td>

</tr>

</table>



- Unión vertical de celdas: Con el atributo rowspan de la etiqueta <td> se puedne unir celdas contiguas de una misma columna.

Ejemplo:

<!DOCTYPE html>

<html lang=”es-es”>

<head>

<title>Fusión filas</title>

<meta charset=”UTF-8”>

</head>

<body>

<table style="border:1px solid #000">

<tr>

<td rowspan=2 style="border:1px solid #000"> filas 1 y 2 celda 1 </td>

<td style="border:1px solid #000"> fila 1 celda 2 </td>

<td style="border:1px solid #000"> fila 1 celda 3 </td>

</tr>

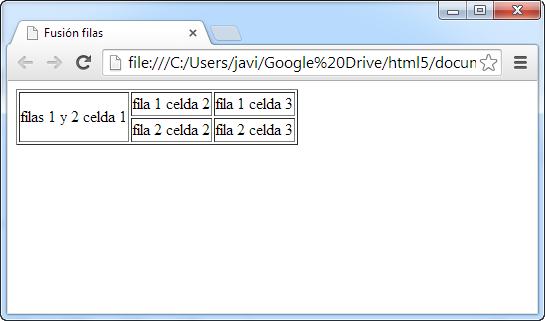
<tr>

<td style="border:1px solid #000"> fila 2 celda 2 </td>

<td style="border:1px solid #000"> fila 2 celda 3 </td>

</tr>

</table>



</body>

</html>

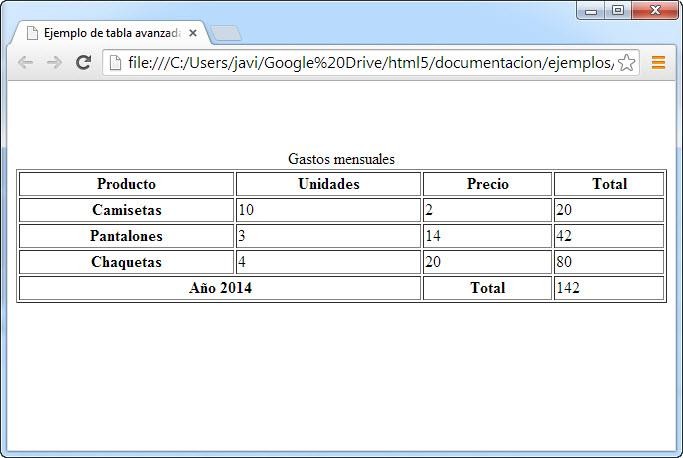
*style=”border:1px solid #000;”* permite poner un borde a la tabla (elemento <table>) o a una celda (elemento <td>) de un pixel de grosor y de color negro.

**Ejercicio nº. 10:** Fíjate en la imagen de la tabla inferior y crea una página HTML5 que reproduzca tal cual está dicha tabla. El ancho de la tabla es el ancho de la ventana por lo que usa el siguiente estilo tanto para la tabla como para las celdas:

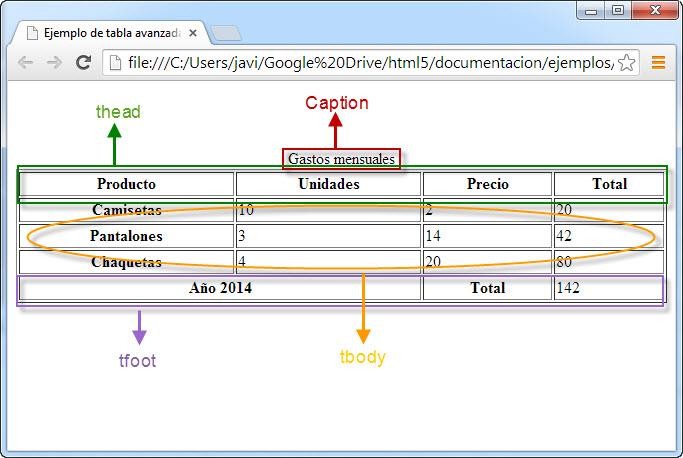
*style=”border:1px solid #000;witdh:100%;”*

Este estilo dibuja un borde de 1 pixel y que ocupa todo el ancho de la ventana en el caso de la table (<table>) y todo el ancho que pueda dependiendo del contenedor en el que está en el caso de la cela (<td>). No pienses mucho en cómo funciona el estilo anterior ya que lo veremos más profundamente posteriormente en el curso. Sin embargo, empléalo para que tengas una mejor perspectiva de lo que estás haciendo.

Ten en cuenta también que no se pueden hacer cálculos, es decir, las operaciones son datos que debes poner directamente sin utilizar ninguna fórmula o cálculo. Lo importante de este ejercicio es que identifiques las partes de una tabla y que sepas su utilidad.

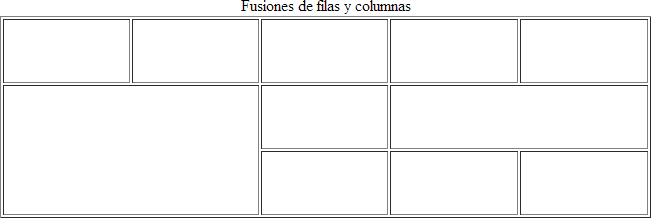


Como guía básica fíjate en la siguiente imagen para que veas las partes de la tabla



El código lo tienes aquí (enlace 5table.html)

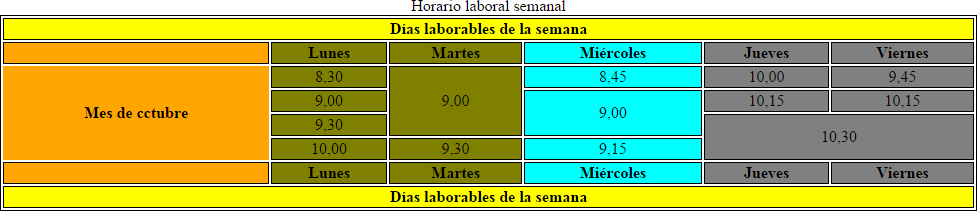
**Ejercicio nº. 11:** Realiza la siguiente tabla fusionando filas y columnas. Usaremos también el estilo anterior para que sea más claro el ejercicio.



Enlace al ejercicio 6table.html

**Ejercicio nº. 12:** Usando todos los conceptos que hemos visto en este tema, crea la tabla de la imagen inferior. Aquí tienes unas pequeñas pautas:

* La tabla ocupa todo el ancho de la página y tiene un borde de color negro de 1 pixel de ancho (style="border:1px solid #000;width:100%;").
* Todas las celdas de la tabla tienen borde de 1 pixel de color negro (style=”border:1px solid #000;”).
* La cabecera de la tabla, <tbody>, ocupa las dos primeras filas, el pie, <tfoot>, tiene otras dos filas y el cuerpo, <tbody>, el resto.
* Hay una agrupación de columnas (<colgroup>) que permiten dar los colores: orange, olive, aqua y grey a los conjuntos de columnas (<col>).
* Las celdas con contenido en negrita y centrado se definen con la etiqueta <th>.
* Las celdas que no están en negrita se definen con la etiqueta <td>.
* Para centrar un texto en una celda <td> se usa el estilo *text-align:center.*



# LISTAS

Una lista es un párrafo estructurado que contiene una serie de elementos HTML que tienen más significado de forma conjunta. Los tipos de listas son:

* **Listas no ordenadas**: Este tipo de lista se usa para enumerar elementos que no tengan un orden definido. Su etiqueta de definición es <ul> y todo lo que se incluya dentro a partir de esa etiqueta y la de cierre </ul>, será el contenido de la lista con los componentes de la misma dentro de etiquetas <li>. También puede contener un título de lista definido por la etiqueta <lh>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <ul> | | | | | | |
| Descripción | Define una lista no ordenada | | | | | | |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios |  | | | | | | |
| Diferencia entre html  4.01 y html5 | Los atributos compact y type están suprimidos en HTML5 | | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <li> | | | | | | |
| Descripción | Define un elemento de una lista ordenada, no ordenada o menú | | | | | | |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios | value | | | | | | |
| Diferencia | El atributo “type” está suprimido en HTML5 y el atributo “value” que está | | | | | | |

|  |  |
| --- | --- |
| entre html  4.01 y html5 | obsoleto en HTML4.01 pero es soportado en HTML5 |

En la siguiente tabla están los atributos propios de <li>

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| value | Número | Especifica el valor de la lista en ese elemento. Solo para listas ordenadas (<ol>) |



*<li> define elementos de listas ordenadas, <ol>, no ordenadas, <ul> y menú, <menu>*



Ejemplo:

*Usa estilos CSS para definir el tipo de lista*

|  |
| --- |
| <!DOCTYPE html>  <html lang=”es-es”>  <head>  <title>Col y colgroup</title>  <meta charset="UTF-8">  </head>  <body>  <ul>  <lh> Título de la lista </lh>  <li> Elemento 1</li>  <li> Elemento 2 </li>  </ul> |

|  |
| --- |
| </body>  </html> |
|  |

* **Listas ordenadas**: En este tipo de lista se usa un número o letra para indicar el orden de cada elemento de la lista. Su etiqueta de definición es <ol> y su uso es idéntico a <ul>

*La listas ordenadas no se ordenan bajo ningún criterio, es decir, solamente siguen el orden de la lista independientemente de su contenido*



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <ol> | | | | | | |
| Descripción | Lista ordenada | | | | | | |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos | Reversed, start, type | | | | | | |

|  |  |
| --- | --- |
| propios |  |
| Diferencia entre html  4.01 y html5 | Los atributos “start” y “type” de HTML 4.01 están obsoletos, pero los soporta HTML5.  El atributo “reversed” es nuevo en HTML5  El atributo “compact” está suprimido en HTML5 |

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| reversed | reversed | Especifica que el orden de la lista debería ser descendente. No soportado por Safari 5 y versiones anteriores |
| start | número | Especifica el primer valor de la lista |
| type | 1  A  a I  i | Especifica el tipo de marcador que se usa en la lista. |

  Nuevo en HTML5 Ejemplo:

|  |
| --- |
| <!DOCTYPE html>  <html lang=”es-es”  <head>  <title> Listas Ordenadas</title>  <meta charset=”UTF-8”>  </head>  <body>  <ol>  <lh> Título de la lista </lh>  <li> Primer elemento </li> |

|  |
| --- |
| <li> Segundo elemento </li>  </ol>  </body>  </html> |
|  |

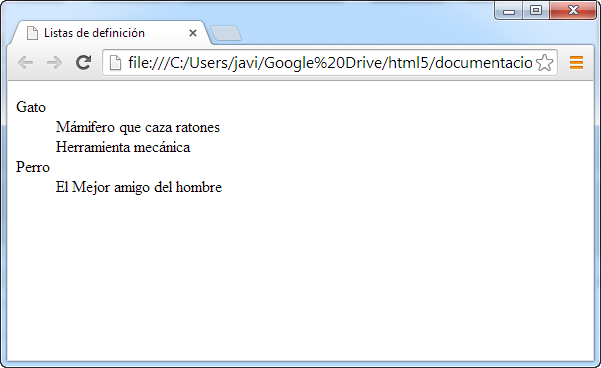
* **Listas de descripción**: En este tipo de lista se indican uno o varios términos y se dan su definiciones. Su etiqueta principal es <dl> y entre ella y la de cierre </dl> se indica el término con la etiqueta <dt> y se define con una o varias etiquetas <dd>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <dl> | | | | | | |
| Descripción | Define una lista de descripción | | | | | | |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios |  | | | | | | |

|  |  |
| --- | --- |
| Diferencia entre html  4.01 y html5 | En HTML 4.01 define una lista de definición y en HTML5 define una lista de descripción |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <dt> | | | | | | |
| Descripción | Se emplea para indicar los términos de una lista de descripción | | | | | | |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios |  | | | | | | |
| Diferencia entre html  4.01 y html5 | En HTML 4.01 define los términos de una lista de definición y en HTML5 define los términos de una lista de descripción | | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <dd> | | | | | | |
| Descripción | Define los términos de una lista de descripción | | | | | | |
| Elemento | Bloque | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios |  | | | | | | |
| Diferencia entre html  4.01 y html5 | En HTML 4.01 define los términos de una lista de definición y en HTML5 define los términos de una lista de descripción | | | | | | |



Ejemplo:

<!DOCTYPE html>

<html lang=”es-es”

<head>

<title> Listas Ordenadas</title>

<meta charset=”UTF-8”>

</head>

<body>

<dl>

<dt> Gato

<dd> Mámifero que caza ratones

<dd> Herramienta mecánica

<dt> Perro

<dd> El Mejor amigo del hombre

</dl>

</body>

</html>

#### Anidamiento de listas

Una lista puede tener definidas otras listas de cualquier tipo dentro de alguno de sus elementos entonces hablamos de listas anidadas y de niveles, donde nivel es la profundidad de una lista o el número de listas que tiene por encima alguna de ellas.

Ejemplo de lista de nivel 3

<!DOCTYPE html>

<html lang=”es-es”

<head>

<title> Listas Ordenadas</title>

<meta charset=”UTF-8”>

</head>

<body>

<ol>

<lh> Provincias, comarcas y pueblos </lh>

<li> Soria

<ul>

<li> Pinares

<ul>

<li> Covaleda</li>

<li> Vinuesa </li>

</ul>

</li>

<li> El Valle

<ul>

<li> El Royo</li>

<li> Molinos de Razón</li>

</ul>

</li>

</body>

</html>

</ul>

</li>

<li> Zaragoza

<ul>

<li> Valdejalón

<ul>

<li>Calatorao</li>

<li>Épila </li>

<li>Chodes</li>

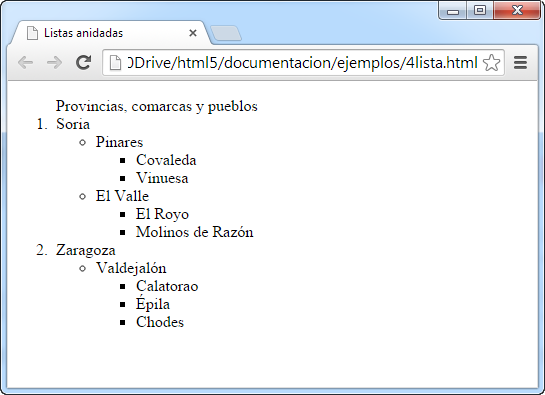
</ul>

</li>

</ul>

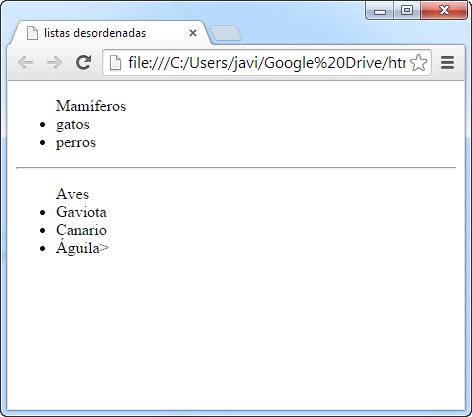
</li>

</ol>



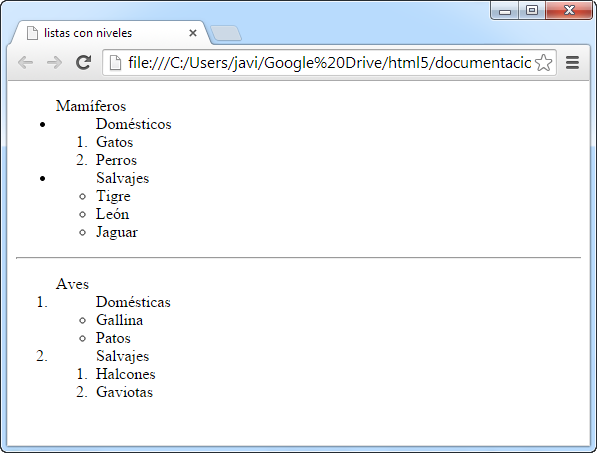
En este ejemplo Covaleda, Vinuesa, Calatorao, etc. son listas de nivel 3 ya que tienen dos niveles por encima de ellas, Pinares, El Valle y Valdejalón son listas de nivel 2 y Soria y Zaragoza son de nivel 1

**Ejercicio 13:** Realiza el siguiente ejercicio con dos listas desordenadas



Para separar ambas listas usa la etiqueta <hr> que dibuja una línea horizontal

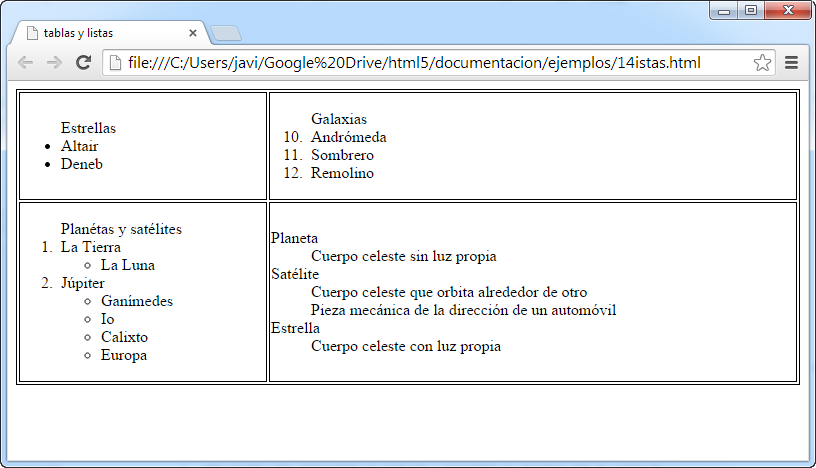
**Ejercicio 14:** Fíjate en la imagen e intenta hacerla tal cual identificando bien los tipos de listas, sus títulos y sus niveles.



**Ejercicio 15:** Utilizando una tabla de dos filas y dos columnas realiza el siguiente ejercicio empleando el siguiente estilo para la tabla

style="width:100%; border:1px solid #000;" y este para cada celda

style=”border:1px solid #000;"



# ENLACES

Los enlaces, vínculos o hipervínculos se utilizan para establecer relaciones [entre](http://librosweb.es/xhtml/capitulo_4.html) dos objetos que pueden ser páginas web, imágenes, documentos, etc., es decir, que pueden ser archivos.

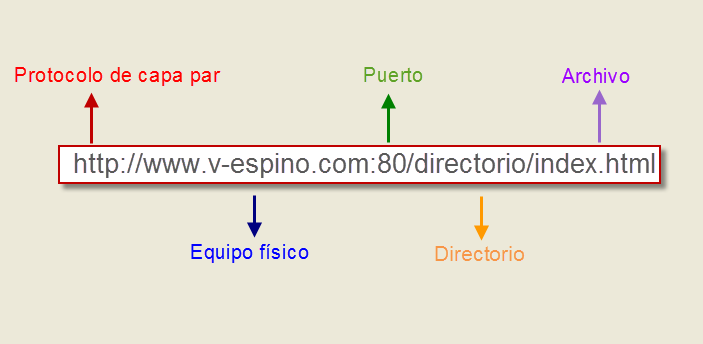
Una característica importante de los enlaces es que están formados por un origen, un destino y una dirección. En otras palabras, el enlace comienza en un objeto origen y apunta hacia otro, que es el destino, iniciando el acceso a este último a través de un clic de ratón sobre el enlace.

#### URL (localizador uniforme de recursos).

URL es un texto que cumple unas normas muy concretas y que sirve para referenciar o identificar un recurso así como localizar el equipo de una red de comunicaciones como, por ejemplo, Internet donde se aloja dicho recurso.

La URL de un objeto tiene dos fines importantes:

* Identificarlo y diferenciarlo claramente de los demás objetos.
* Localizarlo de la forma más rápida a través de la información que suministra la propia URL.



En la imagen anterior vemos un ejemplo de URL que cumple la normativa TCP/IP que es la usada en Internet. Las URL están formadas por las siguientes partes:

* Protocolo de capa par: Normas que debe cumplir el navegador para visualizar sin errores la información del archivo. Otros protocolos de capa par son ftp, DNS, https, pop3, etc.
* Equipo físico: Equipo en el que se encuentra alojado el archivo al que se quiere acceder. Está representado por un nombre DNS o por una dirección ip.
* Puerto: Identificador dentro del sistema de la aplicación servidora que ofrece el archivo.
* Directorio: Directorio donde se encuentra el archivo y que forma parte de la estructura que controla la aplicación servidora que la ofrece.
* Archivo: Fichero al que referencia la URL.

- *URL absolutas.*

Son las que definen todas las partes de la URL por lo que se tiene todo lo necesario para acceder al objeto destino.

Ejemplo:

*Las URL absolutas tienen el inconveniente de que si el recurso cambia de ubicación todas las URL hay que cambiarlas también*

<http://v-espino.com/index.html>



- *URL relativas.*

Son las que no definen todas las partes de la URL con el fin de hacerlas más simples. En realidad, son URL incompletas que se crean a partir de URL absolutas y que obligan a conocer la URL origen del enlace para que funcionen sin problemas.

Ejemplo:

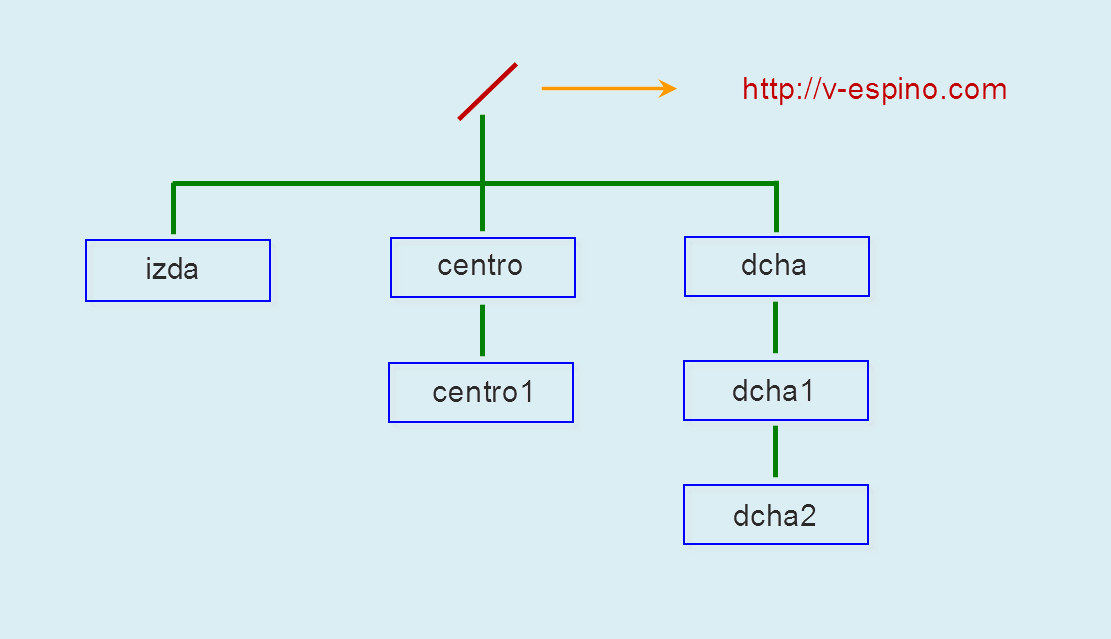
/index.html

Como vemos esta URL no es muy significativa ya que no sabemos en qué ubicación está el archivo. Por eso es importante conocer que la URL absoluta del ejemplo es [http://v-](http://v-espino.com/index.html) [espino.com/index.html](http://v-espino.com/index.html) lo que indica que index.html está en la raíz del servidor v- espino.com.



*Las URL relativas sirven para referirse a recursos dentro del mismo ordenador que pertenecen a la estructura normalmente de una aplicación servidora*

Fíjate en la imagen siguiente y supón que tenemos dos archivos uno llamado origen.html y otro llamado destino.html y que origen.html tiene un enlace que va a destino.html. Vamos a ponerlos en diferentes partes de la estructura y vamos a ver como quedarían las URL relativas y absolutas cuando se usan en el archivo origen.html para referirse a destino.html



|  |  |  |
| --- | --- | --- |
| Carpeta de  origen.html | Carpeta de  destino.html | URL ABSOLUTA de destino.html con respecto a la carpeta de origen.html |
| / | / | <http://v-espino.com/destino.html> |
| /izda | /izda | <http://v-espino.com/izda/destino.html> |
| /izda | /centro | <http://v-espino.com/centro/destino.html> |
| /centro/centro1 | /dcha | <http://v-espino.com/dcha/destino.html> |
| /dcha/dcha1 | /centro/centro1 | <http://v-espino.com/centro/centro1/destino.html> |
| / | /dcha/dcha1/dcha2 | [http://v-](http://v-/) espino.com/dcha/dcha1/dcha2/destino.html |

Como se ve en la tabla anterior la URL absoluta de destino.html solo depende de su ubicación independientemente de donde esté origen.html.

Si llevamos las páginas a otro servidor que no sea v-espino.com entonces no funcionará ninguna de la URL anteriores.

Antes de pasar a las URL relativas hay que saber que “..” representa la carpeta inmediatamente superior y no hace falta usar el nombre de dicha carpeta ya que una carpeta solo puede tener otra por encima a excepción de la carpeta / o raíz que, evidentemente, no tiene ninguna.

|  |  |  |
| --- | --- | --- |
| Carpeta de  origen.html | Carpeta de  destino.html | URL RELATIVA de destino.html con respecto a la carpeta de origen.html |

|  |  |  |
| --- | --- | --- |
| / | / | destino.html |
| /izda | /izda | destino.html |
| /dcha/dcha1/dcha2 | /dcha/dcha1/dcha2 | destino.html |
| /izda | / | ../destino.html |
| /izda | /centro | ../centro/destino.html |
| /izda | /centro/centro1 | ../centro/centro1/destino.html |
| /izda | /dcha | ../dcha/destino.html |
| /izda | /dcha/dcha1/dcha2 | ../dcha/dcha1/dcha2/destino.html |
| /centro/centro1 | / | ../../destino.html |
| /centro/centro1 | /izda | ../../izda/destino.html |
| /centro/centro1 | /dcha/dcha1 | ../../dcha/dcha1/destino.html |
| /centro/centro1 | /centro | ../destino.html |
| /dcha/dcha1/dcha2 | / | ../,,/../destino.html |
| /dcha/dcha1/dcha2 | /izda | ../../../izda/destino.html |
| /dcha/dcha1/dcha2 | /centro/centro1 | ../../../centro/centro1/destino.html |
| /dcha/dcha1/dcha2 | /dcha | ../../destino.html |
| /dcha/dcha1/dcha2 | /dcha/dcha1 | ../destino.html |
| / | /dcha/dcha1/dcha2 | dcha/dcha1/dcha2/destino.html |
| /dcha | /dcha/dcha1/dcha2 | dcha1/dcha2/destino.html |
| /dcha/dcha1 | /dcha/dcha1/dcha2 | cha2/destino.html |

Las conclusiones que se sacan de la tabla anterior son:

* Si ambas páginas están en la misma carpeta con solo poner el nombre de la página de destino vale.
* Si la página destino está justo en la carpeta superior de la del origen con poner

../nombre de la página de destino vale.

* Si la página destino está dos carpetas por encima de la de origen con poner

../../nombre de la página de destino vale.

* Si la página destino está una carpeta por debajo de la de origen con poner el nombre de la carpeta seguido del nombre de la página destino vale.
* Si la página destino está varias carpetas por debajo de la de origen con poner las carpetas sucesivamente y luego el nombre de la página destino vale.
* Si la página destino se encuentra en una carpeta que no está ni encima ni debajo de la de origen se ponen tantos “..” como carpetas hay hasta la raíz y luego se van poniendo los nombres de las carpetas intermedias a medida que se va descendiendo hasta la carpeta de la página de destino seguido del nombre de la página destino.
* Si llevamos las páginas a otro servidor que no sea v-espino.com las URL serían perfectamente válidas siempre que dicho servidor tenga una estructura igual de carpetas.

#### Etiqueta <a>

Para crear un enlace de cualquier tipo se emplea la etiqueta <a> cuya principal función es especificar la dirección del destino ya sea en la propia página (enlace interno) o fuera de la misma (enlace externo).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <a> | | | | | | |
| Descripción | Crea un enlace a un objeto | | | | | | |
| Elemento | En línea | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios | href, hreflang, rel, target, download, type | | | | | | |
| Diferencia entre html  4.01 y html5 | Los atributos compact, name y de aspecto están suprimidos en HTLM5 y en los enlaces internos se usa el atributo global *id* (HTML5) en vez de *name* (HTML4) | | | | | | |

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| [download](http://www.w3schools.com/tags/att_a_download.asp) | Nombre de fichero | Especifica que el fichero indicando será descargado cuando se pulse sobre el enlace |
| [href](http://www.w3schools.com/tags/att_a_href.asp) | URL | Especifa la URL a la que apunta el enlace |
| [hreflang](http://www.w3schools.com/tags/att_a_hreflang.asp) | Idioma | Especifica el lenguaje del documento destino |

|  |  |  |
| --- | --- | --- |
| [media](http://www.w3schools.com/tags/att_a_media.asp) | Dispositivo | Especifica cual es el dispositivo óptimo para usar con el documento enlazado |
| [rel](http://www.w3schools.com/tags/att_a_rel.asp) | alternate author bookmark help license next nofollow noreferrer prefetch prev search  tag | Especifica la relación entre el documento origen y el documento enlazado |
| [target](http://www.w3schools.com/tags/att_a_target.asp) | \_blank  \_parent  \_self  \_top  Nombre marco | Especifica dónde se abrirá el documento enlazado |
| [type](http://www.w3schools.com/tags/att_a_type.asp) | Tipo MIME | Especifica el tipo MIME del documento enlazado |

  Nuevo en HTML5

*El atributo más importante de <a> es* ***href*** *ya que indica la URL del fichero destino en los enlaces externos o la URL más el ancla o marcador en los enlaces internos.*



En general los enlaces tienen la siguiente estructura

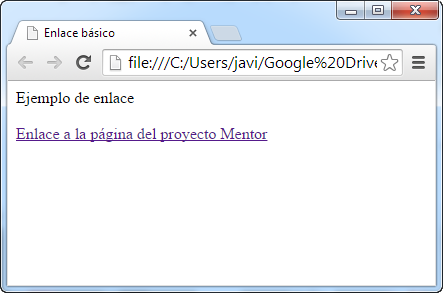
#### <a href=”URL”>Texto del enlace </a>

Donde URL es la dirección del enlace y “Texto del enlace” es lo que se visualizará en el navegador subrayado normalmente en azul y que se pulsa para ir al destino.

*Se puede emplear una imagen o cualquier elemento HTML5 en vez de texto para hacer clic sobre él en un enlace.*



#### Ejemplo:



!DOCTYPE html>

<html>

<head>

<title>Enlace básico</title>

<meta charset="UTF-8">

</head>

<body>

Ejemplo de enlace <br><br>

</body>

</html>

<a href=["http](http://www.mentor.mec.es/):[//www.mentor.mec.es](http://www.mentor.mec.es/)">Enlace a la página del proyecto Mentor</a>

En este ejemplo si pulsamos sobre “Enlace a la página del proyecto Mentor” abrirá la página correspondiente a la dirección [http://www.mentor.mec.es.](http://www.mentor.mec.es/)

Ejemplo:

!DOCTYPE html>

<html>

<head>

<title>Enlace básico con imagen</title>



<meta charset="UTF-8">

</head>

<body>

Ejemplo de enlace <br><br>

</body>

</html>

<a href="[http://www.mentor.mec.es](http://www.mentor.mec.es/)"><img src=”mentorhtml5.png”>

</a>

En este ejemplo se hace clic sobre la imagen mentorhtml5.png que se encuentra en la misma carpeta que el documento que tiene el enlace y nos abre la página del Proyecto Mentor (http://www.mentor.mec.es).

Por defecto, los enlaces pueden aparecer en diferentes colores según hayan sido utilizados

o vayan a serlo.

o Un enlace que no se haya pulsado, o sea, que no haya sido visitado, estará subrayado y de color azul.

* Un enlace pulsado, o sea, visitado, estará subrayado y de color púrpura.
* Un enlace activo o que acaba de ser pulsado estará subrayado y de color rojo.



El destino del enlace determina en qué ventana va a ser abierta la página enlazada y se especifica a través del parámetro *target*. Sus valores son

*Se pueden cambiar los anteriores colores a través de estilos CSS*

|  |  |
| --- | --- |
| Valor de target | Descripción |
| \_blank | Abre la página enlazada en una nueva ventana o pestaña del navegador |
| \_self | Abre la página enlazada en la misma ventana que la página que contiene el enlace (es la opción por defecto) |
| \_parent | Abre la página enlazada en el marco padre de la página que contiene el enlace |
| \_top | Abre la página enlazada en la ventana completa del navegador |
| Nombre de marco | Abre la página enlazada en un marco concreto |

Prueba en alguno de tus navegadores el siguiente ejemplo para que veas cómo funciona la visualización de enlaces. Los valores de *target* referidos a marcos no los vamos a ver porque los marcos están en desuso.

|  |
| --- |
| !DOCTYPE html>  <html>  <head>  <title>Enlace básico</title>  <meta charset=”UTF-8”>  </head>  <body>  Ejemplo de enlace <br><br>  <a href=”[http://www.mentor.mec.es](http://www.mentor.mec.es/)” target=”\_self””>Enlace a la página del proyecto Mentor</a> <br>  <a href=”http://”v-espino.com” target=”\_blank”> Enlace a la página del  IES Virgen del Espino</a><br> |

|  |
| --- |
| <a href=”[http://www.w3c.es](http://www.w3c.es/)” target=”\_top”>Enlace a la página de la Del consorcio w3c</a>  </body>  </html> |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | *Una URL siempre apunta a un archivo en un servidor. Si se omite el* |  |
| *nombre del fichero normalmente el archivo al que se refiere se llama* |  |
| *index.html, index.htm o index.php entre otros (depende de la configuración* |  |
| *del servidor donde esté alojado dicho fichero)* | | |

#### - Enlace externo.

Es el enlace que va a una página u objeto que se encuentra fuera del propio servidor o equipo donde está la página que lo referencia.

#### Ejemplo:

<a href=”[http://www.mentor.mec.es](http://www.mentor.mec.es/)”> Enlace a la página del proyecto mentor</a>

Este tipo de enlaces no tienen que ser siempre a páginas web sino que pueden servir para abrir clientes como los de correo (en este caso servirían para enviar un e-mail) o referencias a archivos para poderlos descargar.

#### Ejemplos:

<a href=”<mailto:uno@gmail.com>”>Manda un e-mail a esa dirección de correo”</a>

EEn este caso se abrirían el cliente de correo que tenemos configurado por defecto (Outlook express, kmail, etc.) con la opción de mensaje nuevo y en el destinatario la dirección que tenemos en el enlace.

< a href=”archivo.zip”>Descargar este archivo”</a>

En este ejemplo al pulsar “Descargar este archivo” se inicia el proceso de descarga del archivo.zip desde el servidor a nuestro equipo

<a href=”ftp://ftp.v-espino.com/directorio/archivo.zip”>

Descarga archivo por ftp</a>

En este ejemplo al pulsar “Descarga archivo por ftp” se inicia el proceso de descarga anónima de archivo.zip desde el servidor ftp que está corriendo en el equipo ftp.v- espino.com

#### - Enlace interno.

Estos enlaces nos llevan a una parte determinada de un documento donde hay definida un ancla o marcador por lo que la URL se sustituye por el nombre del ancla o marcador precedida por el símbolo “#”.

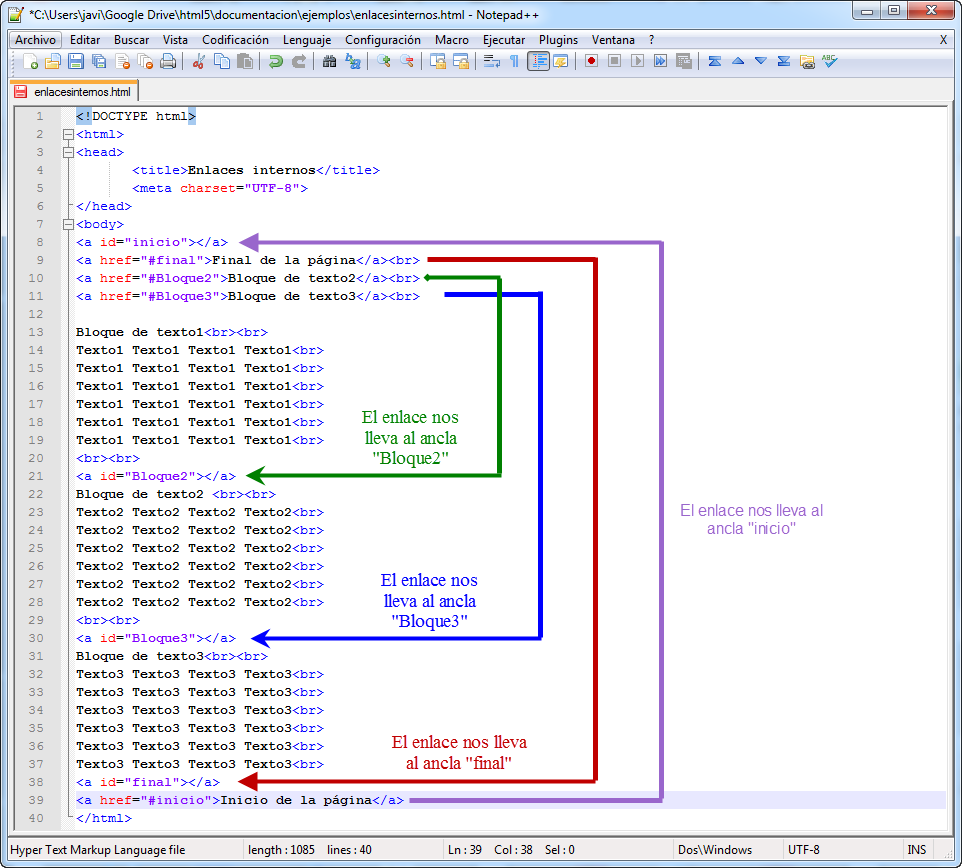
Se utilizan cuando un documento html es tan grande que no se puede visualizar toda la información por lo que podemos marcar una posición dentro de dicho documento y acceder a ella a través de un enlace.

#### Ejemplo:

<a href=”#nombre del ancla”>Ir al marcador </a>

El ancla se coloca en el lugar donde queremos que vaya el enlace con la etiqueta <a id=”nombre del ancla”></a>

#### Ejemplo completo:



También se pueden crear enlaces internos que apunten a partes de otros documentos por lo que un enlace de este tipo abriría el archivo del enlace y nos situaría en el lugar donde esté definida el ancla.

#### Ejemplo:

En este ejemplo suponemos que esta línea está dentro de un documento html que no es [http://www.mentor.mec.es/index.html,](http://www.mentor.mec.es/index.html) por lo que, cuando se pulsa el enlace, se abre la página anterior y se posiciona en el ancla “ejercicios” que hay en ese archivo

<a href=”<http://www.mentor.mec.es/index.html#ejercicios>”> Enlace a la parte ejercicios de la página index.html</a>

#### Imágenes

Las imágenes son uno de los elementos que más se emplean en las páginas web debido a que les da un mejor aspecto que si fueran solo texto. Existen dos tipos de imágenes:

* De contenido: Proporcionan información y complementan el texto de una página.

Se incluyen directamente en la página con la etiqueta <img>

* Estéticas: Se emplean para mejorar el aspecto estético de la página y suelen insertarse a través de atributos de etiquetas u hojas de estilos CSS. Un ejemplo sería el fondo de un documento html.



*Las imágenes mejoran la calidad de una página web pero hay que tener cuidado con los aspectos estéticos a la hora de colocarlas en nuestros documentos html.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <img> | | | | | | |
| Descripción | Inserta una imagen | | | | | | |
| Elemento | In line | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios | Alt. Height, crossorigin, src,ismap,usermap,width | | | | | | |
| Diferencia entre html  4.01 y html5 | Los atributos align, border, hspace, longdesc and vspace no están soportados en HTML5 pero es de los pocos elementos en el que se pueden usar los atributos height y witdh sin tener que utilizar estilos CSS | | | | | | |

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| [alt](http://www.w3schools.com/tags/att_img_alt.asp) | texto | Especifica el texto alternativo a la imagen si no se visualiza correctamente |
| crossorigin | anonymous  use-credentials | Permite el uso de imágenes de sitios de terceros en <canvas> |
| [height](http://www.w3schools.com/tags/att_img_height.asp) | pixels | Especifica la altura de la imagen |
| [ismap](http://www.w3schools.com/tags/att_img_ismap.asp) | ismap | Especifica que es un mapa de imagen |

|  |  |  |
| --- | --- | --- |
|  |  | gestionada por el servidor e implica que <img> debe estar dentro de un enlace |
| [src](http://www.w3schools.com/tags/att_img_src.asp) | URL | Especifica la URL de la imagen |
| [usemap](http://www.w3schools.com/tags/att_img_usemap.asp) | #nombremapa | Especifica que la imagen es un mapa del lado cliente |
| [width](http://www.w3schools.com/tags/att_img_width.asp) | pixels | Especifica el ancho de la imagen |

  Nuevo en HTML5

Los únicos atributos obligatorios son src y alt aunque es muy recomendable especificar el tamaño de la imagen a visualizar bien por estilos o bien por los atributos width y height para poder organizar mejor los contenidos de la página web.

<img> es de los pocos elementos HTML5 en la que existe la posibilidad de manipular el ancho y la altura de la imagen con los atributos width y height ya que, en las demás, esto solo se hace con estilos CSS.



El formato general y más básico que tiene un elemento <img> es el siguiente:

<img> es una etiqueta vacía, es decir, no tiene etiqueta de cierre

<img src=”URL” alt=”Texto alternativo”>

Donde URL cumple todo lo visto en el tema de enlaces e identifica y localiza un archivo imagen.

HTML no impone condiciones sobre el formato gráfico que se puede emplear en las imágenes, por lo que <img> puede incluir cualquier tipo de imagen. No obstante, hay que tener en cuenta que depende de los navegadores el que se pueda mostrar o no una imagen por lo que se recomienda usar los formatos gráficos GIF, JPEG y PNG para que no haya problemas con su visualización.

Hay que tener cuidado con el tamaño real de la imagen que se va a insertar en un documento HTML ya que si dicha imagen es muy grande ralentizará la carga de la página en el navegador



#### Ejemplo:

La unidad de tamaño de una imagen es el pixel o punto de color del monitor.

<!DOCTYPE html>

<img

src=”mentorhtml5.png”

alt=“imagen”

width=”80” height=”50”>

<img

src=”mentorhtml5.png”alt=”imagen” style=”witdh=80px; height=50px;”>

<html lang=”es-es”>

<head>

<title>Imagen</title>

<meta charset="UTF-8">

</head>

<body>

<!DOCTYPE html>

<html lang=”es-es”>

<head>

<title>Imagen</title>

<meta charset="UTF-8">

</head>

<body>

</body>

</html>

</body>

</html>



En este ejemplo, usemos la versión con estilos o la que no los tiene el resultado es el mismo, es decir, acorta la visualización de la imagen a un ancho de 80 pixels y a una altura de 50 pixels. Esta forma de visualizar una imagen nunca modifica el tamaño real de la imagen que ocupará lo mismo independientemente de los valores de ancho y largo.

Permite visualizar un documento HTML dentro de un área de otro documento HTML.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <iframe> | | | | | | |
| Descripción | Visualiza un documento HTML dentro de un área de otro documento HTML | | | | | | |
| Elemento | En línea | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios | name, sandbox,seamless, src, srcdoc, width y height | | | | | | |
| Diferencia | Los atributos align, franeborder, longdesc, marginheight, marginwidth y | | | | | | |

|  |  |
| --- | --- |
| entre html  4.01 y html5 | scrolling no están soportados en HTML5 pero es de los pocos elementos en el que se pueden usar los atributos height y witdh sin tener que utilizar estilos CSS |

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| height | pixels | Especifica la altura del elemento |
| name | texto | Especifica el nombre de un marco |
| sandox | “”  Allow-forms | Inidca un conjunto de restricciones extras para el contenido del marco |
|  | Allow-same-origin |  |
|  | Allow-scripts |  |
|  | Allow-top-navigation |  |
| seamless | seamless | Especifica si el contenido del marco debe parecer como una parte de la página contenedora sin bordes ni barras de scroll |
| src | URL | Especifica la URL de la página a visualizar en el marco |
| [src](http://www.w3schools.com/tags/att_img_src.asp)doc | Código html | Especifica código html para ser visualizado dentro del marco |
| [width](http://www.w3schools.com/tags/att_img_width.asp) | pixels | Especifica el ancho del marco |

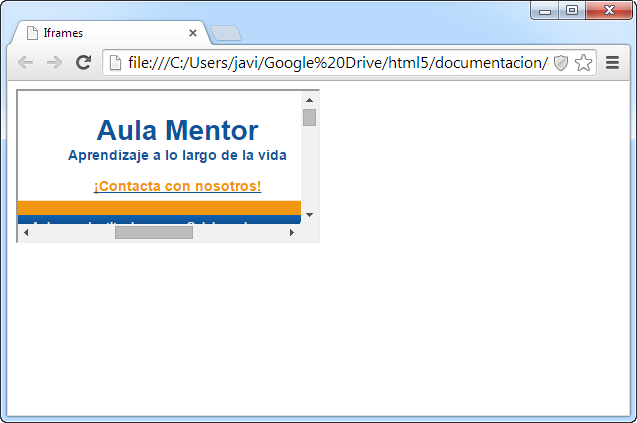
  Nuevo en HTML5

#### Ejemplo básico:

<!DOCTYPE html>

<html>

<head>



<title>Iframes</title>

</head>

<body>

<iframe src="[http://www.mentor.mec.es](http://www.mentor.mec.es/)">

<p>Tu navegador no soporta iframes.</p>

</iframe>

</body>

</html>

Como se ve en este ejemplo, se crea un espacio reservado para visualizar la página [http://www.mentor.mec.es](http://www.mentor.mec.es/) dentro de la que hemos creado. Al no poner ninguna dimensión al iframe, el navegador lo ha generado con ese ancho y largo. También hay que fijarse en el texto (“Tu navegador no soprota iframes”) que se visualizará si el navegador no soporta el elemento <iframe> y que se escribe entre la etiqueta de apertura y la cierre.

En base al ejemplo anterior vemos que los iframe nos permiten incrustar una página web dentro de otra lo que implica que todos los elementos de esa página (enlaces, imágenes, etc.) se verán dentro del marco creado para ello.

No hay que confundir **iframe** con **frame** o marco que se empleaban en HTML4 (en HTML5 no se usan) ya que, aunque pueda parecer que hacen lo mismo, iframe es un elemento normal que no necesita una estructura especial como había que hacer con los frame.

#### Iframes

Permite visualizar un documento HTML dentro de un área de otro documento HTML.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etiqueta | <iframe> | | | | | | |
| Descripción | Visualiza un documento HTML dentro de un área de otro documento HTML | | | | | | |
| Elemento | Inline | | | | | | |
| Navegadores que la  soportan |  |  |  |  |  |  |  |
| Chrome | Firefox | IExplorer | Ópera | Safari |  |
| Atributos | Globales y eventos | | | | | | |
| Atributos propios | name, sandbox,seamless, src, srcdoc, width y height | | | | | | |
| Diferencia entre html  4.01 y html5 | Los atributos align, franeborder, longdesc, marginheight, marginwidth y scrolling no están soportados en HTML5 pero es de los pocos elementos en el que se pueden usar los atributos height y witdh sin tener que utilizar estilos CSS | | | | | | |

|  |  |  |
| --- | --- | --- |
| Atributos propios | Valor | Descripción |
| height | pixels | Especifica la altura del elemento |
| name | texto | Especifica el nombre de un marco |
| sandox | “”  Allow-forms | Inidca un conjunto de restricciones extras para el contenido del marco |
|  | Allow-same-origin |  |
|  | Allow-scripts |  |
|  | Allow-top-navigation |  |
| seamless | seamless | Especifica si el contenido del marco debe parecer como una parte de la página contenedora sin bordes ni barras de scroll |

|  |  |  |
| --- | --- | --- |
| src | URL | Especifica la URL de la página a visualizar en el marco |
| [src](http://www.w3schools.com/tags/att_img_src.asp)doc | Código html | Especifica código html para ser visualizado dentro del marco |
| [width](http://www.w3schools.com/tags/att_img_width.asp) | pixels | Especifica el ancho del marco |

  Nuevo en HTML5

#### Ejemplo básico:

<!DOCTYPE html>

<html>

<head>

<title>Iframes</title>

</head>

<body>

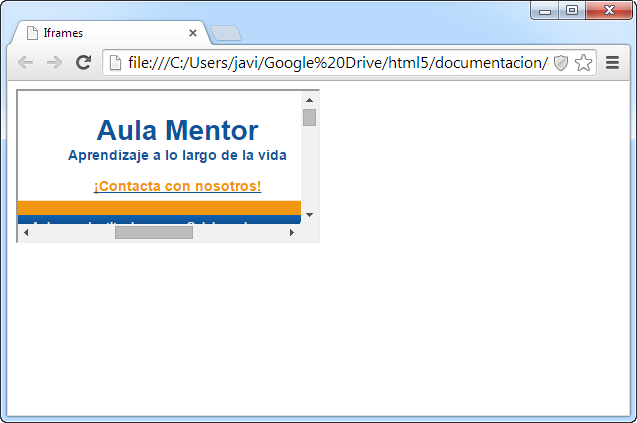
<iframe src[="http://w](http://www.mentor.mec.es/)ww[.mentor.mec.es](http://www.mentor.mec.es/)">

<p>Tu navegador no soporta iframes.</p>

</iframe>

</body>

</html>



Los frames (frame en inglés significa marco) son unas herramientas que han tenido una historia dilatada en el desarrollo de páginas web con HTML. De ser una etiqueta no estándar ha pasado a ser soportada por todos los navegadores y formar parte de las especificaciones de HTML. Por otra parte, el frame siempre ha sido una utilidad para hacer páginas web de uso, cuando menos, controvertido, puesto que tiene ciertas desventajas que muchas veces son más importantes que la indudable practicidad.

En cualquier caso, en DesarrolloWeb.com ya hemos hablado suficiente sobre la etiqueta Frame y hemos tratado ampliamente sus ventajas e inconvenientes. En este artículo vamos a hablar de una etiqueta "hermana" que es a día de hoy mucho más usada, porque resulta más útil y menos problemática que los propios frames. Se trata de iframe, un tag incluido en las especificaciones de HTML 4.0.

**Nota**: iframe viene de inline frame, pero en castellano a veces se les llama frames flotantes

Referencias sobre la etiqueta frame:

[Frames en HTML](http://www.desarrolloweb.com/articulos/791.php)

[Ventajas e inconvenientes del uso de frames](http://www.desarrolloweb.com/articulos/936.php) [Frames - Explicación básica](http://www.desarrolloweb.com/articulos/792.php)

[Control de frames en Javascript](http://www.desarrolloweb.com/articulos/1146.php) [Frames sin bordes](http://www.desarrolloweb.com/articulos/813.php)

[Frames - Atributos avanzados](http://www.desarrolloweb.com/articulos/812.php) [Actualizar dos frames con un solo enlace](http://www.desarrolloweb.com/articulos/1164.php)

[Crear variables o funciones globales a todos los frames](http://www.desarrolloweb.com/articulos/1199.php) [Frames - Una página en cada marco](http://www.desarrolloweb.com/articulos/794.php)

...

En concreto iframe sirve para crear un espacio dentro de la página donde se puede incrustar otra web. Es un cuadrado cuyas dimensiones debe especificar el desarrollador en la propia etiqueta iframe, que tiene asociada una página web que se carga en dicho espacio. Esa página web tendrá sus propios contenidos y estilos, independientes del contexto donde se está mostrando. Además será perfectamente funcional: si tiene enlaces se mostrarán en ese mismo espacio y si tiene scripts o aplicaciones dentro se ejecutarán también de manera autónoma en el espacio reservado al iframe.

#### EJEMPLOS DE USO DE IFRAME

Iframe se utiliza en muchos contextos. Dentro de un iframe podemos mostrar contenidos de otras páginas, como si estuvieran en la nuestra, por lo que sirven para ejemplos como:

* Códigos de banner, que se invocan por medio de un iframe pidiendo los datos del banner generalmente a un servidor de banners que puede estar en otra red.
* Visualizar contenidos de terceros, como bloques de noticias o novedades que ofrecen en otras webs.
* Interfaces de usuario, en el que ciertas actividades se realizan de forma autónoma y el procesamiento está en otra página web.

#### USO DE IFRAME FRENTE A FRAME

Actualmente la etiqueta iframe se utiliza más a menudo que la etiqueta frame, porque no da tantos problemas como esta. La diferencia principal está basada en que la etiqueta iframe no necesita una declaración de los espacios de los frames o frameset, porque se incrusta en el código HTML de la página. El iframe, por tanto, no provoca problemas de navegación, como los que ocurren con los frames normales si no se entra correctamente a través del frameset.

También, ya que no existe el frameset en los iframe, no adolece de los problemas del uso de frames, sobretodo a la hora en que la página es indexada en los motores de búsqueda.

Por decirlo de alguna manera, trabajar con iframe o frames flotantes es tan sencillo como hacer una tabla, que se codifica dentro de la maqueta HTML, con su espacio reservado

dentro de la página. Así, la única diferencia con respecto a una tabla es que el contenido del iframe se extrae de otra página web.

#### CONSTRUCCIÓN DE LA ETIQUETA IFRAME

Como decimos, el iframe se coloca directamente en el código HTML, en el lugar donde queremos que aparezca.

Se colocaría con un código como este:

<iframe src="pagina\_fuente.html" width=290 height=250>Texto para cuando el navegador no conoce la etiqueta iframe</iframe>

Como se ve, los atributos principales de iframe son la página web que se va a mostrar en el espacio y el ancho y alto del recuadro que reservemos para el frame flotante.

Como se puede ver, la etiqueta iframe tiene su correspondiente etiqueta de cierre. Todo el texto que coloquemos entre la etiqueta de inicio y la de cierre es texto alternativo, que sólo se mostrará en caso que el navegador del visitante no acepte la etiqueta iframe.

#### TODOS LOS ATRIBUTOS DE IFRAME

Estos serían los atributos disponibles para la etiqueta iframe:

**src:** Para indicar la página web que se mostrará en el espacio del frame flotante.

**width:** Para definir la anchura del recuadro del iframe.

**height:** Para definir la altura del iframe.

**name:** Para especificar el nombre del frame, que podemos utilizar luego para referirnos a él con el target de los links, o mediante javascript.

**id:** Para indicar el identificador del iframe, y poder referirnos a él desde javascript. **frameborder:** para definir si queremos o no que haya un borde en el frame. Los valores posibles son 0 | 1. frameborder=0 indicaría que no queremos borde y frameborder=1 que sí.

**scrolling:** indica si se quiere que aparezcan barras de desplazamiento para ver los contenidos del iframe completo, en el caso que no aparezcan en el espacio reservado para el iframe. Los valores posibles son: yes | no | auto. El valor "yes" es para que aparezcan siempre las barras de desplazamiento o barras de scroll, "no" sirve para que no aparezcan nunca y "auto" es para que aparezcan sólo cuando son necesarias (es el valor por defecto).

**marginwidth:** Para definir el margen a izquierda y derechar que debe tener la página que va dentro del iframe, con respecto al borde. Este margen va en pixels, pero prevalecerá el margen que pueda tener asignada la página web que mostremos en el frame flotante. **marginheight:** lo mismo que marginwidth, pero en este caso para el tamaño del margen por la parte de arriba y abajo.

**margin:** para especificar alineación del frame, igual que se especifica para las imágenes. **style y class:** los atributos para definir el aspecto del iframe por medio de hojas de estilo css.

Para acabar, aquí vemos otro ejemplo de iframe con unos cuantos atributos más:

<iframe name=miframeflotante src="colabora.htm" width=400 height=550 frameborder="0" scrolling=yes marginwidth=2 marginheight=4 align=left>Tu navegador no soporta frames!!</iframe>

ero aparecen nuevos problemas. Por un lado, un tema de seguridad, habría que controlar que nadie pase en el argumento page una dirección de un archivo que no debería poder ver. Para esto habrá que agregar unos ifs analizando el contenido. Por otro lado, absolutamente todas las páginas de contenido que quiera mostrar deberían pasar a través de este archivo. Entonces, si una página de contenido tenía un link a "descargas.html", ahora debe apuntar a "index.php?page=descargas.html". Nuevamente google me dió una buena solución. Resulta que php también tiene una par de funciones para que la salida del script vaya temporalmente a un buffer en lugar de ir directamente al navegador. Entonces, enviamos la salida del include al buffer, buscamos los links en el buffer, todo lo que empieza con href=", y los actualizamos reemplazando ese comienzo por href="index.php?page=. El result.

#### POR QUÉ PASAMOS DE MARCOS

Aunque por el título pueda parecer lo contrario, no tengo nada contra nadie que se llame Marcos. Hablamos, naturalmente, de los marcos de HTML, los famosos *frames*, uno de los más odiados y más amados inventos de Netscape.

Se sabía. Se veía venir. [Se lo dijeron.](http://www.nyct.net/~aray/htmlwg/95q3/781.html) Los marcos eran una mala idea. Pero aun así[,los](http://www.nyct.net/~aray/htmlwg/95q3/1083.html) [implementaron.](http://www.nyct.net/~aray/htmlwg/95q3/1083.html)

¿Y por qué es tan mala idea?, te estarás preguntando. He aquí el porqué.

#### LOS MARCOS VAN EN CONTRA DE LA PROPIA ESENCIA DEL HTML.

Es como si inventaras el tapón para mangueras.

El objetivo del lenguaje HTML es dotar a un texto de *estructura lógica*: párrafos, listas, citas, etc. En HTML, no dices **cómo** se representa cada cosa; únicamente **qué** es cada cosa. El cómo se representa depende del navegador que estés usando. En un navegador gráfico,

los títulos saldrán más grandes, los párrafos separados entre sí, saldrán marcadores en las listas, etc. Un sintetizador de voz hará pausas, inflexiones en la voz, utilizará sonidos auxiliares, etc. Un motor de búsqueda lo ignorará todo y se dedicará simplemente a indexar las palabras. Etc., etc.

Esto es el punto fuerte de HTML: es flexible, cualquier documento se adapta a cualquier modo de navegar. Cualquier aplicación puede obtener el documento y representarlo de la mejor manera posible.

Los marcos rompen con ese esquema y lo vuelven inservible. Los marcos no son información lógica, sino de presentación. No dice **qué** es cada cosa, sino **dónde** está cada cosa. Esto, obviamente, sólo tiene sentido para navegadores gráficos. Para todas las demás aplicaciones, en las que el concepto de "dónde" no tiene ningún significado (motores de búsqueda, sintetizadores de voz, etc.), el documento puede fácilmente tornarse inaccesible, o difícilmente accesible en el mejor de los casos.

En cualquier caso, introducen una componente de presentación en un lenguaje cuya misión es describir estructuras lógicas. Si lo que queremos es especificar el modo en que se representan nuestros documentos, debemos usar hojas de estilo, que son el complemento del HTML y la herramienta apropiada para ello.

Pero si esto no es razón suficiente para ti (lo cual me hace sospechar que no tienes muy claro por qué estás haciendo páginas para la Web), aquí tienes una ristra más de buenos motivos para evitar el uso de marcos.

#### NO SE PUEDE ENLAZAR A UNA COMBINACIÓN PARTICULAR DE MARCOS

Una de las primeras cosas que se aprende del diseño web, es que cada recurso tiene un "nombre propio" que lo identifica y lo distingue de todos los demás. Esto muy importante, porque la esencia de la Web es que todos los recursos que forman parte de ella se encuentran enlazados entre sí (y de ahí su nombre). Naturalmente, para que un recurso sea referenciado por otro, necesita algo que lo identifique. En la red, ese identificador es un *URI* (identificador uniforme de recursos).

Las páginas web suelen identificarse con un tipo especial de URI llamado *URL*(localizador uniforme de recursos), que especifica dónde está localizado dicho recurso. Es la cadena que aparece en la barra de direcciones (por ejemplo, el URL de este documento es <http://html.conclase.net/articulos/problemasmarcos>).

Pues bien, resulta que este esquema se rompe con las páginas con marcos.

Una página con marcos está compuesta de dos grupos de elementos: por un lado, el documento HTML que contiene la *definición* de los marcos (el elemento [FRAMESET](http://html.conclase.net/w3c/html401-es/present/frames.html#h-16.2.1)) junto con el estado inicial de la página, y por otro lado los documentos HTML que constituyen el *contenido* de cada marco, y que se almacenan por separado. El URL de la página con marcos es siempre el del documento que contiene el FRAMESET. Si comenzamos a navegar por el sitio, se van cargando páginas nuevas en cada marco, pero el URL es siempre el mismo. Si cualquier persona escribe ese URL en la barra de direcciones de su navegador, siempre llegará al estado inicial de la página web.

Eso quiere decir que si queremos hacer un enlace a una página particular... no podemos. Es decir, los marcos están impidiendo en gran medida el objetivo primario de la Web: enlazar unos documentos con otros. A lo sumo, lo más que podemos hacer es enlazar con el FRAMESET y dar instrucciones para avanzar desde allí.

Naturalmente, podríamos enlazar directamente con el documento HTML que nos interesa, pero como hemos quitado deliberadamente los elementos de navegación para ponerlos en un marco separado, llegaremos a una página sin salida, sin herramientas de navegación.

#### DAN PROBLEMAS CON LOS BUSCADORES

Actualmente casi todos los motores de búsqueda conocen la sintaxis de los marcos y son capaces de seguir e indexar todos los documentos enlazados desde los documentos que forman el estado inicial de la página. Sin embargo, imagina esta situación: alguien busca algo en un buscador y da con uno de tus documentos. No con el FRAMESET, sino con uno de los documentos que, en teoría, debería ir en un marco. El usuario pincha, llega al documento y... no hay marcos, no hay elementos de navegación.

#### SÓLO SE PUEDE CAMBIAR EL CONTENIDO DE UN MARCO AL MISMO TIEMPO

Imagina esta situación. Tenemos tres marcos. En el primero tenemos un índice general de capítulos. En el segundo tenemos un subíndice de secciones, y en el tercero los contenidos de la sección que queremos presentar. Al elegir un capítulo del índice de capítulos, queremos que cambien los contenidos del índice de secciones de ese capítulo, y que se muestren los contenidos correspondientes a la primera sección del capítulo elegido.

Pero eso no es posible, porque al elegir uno de los capítulos del índice general, sólo podemos cambiar el contenido de uno de los otros dos marcos.

Añaden más complejidad de la necesaria

Esto es especialmente cierto en páginas con más de dos marcos, sobre todo si queremos que cambien a la vez los contenidos de más de uno de ellos con los métodos de los que hablamos más adelante.

Hay que tener en cuenta que se tarda más en cargar dos documentos pequeños por separado que uno solo más grande que contuviera la suma de los dos (esto es así debido a la naturaleza del protocolo de red utilizado en Internet). Esto es especialmente sensible en lugares con conexiones muy lentas.

#### MÁS PROBLEMILLAS

* El [título](http://html.conclase.net/w3c/html401-es/struct/global.html#idx-title) deja de ser informativo. El título del documento es siempre el del documento que contiene el FRAMESET, y no el del documento individual que estamos leyendo.
* En general no es posible almacenar una combinación específica de marcos en la lista de favoritos. Algunos navegadores sí pueden, pero otros no.
* Los [identificadores de fragmento](http://html.conclase.net/w3c/html401-es/intro/intro.html#fragment-uri) dejan de ser útiles, porque no se puede apuntar hacia ellos, ya que si lo hacemos nos volvemos a quedar sin barra de navegación.
* En cuanto hay más de dos marcos, la navegación por la historia del navegador (con los botones *Atrás* y *Adelante*) se hace bastante confusa (por no decir inservible).

Y las soluciones no son precisamente buenas.

Más que soluciones son trucos, y a veces crean más problemas de los que resuelven.

El problema de cambiar los contenidos de más de un marco a la vez se suele resolver anidando FRAMESETs. En el [ejemplo de arriba,](http://html.conclase.net/articulos/problemasmarcos#t04) en vez de definir tres marcos definiríamos sólo dos. En el primero tendríamos el índice de capítulos y en el segundo cargaríamos un nuevo frameset con dos marcos, en uno de ellos cargaríamos el índice de secciones y en el segundo presentaríamos cada sección. Esto tiene la ventaja de que se puede apuntar a un capítulo en particular, aunque nos quedaríamos sin el índice de capítulos, y seguiríamos sin poder apuntar a una sección en particular o a un fragmento de ella. Además sigue sin resolver el problema de los motores de búsqueda y hará que la navegación por el sitio sea más lenta, ya que cada vez que cambiemos de capítulo habrá que cargar tres nuevos documentos en lugar de sólo dos.

Este método se puede extender al límite, y hacer que cada documento sea en realidad un FRAMESET, de modo que cada vez que queramos abrir un documento, la ventana se vacía (usando [target="\_top"](http://html.conclase.net/w3c/html401-es/types.html#type-frame-target)) y se vuelven a dibujar todos los marcos y a rellenarlos con los documentos apropiados. Ahora podemos cambiar todos los marcos que queramos al mismo tiempo, y tenemos más libertad para apuntar a diferentes combinaciones de marcos, pero a cambio la navegación se hace mucho más lenta, ya que cada vez que seguimos un vínculo hay que volver a cargar un FRAMESET y uno o normalmente más documentos en cada marco. Además, persiste el problema de los motores de búsqueda y el de los identificadores de fragmento. Y además uno de los motivos por el que mucha gente decide usar marcos es que un documento queda fijo en la pantalla sin necesidad de volver a cargarlo cada vez, cosa que con este método ya no ocurre.

Por último, existen scripts de JavaScript para resolver algunos problemas (por ejemplo, cargar automáticamente el FRAMESET cuando se accede directamente a uno de los documentos desde un motor de búsqueda). Sin embargo, uno de los principios del diseño web es que cualquier página debería funcionar, aunque el usuario no pueda o no quiera activar JavaScript en su navegador. Por tanto, estos métodos constituyen una solución parcial, no una solución de ningún problema.

#### En consecuencia:

A primera vista, los marcos son una buena idea. En el fondo, no dan más que problemas y no son tan necesarios como te podrías imaginar. Mi consejo es que, siempre que puedas, los evites.

Si de todos modos vas a usarlos o tienes que hacerlo, hazte un favor: no te unas al [club de](http://www.google.com/search?q=%22browser%2Bdoes%2Bnot%2Bsupport%2Bframes%22) [los 500.000 maleducados.](http://www.google.com/search?q=%22browser%2Bdoes%2Bnot%2Bsupport%2Bframes%22) Está muy bien que utilices el elemento [NOFRAMES](http://html.conclase.net/w3c/html401-es/present/frames.html#h-16.4) para los navegadores que no soportan marcos, pero es absurdo utilizarlo para decir Su navegador no acepta marcos. Y es una auténtica falta de respeto decir Actualice su navegador, o Quizá sea un buen momento para modernizarse cuando no sabes si la persona que está leyendo tu mensaje puede realmente actualizar su navegador o no. Incluso podría estar usando un lector de pantalla carísimo para utilizar con un navegador de texto. La verdad es que Lo siento, no soporto su navegador está más cercano a la realidad. Pero en todo caso, si vas a poner algo en el elemento NOFRAMES, que sea información. Recuerda que todo esto va de transmitir información. Pon un vínculo a una página, o a varias. Sé amable con todos tus visitantes y no despidas a algunos de ellos con un mensaje grosero sólo porque no navegan como tú. Y piensa en ir diseñando una nueva versión de tu página... sin marcos.

#### Más información:

Este artículo se basa en el [tutorial sobre marcos](http://www.webreference.com/html/tutorial15/) de Stephanos Piperoglou en[webreference.com.](http://www.webreference.com/) En [allmyfaqs.com](http://www.allmyfaqs.com/) puedes encontrar [muchos más documentos](http://www.allmyfaqs.com/cgi-bin/wiki.pl?Problems_with_using_frames) con

información sobre los problemas de los marcos, algunos más modernos que otros, pero todos ellos con información interesante.