

# HTML



# Transmisión de la información

## Cientes y Servidores

En internet la información se transmite mediante páginas HTML desde un ordenador Servidor hacia nuestro ordenador Cliente



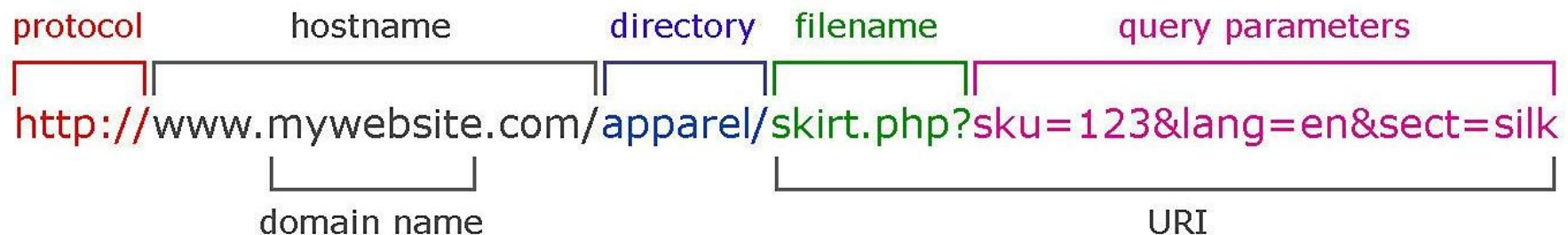
## **PARA QUE NUESTRAS PÁGINAS WEB ESTÉN DISPONIBLES EN INTERNET SON NECESARIOS VARIOS REQUISITOS:**

- Que se encuentren en un ordenador permanentemente conectado a Internet.
- Que dispongamos de una dirección IP fija y conocida por los ordenadores que se encargan de traducir las peticiones de los internautas a direcciones IP.
- Que el ordenador en el que se alojan nuestras páginas disponga del software adecuado para recibir conexiones exteriores y sea capaz de poner a disposición de los visitantes aquello que solicitan: esa aplicación es Apache o Nginx o IIS
- ...lo que necesitamos es un Servidor WEB (llamamos así al ordenador que cumple los requisitos anteriores).

## ¿CÓMO FUNCIONA EL PAR CLIENTE-SERVIDOR?

1. Un internauta solicita una determinada información tecleando el **URL** de la página o haciendo clic en un enlace.
2. El proveedor de acceso del internauta consulta a su servidor de nombres de dominio (**DNS**) y traduce la petición recibida del navegador a **la dirección IP del servidor** correspondiente y se pone en contacto con dicho ordenador.
3. El servidor WEB recibe la petición y si la información solicitada existe y está disponible la envía al ordenador que le ha realizado la petición.
4. El navegador del internauta recibe todos los recursos relacionados con la página solicitada y los organiza para presentarlos en la pantalla según las instrucciones que contiene el código HTML de la página.
5. Dichos recursos están formados por el texto y los elementos multimedia que componen la página: imágenes fijas, sonidos, imágenes en movimiento, pequeños programas.

- **Dirección IP v.4:** Números que determinan de forma única a cada uno de los equipos conectados a internet. Ej.: **64.235.18.129** (se comporta como el DNI de una persona)
- **Servidor de DNS:** es un equipo conectado a Internet que se encarga de traducir las direcciones url en las direcciones ip del ordenador al que se refiere.
- **URL** (Uniform Resource Locator): es el método que tenemos para identificar cualquier recurso (página, imagen,...) que hay en internet. Consta de varias partes



# ¿Qué es HTML?

- **HTML (Hyper Text Markup Language)** es un lenguaje de marcas de hipertexto.
- Permite escribir texto de forma estructurada, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, imágenes, sonidos, videos, etc...).
- Los archivos HTML por lo general, **tienen la extensión .html o .htm**
- Desarrollado por el World Wide Web Consortium (**W3C**) .

# Etiquetas y atributos

- Las páginas web se construyen encerrando los distintos elementos que las componen (títulos, párrafos, listas , artículos, secciones,...) entre etiquetas.
- Las etiquetas se caracterizan por estar encerradas entre los símbolos `<` `>`
- Cada elemento de la página tiene que venir escrito entre una **etiqueta de apertura** y una de **cierre** (esta última se distingue de la primera en que lleva una barra inclinada `< / >`)

`<p>` Este sería un párrafo de una página `</p>`

- HTML define las siguientes etiquetas para *marcar* los diferentes elementos que componen una página:

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

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



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

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

Los elementos **en línea** definidos por HTML son:

a, abbr, acronym, b, basefont, bdo, big, br, cite, code, dfn, em, font, i, img, input, kbd, label, q, s, samp, select, small, span, strike, strong, sub, sup, textarea, tt, u, var.

Los elementos de **bloque** definidos por HTML son:

address, blockquote, center, dir, div, dl, fieldset, form, h1, h2, h3, h4, h5, h6, hr, isindex, menu, noframes, noscript, ol, p, pre, table, ul.

(Se consideran también bloque: dd, dt, frame-set, li, tbody, td, tfoot, th, thead, tr.)

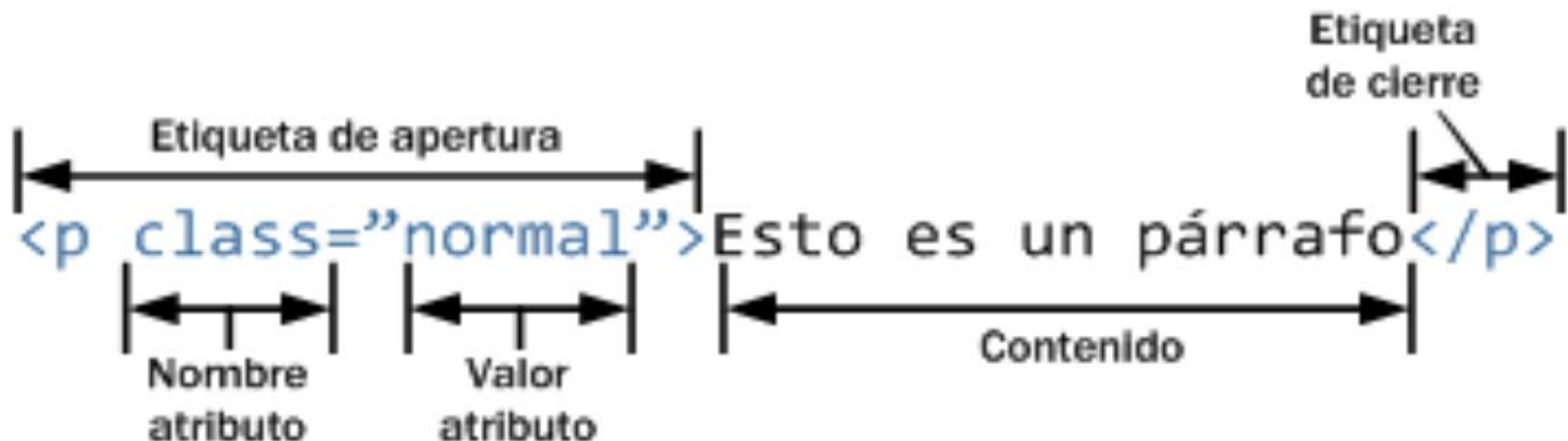
Los siguientes elementos pueden ser en línea y de bloque según las circunstancias: button, del, iframe, ins, map, object, script.

- Las etiquetas pueden incluir cero, uno o varios atributos que siempre siguen esta sintaxis:

`<a href="contenido1" target="contenido2"> Ir a wikipedia </a>`

Los atributos tienen un nombre (ej. `href`) y van seguidos de un símbolo igual ( `=` ) y a continuación de un valor entre comillas (`"contenido1"`)

Los atributos irán separados entre ellos y del nombre de la etiqueta por un espacio en blanco.



- Los atributos **sólo se ponen dentro de la etiqueta de apertura**  
(no en la de cierre)
- Cada etiqueta tiene sus propios atributos.

# Versión estricta del lenguaje: XHTML

A continuación se muestran las cinco restricciones básicas que introduce XHTML respecto a HTML en la sintaxis de sus etiquetas

1) Las etiquetas se tienen que cerrar de acuerdo al orden en el que se abren:

Ejemplo correcto en XHTML:

`<p>Este es un párrafo con <a>un enlace</a></p>`

Ejemplo incorrecto en XHTML (pero correcto en HTML):

`<p>Este es un párrafo con <a>un enlace</p></a>`

## 2) Los nombres de las etiquetas y atributos siempre se escriben en minúsculas:

Ejemplo correcto en XHTML:

```
<p>Este es un párrafo con <a href="http://www.google.com">un enlace</a></p>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<P>Este es un párrafo con <A HREF="http://www.google.com">un enlace</A></P>
```

## 3) El valor de los atributos siempre se encierra con comillas:

Ejemplo correcto en XHTML:

```
<p>Este es un párrafo con <a href="http://www.google.com">un enlace</a></p>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<p>Este es un párrafo con <a href=http://www.google.com>un enlace</a></p>
```

#### 4) Los atributos no se pueden comprimir:

Ejemplo correcto en XHTML:

```
<dl compact="compact">...</dl>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<dl compact>...</dl>
```

Este tipo de atributos en los que el nombre coincide con su valor no son muy habitual es.

## 5) Todas las etiquetas deben cerrarse siempre:

La mayoría de etiquetas HTML encierran un contenido de texto entre la etiqueta de apertura y la etiqueta de cierre. Sin embargo, algunas etiquetas especiales llamadas "**etiquetas vacías**" no necesitan encerrar ningún texto.

La etiqueta `<br>` por ejemplo, se utiliza para indicar el comienzo de una nueva línea, y nunca encierra ningún contenido de texto. Como el estándar XHTML obliga a cerrar todas las etiquetas abiertas, siempre que se incluya la etiqueta `<br>` se debería cerrar de forma seguida: `<br></br>`.

Para que el código resulte más cómodo de escribir, XHTML permite en estos casos escribir de forma abreviada una etiqueta que se abre y se cierra de forma consecutiva: `<br/>` para indicar que es una etiqueta vacía que se abre y se cierra en ese mismo punto.

Ejemplo correcto en XHTML:

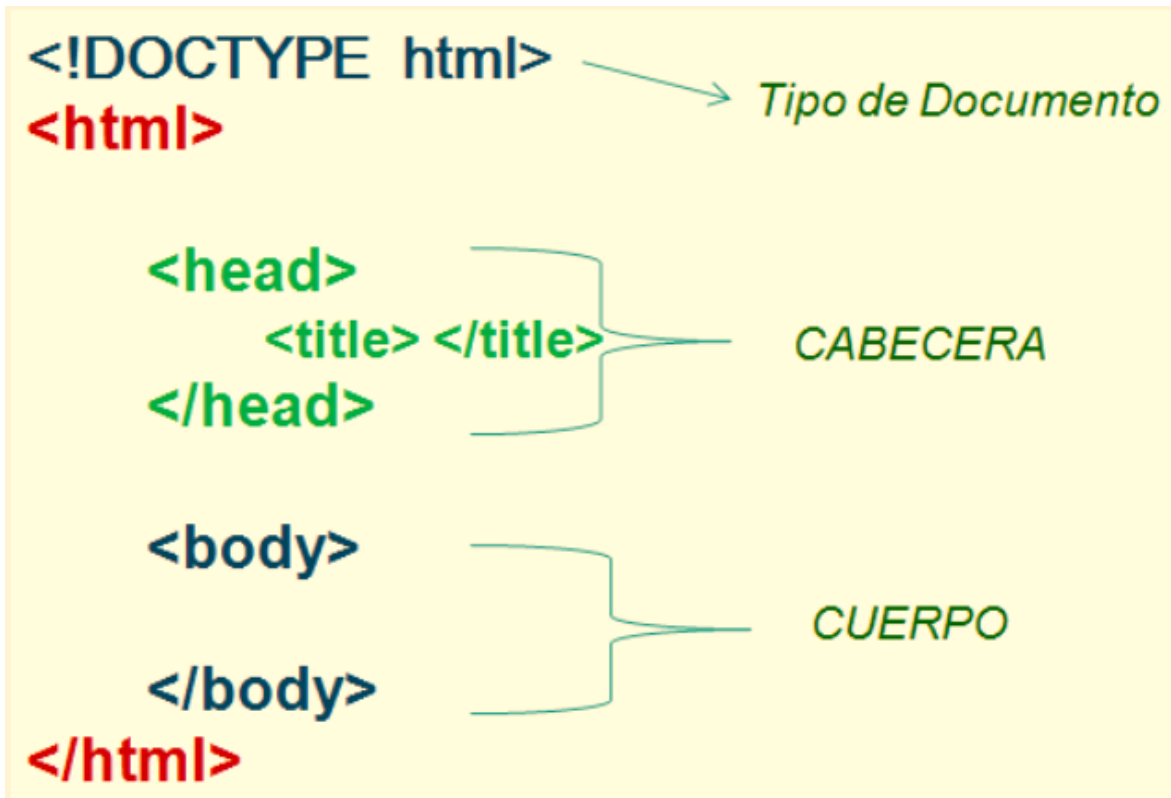
`<br/>`

Ejemplo incorrecto en XHTML (pero correcto en HTML):

`<br>`



# Estructura global de una página web



Una página web consta de dos partes:

- **Cabecera** (elemento `<head>`): parte no visible de la página pero con información relevante para los motores de búsqueda y el navegador.
- **Cuerpo** (`<body>`): parte de la página visible en el navegador.

## TODO DOCUMENTO HTML5 DEBE COMENZAR CON

**<!DOCTYPE html>**

- Esta línea debe ser la primera línea del archivo, sin espacios o líneas que la precedan e **indica el tipo de documento** que estamos creando para luego, con un validador de código, comprobar que se ajusta al DTD correspondiente.
- La declaración anterior indica que se trata de un documento que usa **HTML versión 5**

- Luego de declarar el tipo de documento, debemos comenzar a construir la estructura HTML. La estructura tipo árbol de este lenguaje tiene su raíz en el elemento **<html>**

Este elemento envolverá al resto del código:

```
<html lang="es">Aquí dentro va todo el contenido </html>
```

El atributo *lang* en la etiqueta de apertura **<html>** es el único atributo que necesitamos especificar en HTML5.

Este atributo define el idioma humano del contenido del documento que estamos creando, en este caso **es por español**.

# Elementos de la cabecera <head>

La cabecera contiene información de identificación y control que en general no se muestra en la ventana del navegador, aunque la pueden utilizar los motores de búsqueda. Puede incluir una o varias etiquetas:

- **<title> ...</title>** Etiqueta del título: es obligatoria y debe incluirse en todas las páginas web. Contiene el texto que se muestra en la pestaña del navegador.
- Etiquetas **<meta>** pensadas para proporcionar información sobre el documento a los programas que analicen la página y por ello existen muchas etiquetas <meta> diferentes

**<meta charset="utf-8">** especifica el conjunto de caracteres que se utilizará para presentar el texto en pantalla

Las siguientes etiquetas meta son importantes para motores de búsqueda y obtener un sumario de la información que contiene

**<meta name="description" content="Ejemplo de HTML5">**

**<meta name="keywords" content="HTML5, CSS3, Javascript">**

**<meta name="author" content="José María">**

Otra etiqueta meta interesante es la que se utiliza para mejorar la visualización de la página en las pantallas de los teléfonos móviles:

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**



Las etiquetas <meta> no tienen etiqueta de cierre, pero es conveniente cerrar todas las etiquetas. Por tanto lo anterior debemos escribirlo así:

```
<meta charset="utf-8" />
```

```
<meta name="description" content="Ejemplo de HTML5" />
```

```
<meta name="keywords" content="HTML5, CSS3, JavaScript" />
```

# Estructura global <body>

- La estructura del cuerpo (el código entre las etiquetas <body>) **generará la parte visible del documento**. Este es el código que producirá nuestra página web
- HTML se debe dedicar exclusivamente a estructurar la información dentro del cuerpo de un documento, y no debe encargarse para nada del cómo se va a ver la página.
- Los principales elementos que permiten organizar la información dentro de la página son:
  -  <p> para marcar los párrafos
  -  <h1> ... <h6> para los distintos niveles de títulos

✚ **<table>** Las tablas permitían a los diseñadores acomodar datos, texto, imágenes y herramientas dentro de filas y columnas de celdas.

✚ **<div>** Marca un bloque dentro de la página. El uso de **<div>** para marcar las distintas partes de una página se volvió una práctica común en el html4. Desde imágenes a menús, textos, enlaces, códigos, formularios, cualquier cosa puede ir entre las etiquetas de apertura y cierre de un elemento **<div>**. En otras palabras, la palabra clave div solo especifica una división en el cuerpo, como la celda de una tabla, pero no ofrece indicio alguno sobre qué clase de división es, cuál es su propósito o qué contiene.

✚ **<span>** para marcar trozos dentro de una línea (inline)



# Organización del cuerpo de la página

Aquí vemos un diseño común de la mayoría de los sitios webs actuales.



- En la parte superior, **Cabecera**, se encuentra el espacio donde usualmente se ubica el logo, título, subtítulos y una corta descripción del sitio web o la página.
- Inmediatamente debajo, podemos ver la **Barra de Navegación** donde de ofrece un menú o lista de enlaces con el propósito de facilitar la navegación a

través del sitio. Los usuarios son guiados desde esta barra hacia las diferentes páginas o documentos, normalmente pertenecientes al mismo sitio web.

- El contenido más relevante de una página web se encuentra ubicado en su centro. El contenido presentado en esta parte del diseño es usualmente de alta prioridad. En el diseño de ejemplo, **Información Principal** podría contener una lista de artículos, descripción de productos, entradas de un blog o cualquier otra información importante. **La Barra Lateral** podría mostrar una lista de enlaces apuntando hacia cada uno de esos elementos. En un blog, por ejemplo, esta última columna ofrecerá una lista de enlaces apuntando a cada entrada del blog, información acerca del autor, etc...

- En la base de un diseño web clásico siempre nos encontramos con una barra más que aquí llamamos **Institucional**. La nombramos de esta manera porque esta es el área en donde normalmente se muestra información acerca del sitio web, el autor o la empresa.



En el formato antiguo (html4), para organizar la página de esa forma, se marcaban las distintas partes de la siguiente forma:

# HTML4

`<div id="header">`

`<div id="nav">`

`<div id="main">`

`<div id="aside">`

`<div id="article">`

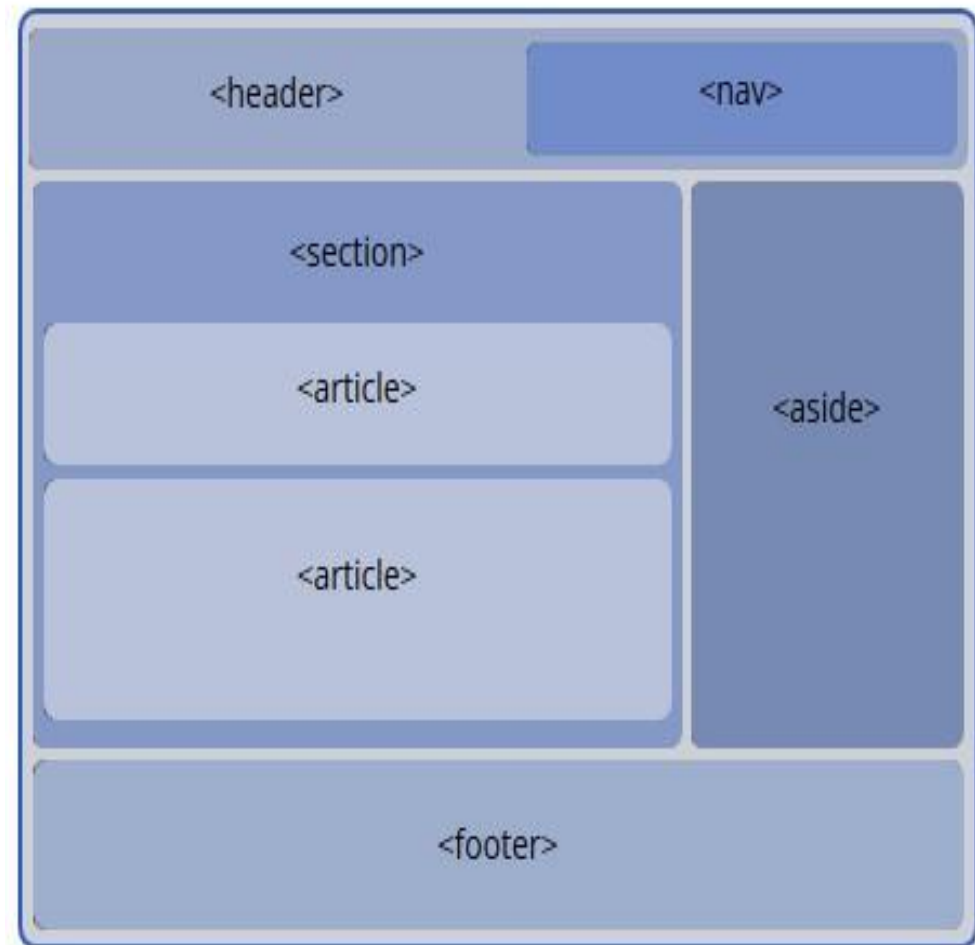
`<div id="article">`

`<div id="footer">`

# Mejoras en la estructura con HTML5

Se introducen nuevas etiquetas para marcar las distintas partes de la página

- **<header>** Es el equivalente a la **cabecera** de la página web. Contiene el título o nombre de la empresa/titular de la página, logo e información relacionada.
- **<nav>** Contiene los enlaces (barra de navegación) externos o internos de la página.
- **<section>** Es una gran caja



que sirve para mostrar grandes bloques de contenido de la página. Puede contener diferentes subapartados de diferentes temas (de tipo **<article>**).

- **<article>** Es una caja independiente de contenido que puede estar contenida (o no) dentro de un **<section>**. Normalmente utilizada para contenidos no demasiado extensos.
- **<aside>** Define un bloque de contenido relacionado de manera indirecta con el contenido principal, pero que no es esencial para la comprensión del mismo.
- **<footer>** Equivale al pie de página de un apartado concreto (**<section>**, **<article>**...) o de la página web en general.



- article



- aside



- header



- nav



- section

