

Ejemplo:

```
<meta http-equiv="pics-Label" content='(pics-1.1
"http://www.icra.org/pics/vocabularyv03/" 1 gen true for
"http://juanmacr.es" r (n 0 s 0 v 0 l 0 oa 0 ob 0 oc 0 od 0 oe
0 of 0 og 0 oh 0 c 1))'>
```

Es recomendable su uso, sobretodo cuando el contenido no es apto para menores.

Para más información consultar: <http://www.fosi.org/icra/>

- **Manejo de la caché:**

La memoria caché sirve para almacenar la respuesta que envía el servidor, ante una petición del cliente como páginas HTML, imágenes o archivos. De esta forma cuando el cliente solicita de nuevo la misma URI, la caché devuelve la respuesta almacenada en lugar de hacer otra petición al servidor.

El objetivo es mejorar el tiempo de respuesta y reducir el tráfico de red.

Esta memoria puede ubicarse en el navegador del cliente, en servidores Proxy directos como en nuestro ISP (Internet Service Provider - Proveedor de Servicios de Internet) o en servidores Proxy inversos (que sirven de respaldo a sitios web con muchas consultas).

- **cache-control**

Sirve para controlar el cacheo que va a recibir el documento.

Se admiten las siguientes opciones en **content**:

- **max-age** = *nº segundos*

Indica el número de segundos durante los que el contenido es considerado como válido.

- **s-maxage** = *nº segundos*

Similar a la directiva max-age, pero aplicable solo para cachés compartidas.

- **public**

Indica que el documento puede ser guardado por proxies intermedios.

- **private**

Indica que el archivo puede ser almacenado por el navegador del usuario, pero no por proxies intermedios (la página está personalizada para cada usuario).

- **no-cache**

Significa que no se debe consultar la caché, por tanto, obliga al navegador a realizar una petición al servidor.

- **no-store**

Indica al navegador que no guarde el documento en caché después de visualizarlo.

- **must-revalidate**

Comunica a las cachés que deben cumplir todas nuestras reglas de cacheo.

- **proxy-revalidate**

Similar a la anterior, para servidores Proxy.

Ejemplos:

```
<meta http-equiv="cache-control" content="max-age=3600">
```

```
<meta http-equiv="cache-control" content="no-cache">
```

- **pragma**

Sirve para evitar que la página sea consultada en memoria caché, por tanto, el navegador debe solicitar la página al servidor cada vez que queremos visualizarla.

Ejemplo:

```
<meta http-equiv="pragma" content="no-cache">
```

Equivale a cache-control="no-cache", pero se utiliza por si hay servidores proxy intermedios con protocolo http 1.0.

- **last modified**

Con esta etiqueta el servidor comprueba si la respuesta que el navegador tiene almacenada en caché es válida. El desarrollador web debe modificar la fecha cada vez que actualice la página.

Ejemplo:

```
<meta http-equiv="last-modified"
      content="Mon, 12 Nov 2012 11:35:00 GMT">
```

El navegador envía la fecha "last-modified" junto con la petición de página al servidor web, y éste comprueba si la página está obsoleta o es válida.

Si está obsoleta, envía la página al cliente, pero si es válida envía un mensaje "not modified", entonces el navegador sirve la página que tiene en caché.

- **expires**

Se utiliza para indicar en qué fecha expira la página.

Ejemplo:

```
<meta name="expires" content="Mon, 29 Oct 2012">
```

El formato de fecha según RFC 1123: "día de la semana con tres letras, día del mes con tres letras, año con 4 dígitos, hh:mm:ss GMT".

A veces se utiliza con el valor `content="0"` para evitar que el navegador consulte la caché, pero esto se debe hacer con `cache-control`.

2.4.3.3. El protocolo Open Graph

Es un protocolo que permite identificar los elementos que contiene nuestra página web dentro de una red social. Por ejemplo, para permitir que nuestra web sea conocida en facebook®, y entonces disponga de las mismas ventajas que el resto de páginas.

Podemos incluir un botón “Me gusta” para enviar el contenido de las metaetiquetas. Una vez enviadas los recursos de la página estarán accesibles desde facebook.

Para empezar hay que incluir los siguientes atributos en html:

```
xmlns:fb="http://www.facebook.com/2008/fbml"
```

```
xmlns:og="http://ogp.me/ns#"
```

Después las etiquetas meta con la sintaxis:

```
<meta property="og:nombre" content="valor" />
```

- Las propiedades básicas son:

og:title = El título tal y como aparecerá en la red.

og:type = El tipo de objeto, por ejemplo "article".

og:image = La URL de una imagen representativa del objeto.

og:url – La URL canónica del objeto que será su identificador en la red, por ejemplo, "http://www.mipagina.es/facebook/1618033988".

- Algunas propiedades opcionales:

og:audio = La URI a un fichero de audio.

og:description = La descripción con un par de frases.

og:locality = La localidad del sitio.

og:site_name = Nombre del sitio.

og:video = La URI a un fichero de video.

fb:admins = El identificador del administrador del sitio en Facebook.

Ejemplos:

```
<meta property="og:title" content="mipagina" />
```

```
<meta property="og:type" content="article" />
```

```
<meta property="og:url" content="http://mipagina.es" />
```

```
<meta property="og:image" content="http://foto.jpg" />
```

```
<meta property="fb:admins" content="1258775867" />
```

2.4.4. <link>

Definición: define un vínculo a otro documento indicado por **href**.

Aparición: sin etiqueta de cierre (V).

Atributos: **lang**, **dir**, **class**, **id**, **style**, **title**, **eventos intrínsecos**, **charset**, **href**, **hreflang**, **type**, **rel**, **rev** y **media**.

- **charset**: establece el conjunto de caracteres usado para el documento.
- **href**: indica la URI del documento vinculado.
- **hreflang**: indica el lenguaje del documento vinculado.
- **type**: indica el tipo de contenido del documento vinculado, para que el navegador no lo abra en caso de no tener soporte.

Los tipos más comunes son: text/html, text/css, image/png, image/gif, image/x-icon, video/mpeg y audio/basic.

Tipos para documentos xml: application/rss+xml, application/atom+xml.

Otras: application/x-shockwave-flash, application/opensearchdescription+xml.

- **rel**: establece la relación entre los documentos origen y destino.
- **rev**: indica un vínculo inverso, es decir, desde el destino al origen.
- **media**: indica el medio al que se refiere los datos de estilo en el documento destino.

Tipos de medios: screen, tty, tv, projection, handheld, print, braille, aural, all.

Valores básicos para el atributo **rel**:

- **stylesheet** = se refiere a una hoja de estilos externa.
- **alternate** = designa una versión alternativa del documento actual.

Cuando se usa con el atributo **hreflang** implica que hay una versión traducida del documento. Cuando se usa con el atributo **media**, implica que hay una versión diseñada para un medio diferente. Cuando se usa con **type = application** se trata de un feed.

- **start** = se refiere al primer documento de un conjunto de documentos.
- **next** = se refiere al siguiente documento en una serie ordenada de documentos.
- **prev** = se refiere al documento anterior en una serie ordenada de documentos.
- **contents** = se refiere a un documento que sirve como tabla de contenidos.
- **index** = se refiere a un documento que es un índice para el documento actual.
- **glossary** = se refiere a un documento que proporciona un glosario de términos.
- **copyright** = se refiere al aviso de copyright del documento actual.
- **chapter** = se refiere a un documento que actúa como capítulo en una serie.

- **section** = se refiere a un documento que actúa como sección en una serie.
- **subsection** = se refiere a un documento que actúa como subsección en una serie.
- **appendix** = se refiere a un documento que actúa como apéndice en una serie.
- **help** = se refiere a un documento que ofrece ayuda.
- **bookmark** = se refiere a una señal de lectura, es decir, un vínculo a una entrada importante dentro de un documento extenso.

Otros valores no predefinidos para usar con **rel**:

- **shortcut icon** = sirve para incluir un logo en la barra/pestaña del navegador.
- **canonical** = indica la página principal en caso de duplicación con alternate.
- **offline** = establece un fichero de tipo CDF (Channel Definition Format).
- **search** = indica una página de material relacionado con el documento actual.
- **service.feed**, **service.post**, **editUri**, etc. para las tecnologías RSS y Atom.

Ejemplos:

Para indicar que hay una versión de la página en otro idioma:

```
<link rel="alternate" hreflang="en" title="Mi página en Inglés"
      type="text/html" href="http://mipagina.es/ingles.html">
```

Para indicar que hay un archivo rss:

```
<link rel="alternate" type="application/rss+xml"
      title="Resumen de todas las secciones" href="/feed.xml">
```

Para insertar un icono en la barra de título o pestaña del navegador:

```
<link rel="shortcut icon" type="image/x-icon"
      href="http://mipagina.es/imagenes/logo.ico">
```

Para vincular una hoja de estilos con el documento actual:

```
<link rel="stylesheet" type="text/css"
      href="http://mipagina.es/estilos.css">
```

Para especificar la página principal cuando hay duplicación en diferentes idiomas:

```
<link rel="canonical" href="http://mipagina.es/index.html">
```

(En todos los casos se puede especificar una ruta relativa en **href** dependiendo de la estructura de directorios)

Actividad 2.12:

- En tu sitio crea la carpeta “imagenes” (no uses tilde ni espacio para nombres de carpeta).
 - Crea un logo relacionado con tu empresa, dale tamaño 80x80 px, llámalo logo.ico y súbelo a la carpeta /imagenes.
 - Inserta el enlace tipo “shortcut icon” en tu página.
 - Comprueba que aparece el logo junto al título, en la pestaña del navegador.
-

<object>

Su uso dentro de la cabecera es poco habitual, trataremos este elemento en el siguiente apartado, dentro del cuerpo del documento.

2.4.5. <script>

Definición: inserta un script dentro del documento.

Puede aparecer varias veces, tanto en la cabecera como en el cuerpo y puede ser interno al documento, o bien un fichero externo.

Aparición: las etiquetas de inicio y cierre son obligatorias.

Atributos: **charset**, **type**, **src**, **defer**.

- **charset**: codificación de caracteres del script indicado por **src**.
- **type**: indica el lenguaje de programación en el cual está escrito el script.
Como ya sabemos, con la etiqueta **<meta>** se puede determinar el lenguaje script por defecto para el documento. Si no se hace así, es conveniente indicar el lenguaje cada vez mediante el atributo **type**.
- **src**: indica la URI donde está alojado el script externo (del inglés **source**).
- **defer**: es un booleano, su presencia indica al navegador que el script no va a generar ningún contenido en el documento.

Ejemplos:

- Interno:

```
<script type="text/javascript">
...
</script>
```

- Externo:

```
<script type="text/javascript" src="/js/banner.js"></script>
<script type="text/javascript"
src="https://www.google.com/jsapi"></script>
```

Actividad 2.13:

- Busca un código javascript que permita visualizar fecha y hora del sistema.
 - Inserta el código para la fecha en tu página.
 - Guarda el código para la hora en un archivo externo hora.js
 - Inserta la llamada al archivo externo mediante la etiqueta <script... src>
 - Visualiza el resultado de la forma FECHA: HORA:
-

2.4.6. <style>

Definición: permite insertar una hoja de estilo interna en la cabecera del documento.

Aparición: etiquetas de inicio y fin obligatorias.

Atributos: lang, dir, type, media, title.

- **type:** indica el tipo de lenguaje de estilos.
Normalmente será "text/css".
- **media:** indica el tipo de medio al que se dirige la información de estilo.
Por defecto será "screen". Otros valores: tty, tv, handheld, print, braille, aural, all.
 - handheld = para dispositivos móviles.
 - print = para impresoras.
- **title:** es un texto descriptivo de la información de estilo.

Ejemplo:

```
<style type="text/css">
  body {
    margin-left: 40px;
    margin-top: 40px;
    margin-right: 40px;
  }
</style>
```

Actividad 2.14:

Inserta este código en tu página y observa los cambios producidos.

2.5. Contenido del cuerpo

En HTML la mayoría de elementos del cuerpo se pueden clasificar en:

- **Elementos de bloque.**

Son elementos que crean estructuras más grandes que pueden contener a los elementos de línea y a otros elementos de bloque. Además siempre comienzan en una línea nueva.

- **Elementos de línea.**

Estos elementos solo pueden contener datos y a otros elementos de línea.

No comienzan con línea nueva.

2.5.1. Manejo del texto

<h1 | h2 | h3 | h4 | h5 | h6>

Para empezar vamos a ver los elementos **h1** al **h6** que sirven para establecer encabezados, es decir, el tamaño de los títulos.

Hay seis niveles de encabezados: **h1**, **h2**, **h3**, **h4**, **h5** y **h6** ordenados estrictamente por el tamaño. Los encabezados son elementos de bloque.

Aparición: las etiquetas de apertura y cierre son obligatorias.

Atributos: %attrs, align.

Ejemplo:

```
<body>
  <h1>Encabezado H1</h1><h2>Encabezado H2</h2>
  <h3>Encabezado H3</h3><h4>Encabezado H4</h4>
  <h5>Encabezado H5</h5><h6>Encabezado H6</h6>
</body>
```

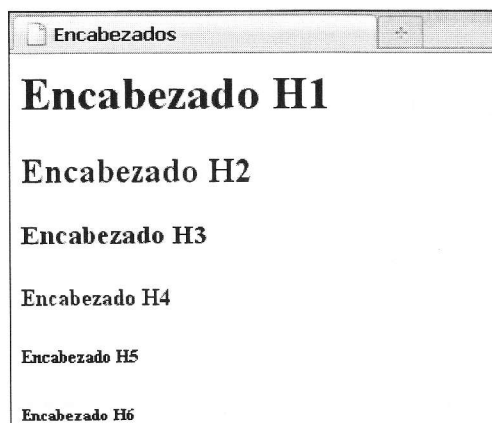


Figura 2.3: Encabezados en HTML

<p>

Definición: Otro de los elementos de bloque más típicos es <p> de párrafo.

De la misma forma que distribuimos el texto mediante párrafos en un documento Word, las líneas de texto en un documento HTML deben agruparse dentro de <p>. Cada párrafo en HTML se compone de una única línea de texto, salvo que introduzcamos un retorno de carro explícito usando el elemento
. El navegador se encarga de que la línea de texto ocupe todo el ancho disponible, y si el ancho cambia, el texto se ajusta al nuevo ancho ocupando más o menos líneas.

Todo texto de un documento HTML, que no sea encabezado, lista o celda de tabla debe estar dentro del elemento <p>.

Aparición: la etiqueta de cierre es opcional. No puede contener elementos de bloque.

Atributos: %attrs, align.

Ejemplo:

```

<body>
  <h1>Evolución de Networking</h1>
  <h3>Fase1: Ordenadores personales</h3>
  <p>La aparición de ordenadores personales en las empresas fue
    lenta al principio, pero el lanzamiento de programas de
    gestión como Lotus y otras aplicaciones específicas de
    empresa, impulsaron el desarrollo del PC.</p>
  <h3>Fase2: Red a pie</h3>
  <p>Al principio los ordenadores eran dispositivos autónomos a
    los que a veces se conectaban impresoras. Cuando un empleado
    que no tenía impresora conectada a su ordenador, tenía que
    copiar los ficheros a disquete, cargarlos en el ordenador de
    algún compañero que tuviera impresora e imprimirlos desde
    allí. A este tipo de red rudimentaria, se la conoció con el
    nombre de "red a pie".</p>
  <h3>Fase 3: Red LAN</h3>
  <p>Para resolver este inconveniente, las empresas instalaron
    redes de área local (LAN). Esto permitía a los usuarios de
    un mismo departamento transferir rápidamente información a
    través de la red. Las impresoras locales fueron sustituidas
    por impresoras de red.</p>
</body>

```

El resultado se ve en la Figura 2.4.

Un asunto a tener en cuenta es el espacio en blanco, si queremos insertar un espacio más del normal entre dos palabras, no vale utilizar la barra espaciadora porque es ignorada por el navegador. Es necesario insertar el carácter (no-break space) por cada espacio que queramos añadir.

Evolución de Networking

Fase1: Ordenadores personales

La aparición de ordenadores personales en las empresas fue lenta al principio, pero el lanzamiento de programas de gestión como Lotus y otras aplicaciones específicas de empresa, impulsaron el desarrollo del PC.

Fase2: Red a pie

Al principio los ordenadores eran dispositivos autónomos a los que a veces se conectaban impresoras. Cuando un empleado no tenía impresora conectada a su ordenador, tenía que copiar los ficheros a disquete, cargarlos en el ordenador de algún compañero que tuviera impresora e imprimirlos desde allí. A este tipo de red rudimentaria, se la conoció con el nombre de “red a pie”.

Fase 3: Red LAN

Para resolver este inconveniente, las empresas instalaron redes de área local (LAN). Esto permitía a los usuarios de un mismo departamento transferir rápidamente información a través de la red. Las impresoras locales fueron sustituidas por impresoras de red.

Figura 2.4: Texto con la etiqueta <p>

A veces es preciso introducir un salto de línea dentro del mismo párrafo, en ese caso debemos insertar la etiqueta
 de “break” que sirve para “romper” la línea de texto. No sirve insertar un salto de línea usando enter porque también es ignorado por el navegador.

Usando el ejemplo anterior, para evitar que una frase quede partida y mejorar la legibilidad del texto, podríamos introducir saltos de línea detrás de punto y seguido.

Aparición: sin etiqueta de cierre (V).

Evolución de Networking

Fase1: Ordenadores personales

La aparición de ordenadores personales en las empresas fue lenta al principio, pero el lanzamiento de programas de gestión como Lotus y otras aplicaciones específicas de empresa, impulsaron el desarrollo del PC.

Fase2: Red a pie

Al principio los ordenadores eran dispositivos autónomos a los que a veces se conectaban impresoras. Cuando un empleado no tenía impresora conectada a su ordenador, tenía que copiar los ficheros a disquete, cargarlos en el ordenador de algún compañero que tuviera impresora e imprimirlos desde allí. A este tipo de red rudimentaria, se la conoció con el nombre de “red a pie”.

Fase 3: Red LAN

Para resolver este inconveniente, las empresas instalaron redes de área local (LAN). Esto permitía a los usuarios de un mismo departamento transferir rápidamente información a través de la red. Las impresoras locales fueron sustituidas por impresoras de red.

Figura 2.5: Texto con las etiquetas <p> y
