

UT2. HTML

UT2. HTML

2.1 Introducció

2.2. Evolució històrica

2.3. Elementos de HTML

2.4 Contenido de la cabecera

2.1 Introducció

El **lenguaje de marcado de hipertexto** o **HTML** es el lenguaje de marcado estándar para documentos diseñados para mostrarse en un navegador web.

Puede ser asistido por tecnologías como hojas de estilo en cascada (**CSS**) y lenguajes de secuencias de comandos como **JavaScript**.

Los navegadores web reciben documentos HTML de un servidor web o del almacenamiento local y convierten los documentos en páginas web multimedia.

2.2 Evolución histórica

HTML (HyperText Markup Language) aparece a principios de los años noventa cuando el investigador Tim Berners Lee, lo desarrolla para potenciar la colaboración entre físicos e investigadores de la energía nuclear.

El objetivo principal era enlazar los documentos mediante links (**hipervínculos**), de forma que al hacer clic sobre algún texto, se abriera otro documento relacionado con la información seleccionada.

En 1991 se publica el primer documento formal con el nombre de "**HTML tags**". Esta primera versión tiene muchas limitaciones, por ello organismos internacionales colaboraron para desarrollar mejoras en el lenguaje.a

En 1993 hubo un primer intento de realizar una especificación oficial de HTML, sin embargo no fue reconocida. Se trataba de la versión HTML 1.2.

En 1995 se publica el primer estándar oficial llamado **HTML 2.0** por el organismo IETF (Internet Engineering Task Force).

2.2 Evolución histórica

30 años de la World Wide Web: ¿cuál fue la primera página web de la historia y para qué servía?



2.2 Evolución histórica

En 1997 se publica la versión **HTML 3.2** por la organización **W3C (World Wide Web Consortium)**, que será a partir de ahora y hasta nuestros días, el organismo oficial encargado de publicar los estándares oficiales. Incluye tablas y applets de Java.

En 1998 aparece la versión **HTML 4.0** que incluye muchas mejoras como **soporte para lenguajes script, hojas de estilo css**, facilidades de impresión, accesibilidad para discapacitados, etc. En este mismo año **aparece también la primera especificación de XML**. Al año siguiente se publica **HTML 4.01** la última especificación oficial hasta el momento, se trata de una mera revisión que no incluye novedades significativas.

En el año 2000 el W3C publica XHTML 1.0, se trata de una reformulación de HTML 4.0 basada en XML. Cumpliendo unas directrices de compatibilidad puede funcionar sobre aplicaciones creadas para HTML, pero a la vez es conforme a XML, lo que le proporciona gran flexibilidad para añadir nuevos elementos y módulos.

2.2 Evolución histórica

XHTML 1.0 se convirtió en un estándar aceptado y ampliamente utilizado por la comunidad de desarrolladores Web. **W3C** siguió trabajando por este camino en un nuevo lenguaje XHTML 2.0, pero que ya no era compatible ni con HTML ni siquiera con XHTML 1.0.

En el año 2004 las empresas Apple, Mozilla y Opera crearon un grupo de trabajo llamado WHATWG con el objetivo de continuar el desarrollo de HTML.

Una de sus premisas de trabajo era mantener total compatibilidad con versiones anteriores.

2.2 Evolución histórica

En el 2007 los grupos de trabajo W3C y WHATWG deciden unirse para el desarrollo de una especificación de **HTML 5**, lo que permite publicar a principios de 2008 el primer borrador de HTML5.

En el 2009 el consorcio W3C decide poner fin al desarrollo del estándar XHTML 2.0 a favor de HTML5. HTML5 sale a la luz el 28 de octubre de 2014.

2.2 Evolución histórica

HTML 4.01 recomienda separar por completo el contenido del formato.

En HTML la declaración es más simple:

```
<!DOCTYPE html>
```

DTD estricta:
<p>Si cumplimos escrupulosamente con esta recomendación, almacenando toda la información sobre el formato en una hoja de estilos CSS, entonces podemos utilizar la declaración estricta:</p> <pre><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN" "http://www.w3.org/TR/html4/strict.dtd"></pre>
DTD de transición:
<p>Si queremos combinar contenido e información de formato en nuestro documento HTML, entonces debemos utilizar la declaración de transición:</p> <pre><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"></pre>
DTD con marcos:
<p>Si queremos incluir marcos en nuestro documento, entonces tenemos que usar una DTD que es muy similar a la anterior, pero que sustituye el elemento body por frameset, en este caso la declaración sería:</p> <pre><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd"></pre>

2.2 Evolución histórica

Actividad: Copia este código en un archivo de texto, guárdalo con el nombre **mipagina.html** y luego observa el resultado en el navegador.

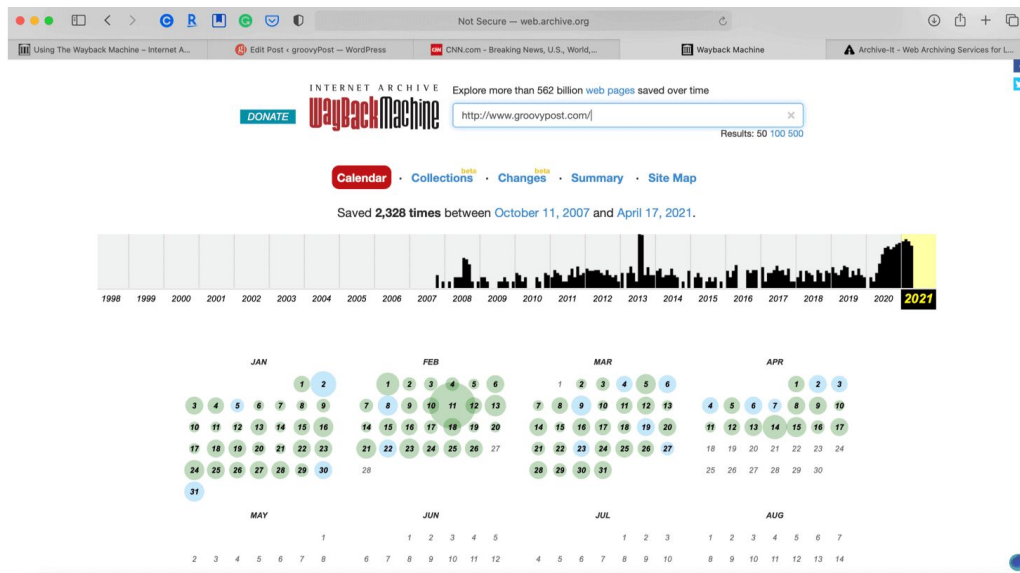
```
<!DOCTYPE html>
<html>
  <head>
    <title>Mi primera página</title>
  </head>
  <body>
    <strong>Hola Mundo </strong>
  </body>
</html>
```



2.2 Evolució històrica

Actividad: Empra l'eina Wayback Machine i visualitza com han anat canviat les pàgines web al llarg del temps.

Exemple: `https://www.whitehouse.gov`



2.3 Elementos de HTML

Los **elementos** son los componentes básicos del lenguaje HTML. Cada elemento puede tener **atributos** y **contenido**. La mayoría de elementos tienen una **etiqueta de inicio**, una **etiqueta de fin** y un contenido encerrado entre ambas. Si no tienen contenido se les llama vacíos.

Los atributos de un elemento se colocan en la etiqueta de apertura.

Sintaxis:

- La sintaxis básica para un elemento no vacío tiene la forma:

```
<nombre_elemento atributo1="valor1" atributo2="valor2" ... >  
    contenido del elemento  
</nombre_elemento>
```

- La sintaxis para un elemento vacío:

```
<nombre_elemento atributo1="valor1" atributo2="valor2"... >
```

2.3 Elementos de HTML

Las **etiquetas de inicio** deben tener el siguiente formato:

- El primer carácter de una etiqueta de inicio debe ser un carácter <.
- Los siguientes caracteres de una etiqueta de inicio deben ser el nombre de la etiqueta del elemento.
 - Si va a haber algún atributo en el siguiente paso, primero debe haber uno o más espacios en blanco ASCII.
 - Entonces, la etiqueta de inicio puede tener varios atributos, cuya sintaxis se describe a continuación.
 - Los atributos deben estar separados entre sí por uno o más espacios en blanco ASCII.
 - Después de los atributos, o después del nombre de la etiqueta si no hay atributos, puede haber uno o más espacios en blanco ASCII.
 - Entonces, si el elemento es uno de los elementos vacíos, o si el elemento es un elemento externo, entonces puede haber un solo carácter /, que en los elementos externos marca la etiqueta de inicio como de cierre automático.
 - *En los elementos vacíos, no marca la etiqueta de inicio como de cierre automático, sino que es innecesario y no tiene ningún tipo de efecto.* Para tales elementos vacíos, debe usarse solo con precaución, especialmente porque, si está precedido directamente por un valor de atributo sin comillas, se convierte en parte del valor del atributo en lugar de ser descartado por el analizador.
- Finalmente, las etiquetas de inicio deben cerrarse con un carácter >.

2.3 Elementos de HTML

Las **etiquetas finales** deben tener el siguiente formato:

- El primer carácter de una etiqueta final debe ser un carácter <.
- El segundo carácter de una etiqueta final debe ser un carácter /.
- Los siguientes caracteres de una etiqueta final deben ser el nombre de la etiqueta del elemento.
- Después del nombre de la etiqueta, puede haber uno o más espacios en blanco ASCII.
- Finalmente, las etiquetas finales deben cerrarse con un carácter >.

2.3.1 html

html Delimita el contenido del documento.

Atributos: **dir**, **lang**, ...

- **dir**: indica el sentido de lectura de texto cuando no corresponde al habitual (árabe).
Valores: ltr (de izquierda a derecha) y rtl (de derecha a izquierda).
- **lang**: indica el idioma por defecto del documento.
Valores: de (alemán), en (inglés), es (español), fr (francés), etc.

2.3.1 html

html Delimita el contenido del documento. Es el elemento raíz.

Contenido de html: **<head>** y **<body>**

Ejemplo:

```
<html lang="es" dir="ltr">  
<head> ... </head> <body> ... </body>  
</html>
```


2.3.1 html

Actividad: Indica que mipagina.html está en español y con dirección rtl, observa el resultado, después deja la dirección definitiva en ltr.



2.3.2 head

head Delimita la cabecera de un documento.

Contenido de la cabecera: **<title>**, **<base>**, **<meta>**, **<link>**, **<object>**, **<script>** y **<style>**.

El contenido de la cabecera es una información general de todo el documento, por eso se escribe al principio y es lo primero que analiza el navegador.

2.3.3 body

body Delimita el cuerpo de un documento.

Atributos: **id**, **class**, **title**, **style**, ...

- **id**: sirve para asignar un nombre único a un elemento.

Se utiliza para identificar elementos en hojas de estilo y en scripts.

- **class**: sirve para asignar un nombre de clase a un elemento. Es posible usar el mismo nombre de clase para varios elementos.

- **title**: sirve para agregar un comentario asociado a un elemento. Los navegadores muestran este comentario en una ventana emergente cuando el ratón se sitúa sobre el elemento.

- **style**: sirve para aplicar información de estilo a un elemento.

2.3.3 body

body Delimita el cuerpo de un documento.

Ejemplo:

```
<body style="background-color: blue"> ... </body>
```

2.3.3 body

Actividad: Cambia el color de fondo con el atributo style y con el atributo desaprobado bgcolor. Comprueba que se obtiene idéntico resultado.

```
<body style="background-color: blue"> ... </body>
```

```
<body bgcolor="#0000ff"> ... </body>
```



2.3.4 Comentarios

comentarios `<!-- -->` sirve para insertar una o varias líneas con comentarios que no serán interpretadas por el navegador. No se pueden anidar.

Ejemplo:

```
<!-- mipagina.html -->
<html>
  <head> ... </head>
    <!-- Aquí empieza el cuerpo del documento -->
  <body> ... </body>
</html>
```

2.3.4 Comentarios

Actividad: Agrega comentarios a `mipagina.html` para separar las partes del documento.



2.3.5 Conjunto de atributos

Los atributos globales pueden ser especificados para todos los elementos de HTML.

```
%attrs = %i18n + %coreattrs + %events
```

```
%i18n = lang, dir
```

```
%coreattrs = accesskey, autocapitalize, autofocus, contenteditable, dir, draggable,  
enterkeyhint, hidden, inert, inputmode, is, itemid, itemprop, itemref, itemscope,  
itemtype, lang, nonce, spellcheck, style, tabindex, title, translate
```

```
%events = onabort, onblur, oncanplay, oncanplaythrough, onchange, onclick, oncontextmenu,  
oncuechange, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,  
ondragstart, ondrop, ondurationchange, onemptied, onended, onerror, onfocus, oninput,  
oninvalid, onkeydown, onkeypress, onkeyup, onload, onloadeddata, onloadedmetadata,  
onloadstart, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onmousewheel,  
onpause, onplay, onplaying, onprogress, onratechange, onreadystatechange, onreset, onscroll,  
onseeked, onseeking, onselect, onshow, onstalled, onsubmit, onsuspend, ontimeupdate,  
onvolumechange, onwaiting
```


2.3.5 Conjunto de atributos

Eventos

Atributo	Descripción
onload	Ocurre cuando termina la carga completa de un documento
onunload	Ocurre cuando el navegador elimina un documento

Atributo	Descripción
onblur	Ocurre cuando un campo pierde el foco
onchange	Ocurre cuando un campo pierde el foco y su valor se modifica después
onfocus	Ocurre cuando un campo recibe el foco
onreset	Ocurre cuando se reinicia un formulario
onselect	Ocurre cuando se selecciona un texto de un campo de texto
onsubmit	Ocurre cuando se envía un formulario

2.3.5 Conjunto de atributos

Eventos

Eventos de `<body>` y `<frameset>`

Atributo	Descripción
onload	Ocurre cuando termina la carga completa de un documento
onunload	Ocurre cuando el navegador elimina un documento

2.3.5 Conjunto de atributos

Eventos

Eventos de **formulario**

Atributo	Descripción
onblur	Ocurre cuando un campo pierde el foco
onchange	Ocurre cuando un campo pierde el foco y su valor se modifica después
onfocus	Ocurre cuando un campo recibe el foco
onreset	Ocurre cuando se reinicia un formulario
onselect	Ocurre cuando se selecciona un texto de un campo de texto
onsubmit	Ocurre cuando se envía un formulario

2.3.5 Conjunto de atributos

Eventos

Eventos de imagen

Atributo	Descripción
onabort	Ocurre cuando la carga de una imagen se interrumpe

Eventos con el teclado

Atributo	Descripción
onkeydown	Ocurre cuando se pulsa una tecla encima de un elemento
onkeypress	Ocurre cuando se pulsa y se suelta una tecla encima de un elemento
onkeyup	Ocurre cuando se suelta una tecla encima de un elemento

2.3.5 Conjunto de atributos

Eventos

Eventos con el ratón

Atributo	Descripción
onclick	Ocurre cuando se hace click sobre un elemento
ondblclick	Ocurre cuando se hace doble click sobre un elemento
onmousedown	Ocurre cuando se pulsa el botón del ratón encima de un elemento
onmousemove	Ocurre cuando el ratón se mueve estando encima de un elemento
onmouseout	Ocurre cuando el ratón sale de un elemento
onmouseover	Ocurre cuando el ratón se sitúa sobre un elemento
onmouseup	Ocurre cuando el botón del ratón se suelta estando encima

2.3.5 Conjunto de atributos

Para utilizar estos eventos intrínsecos hay que conocer un lenguaje script, que sirva para ejecutar una determinada acción cuando se produzca el evento.

Ejemplo:

```
<body onload="alert('Carga completa');">
```

2.4 Contenido de la cabecera

Elementos:

- `<base>`
- `<link>`
- `<meta>`
- `<noscript>`
- `<script>`
- `<style>`
- `<template>`
- `<title>`

2.4 Contenido de la cabecera

title indica el nombre del sitio Web en la barra superior o pestaña del navegador.

Etiquetas de apertura y cierre obligatorias. No debe haber más de un title por documento.

Ejemplo:

```
<head>  
  <title> Mi primera página </title>  
</head>
```


2.4 Contenido de la cabecera

base indica la dirección raíz del sitio Web, la cual permite resolver las direcciones relativas.

Sin etiqueta de cierre. Sólo debe haber un elemento base por documento. Si base tiene el atributo href entonces debe ir delante de cualquier elemento que use direcciones URL.

Atributos:

- **href**: se usa siempre para indicar la dirección raíz del documento.

Ejemplo: <base href="<http://www.juanroacr.es/>">

- **target**: Contexto por defecto del navegador para la navegación por enlaces o el envío de formularios.

2.4 Contenido de la cabecera

base indica la dirección raíz del sitio Web, la cual permite resolver las direcciones relativas.

Sin etiqueta de cierre. Sólo debe haber un elemento base por documento.

Si base tiene el atributo href entonces debe ir delante de cualquier elemento que use direcciones URL.

Si base tiene el atributo target, este debe entonces debe contener un nombre de contexto de navegador válido: _self, _top, _blank, _parent

Atributos:

- **href:** se usa siempre para indicar la dirección raíz del documento.

Ejemplo: `<base href="http://www.cifpfbmoll.eu/">`

- **target:** Contexto por defecto del navegador para la navegación por enlaces o el envío de formularios.

2.4 Contenido de la cabecera

meta Indica un conjunto de propiedades generales del documento como el autor, descripción, palabras clave, herramienta utilizada, tipo de contenido, etc.

Estas propiedades evolucionan continuamente, es decir, se crean nuevas propiedades para ampliar la información asociada al documento, las redes sociales son un ejemplo.

Normalmente estas propiedades pasan desapercibidas para el usuario habitual.

Sin etiqueta de cierre.

Atributos: lang, dir, name, content, http-equiv, scheme.

2.4 Contenido de la cabecera

meta

Atributos:

- **name:** indica el nombre de una propiedad, no hay una lista oficial de propiedades pero podemos citar los siguientes valores: abstract, author, copyright, date, description, distribution, expires, generator, google-site-verification, keywords, language, noemail-collection, organization, rating, reply-to, revisit-after, robots y la familia de etiquetas Dublin Core.
- **content:** contiene el valor de la propiedad indicada antes por name, por tanto, estos dos atributos funcionan simultáneamente proporcionando parejas de valores.

Ejemplo:

```
<meta name="description" content="Lenguajes de Marcas">
```

2.4 Contenido de la cabecera

meta

- **http-equiv**: indica una propiedad al navegador en forma de cabecera http, como si el propio servidor http (Web) hubiera generado dicha cabecera, de ahí viene el nombre "http-equivalent".

Valores: cache-control, content-type, set-cookie, content-disposition, pics-label, pragma, refresh, resource-type, content-script-type, content-style-type, window-target.

Tiene muchas utilidades como por ejemplo, indicar el conjunto de caracteres a utilizar en nuestro sitio Web, sin necesidad de modificar el servidor. También funciona en coordinación con el atributo content para proporcionar parejas de valores.

Ejemplo:

```
<meta http-equiv="content-type" content="text/html;  
charset=iso-8859-15">
```

Esto indica que el contenido de la página es texto plano HTML con el conjunto de caracteres iso-8859-15.

2.4 Contenido de la cabecera

meta

- **charset**: El atributo charset especifica la codificación de caracteres utilizada por el documento. Esta es una declaración de codificación de caracteres. Si el atributo está presente, su valor debe ser una coincidencia ASCII que no distinga entre mayúsculas y minúsculas para la cadena "utf-8".

Ejemplo:

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

2.4 Contenido de la cabecera

meta

- **scheme**: indica al navegador que tiene que interpretar los metadatos atendiendo al perfil especificado, para evitar ambigüedad.

Ejemplo:

```
<meta scheme="Europe" name="date" content="24-10-2012">
```

Esto indica que la fecha se lee según el perfil "Europe", es decir: "dd-mm-aaaa".

2.4 Contenido de la cabecera

Ahora vamos a estudiar las principales meta etiquetas en función de su utilidad.

Se clasifican en dos grupos (por sus dos atributos principales, aunque hay más):

1. Las `<meta name>`: se utilizan principalmente para optimizar el resultado de los motores de búsqueda, es decir, para ayudar a los buscadores de páginas Web.

La información que manejan no se presenta en pantalla.

2. Las `<meta http-equiv>`: se utilizan para dar instrucciones a los navegadores, es decir, ejercen cierto control sobre ellos.

Tienen influencia sobre lo que aparece en pantalla.

2.4 Contenido de la cabecera

1) Meta name

- Información general sobre la página: abstract, author, copyright, date, generator
- Información específica para buscadores: description, distribution, keywords, locality, rating, revisit-after, robots
- Para administración del sitio: google-site-verification, reply-to

2) Meta http-equiv

- Tipo de Contenido: content-type, content-language, content-script-type
- Manejo de cookies: set-cookie
- Actualización/Redirección de la página: refresh
- Control parental: pics-label
- Manejo de la cache: cache-control, pragma, last modified, expires

2.4 Contenido de la cabecera

- **abstract** Proporciona un resumen muy corto de la etiqueta "description". No es utilizada por todos los motores de búsqueda al contrario que description.
- **author** Proporciona el nombre del autor del documento actual. Es recomendable indicar el nombre completo con nombre y apellidos.

```
<meta name="author" content="Jaume Smith Jones">
```

- **copyright** Sirve para indicar que hay información bajo derechos de autor.

```
<meta name="copyright" content="Editorial Sybex">
```

Es recomendable sin embargo, utilizar un icono o enlace fácilmente visible para especificar que el documento está bajo derechos de autor. Para eso puedes usar por ejemplo el creador de licencias de [Creative Commons](#).

- **date** Sirve para indicar la fecha de creación de la página.

```
<meta name="date" content="24 de Febrero de 2010">
```

- **generator** Sirve para indicar la herramienta utilizada para construir el sitio web.

```
<meta name="generator" content="KompoZer">
```

2.4 Contenido de la cabecera

- **description** Sirve para explicar el contenido y objetivos del sitio web. Esta información es utilizada por los motores de búsqueda para indexar la página, así que resulta importante el texto que coloquemos.

```
<meta name="description" content="Lenguajes de marcas y sistemas de gestión de información.">
```

- **distribution** Se utiliza para indicar el nivel de distribución que queremos para nuestro sitio web. Niveles: Global (Internet), Local (reservado para la red local) IU (Uso interno).

```
<meta name="distribution" content="global">
```

No es recomendable utilizar esta etiqueta, la forma adecuada de restringir el acceso a nuestro sitio es configurar adecuadamente el archivo .htaccess de nuestro servidor http, o bien crear un archivo robots.txt, para que los motores de búsqueda indexen según nuestras preferencias.

- **keywords** Sirve para indicar una lista de palabras clave acerca de nuestro sitio web.

Es una etiqueta muy útil para que nuestro sitio sea localizable, sin embargo debe utilizarse correctamente. La inclusión de palabras no relacionadas con nuestro sitio con el fin de mejorar el posicionamiento web, se considera spam y puede ser penalizado por los motores de búsqueda.

```
<meta name="keywords" content="Ciclos, Formativos, Informática, FP, DAW, DAM, ASIR, SMR">
```

2.4 Contenido de la cabecera

- **locality** Sirve para indicar al robot de indexación la localidad donde se ubica el sitio web.

```
<meta name="locality" content="Palma, España">
```

- **revisit-after** Sirve para indicar a los motores de búsqueda, que revisiten nuestra página para indexarla de nuevo, ya que ha sido modificada.

```
<meta name="revisit-after" content="3 days">
```

Los motores de búsqueda suelen tener su propio patrón de retorno a las páginas ya indexadas, así que no siempre cumplen el contenido de esta etiqueta.

2.4 Contenido de la cabecera

- robots

Sirve para controlar los robots de los motores de búsqueda. Los valores posibles son:

Valor	Utilidad
index	Indica al robot que indexe esta página
follow	Indica al robot que siga los hipervínculos de esta página para indexarlos también.
all	Equivale a los dos anteriores juntos.
noindex	No indexa la página.
nofollow	No sigue los hipervínculos.
none	Equivalente a las dos anteriores juntas.
noarchive	No permite a los buscadores almacenar la página en caché
noodp	No permite usar la descripción oficial de la página que hay en Wikipedia (si la hubiera).
noydir	Equivalente a la anterior para Yahoo.

Ejemplo: `<meta name="robots" content="index, nofollow">`

2.4 Contenido de la cabecera

- **locality** Sirve para indicar al robot de indexación la localidad donde se ubica el sitio web.

```
<meta name="locality" content="Palma, España">
```

- **revisit-after** Sirve para indicar a los motores de búsqueda, que revisiten nuestra página para indexarla de nuevo, ya que ha sido modificada.

```
<meta name="revisit-after" content="3 days">
```

Los motores de búsqueda suelen tener su propio patrón de retorno a las páginas ya indexadas, así que no siempre cumplen el contenido de esta etiqueta.

2.4 Contenido de la cabecera

Además de usar esta etiqueta, los motores de búsqueda se comunican con el sitio Web mediante el archivo **robots.txt**, que debe crear el administrador del sitio y colocarlo en el directorio raíz.

El archivo robots.txt es de fácil creación, se trata de texto plano que se puede generar con el bloc de notas. Su objetivo primordial es limitar la indexación generalizada de las páginas web de nuestro sitio.

Utiliza unas pocas palabras clave como:

User-agent (robot), **Allow** (permitir), **Disallow** (prohibir)

y los valores:

* (todos los robots), / (todos los directorios).

Para evitar directorios o archivos en particular hay que deshabilitados por separado.

El archivo se lee de arriba abajo ordenadamente, para cada robot se pueden prohibir los directorios y archivos deseados y cada user-agent es independiente del siguiente.

2.4 Contenido de la cabecera

Ejemplos:

User-agent: * Allow: /	Permite a todos los buscadores indexar todas las páginas, sería equivalente a no crear el archivo.
User-agent: * Disallow:	Lo mismo que el anterior.
User-agent: * Disallow: /imagenes/ Disallow: /js/	Impide a todos los buscadores indexar los directorios: imagenes y js.
User-agent: * Disallow: / User-agent: googlebot Disallow: / Allow: public	Impide a todos los robot indexar todas las carpetas y permite al robot googlebot indexar solamente el directorio /public.

Hay que tener en cuenta que a pesar de evitar la indexación de una carpeta, si algún recurso es referenciado por un enlace, entonces aparecerá en las respuestas del buscador. Para evitar esto hay que utilizar la metaetiqueta robots con "noindex, nofollow".

2.4 Contenido de la cabecera

- **google-site-verification** Sirve para identificar al administrador del sitio en las herramientas de administración de sitios de google. Estas herramientas permiten a un administrador analizar varios parámetros: como su visibilidad, las búsquedas que recibe el sitio, sitemaps, errores, etc.

```
<metaname="google-site-verification"  
content="8EYm4qdOa4UOL8tRw9clHsjSbo_JOKOJjlxaa40tQAA">
```

- **reply-to** Sirve para indicar una dirección email para contactar con el administrador del sitio.

No es nada recomendable usar esta etiqueta pues puede utilizarse como herramienta de spam, la alternativa es construir un formulario de contacto.

- **Dublin-Core** Este proyecto pretende crear un estándar para el uso de metaetiquetas.

```
<meta name="dc.title" content="Alpargatas 3000 S.A.">
```

2.4 Contenido de la cabecera

- **content-type** Sirve para indicar el tipo de los datos contenidos en el cuerpo de la página y en caso de ser texto también indica el juego de caracteres utilizado. Según el RFC 822 el campo content-type se define:

```
content-Type := tipo "/" subtipo [";" atributo = valor]
```

Los tipos MIME (Multipurpose Internet Mail Extensions) representan los tipos de formato válidos para un archivo.

```
<meta http-equiv="content-type"
content="text/html; charset=iso-8859-1">
```

En este caso usamos el juego de caracteres latín nº1 que es adecuado para nuestro idioma.

Si queremos hacer más internacional nuestra página, escribiendo caracteres que no pertenecen a nuestro alfabeto, necesitamos el charset utf-8.

Tipo MIME	Utilidad
application	Datos binarios o información para programas de correo Subtipos: octet-stream, oda, postscript, x-shockwave-flash, pdf
audio	Archivos de sonido, requieren de un periférico para su salida Subtipos: basic, wav, mp3
image	Imágenes, requieren de un periférico para su salida Subtipos: jpeg, gif, x-icon
message	Un mensaje encapsulado Subtipos: rfc828, partial, external-body
multipart	Datos formados por varias partes con diferentes tipos de datos Subtipos: mixed, alternative, parallel, digest
text	Información en forma de texto Subtipos: plain, richtext, html
video	Datos de video, requiere software y hardware específico Subtipos: mpeg, avi

2.4 Contenido de la cabecera

El UTF-8 (Unicode Transformation Format) contiene todos los caracteres posibles y cada uno tiene un código diferente.

Para nuestros caracteres específicos como los acentos o la ñ debemos utilizar los siguientes códigos:

á = ´

é = é

í = í

ó = ó

ú = ú

ñ = ñ

Named character references.

<https://html.spec.whatwg.org/#named-character-references>

2.4 Contenido de la cabecera

- **content-language** Sirve para indicar el lenguaje principal del documento como en lang.

```
<meta http-equiv="content-language" content="es_es">
```

Puede ser utilizado por los robots para categorizar el sitio web por el idioma. La lista oficial de códigos de idioma está definida por el IANA.

- **content-script-type, content-style-type** Sirve para indicar el lenguaje de programación por defecto usado en el script y el lenguaje de hoja de estilos por defecto en el documento.

Normalmente no se usan porque los navegadores los detectan automáticamente.

2.4 Contenido de la cabecera

- **set-cookie** Sirve para guardar una cookie en el navegador del usuario.

Una cookie es información personal, como el nombre de usuario o sus preferencias, que sirve para mejorar la navegación cuando revisitamos un sitio web.

Si se especifica la fecha de expiración, la cookie será almacenada hasta el momento indicado, si no se especifica se borrará al cerrar el navegador.

No es lo más adecuado usar esta etiqueta, el manejo de cookies hoy en día se realiza mediante lenguajes de lado servidor como PHP.

2.4 Contenido de la cabecera

- **refresh** Sirve para indicar el tiempo de espera en segundos hasta una actualización o redirección automática de la página.

```
<meta http-equiv="refresh" content="10">
```

Se utiliza habitualmente en páginas que presentan información en tiempo real.

```
<meta http-equiv="refresh" content="10; URL=http://www.dominio.com/otro_sitio.html">
```

Puede utilizarse para redireccionar a otro sitio web.

En el caso de que hayamos cambiado de dominio, es preferible insertar un enlace para que el usuario se redirija o el método de redirección 301 (que requiere insertar código en el lado servidor).

2.4 Contenido de la cabecera

- **pics-label** Sirve para indicar la calificación moral del contenido atendiendo a una serie de parámetros, como desnudez, violencia, lenguaje, comportamientos sociales, etc.

Existen varias organizaciones de calificación, una de las más conocidas es el ICRA (Internet Content Rating Association), la cual proporciona un generador gratuito de etiquetas.

Ejemplo:

```
<meta http-equiv="pics-Label" content='(pics-1.1 "http://www.icra.org/pics/vocabularyv03/"  
1 gen true for "http://miweb.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/>

2.4 Contenido de la cabecera

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 (Proveedor de Servicios de Internet) o en servidores Proxy inversos (que sirven de respaldo a sitios web con muchas consultas).

2.4 Contenido de la cabecera

- **cache-control** Sirve para controlar el cacheo que va a recibir el documento. Se admiten las siguientes opciones en content:

- **max-age = n° segundos** Indica el número de segundos durante los que el contenido es considerado como válido.
- **s-maxage = n° 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.

2.4 Contenido de la cabecera

- **cache-control** Se admiten las siguientes opciones en content:
 - **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">
```

2.4 Contenido de la cabecera

- **pragma** Sirve para evitar que la pagina 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.

2.4 Contenido de la cabecera

- **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.

2.4.1 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.

Las etiquetas meta con la sintaxis:

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

2.4.1 El protocolo Open Graph

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.

2.4.1 El protocolo Open Graph

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" />
```

Enlace:

The Open Graph protocol



2.4.1 El protocolo Open Graph

link define un vínculo a otro documento indicado por href. Sin etiqueta de cierre.

Atributos:

- **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. Ex: text/css
- **rel**: establece la relación entre los documentos origen y destino.
- **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, aH.

2.4 Contenido de la cabecera

link define un vínculo a otro documento indicado por href. Sin etiqueta de cierre.

Atributos:

- **rel**: establece la relación entre los documentos origen y destino.

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.

2.4 Contenido de la cabecera

link define un vínculo a otro documento indicado por href. Sin etiqueta de cierre.

Ejemplos:

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

```
<link rel="alternate" hreflang="en" title="Mi pagina en Ingles" 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">
```

2.4 Contenido de la cabecera

Actividad: 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.



2.4 Contenido de la cabecera

link 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.

Las etiquetas de inicio y cierre son obligatorias.

Atributos:

- **charset**: codificación de caracteres del script indicado por src.
- **type**: Indica el lenguaje de programación en el cual está escrito el script.
- **src**: Indica la URI donde está alojado el script externo.
- **async**: Ejecuta el script cuando esté disponible, sin bloquear.
- **defer**: Es un booleano, su presencia indica al navegador que el script no va a generar ningún contenido en el documento.

2.4 Contenido de la cabecera

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>
```

2.4 Contenido de la cabecera

Actividad: Busca un código javascript que permita visualizar la 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 Contenido de la cabecera

style Permite insertar una hoja de estilo interna en la cabecera del documento. Las etiquetas de inicio y fin son obligatorias.

Atributos:

- **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.

2.4 Contenido de la cabecera

style Permite insertar una hoja de estilo interna en la cabecera del documento. Las etiquetas de inicio y fin son obligatorias.

Atributos:

- **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.

2.4 Contenido de la cabecera

style Permite insertar una hoja de estilo interna en la cabecera del documento. Las etiquetas de inicio y fin son obligatorias.

Ejemplo:

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