



HTML

Capítulo 10

Lenguajes de Marcas y Sistemas de Gestión de Información

Curso 2020/2021

1. Introducción
2. Estructura de la Cabecera
3. Metadatos
4. DOCTYPE



1. INTRODUCCIÓN

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 de la etiqueta `<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 las próximas secciones se explica cómo incluir la metainformación y se introduce un concepto relacionado llamado `DOCTYPE`.



2. ESTRUCTURA DE LA CABECERA

Como ya se explicó anteriormente, las páginas XHTML se dividen en **dos partes** denominadas cabecera y cuerpo. La sección de la cabecera está formada por todas las etiquetas encerradas por la etiqueta `<head>`:

Etiqueta	<code><head></code>
Atributos comunes	internacionalización
Atributos propios	<ul style="list-style-type: none">• <code>profile = "url"</code> - Especifica la URL del perfil o perfiles que utilizan los metadatos• <code>lang = "codigo_de_idioma"</code> - Especifica el idioma principal de los contenidos de la página
Tipo de elemento	-
Descripción	Define la cabecera del documento HTML



2. ESTRUCTURA DE LA CABECERA

```
<head>

  <!-- Zona de etiquetas META -->

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


  <!-- Zona de título -->

  <title>El título del documento</title>


  <!-- Zona de recursos enlazados (CSS, RSS, JavaScript) -->

  <link rel="stylesheet" href="#" type="text/css" media="screen" />

  <link rel="stylesheet" href="#" type="text/css" media="print" />

  <link rel="alternate" type="application/rss+xml" title="RSS 2.0" href="#" />

  <script src="#" type="text/javascript"></script>

</head>
```



2. ESTRUCTURA DE LA CABECERA

La etiqueta `<title>` establece el título de la página. Los navegadores muestran este título como título de la propia ventana del navegador. Los buscadores utilizan este título como título de sus **resultados de búsqueda**.

El valor de `<title>` no sólo es importante para los usuarios, sino que también es importante para que los usuarios encuentren las páginas a través de los **buscadores**. Un error común de muchos sitios web consiste en mostrar un mismo título genérico en todas sus páginas. Cada página debe mostrar un **título corto**, adecuado, único y que describa inequívocamente los contenidos de la página.



2. ESTRUCTURA DE LA CABECERA

Las páginas XHTML deben tener definido un **título** y sólo uno, por lo que todas las páginas web deben incluir **obligatoriamente** una etiqueta `<title>`, cuya definición formal se muestra a continuación:

Etiqueta	<code><title></code>
Atributos comunes	internacionalización
Atributos propios	<ul style="list-style-type: none">• <code>lang = "codigo_de_idioma"</code> - Especifica el idioma principal del título de la página
Tipo de elemento	-
Descripción	Define el título del documento HTML

2. ESTRUCTURA DE LA CABECERA

Por último, la etiqueta `<head>` permite definir en el atributo `profile` la URL de un documento externo que contiene el **perfil que siguen los metadatos de la cabecera**. *Los blogs creados con el programa WordPress incluyen por ejemplo el siguiente perfil en su cabecera:*

```
<head profile="http://gmpg.org/xfn/11">
```

...

```
</head>
```

El documento `http://gmpg.org/xfn/11` es un perfil que define **atributos adicionales** para establecer la relación entre sitios web.



3. METADATOS

Permiten incluir cualquier **información relevante** sobre la propia página. La especificación oficial de HTML no define la lista de metadatos que se pueden incluir, por lo que las páginas tienen **libertad absoluta** para definir los metadatos que consideren adecuados. La etiqueta empleada para la definición de los metadatos es `<meta>`.

Etiqueta	<code><meta></code>
Atributos comunes	internacionalización
Atributos propios	<ul style="list-style-type: none">• <code>name = "texto"</code> - El nombre de la propiedad que se define (no existe una lista oficial de propiedades)• <code>content = "texto"</code> - El valor de la propiedad definida (no existe una lista de valores permitidos)• <code>http-equiv = "texto"</code> - En ocasiones, reemplaza al atributo "name" y lo emplean los servidores para adaptar sus respuestas al documento• <code>scheme = "texto"</code> - Indica el esquema que se debe emplear para interpretar el valor de la propiedad
Tipo de elemento	-
Descripción	Permite definir el valor de los metadatos que forman la metainformación del documento



3. METADATOS

Los metadatos habituales utilizan solamente los atributos `name` y `content` para definir el **nombre** y el **valor** del metadato:

```
<meta name="autor" content="Enrique GC" />
```

No obstante, algunas etiquetas `<meta>` muy utilizadas hacen uso del atributo `http-equiv` para indicar que el valor establecido por este metadato puede ser **utilizado por el servidor** al entregar la página al navegador del usuario. *El siguiente metadato indica al servidor que el contenido de la página es código HTML y su codificación de caracteres es UTF-8:*

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

3. METADATOS

El atributo `schema` no suele utilizarse, aunque permite proporcionar **información de contexto** para que el navegador interprete correctamente el valor del metadato. *En el siguiente ejemplo, el atributo `schema` indica al navegador que el valor del metadato hace referencia al código ISBN:*

```
<meta schema="ISBN" name="identificador"
      content="789-1392349610">
```

Aunque **no existe una lista oficial** con los metadatos que se pueden definir, algunos de ellos se utilizan en tantas páginas que se han convertido prácticamente en un estándar.



3. METADATOS

- Definir el **autor** del documento:

```
<meta name="author" content="Enrique GC" />
```

- Definir el **programa** con el que se ha creado el documento:

```
<meta name="generator" content="VisualStudioCode" />
```

- Definir la **codificación de caracteres** del documento:

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

```
<meta http-equiv="Content-Type" content="text/html;  
charset=ISO-8859-1" />
```

- Definir el **copyright** del documento:

```
<meta name="copyright" content="mipagina.es" />
```



3. METADATOS

- Definir el **comportamiento** de los **buscadores**:

```
<meta name="robots" content="index, follow" />
```

- Definir las **palabras clave** que definen el contenido del documento:

```
<meta name="keywords" content="diseño, css, hojas de  
estilos, web, html" />
```

- Definir una **breve descripción** del sitio:

```
<meta name="description" content="Artículos sobre diseño  
web, usabilidad y accesibilidad" />
```

4. DOCTYPE

El estándar XHTML deriva de XML, por lo que comparte con el muchas de sus **normas y sintaxis**. Uno de los conceptos fundamentales de XML es la utilización del **DTD** o *Document Type Definition* ("Definición del Tipo de Documento").

Un DTD es un documento que recoge el conjunto de **normas y restricciones** que deben cumplir los documentos de un determinado tipo.

Ej: se define un DTD para los documentos relacionados con libros, se puede fijar como norma que cada libro tenga un título y sólo uno, que tenga uno o más autores, que el número de páginas pueda ser opcional, etc.

4. DOCTYPE

El conjunto de normas, obligaciones y restricciones que se deben seguir al crear un documento de un determinado tipo, se recogen en su correspondiente DTD. El estándar XHTML define el DTD que deben seguir las páginas y documentos XHTML. En este documento se definen las **etiquetas** que se pueden utilizar, los **atributos** de cada etiqueta y el **tipo de valores** que puede tener cada atributo.

En realidad, la versión 1.0 del estándar de XHTML define **tres DTD diferentes**. Para indicar el DTD utilizado al crear una determinada página, se emplea una etiqueta especial llamada `doctype`. La etiqueta `doctype` es el único elemento que se incluye **fuera de la etiqueta `<html>` de la página**. De hecho, la declaración del `doctype` es lo **primero** que se debe incluir en una página web, antes incluso que la etiqueta `<html>`.

4. DOCTYPE

Para que una página XHTML sea **correcta** y **válida** es imprescindible que incluya el correspondiente `doctype` que indica el DTD utilizado. A continuación se muestran los tres DTD que se pueden utilizar al crear páginas XHTML:

XHTML 1.0 Estricto

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

Se trata de la variante con las normas más **estrictas** y las restricciones más **severas**. Las páginas web que incluyan este `doctype`, no pueden utilizar atributos relacionados con el aspecto de los contenidos, por lo que requiere una separación total de código HTML y estilos CSS.

XHTML 1.0 Transitorio

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

Se trata de una variante menos estricta que la anterior, ya que permite el uso de **algunos atributos** HTML relacionados con el aspecto de los elementos.

XHTML 1.0 Frameset

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

Esta última variante la utilizan las páginas que están formadas por frames, una práctica completamente **desaconsejada** y que hoy en día sólo utilizan los sitios web obsoletos.

4. DOCTYPE

Si no tienes claro el DTD que más te conviene, deberías utilizar el **XHTML 1.0 Transitorio**, ya que es más fácil crear páginas web válidas. Si tienes conocimientos avanzados de XHTML, puedes utilizar XHTML 1.0 Estricto.

Por otra parte, además del DOCTYPE apropiado, también es necesario que las páginas web indiquen el namespace asociado. Un namespace en un documento XML permite diferenciar las **etiquetas** y **atributos** que pertenecen a cada **lenguaje**.

Si en un mismo documento se mezclan etiquetas de dos o más lenguajes derivados de XML (*XHTML* y *SVG*) y que tienen el mismo nombre, no se podría determinar a qué lenguaje pertenece cada etiqueta y por tanto, no se podría **interpretar** esa etiqueta o ese atributo. Los namespaces se indican mediante una URL.



4. DOCTYPE

El namespace **que utilizan todas las páginas XHTML (independientemente de la versión y del DOCTYPE)** es `http://www.w3.org/1999/xhtml` y se indica de la siguiente manera:

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
...
```

```
</html>
```

De esta forma, es habitual que las páginas XHTML comiencen con el siguiente código:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="es" xml:lang="es">
```



4. DOCTYPE

La etiqueta en HTML5 queda así:

<!DOCTYPE html>

Lo importante del doctype en HTML5

- El doctype en HTML5 es más sencillo que nunca, por lo que ya no hay excusas para utilizarlo!
- En doctype informas del DTD que utilizas en tu documento, y HTML5 acepta todo lo que admite XHTML, además de las nuevas etiquetas.
- El DTD nos indica la estructura de etiquetas del documento, los atributos que pueden tener las etiquetas, los eventos, etc.
- Tenemos que crear nuestro documento respetando el DTD indicado en nuestro doctype. Esto no es más que conocer y respetar los estándares web de la versión html definida en doctype.
- Podemos comprobar si nuestro documento sigue el estándar con el validador web, de la W3C