

Etiquetas cabecera HTML

Las etiquetas de cabecera o head no tienen representación o visualización en el contenido que renderiza (pintar) el navegador. Por ello son meta-información adicional.

Etiquetas de relación

Bienvenidos a la sección de **cabeceras** de HTML. Cuando hablamos de la estructura de un documento de **HTML5** una parte que tratamos fue la cabecera o `<head>` ahora vamos a ver las etiquetas más importantes que se suelen usar dentro de estas.

El `<head>` elemento es un contenedor de metadatos (datos sobre datos). Los metadatos HTML son datos que no se muestran al usuario sobre el documento HTML. Además, contiene el título del documento y el enlace a las hojas de estilos que asociemos.

En la etiqueta `<head>` se sitúa el título del documento. Aquí también encontraremos el enlace a la hoja de estilos que asociemos a nuestro documento.

En resumen, el elemento HTML `<head>` se coloca entre la `<html>` etiqueta y la `<body>` etiqueta y es un contenedor para los siguientes elementos: `<title>`, `<style>`, `<meta>`, `<link>`, `<script>` y `<base>`.

Etiquetas de relación: Título y Codificación

Dentro de `<head>` es aconsejable usar las etiquetas `<title>` y `<meta>`.

```
<head>
  <title>Título</title>
  <meta charset="utf-8" />
</head>
```

La etiqueta `<meta>` con el atributo `charset` indica al navegador que utilice la codificación UTF-8 para mostrar el texto. De este modo evitaremos problemas con vocales acentuadas, o caracteres como "ñ".

Y el `<title>` es el nombre que se muestra en la ventana de nuestro navegador. Si lo cambiamos y vamos al navegador veréis su uso.

Etiquetas de relación: Favicon

El Favicon es un icono que hace referencia a nuestra aplicación, es una pequeña imagen que se muestra junto al título de la página en la pestaña del navegador. Por defecto nuestro navegador tiene un elemento en el caso de chrome una esfera pero podemos colocar el que queramos.



Para agregar un favicon a nuestra web, tenemos que crear una carpeta en la raíz de nuestro proyecto llamada `assets` y guardar la imagen de favicon en esta carpeta (usaremos la imagen del logo de ThePower para el ejemplo). Un nombre común para una imagen de favicon es `"favicon.ico"`.

Para ello tenemos que hacer uso de la etiqueta `<link>` con dos atributos `rel="icon"` y `href="ruta-imagen"` como en el ejemplo:

```
<head>
  <link rel="icon" href="/favicon.ico" />
</head>
```

Si vamos al navegador veréis que donde teníamos la esfera ahora vemos el logo de ThePower.

Etiquetas de relación: Añadir hoja de estilos CSS y Script

También dentro de nuestra cabecera enlazamos tanto los ficheros de estilos como los de scripting, para ello vamos a crear un fichero CSS3 en la raíz del proyecto llamado `styles.css` y otro JS en la raíz del proyecto también llamado `main.js`. Y los vamos a enlazar.

```
<link rel="stylesheet" href="styles.css" type="text/css" />
<script type="text/javascript" src="./main.js"></script>
```

Como veis para estilos usamos la etiqueta `<link>` y para javascript la etiqueta `<script>`. Para comprobar que ya están enlazados vamos a seguir los pasos a continuación:

```
<!-- En el index.html -->

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script type="text/javascript" src="./main.js"></script>
  <link rel="stylesheet" href="./styles.css">
  <title>My first document</title>
</head>
<body>
  <p>Esto es una prueba de que funciona la conexión entre archivos</p>
</body>
</html>
```

```
/* Styles.css */

p {
  color: blueviolet;
}
```

Una vez creados todos los documentos con su información, abrimos el `index.html` en el navegador para ver si hemos podido conectar o no nuestros ficheros.

Etiquetas de relación: Meta Viewport

El `name="viewport"` dentro de la etiqueta `<meta>` es el área visible del usuario de una página web. Varía según el dispositivo: será más pequeño en un teléfono móvil que en la pantalla de un portátil...

Cuando tenemos una aplicación web optimizada para dispositivos móviles deberemos añadir la siguiente etiqueta:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Esto le da al navegador instrucciones sobre cómo controlar las dimensiones y la escala de la página.

`width=device-width` establece el ancho de la página (que variará según el dispositivo).

La `initial-scale=1.0` parte establece el nivel de zoom inicial cuando el navegador carga la página por primera vez.

Os dejamos un ejemplo visual de la vista de móvil usando o no la etiqueta meta para que lo tengáis más claro:



Sin la etiqueta meta de la ventana gráfica



Con la metaetiqueta de ventana gráfica

Etiquetas de relación: Compatibilidad con navegadores

X-UA-compatible en html es otro meta-tag útil en la página HTML y este meta-tag se relaciona mayormente con el navegador. **El modo Edge** le dice a Internet Explorer que muestre el contenido en el modo más alto disponible.

Y se representa del siguiente modo:

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

Etiquetas de relación: ¿Qué hemos aprendido?

1. Qué son las etiquetas de cabeceras y los tipos que existen.
2. Que el **<head>** contiene metadatos o cosas que como usuarios no vemos en la web y el **<body>** contendrá toda la parte visual.
3. Los elementos que podemos usar dentro de la **<head>** etiqueta

Etiquetas metadatos

Esta etiqueta de metadatos será `<meta>` y ofrece una gran cantidad de opciones dentro de la cabecera de un documento HTML. Puede contener los atributos `name` y `content` y servirán para indicar metadatos al documento.

Son usados por los buscadores para definir la información principal de nuestra web (temática, descripción), por lo que será muy importante que lo tengamos correctamente configurados.

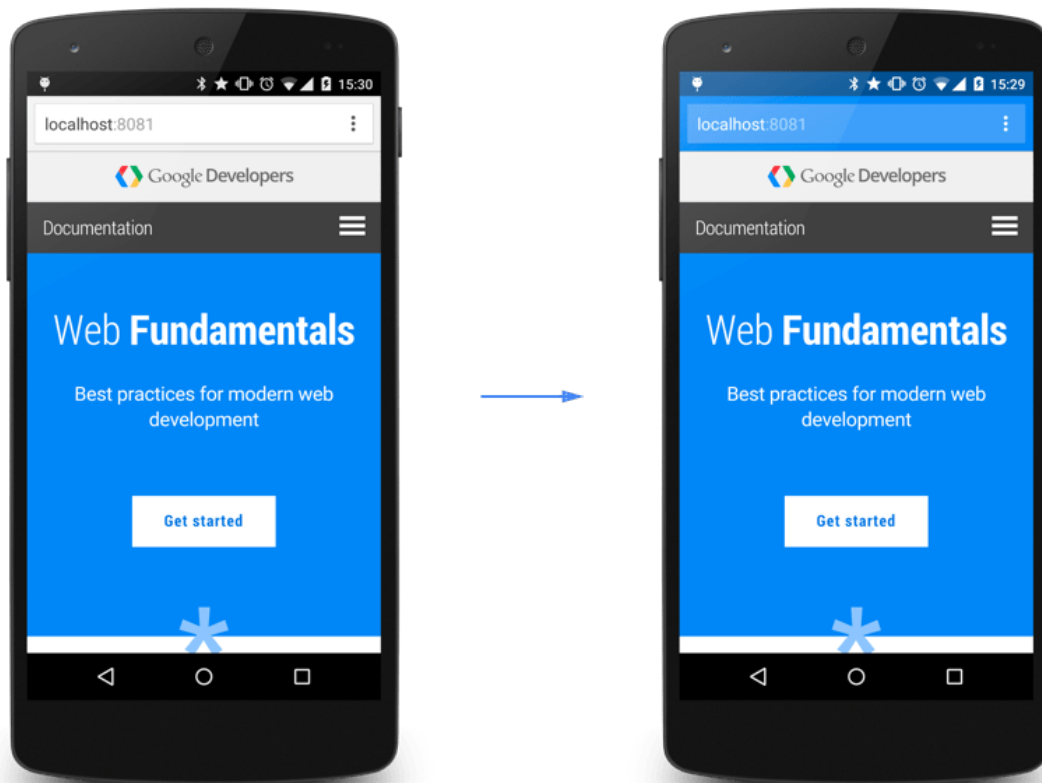
Algunos de los metadatos que podemos indicar son:

Atributo	Contenido	Descripción
description	texto	Indica la descripción de la página que aparece en buscadores.
keywords	palabras clave	Lista de palabras clave separadas por comas. Google no la tiene en cuenta.
author	nombre	Indica el nombre del autor de la página.
theme-color	color	Color en formato hexadecimal de la barra de navegación del navegador.
viewport	parámetros	Comportamiento de la región visible del navegador. Ver Responsive.

Además, en el navegador Google Chrome está disponible el atributo `theme-color`, que cambiará el color de la barra de dirección del navegador. Además, os dejamos un ejemplo de cada atributo:

```
<head>
  <meta name="theme-color" content="#4285f4" />
  <meta name="description" content="Esta es la descripción en metadatos" />
```

```
<meta name="keywords" content="web, programar, site, metadatos" />
<meta name="author" content="Alberto Rivera" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
</head>
```



Etiquetas metadatos: Otros metadatos incluidos por Google son

usando el atributo `google`, los valores `nositelinkssearchbox`, que indica a google que no muestre el buscador en sitelinks y el valor `notranslate` para indicar que no se debe traducir la página.

Por último, el atributo `robots` que recibirá parámetros y se le indica al robot de google si debe indexar la página o no. Estos robots de Google son como “spiders” que se encargan de entrar en nuestra web y establecer un puntaje sobre accesibilidad,

experiencia de usuario y recolectan datos de la página. Simulan ser un usuario que hace clic en enlaces, navega por la web y demás acciones.

Algunos de los parámetros que se le pueden pasar a estos robots son:

Parámetro	Significado
index	Sugiere a Google que se quiere indexar el contenido de la página en el buscador.
noindex	Indica a Google que no debe indexar el contenido de la página en el buscador.
follow	Sugiere a Google que siga los enlaces que encuentre en la página.
nofollow	Indica a Google que no debe seguir los enlaces que encuentre en la página.
nosnippet	Indica a Google que no muestre fragmentos (descripción, etc...).
noimageindex	Indica a Google que no indexe la página en resultados de Google Imágenes.
none	Equivalente a indicar los parámetros noindex, nofollow.

```
<head>
  <meta name="google" content="nositelinkssearchbox" />
  <meta name="google" content="notranslate" />
  <meta name="robots" content="index, nofollow" />
  <meta name="robots" content="none" />
  <meta name="googlebot-news" content="nosnippet">
</head>
```

Existe además un último atributo [http-equiv](#) usado para que los robots del buscador sepan que se deben cambiar las cabeceras HTTP al realizar ciertas acciones. Hay que tener en cuenta que este elemento no tiene soporte en algunos navegadores.

```
<head>
  <!-- refresh redirige a la URL pasados los segundos indicados -->
```



```
<meta http-equiv="refresh" content="300;url=http://www.google.com/" />
<!-- Expires es la fecha a partir de la que consideramos la página expirada -->
<meta http-equiv="expires" content="Fri, 29 Apr 2016 12:56:00 GMT" />
<!-- Pragma indica si queremos que se guarde o no la caché -->
<meta http-equiv="pragma" content="no-cache" />
<!-- Este es el especial que indica a Internet si debe o no guardar la caché -->
<meta http-equiv="cache-control" content="no-cache" />
</head>
```

Etiquetas metadatos: Qué hemos aprendido

1. Qué son las etiquetas de metadatos y para qué sirven.
2. Los atributos especiales de Google.
3. Cómo funcionan los robots de Google.
4. Cómo usar el atributo http-equiv.

RRSS

Igual que para añadir título y descripción a una página web lo hacemos con etiquetas de metadatos, existen un tipo de etiquetas de metadatos especiales orientadas a redes sociales. Así, se pueden añadir títulos y descripciones especiales para estos sites.

Por ejemplo Facebook utiliza el protocolo Open Graph, que lo que hace es que, cuando nosotros insertamos en nuestra página web un enlace de facebook este protocolo lo analizará en busca de metadatos que puedan incluir información adicional.

En caso de encontrarlos, creará sus propios metadatos. Lo que esto nos posibilita es personalizar nuestra web aún más ya que convierte este enlace insertado en información como si la vieses en la propia red social.

Las propiedades obligatorias en Open Graph serán **og:title**, que hace referencia al título del objeto, **og:type**, que es el tipo de objeto que insertamos. **og:image**, que será la

imagen que represente al objeto que estamos insertando y **og:url**, que será la url del objeto que se usará como ID.

Un ejemplo de la web de Open Graph es:

```
<head>
<title>The Rock (1996)</title>
<meta property="og:title" content="The Rock" />
<meta property="og:type" content="video.movie" />
<meta property="og:url" content="https://www.imdb.com/title/tt0117500/" />
<meta property="og:image" content="https://ia.media-imdb.com/images/rock.jpg" />
</head>
```

Por su parte Twitter utiliza Twitter Cards, que servirá para lo mismo que la anterior herramienta. Esta permitirá modificar datos como el formato y tipo de la información mostrada, el nombre del autor de la página del enlace, el título, descripción e imagen de presentación, etc.

```
<head>
  <meta property="twitter:card" content="mi_imagen_perfil" />
  <meta property="twitter:creator" content="@Alberto" />
  <meta property="twitter:title" content="Título" />
  <meta property="twitter:description" content="Descripción" />
  <meta property="twitter:image:src" content="mi_imagen.jpg" />
</head>
```

Hay que aclarar que para usar Twitter Cards debemos registrarnos previamente en Twitter Developers y deberán analizar si nuestra web cumple o no los requisitos.

RRSS: Qué hemos aprendido

1. Qué son las etiquetas de redes sociales.
2. Cómo usar las herramientas que las redes sociales ponen a nuestra disposición.