



# **HTML**

## **Capítulo 4**

*Lenguajes de Marcas y Sistemas de Gestión de Información*

*Curso 2019/2020*

# + **Capítulo 4: ENLACES**

1. Introducción
2. URL
3. Enlaces Relativos y Absolutos
4. Enlaces Básicos
5. Enlaces Avanzados
6. Otros Tipos de Enlaces
7. Ejemplos de Enlaces



# 1. INTRODUCCIÓN

La incorporación del hipertexto es una de las claves del éxito del lenguaje HTML, ya que permite crear **documentos interactivos** que proporcionan información adicional cuando se solicita. El elemento principal del hipertexto es el "hiperenlace", también llamado "enlace web" o simplemente "**enlace**".

Los enlaces se utilizan para establecer **relaciones entre dos recursos**. La mayoría de enlaces relacionan páginas web, pero es posible enlazar imágenes, documentos o archivos.

Una característica que no se suele tener en cuenta en los enlaces es que están formados por **dos extremos** y un sentido. En otras palabras, el enlace comienza en un recurso y apunta hacia otro recurso. Cada uno de estos extremos se llaman "*anchors*".



## 2. URL

Antes de empezar a crear enlaces, es necesario comprender y dominar el concepto de **URL**.

Las URL son esenciales para crear los enlaces, pero también se utilizan en otros elementos HTML como las imágenes y los formularios.

La URL de un recurso tiene dos **objetivos** principales:

- **Identificar** de forma única a ese recurso.
- **Permitir** localizar de forma eficiente ese recurso.



## 2. URL

Las URL permiten que cada página HTML publicada en Internet tenga un nombre único que permita diferenciarla de las demás → Es posible crear enlaces que apunten de forma inequívoca a una determinada página.

Si se accede a la página principal de Google, la dirección que muestra el navegador es:

`http://www.google.com`

La cadena de texto `http://www.google.com` es la URL completa de la página principal de Google. La URL de las páginas es imprescindible para crear los enlaces, ya que permite distinguir una página de otra.



## 2. URL

El segundo objetivo es el de permitir la localización eficiente de cada recurso de Internet. Una URL sencilla siempre está formada por las mismas partes:

`http://www.nombrepagina.es/html/ejemplo.html`

- **Protocolo** (`http://`): Mecanismo que usa el navegador para acceder a ese recurso. Las webs seguras utilizan `https://`.
- **Servidor** (`www.nombrepagina.es`): Servidor en el que se encuentra guardada la página que se quiere acceder. Los navegadores son capaces de obtener la dirección de cada servidor a partir de su nombre (DNS).
- **Ruta** (`/html/ejemplo.html`): Camino que se debe seguir, una vez que se ha llegado al servidor, para localizar el recurso.



## 2. URL

Existen URL complejas formadas por más partes:

`http://www.pagina.com/ejemplo/web2019?page=1#50`

- **Protocolo** (`http://`).
- **Servidor** (`www.pagina.com`).
- **Ruta** (`/ejemplo/web2019`).
- **Consulta** (`?page=1`): Información adicional necesaria para que el servidor localice correctamente el recurso que se quiere acceder. Siempre comienza con el **carácter** `?` y contiene una sucesión de palabras separadas por `=` y `&`.
- **Sección** (`#50`): Permite que el navegador se posicione automáticamente en una sección de la página web. Siempre comienza con el **carácter** `#`.



## 2. URL

Las URL utilizan los caracteres `:`, `=`, `&` y `/` para separar sus partes. Estos caracteres están **reservados** y no se pueden utilizar libremente.

Si es necesario incluir estos caracteres reservados y especiales en una URL, se sustituyen por **combinaciones de caracteres seguros**. Esta sustitución se denomina **codificación de caracteres** y el servidor realiza el proceso inverso (decodificación) cuando le llega una URL con los caracteres codificados.

Carácter original	Carácter codificado	Carácter original	Carácter codificado
/	%2F	?	%3F
:	%3A	@	%40
=	%3D	&	%26
"	%22	\	%5C
'	%27	~	%7E
(espacio en blanco)	%20	#	%23



## + 2. URL

Aunque desde hace tiempo ya es posible incluir en las URL caracteres de otros idiomas que no sean el inglés, aún **no es completamente seguro** utilizar estos caracteres en las URL. Si se utilizan letras como ñ, á, é o ç es posible que algunos navegadores no las interpreten de forma correcta.

Carácter original	Carácter codificado	Carácter original	Carácter codificado
ñ	%F1	Ñ	%D1
á	%E1	Á	%C1
é	%E9	É	%C9
í	%ED	Í	%CD
ó	%F3	Ó	%D3
ú	%FA	Ú	%DA
ç	%E7	Ç	%C7



## 2. URL

Es fácil crear URL correctas sin caracteres problemáticos:

`<!-- URL problemática -->`

`http://www.ejemplo.com/estaciones/otoño.html`

`<!-- URL correcta -->`

`http://www.ejemplo.com/estaciones/oto%F1o.html`

`<!-- URL problemática -->`

`http://www.ejemplo.com/ruta/nombre página.html`

`<!-- URL correcta -->`

`http://www.ejemplo.com/ruta/nombre%20p%Elgina.html`

# 3. ENLACES RELATIVOS Y ABSOLUTOS

Las páginas web habituales suelen contener decenas de enlaces de diferentes tipos.

20  
minutos

Tu Ciudad Nacional Internacional Deportes Cultura Opinión Gente y TV Más ▾

ANDALUCÍA > ALMERÍA CÁDIZ CÓRDOBA GRANADA MÁS ▾



¿CUÁNTA GENTE ES CAPAZ DE VER LA REALIDAD DE UNA NOTICIA ECONÓMICA?

**LA iNFORMACiÓN**  
UN NUEVO ENFOQUE PARA LA ECONOMÍA

RIGOR. INDEPENDENCIA. ENFOQUES PROPIOS. EXCLUSIVAS, ACTUALIDAD y MUCHO MÁS en [WWW.LAINFORMACION.COM](http://WWW.LAINFORMACION.COM)

DIRECTO 1-0

Sánchez: "El 155 lo puede aplicar un Gobierno en funciones sin problemas"

## SEVILLA



UNIA.- La sede Cartuja de la UNIA abre este martes sus cursos de verano

Vox pide que el Pacto por la

### BLOGS DE 20MINUTOS



CAPEANDO LA CRISIS  
Una campaña de luchadores contra el despilfarro de alimentos



EN BUSCA DE UNA SEGUNDA OPORTUNIDAD  
¿Sabías que los gatos tricolor y carey casi siempre son hembras?



LA CIMA 2030  
Las vacas no dijeron ni mu



## 3. ENLACES RELATIVOS Y ABSOLUTOS

Cuando se pincha sobre algunos enlaces, el navegador abandona el sitio web para acceder a páginas que se encuentran en otros sitios. Estos enlaces se conocen como "**enlaces externos**". Sin embargo, la mayoría de enlaces de un sitio web apuntan a páginas del propio sitio web, por lo que se denominan "**enlaces internos**".

La otra característica que diferencia a los enlaces es si el enlace es absoluto o relativo:

- ❑ **URL Absolutas:** Incluyen todas las partes de la URL (protocolo, servidor y ruta) por lo que no se necesita más información para obtener el recurso enlazado.
- ❑ **URL Relativas:** Prescinden de algunas partes de las URL para hacerlas más breves. Es necesario disponer de información adicional para obtener el recurso enlazado. Las URL relativas se construyen a partir de las URL absolutas y prescinden de la parte del protocolo, del nombre del servidor e incluso de parte o toda la ruta del recurso enlazado.

## + 3. ENLACES RELATIVOS Y ABSOLUTOS

Disponemos de una página publicada en:

```
http://www.ejemplo.com/ruta1/ruta2/pagina1.html
```

Queremos incluir en ella un enlace a otra página:

```
http://www.ejemplo.com/ruta1/ruta2/pagina2.html
```

El **enlace** debería utilizar la URL completa de la segunda página.

Las **URL completas** también se llaman **URL absolutas**, ya que el navegador no necesita disponer de información adicional para localizar el recurso enlazado. Si se utilizan siempre las URL absolutas, los enlaces están completamente definidos.



# 3. ENLACES RELATIVOS Y ABSOLUTOS

Escribir siempre las URL completas es bastante tedioso, cuesta mucho tiempo y hace imposible cambiar la ubicación de los contenidos de un sitio web → Casi todos los sitios web de Internet utilizan **URL relativas** siempre que es posible.

Una URL relativa es una **versión abreviada de una URL absoluta**. Su objetivo es eliminar todas las partes de la URL absoluta que se pueden adivinar a partir de la información de contexto de la página web.

Las URL relativas aprovechan la inteligencia de los navegadores para crear URL incompletas que los navegadores pueden completar deduciendo la información que falta.



## 3. ENLACES RELATIVOS Y ABSOLUTOS

Considerando el ejemplo anterior, la URL a la que se quiere enlazar utiliza el mismo protocolo y se encuentra en el mismo servidor que la página origen, por lo que la URL relativa puede prescindir de esas partes:

### ■ *URL Absoluta*

```
http://www.ejemplo.com/ruta1/ruta2/pagina2.html
```

### ■ *URL Relativa*

```
/ruta1/ruta2/pagina2.html
```

## 3. ENLACES RELATIVOS Y ABSOLUTOS

Las dos URL son equivalentes porque cuando no se indica el protocolo y el servidor de una URL, los navegadores suponen que son los mismos que los de la página origen. Cuando el navegador encuentra la URL `/ruta1/ruta2/pagina2.html`, realiza lo siguiente:

- ① La URL no es absoluta, por lo que se debe determinar la URL relativa partir de la URL absoluta para poder cargar el recurso enlazado.
- ② A la URL relativa le falta el protocolo y el servidor, por lo que se supone que son los mismos que los de la página origen.
- ③ Se añaden las partes que faltan a la URL relativa para obtener la URL absoluta:

```
http:// + www.ejemplo.com + /ruta1/ruta2/pagina2.html =  
http://www.ejemplo.com/ruta1/ruta2/pagina2.html
```





# 3. ENLACES RELATIVOS Y ABSOLUTOS

## 1) El origen y el destino del enlace se encuentran en el MISMO DIRECTORIO

Si desde una página web se quiere enlazar un recurso que se encuentra en el mismo directorio del servidor, la URL relativa puede prescindir de todas las partes de la URL absoluta salvo el nombre del recurso enlazado.

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>
Página enlazada	Página web llamada <code>pagina2.html</code> y que se encuentra en el mismo directorio
URL absoluta	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina2.html</code>
URL relativa	<code>pagina2.html</code>



## 3. ENLACES RELATIVOS Y ABSOLUTOS

### 2) El destino del enlace se encuentra CERCA de su origen y en un NIVEL SUPERIOR

La URL relativa debe indicar de alguna manera que es necesario subir un nivel en la jerarquía de directorios para llegar hasta el recurso.

Para indicar al navegador que debe **subir un nivel**, se incluyen dos puntos y una barra (../) en la ruta del recurso enlazado → Cada vez que aparece ../ en una URL relativa, significa que se debe subir un nivel.

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>
Página enlazada	Página web llamada <code>pagina2.html</code> y que se encuentra en el directorio superior llamado <code>ruta2</code>
URL absoluta	<code>http://www.ejemplo.com/ruta1/ruta2/pagina2.html</code>
URL relativa	<code>../pagina2.html</code>

**¿SUBIR DOS NIVELES?**



## 3. ENLACES RELATIVOS Y ABSOLUTOS

### 3) El destino del enlace se encuentra CERCA de su origen y en un NIVEL INFERIOR

Es muy similar al anterior, pero más sencillo. Si el recurso enlazado se encuentra en algún directorio inferior al que se encuentra el origen, sólo es necesario indicar el nombre de los directorios a los que debe entrar el navegador.

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>
Página enlazada	Página web llamada <code>pagina2.html</code> y que se encuentra en un directorio inferior llamado <code>ruta4</code>
URL absoluta	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/ruta4/pagina2.html</code>
URL relativa	<code>ruta4/pagina2.html</code>



## 3. ENLACES RELATIVOS Y ABSOLUTOS

### 4) El origen y el destino del enlace se encuentran MUY ALEJADOS

El origen y el destino de un enlace se encuentran muy alejados (pero en el mismo servidor). En estos casos, lo más sencillo es **indicar la ruta completa hasta el recurso enlazado** comenzando desde la raíz del servidor web. Por lo tanto, estas URL relativas sólo omiten el protocolo y el nombre del servidor.

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>
Página enlazada	Página web llamada <code>pagina2.html</code> y que se guarda en un directorio llamado <code>ruta7</code> que se encuentra en la raíz del servidor
URL absoluta	<code>http://www.ejemplo.com/ruta7/pagina2.html</code>
URL relativa	<code>/ruta7/pagina2.html</code>



## 3. ENLACES RELATIVOS Y ABSOLUTOS

Cuando la URL relativa comienza por `/`, el navegador considera que es la ruta completa comenzando desde la raíz del servidor, por lo que sólo le añade el protocolo y el nombre del servidor origen.

Cuatro posibles casos de URL relativas y el procedimiento que sigue el navegador para convertirlas en URL absolutas:

Si la URL relativa...	El navegador la transforma en URL absoluta...
...sólo consiste en el nombre de un recurso	...añadiendo el protocolo, servidor y ruta completa del origen del enlace
...comienza por <code>../</code>	...añadiendo el protocolo y servidor del origen del enlace, subiendo un nivel en la jerarquía de directorios y añadiendo el resto de la ruta incluida en la URL relativa
...comienza por <code>/</code>	...añadiendo el protocolo y servidor del origen del enlace
En cualquier otro caso	...añadiendo el protocolo, servidor y ruta completa del origen del enlace, a la que se añade la ruta incluida en la URL relativa

## + 4. ENLACES BÁSICOS

Los enlaces se crean mediante la etiqueta `<a>`:

Etiqueta	<code>&lt;a&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> , <a href="#">eventos</a> y <a href="#">foco</a>
Atributos propios	<ul style="list-style-type: none"><li>• <code>name = "texto"</code> - Permite nombrar al enlace para que se pueda acceder desde otros enlaces</li><li>• <code>href = "url"</code> - Indica la URL del recurso que se quiere enlazar</li></ul>
Tipo de elemento	En línea
Descripción	Se emplea para enlazar todo tipo de recursos

Las URL de los enlaces pueden ser absolutas, relativas, internas y externas.

Para crear un enlace que apunte a la página principal de Google solamente habría que incluir lo siguiente en un documento HTML:

```
<a href="http://www.google.es">Página de Google</a>
```

## + 4. ENLACES BÁSICOS

El atributo `href` indica una URL así que puede apuntar a cualquier tipo de recurso al que pueda acceder el navegador. El siguiente enlace apunta a una imagen, que se mostrará en el navegador cuando el usuario pinche sobre el enlace:

```
<a href="http://www.ejemplo.com/fondo_escritorio.jpg">Imagen  
    interesante para un fondo de escritorio</a>
```

De la misma forma, los enlaces pueden apuntar directamente a documentos PDF, Word, etc.

```
<a href="http://www.ejemplo.com/informe.pdf">Descargar  
    informe completo [PDF]</a>
```

```
<a href="http://www.ejemplo.com/informe.docx">Descargar  
    informe completo [DOC]</a>
```

## + 4. ENLACES BÁSICOS

Un truco muy útil con los enlaces es el uso de URL relativas para poder **volver al inicio del sitio web** desde cualquier página web interior:

```
<a href="/">Volver al inicio</a>
```

El enlace anterior utiliza una URL relativa con una ruta que apunta directamente a la raíz del servidor. Para obtener la URL absoluta, el navegador añade el mismo protocolo y el mismo nombre de servidor de la página en la que se encuentra el enlace.

El resultado es que cuando se pincha ese enlace, siempre **se vuelve al inicio del sitio web**, independientemente de la página en la que se incluya el enlace.



# + 4. ENLACES BÁSICOS

El otro atributo básico de la etiqueta `<a>` es `name`, que permite definir enlaces dentro de una misma página web. Este tipo de enlaces son especiales, ya que la URL de la página siempre es la misma para todas las secciones y por tanto, debe añadirse otra parte a las URL para identificar cada sección.

```
<a name="primera_seccion"></a>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id ligula eu felis adipiscing ultrices. Duis gravida leo ut lectus. Praesent condimentum mattis ligula.</p>
```

...

```
<a name="segunda_seccion"></a>
```

```
<p>Pellentesque malesuada. In in lacus. Phasellus erat erat, lacinia a, convallis eu, nonummy et, odio. Aenean urna elit, ultrices id, placerat varius, facilisis eget, dolor.</p>
```

...

## + 4. ENLACES BÁSICOS

El atributo `name` permite crear "enlaces vacíos" que hacen referencia a **secciones dentro de una misma página**. Una vez definidos los "enlaces vacíos", es posible crear un enlace que apunte directamente a una sección concreta de una página:

```
<!-- Enlace normal a la página -->
```

```
<a href="http://www.ejemplo.com/pagina1.html">Enlace a la  
página 1</a>
```

```
<!-- Enlace directo a la segunda sección de la página -->
```

```
<a href="http://www.ejemplo.com/  
pagina1.html#segunda_seccion">Enlace a la sección 2 de la  
página 1</a>
```

## + 4. ENLACES BÁSICOS

También es posible utilizar este tipo de enlaces con URL relativas en una misma página. El siguiente ejemplo añade enlaces de tipo "Volver al inicio de la página" en varias secciones:

```
<a name="inicio"></a>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id ligula  
eu felis adipiscing ultrices. Duis gravida leo ut lectus. Praesent condimentum  
mattis ligula.</p>
```

```
<a href="#inicio">Volver al inicio de la página</a>
```

...

```
<p>Pellentesque malesuada. In in lacus. Phasellus erat erat, lacinia a,  
convallis eu, nonummy et, odio. Aenean urna elit, ultrices id, placerat varius,  
facilisis eget, dolor.</p>
```

```
<a href="#inicio">Volver al inicio de la página</a>
```

...

## 5. ENLACES AVANZADOS

La definición completa de `<a>` es muy compleja, ya que dispone de varios atributos específicos importantes:

<b>Etiqueta</b>	<code>&lt;a&gt;</code>
<b>Atributos comunes</b>	<a href="#">básicos</a> , <a href="#">internacionalización</a> , <a href="#">eventos</a> y <a href="#">foco</a>
<b>Atributos propios</b>	<ul style="list-style-type: none"><li>• <code>name = "texto"</code> - Permite nombrar al enlace para que se pueda acceder desde otros enlaces</li><li>• <code>href = "url"</code> - Indica la URL del recurso que se quiere enlazar</li><li>• <code>hreflang = "codigo_idioma"</code> - Idioma del recurso enlazado</li><li>• <code>type = "tipo_de_contenido"</code> - Permite "avisar" al navegador sobre el tipo de contenido que se enlaza (imágenes, archivos, etc.) para que pueda prepararse en caso de que no entienda ese contenido</li><li>• <code>rel = "tipo_de_relacion"</code> - Describe la relación del documento actual con el recurso enlazado</li><li>• <code>rev = "tipo_de_relacion"</code> - Describe la relación del recurso enlazado con el documento actual</li><li>• <code>charset = "tipo_de_charset"</code> - Describe la codificación del recurso enlazado</li></ul>
<b>Tipo de elemento</b>	En línea
<b>Descripción</b>	Se emplea para enlazar todo tipo de recursos

## + 5.1. IDIOMA DEL ENLACES (`hreflang`)

El enlace puede indicar al navegador el **idioma del recurso** que se enlaza. Para establecer el valor del idioma, se utiliza un código estandarizado de dos letras:

Código	Idioma	Variación idiomática
en	Inglés	-
en-AU	Inglés	Australia
en-US	Inglés	Estados Unidos
es	Español	-
es-AR	Español	Argentina
es-ES	Español	España
es-MX	Español	México

## + 5.2. TIPO DE CONTENIDO (`type`)

Se utiliza para notificar al navegador sobre el **tipo de contenido que se enlaza**. Se indica mediante una cadena de texto cuyos posibles valores también están estandarizados.

Los valores de los contenidos más utilizados son los siguientes: `"text/html"` (páginas HTML), `"image/png"` (imágenes con formato PNG), `"image/gif"` (imágenes con formato GIF), `"text/css"` (hojas de estilo CSS), `"application/rss+xml"` (archivos RSS)...

## + 5.3. TIPO DE RELACIÓN (`rel` Y `rev`)

Los atributos `rel` y `rev` permiten indicar la relación que la página actual tiene con la página a la que se enlaza (atributo `rel`) y la relación que tiene la página enlazada con la página actual (atributo `rev`).

Los tipos de relación definidos son los siguientes:

- ❑ `alternate` – Indica que es una versión alternativa al documento actual (puede ser una versión en otro idioma o una versión preparada para otro medio, como una impresora o un dispositivo móvil).
- ❑ `stylesheet` – Indica que se ha enlazado una hoja de estilos.
- ❑ `start` – Indica que se trata del primer documento de una colección de documentos (por ejemplo el primer capítulo de un libro).

## + 5.3. TIPO DE RELACIÓN (`rel` Y `rev`)

- ❑ `next` – Indica que es el documento que sigue al actual dentro de una secuencia lógica de documentos (por ejemplo, los capítulos de un libro).
- ❑ `prev` – Indica que es el documento que precede al actual dentro de una secuencia lógica de documentos (por ejemplo, los capítulos de un libro).
- ❑ `contents` – Indica que el recurso enlazado es el documento que contiene la tabla de contenidos de la colección de documentos (por ejemplo, el índice de un libro).
- ❑ `bookmark` – Establece el enlace actual como un "marcador" o "favorito".



## + 5.4. CODIFICACIÓN DE CARACTERES (charset)

Los valores que se pueden utilizar también están estandarizados y las codificaciones más utilizadas son **UTF-8** y **ISO-8859-1**.

Los ejemplos anteriores de enlaces básicos se pueden rehacer utilizando algunos de los atributos definidos por la etiqueta `<a>`:

```
<a href="http://www.google.com" hreflang="en" type="text/html"
charset="UTF-8">Página principal de Google</a>
```

```
<a href="http://www.ejemplo.com/fondo_escritorio.jpg"
type="image/jpg">Imagen interesante para un fondo de
escritorio</a>
```



## 6. OTROS TIPOS DE ENLACES

La característica más importante de los enlaces anteriores es que el usuario debe activar la carga de los recursos: el navegador no carga ningún recurso enlazado con la etiqueta `<a>` a menos que el usuario pinche sobre el enlace.

Las páginas HTML pueden incluir otro tipo de enlaces que cargan los recursos automáticamente. Si una página HTML utiliza archivos CSS para aplicar estilos a sus contenidos, no es lógico que los enlace con la etiqueta `<a>` y espere a que el usuario pinche sobre el enlace para así **cargar los archivos CSS**. De la misma forma, muchas páginas web dinámicas necesitan que el navegador **cargue varios archivos JavaScript** para funcionar correctamente.

HTML define las etiquetas `<script>` y `<link>` para enlazar recursos que se deben **cargar automáticamente**. Cuando el navegador encuentra alguna de estas dos etiquetas, descarga los recursos enlazados y los aplica a la página web.

## 6. OTROS TIPOS DE ENLACES

La **etiqueta** `<script>` tiene dos modos de funcionamiento, ya que se emplea tanto para insertar un bloque de código JavaScript en la página como para enlazar un archivo JavaScript externo.

Etiqueta	<code>&lt;script&gt;</code>
Atributos comunes	-
Atributos propios	<ul style="list-style-type: none"><li>• <code>src = "url"</code> - Indica la dirección del archivo que contiene el código</li><li>• <code>type = "tipo_de_contenido"</code> - Permite "avisar" al navegador sobre el tipo de código que se incluye (normalmente JavaScript)</li><li>• <code>defer = "defer"</code> - El código no va a modificar el contenido de la página web</li><li>• <code>charset = "tipo_de_charset"</code> - Describe la codificación del código enlazado</li></ul>
Tipo de elemento	Bloque y en línea (también puede ser una etiqueta vacía)
Descripción	Se emplea para enlazar o definir un bloque de código (normalmente JavaScript)



## 6. OTROS TIPOS DE ENLACES

Aunque la etiqueta `<script>` permite enlazar código de varios lenguajes de programación, el uso habitual de `<script>` consiste en enlazar un archivo JavaScript externo:

```
<head>
```

```
    <script type="text/javascript" src="http://www.ejemplo.com/  
js/inicializar.js"></script>
```

```
</head>
```

El atributo `type` utilizado habitualmente para los archivos JavaScript es `"text/javascript"`. El atributo `src` es equivalente al atributo `href` de los enlaces creados con la etiqueta `<a>`. La URL indicada en el atributo `src` puede ser absoluta o relativa y externa o interna.



## 6. OTROS TIPOS DE ENLACES

Además de enlazar un archivo JavaScript externo, la misma etiqueta `<script>` también permite incluir en la página web un **bloque de código JavaScript**:

```
<html>

<head>

  <script type="text/javascript">

    //<![CDATA[

      window.onload = function() { alert("La página se ha cargado completamente"); }

    //]]>

  </script>

</head>

<body>

  ...

</body>

</html>
```

## 6. OTROS TIPOS DE ENLACES

La **segunda etiqueta** de XHTML para enlazar recursos es `<link>`, que permite enlazar y relacionar la página con otros recursos externos:

Etiqueta	<code>&lt;link&gt;</code>
Atributos comunes	<a href="#">básicos</a> , <a href="#">internacionalización</a> y <a href="#">eventos</a>
Atributos propios	<ul style="list-style-type: none"><li>• Los siguientes con el mismo significado que para la etiqueta "a": charset, href, hreflang, type, rel y rev</li><li>• media = "tipo_de_medio" - Indica el medio para el que debe aplicarse la relación</li></ul>
Tipo de elemento	Etiqueta vacía
Descripción	Se emplea para enlazar y establecer relaciones entre el documento y otros recursos



## 6. OTROS TIPOS DE ENLACES

Al contrario que `<script>`, la etiqueta `<link>` solamente se puede **incluir dentro de la cabecera del documento**. Se pueden añadir tantas etiquetas `<link>` como sean necesarias, pero siempre dentro de `<head>...</head>`.

El atributo `media` hace referencia al **medio para el que es válida la relación con el recurso enlazado**. Los medios disponibles también están estandarizados, siendo los más comunes `screen` para los contenidos mostrados en pantalla, `print` para las impresoras y `handheld` para los dispositivos móviles.

## 6. OTROS TIPOS DE ENLACES

El uso habitual de la etiqueta `<link>` es el de enlazar las hojas de estilos CSS utilizadas por las páginas web:

```
<head>
```

```
...
```

```
    <link rel="stylesheet" type="text/css" href="/css/
comun.css" />
```

```
</head>
```

Es habitual establecer los atributos `rel` y `type` para indicar el tipo de recurso enlazado y su relación con la página web. La URL del recurso enlazado se indica en el atributo `href`, que admite tanto URL absolutas como relativas.



## + 7. EJEMPLOS DE ENLACES

- ❖ Enlace al Inicio de un Sitio Web.
- ❖ Enlace a un Correo.
- ❖ Enlace a un Archivo FTP.
- ❖ Enlace a varias Hojas de Estilos.
- ❖ Enlazar el *favicon*.
- ❖ Enlazar Hojas de Estilo CSS, *favicon* y RSS.



## 7.1. ENLACE AL INICIO DE UN SITIO WEB

```
<a href="/">Inicio</a>
```

Al pulsar el enlace anterior desde cualquier página web, se vuelve directamente a la página de inicio, home o página principal del sitio web.



## 7.2. ENLACE A UN CORREO

```
<a href="mailto:nombre@direccion.com" title="Dirección  
de email para solicitar más información">
```

Solicita más información

```
</a>
```

Al pinchar sobre el enlace anterior, se abre automáticamente el programa de correo electrónico del ordenador del usuario y se establece la dirección de envío al valor indicado después de `mailto:`

La sintaxis es la misma que la de un enlace normal, salvo que se cambia el prefijo `http://` por `mailto:`



## 7.3. ENLACE A UN ARCHIVO FTP

Para enlazar un archivo almacenado en un servidor FTP, la parte del protocolo de la URL debe cambiar de

`http://` a `ftp://`:

```
<a href="ftp://ftp.ejemplo.com/ruta/archivo.zip"  
title="Archivo comprimido de los contenidos">
```

Descarga un ZIP con todos los contenidos

```
</a>
```



## 7.4. ENLACE A VARIAS HOJAS DE ESTILOS

```
<link rel="stylesheet" type="text/css" href="/css/  
comun.css" />
```

```
<link rel="stylesheet" type="text/css" href="/css/  
secciones.css" />
```

```
<link rel="stylesheet" type="text/css" href="/css/  
comun.css" media="screen, projection" />
```

```
<link rel="stylesheet" type="text/css" href="/css/  
impresora.css" media="print" />
```

```
<link rel="stylesheet" type="text/css" href="/css/  
movil.css" media="handheld" />
```

## 7.5. ENLAZAR EL FAVICON

El ***favicon*** es el pequeño icono que muestran las páginas en varias partes del navegador. Dependiendo del navegador que se utilice, este icono se muestra en la barra de direcciones, en la barra de título del navegador y/o en el menú de favoritos/marcadores.

```
<link rel="shortcut icon" href="/favicon.ico"
type="image/ico" />
```

Aunque en principio la imagen debería ser de **tipo .ICO** (formato gráfico de los iconos), algunos navegadores soportan *favicons* en otros formatos gráficos más habituales.



## 7.6. ENLAZAR HOJAS DE ESTILOS CSS, FAVICON Y RSS

```
<head>
```

```
...
```

```
<link rel="stylesheet" type="text/css" href="/css/impresora.css"
media="print" />
```

```
<link rel="stylesheet" type="text/css" href="/css/movil.css"
media="handheld" />
```

```
<style type="text/css" media="screen,projection">
```

```
    @import '/css/main.css';
```

```
</style>
```

```
<link rel="shortcut icon" href="/favicon.ico" type="image/ico" />
```

```
<link rel="alternate" type="application/rss+xml" title="Resumen de todos
los artículos del blog" href="/feed.xml" />
```

```
...
```

```
</head>
```