

Enlaces

El lenguaje de marcado HTML se definió teniendo en cuenta algunas de las características que existían en ese momento para la publicación digital de contenidos. Entre los conceptos utilizados en su creación, se encuentra el mecanismo de "*hipertexto*".

De hecho, las letras "HT" de la sigla HTML significan "*hipertexto*" (*hypertext* en inglés), por lo que el significado completo de HTML podría traducirse como "lenguaje de marcado para hipertexto".

La incorporación del *hipertexto* fue una de las claves del éxito del lenguaje HTML, ya que permitió 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. Aunque la mayoría de enlaces relacionan páginas web, también es posible enlazar otros recursos como imágenes, documentos y 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 los dos extremos se llaman "*anchors*" en inglés, que se puede traducir literalmente como "anclas".

URL

Antes de empezar a crear enlaces, es necesario comprender y dominar el concepto de URL. El acrónimo URL (del inglés *Uniform Resource Locator*) hace referencia al identificador único de cada recurso disponible en Internet. 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

En primer lugar, las URL permiten que cada página HTML publicada en Internet tenga un nombre único que permita diferenciarla de las demás. De esta forma 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.

El segundo objetivo de las URL es el de permitir la localización eficiente de cada recurso de Internet. Para ello es necesario comprender las diferentes partes que forman las URL. Una URL sencilla siempre está formada por las mismas tres partes. Si por ejemplo se considera la siguiente URL:

<http://www.apuntes.es/programacion/tema4.html>

Las partes que componen la URL anterior son:

- Protocolo (<http://>): el mecanismo que debe utilizar el navegador para acceder a ese recurso. Todas las páginas web utilizan <http://>. Las páginas web *seguras* (por ejemplo las de los bancos y las de los servicios de email) utilizan <https://> (se añade una letra *s*).
- Servidor (www.apuntes.es): simplificando mucho su explicación, se trata del ordenador 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.
- Ruta ([/programacion/tema4.html](http://www.apuntes.es/programacion/tema4.html)): *camino* que se debe seguir, una vez que se ha llegado al servidor, para localizar el recurso específico que se quiere acceder.

Por tanto, las URL no sólo identifican de forma única a cada recurso de Internet, sino que también proporcionan a los navegadores la información necesaria para poder llegar hasta ese recurso.

La mayoría de URL son tan sencillas como la URL mostrada anteriormente. No obstante, existen URL complejas formadas por más partes.

<http://www.alistapart.com/comments/webstandards2008?page=5#42>

Las cinco partes que forman la URL anterior son:

- Protocolo (<http://>)
- Servidor (www.alistapart.com)
- Ruta ([/comments/webstandards2008](http://www.alistapart.com/comments/webstandards2008))
- Consulta ([?page=5](http://www.alistapart.com/comments/webstandards2008?page=5)): 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 ([#42](http://www.alistapart.com/comments/webstandards2008?page=5#42)): permite que el navegador se posicione automáticamente en una sección de la página web. Siempre comienza con el carácter *#*

Como las URL utilizan los caracteres *:*, *=*, *&* y */* para separar sus partes, estos caracteres están reservados y no se pueden utilizar libremente. Además, algunos caracteres no están reservados pero pueden ser problemáticos si se utilizan en la propia URL.

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.

A continuación se muestra la tabla para codificar los caracteres más comunes:

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

Por otra parte, 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.

La solución consiste en codificar todos los caracteres que no existen en inglés. La siguiente tabla muestra la codificación de los caracteres más utilizados:

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

Teniendo en cuenta las dos tablas anteriores de codificación de caracteres, es fácil crear las 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%E1gina.html
```

Enlaces relativos y absolutos

Las páginas web habituales suelen contener decenas de enlaces de diferentes tipos. La siguiente imagen muestra algunos de los tipos de enlaces de la página principal de un sitio web.



Ejemplo de diferentes tipos de enlaces en la página 456BereaStreet.com

En esa página, 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".

Además de internos/externos, la otra característica que diferencia a los enlaces (y por tanto, también a las URL) es si el enlace es absoluto o relativo. Las **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.

Las **URL relativas** prescinden de algunas partes de las URL para hacerlas más breves. Como se trata de URL incompletas, es necesario disponer de información adicional para obtener el recurso enlazado. En concreto, para que una URL relativa sea útil es imprescindible conocer la URL del origen del enlace.

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. Aunque las URL relativas pueden ser difíciles de entender para los que comienzan con HTML, son tan útiles que todos los sitios web las utilizan.

Imagina que dispones de una página publicada en <http://www.ejemplo.com/ruta1/ruta2/pagina1.html> y quieres incluir en ella un enlace a otra página que se encuentra en <http://www.ejemplo.com/ruta1/ruta2/pagina2.html>. Como las URL identifican de forma única a los recursos de Internet y proporcionan la información necesaria para llegar hasta ellos, 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.

Sin embargo, escribir siempre las URL completas es bastante aburrido, cuesta mucho tiempo y hace imposible cambiar la ubicación de los contenidos de un sitio web. Por ese motivo, 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. En otras palabras, las URL relativas aprovechan la *inteligencia* de los navegadores para crear URL incompletas que los navegadores pueden completar deduciendo la información que falta.

Considerando de nuevo 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>

En el ejemplo anterior, 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. Por lo tanto, cuando el navegador encuentra la URL </ruta1/ruta2/pagina2.html>, realiza el siguiente proceso:

1. La URL no es absoluta, por lo que se debe determinar la URL absoluta a partir de la URL relativa 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 (<http://> y www.ejemplo.com).
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>.

Aunque el ejemplo mostrado es el caso más sencillo de URL relativa, existen otros casos más avanzados en los que se prescinde de parte o toda la ruta del recurso que se enlaza. A continuación se muestran los cuatro tipos diferentes de URL relativas:

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.

Origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Recurso enlazado	Página web llamada pagina2.html y que se encuentra en el mismo directorio
URL absoluta	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina2.html
URL relativa	pagina2.html

Cuando el navegador encuentra una URL relativa que sólo consiste en el nombre de un recurso, supone que el protocolo, servidor y directorio del recurso enlazado son los mismos que los del origen del enlace.

2) El destino del enlace se encuentra cerca de su origen y en un nivel superior

En este caso, el recurso que se enlaza no está en el mismo directorio que el origen del enlace pero sí que está *cerca* y en algún directorio 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. De esta forma, cada vez que aparece `../` en una URL relativa, significa que se debe subir un nivel.

Origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Recurso enlazado	Página web llamada pagina2.html y que se encuentra en el directorio superior llamado ruta2
URL absoluta	http://www.ejemplo.com/ruta1/ruta2/pagina2.html
URL relativa	../pagina2.html

Cuando el navegador encuentra la URL relativa [../pagina2.html](#), sabe que para encontrar el recurso enlazado ([pagina2.html](#)) tiene que subir un nivel desde el lugar en el que se encuentra esa URL relativa. La página que incluye esa URL se encuentra en el directorio [ruta1/ruta2/ruta3](#), por lo que subir un nivel equivale entrar en el directorio [ruta1/ruta2](#).

De la misma forma, si el destino se encuentra un par de niveles por encima, se debe incluir `../` dos veces seguidas:

Origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Recurso enlazado	Página web llamada pagina2.html y que se encuentra en el directorio superior llamado ruta1
URL absoluta	http://www.ejemplo.com/ruta1/pagina2.html
URL relativa	../../pagina2.html

Además de subir niveles, también se puede entrar en otros directorios para obtener los recursos:

Origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Recurso enlazado	Página web llamada pagina2.html y que se encuentra en un directorio llamado ruta4 que se encuentra en la raíz del servidor
URL absoluta	http://www.ejemplo.com/ruta4/pagina2.html
URL relativa	../../../../ruta4/pagina2.html

Si se intentan subir más niveles de los que es posible, el navegador ignora todos los `../` sobrantes. Si la página que tiene el enlace es <http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html> y la URL relativa que se incluye es [../../../../../../../../pagina2.html](#), el navegador realmente la interpreta como [../../../../pagina2.html](#).

Como el objetivo de las URL relativas es crear URL más cortas y sencillas que las URL absolutas, este método sólo se puede utilizar cuando el origen y el destino se encuentran cerca, porque de otro modo la URL relativa se complica demasiado.

3) El destino del enlace se encuentra cerca de su origen y en un nivel inferior

Este caso 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.

Origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Recurso enlazado	Página web llamada pagina2.html y que se encuentra en un directorio inferior llamado ruta4
URL absoluta	http://www.ejemplo.com/ruta1/ruta2/ruta3/ruta4/pagina2.html
URL relativa	ruta4/pagina2.html

De la misma forma, se pueden indicar varios directorios seguidos para que el navegador descienda jerárquicamente por la estructura de directorios:

Origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Recurso enlazado	Página web llamada pagina2.html y que se encuentra en un directorio inferior llamado ruta6 que está dentro del directorio ruta5 y que a su vez está dentro del directorio ruta4
URL absoluta	http://www.ejemplo.com/ruta1/ruta2/ruta3/ruta4/ruta5/ruta6/pagina2.html
URL relativa	ruta4/ruta5/ruta6/pagina2.html

4) El origen y el destino del enlace se encuentran muy alejados

Cuando el origen y el destino de un enlace se encuentran muy alejados (pero en el mismo servidor) las URL relativas se pueden complicar en exceso. Aunque es posible utilizar `../` para subir por la jerarquía de directorios y se puede entrar en cualquier directorio indicando su nombre, las URL relativas que se obtienen son demasiado largas y complicadas.

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.

Origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Recurso enlazado	Página web llamada pagina2.html y que se guarda en un directorio llamado ruta7 que se encuentra en la raíz del servidor
URL absoluta	http://www.ejemplo.com/ruta7/pagina2.html
URL relativa	/ruta7/pagina2.html

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.

A continuación se resumen los 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

Enlaces básicos

Los enlaces en HTML se crean mediante la etiqueta `<a>` (su nombre viene del inglés "*anchor*", literalmente traducido como "ancla"). A continuación se muestra la definición simplificada de `<a>` y más adelante se muestra su definición completa:

<a>	Enlaces
Atributos comunes	básicos, i18n, eventos y foco
Atributos específicos	<p><code>name = "texto"</code> - Permite nombrar al enlace para que se pueda acceder desde otros enlaces</p> <p><code>href = "url"</code> - Indica la URL del recurso que se quiere enlazar</p>
Tipo de elemento	En línea
Descripción	Se emplea para enlazar todo tipo de recursos

El atributo más importante de la etiqueta `<a>` es `href`, que se utiliza para indicar la URL a la que apunta el enlace. Cuando el usuario pincha sobre un enlace, el navegador se dirige a la URL del recurso indicado mediante `href`. Las URL de los enlaces pueden ser absolutas, relativas, internas y externas.

Con la definición anterior, 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.com">Página principal de Google</a>
```

Como el atributo `href` indica una URL, un enlace 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.doc">Descargar informe completo [DOC]</a>
```

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.

El otro atributo básico de la etiqueta `<a>` es `name`, que permite definir enlaces dentro de una misma página web. Si una página es muy larga, puede ser útil mostrar enlaces de tipo "[Saltar hasta la segunda sección](#)", "[Volver al principio de la página](#)", etc.

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>
```

...

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>
```

La sintaxis que se utiliza con estos enlaces es la misma que con los enlaces normales, salvo que se añade el símbolo `#` seguido del nombre de la sección a la que se apunta. Cuando el usuario pincha sobre uno de estos enlaces, el navegador accede a la página apuntada por la URL y baja directamente a la sección cuyo nombre se indica después del símbolo `#`.

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>
```

...

Los enlaces directos a secciones también funcionan con el atributo `id` de cualquier elemento. El siguiente ejemplo es equivalente al ejemplo anterior:

```
<h1 id="inicio">Título de la página</h1>

<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>
...
```

El nombre de la sección que se indica después del símbolo `#` puede utilizar el valor de los atributos `id` de cualquier elemento. De hecho, se recomienda utilizar los atributos `id` de los elementos ya existentes en la página en vez de crear "enlaces vacíos" de tipo ``.



Ejercicio 6

Enlaces avanzados

Incluir enlaces básicos mediante la etiqueta `<a>` es muy sencillo. Sin embargo, la definición completa de `<a>` es muy compleja, ya que dispone de varios atributos específicos importantes. A continuación se muestra la definición completa de `<a>`:

<a>	Enlaces
Atributos comunes	básicos, <code>id</code> , eventos y foco
Atributos específicos	<p><code>name = "texto"</code> - Permite nombrar al enlace para que se pueda acceder desde otros enlaces</p> <p><code>href = "url"</code> - Indica la URL del recurso que se quiere enlazar</p> <p><code>hreflang = "codigo_idioma"</code> - Idioma del recurso enlazado</p> <p><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</p> <p><code>rel = "tipo_de_relacion"</code> - Describe la relación del documento actual</p>

<a>	Enlaces
	<p>con el recurso enlazado</p> <p><code>rev = "tipo_de_relacion"</code> - Describe la relación del recurso enlazado con el documento actual</p> <p><code>charset = "tipo_de_charset"</code> - Describe la codificación del recurso enlazado</p>
Tipo de elemento	En línea
Descripción	Se emplea para enlazar todo tipo de recursos

Idioma del enlace (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. Además del idioma genérico, también se puede indicar una variación idiomática. Ejemplo de códigos de idioma más utilizados:

Código	Idioma	Variación idiomática
<code>en</code>	Inglés	-
<code>en-US</code>	Inglés	Estados Unidos
<code>es</code>	Español	-
<code>es-ES</code>	Español	España
<code>es-AR</code>	Español	Argentina

Otros códigos utilizados son: `fr` (francés), `de` (alemán), `it` (italiano), `nl` (holandés), `el` (griego), `pt` (portugués), `ar` (árabe), `he` (hebreo), `ru` (ruso), `zh` (chino), `ja` (japonés).

La lista completa de códigos de idioma está definida en [el estándar ISO 639](#).

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).

La lista completa de tipos de contenido se define en [los estándares RFC 2045 y RFC 2046](#).

Tipo de relación (rel y rev)

Los enlaces pueden proporcionar información adicional muy útil para los navegadores y para los motores de búsqueda como Google. 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)
- `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". Un marcador es un enlace que constituye un punto de entrada muy importante dentro del documento.

La especificación oficial de HTML define la [lista completa de tipos de relaciones](#) que se pueden utilizar.

Codificación de caracteres (charset)

Además del idioma, tipo de contenido y relación del recurso que se enlaza, los enlaces también pueden indicar la codificación de caracteres que utiliza la página web enlazada.

Los valores que se pueden utilizar también están estandarizados y las codificaciones más utilizadas son [UTF-8](#) y [ISO-8859-1](#), aunque existen decenas de códigos definidos ([ISO-10646-UCS-2](#), [IBM852](#), [Big5-HKSCS](#), [windows-1252](#), [HZ-GB-2312](#)).

El organismo IANA publica la [lista completa de codificaciones de caracteres disponibles](#).

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>
```

Otros tipos de enlaces

Los enlaces mostrados en las secciones anteriores son los más utilizados por las páginas web. Los enlaces creados con la etiqueta `<a>` permiten enlazar cualquier tipo de recurso desde cualquier página. La característica más importante de estos enlaces es que el usuario debe activar la carga de los recursos. En otras palabras, el navegador no carga ningún recurso enlazado con la etiqueta `<a>` a menos que el usuario pinche sobre el enlace.

Además de estos enlaces, 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.

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.

<code><script></code>	Código ejecutable
Atributos comunes	-

<script>	Código ejecutable
Atributos específicos	<p><code>src = "url"</code> - Indica la dirección del archivo que contiene el código</p> <p><code>type = "tipo_de_contenido"</code> - Permite "avisar" al navegador sobre el tipo de código que se incluye (normalmente JavaScript)</p> <p><code>defer = "defer"</code> - El código no va a modificar el contenido de la página web</p> <p><code>charset = "tipo_de_charset"</code> - Describe la codificación del código enlazado</p>
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)

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.

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">
    //
      window.onload = function() { alert("La página se ha cargado completamente"); }
    //]]&gt;
  &lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
  ...
&lt;/body&gt;
&lt;/html&gt;</pre>
</div>
```

Cuando se incluye código JavaScript en la propia página XHTML, se debe insertar dentro de una sección especial llamada `CDATA`. Para ello, el código JavaScript se debe encerrar entre `<![CDATA[` y `]]>`. Cuando el navegador encuentra una sección de este tipo, no procesa su contenido como si fuera XHTML y por tanto no tiene en cuenta los posibles errores de validación de XHTML.

De esta forma, se pueden construir páginas XHTML válidas y código JavaScript correcto. En los capítulos posteriores se profundiza en el concepto de validación de páginas XHTML. Los caracteres `//` al comienzo y al final de la sección `CDATA` son necesarios para los navegadores que no son capaces de procesar correctamente estas secciones.

La etiqueta `<script>` (tanto cuando enlaza, como cuando incluye directamente el código) puede aparecer en cualquier parte del documento HTML, aunque normalmente se incluye dentro de la cabecera de la página (`<head>...</head>`).

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

<link>	Enlazar recursos
Atributos comunes	básicos, i18n y eventos
Atributos específicos	<p>Los siguientes con el mismo significado que para la etiqueta "a": charset, href, hreflang, type, rel y rev</p> <p><code>media = "tipo_de_medio"</code> - Indica el medio para el que debe aplicarse la relación</p>
Tipo de elemento	Etiqueta vacía
Descripción	Se emplea para enlazar y establecer relaciones entre el documento y otros recursos

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.

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>
```

En este caso, 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.

Ejemplos de enlaces habituales

Enlace al inicio del 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.

Enlace a un email

```
<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:`.

La sintaxis de `mailto:` permite utilizarlo para otros ejemplos más complejos:

```
<!-- Envío del correo electrónico a varias direcciones a la vez -->
<a href="mailto:nombre@direccion.com,otro_nombre@direccion.com">Solicita más información</a>

<!-- Añadir un "asunto" inicial al correo electrónico -->
<a href="mailto:nombre@direccion.com?subject=Solicitud de más información">Solicita más información</a>

<!-- Añadir un texto inicial en el cuerpo del correo electrónico -->
<a href="mailto:nombre@direccion.com?body=Estaría interesado en solicitar más información sobre sus productos">Solicita más información</a>
```

Todas las opciones anteriores se pueden combinar entre sí para realizar ejemplos más avanzados. Aunque el uso de `mailto:` puede parecer una ventaja, su uso está desaconsejado. Si se incluye una dirección de correo electrónico directamente en una página web, es muy probable que en poco tiempo esa dirección de email se encuentre llena de correo electrónico basura o "*spam*", ya que existen programas automáticos encargados de rastrear sistemáticamente todas las páginas web de Internet para encontrar direcciones de correo electrónico válidas.

La forma de mostrar las direcciones de correo electrónico en las páginas web consiste en incluir la dirección en una imagen o indicarla de forma que solamente los usuarios puedan entenderlo:

```
<p>La dirección de correo es <strong>nombre (arroba) direccion.com</strong></p>
<p>La dirección de correo es <strong>nombre_arroba_direccion.com</strong></p>
<p>La dirección de correo es <strong>nombreQUITAESTO@direccion.com</strong></p>
<p>La dirección de correo es <strong>nombre(ARROBA)direccion.com</strong></p>
<p>La dirección de correo es <strong>nombre @ direccion . com</strong></p>
```

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>
```

Enlazar varias hojas de estilos CSS

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

Enlazar hojas de estilos CSS para diferentes medios

```
<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" />
```

Enlazar el favicon

El *favicon* o icono para favoritos 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 (como por ejemplo `.PNG`).

Enlazar un archivo RSS

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

Enlazar hojas de estilos, favicon y RSS

En una misma página se pueden incluir varias etiquetas `<link>`, por lo que es habitual que las páginas enlacen hojas de estilos, favicon y archivos RSS de forma conjunta:

```
<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>
```

Indicar que existe una versión de la página en otro idioma

```
<head>
<title>English tutorial</title>
<link lang="es" xml:lang="es" title="El tutorial en español" type="text/html"
rel="alternate" hreflang="es" href="http://www.ejemplo.com/tutorial/espanol.html" />
</head>
```

Indicar que existe una versión de la página preparada para imprimir

```
<head>
<link media="print" title="El tutorial en PDF" type="application/pdf" rel="alternate"
href="http://www.ejemplo.com/tutorial/documento.pdf" />
</head>
```

Indicar que existe una página que es índice de la página actual

```
<head>
<title>Tutorial - Capítulo 5</title>
<link rel="start" title="El índice del tutorial" type="text/html"
href="http://www.ejemplo.com/tutorial/indice.html" />
</head>
```



Ejercicio 7