INTRODUCCIÓN A XHTML

4.Enlaces

© Francisco Pérez
ies_paco@yahoo.es

CONTENIDO

1. Introducción

La incorporación de enlaces, también llamados hipervínculos o hiperenlaces fue una de las claves del éxito del lenguaje HTML ya que permitió crear documentos interactivos que permiten enlazar con otros documentos relacionados, proporcionando así información adicional cuando se solicita.

Los enlaces se utilizan para establecer relaciones entre dos recursos, entendiendo por recurso documentos, imágenes u otros tipos de archivo.

2. URL

Antes de crear enlaces es necesario abordar el concepto de **URL** o ***Uniform Resource Locator***. Una URL hace referencia a un identificador de un recurso disponible a través de la red. Las URL no solo se utilizan para determinar el documento destino de un enlace, sino que también permiten determinar la ubicación de imágenes, formularios y otros tipos de archivo.

Cuando accedemos a un recurso en red como un sitio web utilizamos una URL similar a la siguiente:

```
http://www.eltiempo.es:8080/previsiones/enero2013?region=murcia#resumen
```

En esta URL podemos identificar las siguientes secciones:

- Una primera sección que contiene el **protocolo** utilizado (***http://***).
- Una sección que contiene el **nombre DNS** del servidor que administra el recurso. (***www.eltiempo.es***). En vez del nombre DNS puede utilizarse la **dirección IP** del servidor.
- Una sección que indica el **puerto** por el que escucha peticiones el servidor (***:8080***). Cuando el servidor escucha por el puerto por defecto no es necesario especificar este campo. Los servicios de red habituales como web, FTP o POP tienen definido un puerto por defecto (80 para el servicio web, 21 para el servicio FTP, etc.).
- Una sección que indica **la ruta del recurso** (***/previsiones/enero2013***).
- Una sección de **consulta**, que es un conjunto de parejas parámetro/valor en la forma *?parámetro1=valor1&parámetro2=valor2* , etc. En el ejemplo anterior, la consulta está formada por una única pareja parámetro/valor (***?región=murcia***) aunque es posible utilizar múltiples parejas separándolas con el carácter "&". La información de la consulta es requerida por las aplicaciones web alojadas en el servidor para procesar las solicitudes del cliente correctamente y ofrecerles el documento adecuado a cada petición.
- Finalmente, una sección de **ancla** (***#resumen***) que indica al navegador en qué parte documento debe posicionarse. Esto resulta útil si tenemos un documento muy extenso y queremos que al pulsar sobre el enlace, el cliente navegue hasta una parte concreta de dicho documento en vez de situarse al comienzo del documento.

Para que una URL se considere bien formada debe incluir, como mínimo:

- El protocolo.
- El nombre DNS o dirección IP del servidor.
- El puerto por el que escucha el servidor si éste es diferente del puerto estándar.
- La ruta hasta el recurso al que se quiere acceder.

Como acabamos de ver, en las URLs se utilizan los caracteres especiales siguientes:

Carácter	Utilización
:	Delimitador de número de puerto
=	Asignación de un valor a un parámetro
&	Concatenación de parejas parámetro/valor
/	Separador de directorios

Estos caracteres tienen un significado especial y por ello no pueden usarse libremente.

Por otro lado, el carácter de espacio y cualquier carácter no-plano (como letras acentuadas o la letra 'ñ') pueden resultar problemáticos. Desde hace tiempo es posible incluir en una URL caracteres de idiomas distintos al inglés. Sin embargo, todavía no resulta totalmente seguro ya que es posible que no las interpreten correctamente tanto servidores como navegadores que no estén actualizados. Si necesitamos incluir alguno de estos caracteres en la URL, tendremos que sustituir cada uno de ellos por una combinación de caracteres equivalentes seguros. Esta sustitución se denomina ***codificación*** y el servidor realiza el proceso inverso (***decodificación***) cuando le llega una URL con los caracteres codificados.

Esta es la codificación de los caracteres más habituales:

Carácter original	Carácter codificado	Carácter original	Carácter codificado
/	%2F	?	%3F
:	%3A	@	%40
=	%3D	&	%26
"	%22	\	%5C
'	%60	~	%7E
(espacio en blanco)	%20	|	%7C

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

3. Tipos de URL

Al crear enlaces en (X)HTML, podemos especificar dos tipos de URL:

- **URL absoluta:** incluyen todas las partes de URL (como mínimo protocolo, servidor y ruta) por lo que no se necesita más información para obtener el recurso enlazado.

- **URL relativa:** se construyen prescindiendo de la parte del protocolo, del nombre del servidor e incluso de parte o toda la ruta del recurso enlazado. Esto hace sencillo manejar URLs largas en el código fuente de un documento web, puesto que evita tener que teclear largas cadenas.

  Como hemos visto anteriormente, una URL relativa no se podría considerar una URL completa por lo que al utilizarlas, el navegador debe deducir el resto de elementos omitidos.

Imagina que dispones de una página publicada en:

http://www.ejemplo.com/dir1/dir2/pagina1.html

y quieres incluir en ella un enlace a otra página que se encuentra en:

http://www.ejemplo.com/dir1/dir2/pagina2.html

Para ello deberías utilizar en la primera página la URL completa de la segunda página. Utilizar esta ruta absoluta, resulta tedioso, propenso a errores y produce un código fuente más pesado. Además, la utilización de URLs absolutas puede plantear problemas cuando se cambia la ubicación de los contenidos de un sitio web (o se modifica su nombre DNS). Por este 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 deducir 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 absoluta: http://www.ejemplo.com/dir1/dir2/pagina2.html puede transformarse en la URL relativa: /dir1/dir2/pagina2.html

Estas 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. Para que el navegador pueda reconstruir la ruta absoluta a partir de la ruta relativa, se deben verificar las siguientes reglas de construcción de URL relativas:

1. Si el destino de la URL es un recurso que se encuentra **en el mismo directorio** que el documento que contiene la URL, la URL relativa puede prescindir de todas las partes de la URL absoluta salvo el nombre del recurso enlazado. Por ejemplo:

Página que contiene el enlace:	http://www.ejemplo.com/directorio1/directorio2/pagina1.htm
Recurso enlazado:	Página web llamada 'pagina2.htm'. Esta página se encuentra en el mismo directorio que la página que contiene el enlace.
URL absoluta:	http://www.ejemplo.com/directorio1/directorio2/pagina2.htm
URL relativa:	pagina2.htm

2. Si el destino del enlace se encuentra **en un directorio superior**, la URL relativa debe indicar que es necesario *subir* uno o varios niveles en la jerarquía de directorios para llegar hasta el recurso. Esto se consigue incluyendo en la URL relativa una secuencia formada por dos puntos y una barra (../). De esta forma, cada vez que aparece ../ en una URL relativa, significa que se debe subir un nivel. Por ejemplo:

Página que contiene el enlace:	http://www.ejemplo.com/dir1/dir2/pagina1.htm
Recurso enlazado:	Página web llamada 'pagina2.htm' que se encuentra en el directorio superior llamado 'dir1'.
URL absoluta:	http://www.ejemplo.com/dir1/pagina2.htm
URL relativa:	../pagina2.htm

Página que contiene el enlace:	http://www.ejemplo.com/dir1/dir2/pagina1.htm
Recurso enlazado:	Página web llamada 'pagina2.htm' que se encuentra en el directorio raíz del sitio web.
URL absoluta:	http://www.ejemplo.com/pagina2.htm
URL relativa:	../../pagina2.htm

Página que contiene el enlace:	http://www.ejemplo.com/dir1/dir2/pagina1.htm
Recurso enlazado:	Página web llamada 'pagina2.htm' que se encuentra en el directorio llamado 'otrodir' que se encuentra bajo el directorio raíz del sitio web.
URL absoluta:	http://www.ejemplo.com/otrodir/pagina2.htm
URL relativa:	../../otrodir/pagina2.htm

3. Si el destino del enlace se encuentra **en un directorio de nivel inferior**, sólo es necesario indicar el nombre los directorios a los que debe entrar el navegador y el nombre del recurso al que se quiere acceder. Por ejemplo:

Página que contiene el enlace:	http://www.ejemplo.com/dir1/dir2/pagina1.htm
Recurso enlazado:	Página web llamada 'pagina2.htm' que se encuentra en el directorio inferior llamado 'dir3'.
URL absoluta:	http://www.ejemplo.com/dir1/dir2/dir3/pagina2.htm
URL relativa:	dir3/pagina2.htm

Página que contiene el enlace:	http://www.ejemplo.com/dir1/dir2/pagina1.htm
Recurso enlazado:	Página web llamada 'pagina2.htm' que se encuentra en un directorio dos niveles por debajo 'dir4'.
URL absoluta:	http://www.ejemplo.com/dir1/dir2/dir3/dir4/pagina2.htm
URL relativa:	dir3/dir4/pagina2.htm

4. Si el destino del enlace se encuentra en el mismo sitio web pero "*alejado*" de la página que contiene el enlace, podríamos construir una ruta relativa demasiado larga o compleja (por ejemplo, puede ser difícil entender cuál es el recurso al que apunta una URL cuando tiene muchas secuencias ../ en su ruta). En estos casos, la ruta relativa se suele construir utilizando la ruta desde la raíz del sitio web hasta el recurso de destino, es decir, sólo se omite el protocolo y el nombre del servidor. Por ejemplo:

Página que contiene el enlace:	http://www.ejemplo.com/dir1/dir2/dir3/dir4/dir5/pagina1.htm
Recurso enlazado:	Página web llamada 'pagina2.htm' que se encuentra en un directorio llamado 'otrodir' que se encuentra bajo el directorio raíz del sitio web.
URL absoluta:	http://www.ejemplo.com/otrodir/pagina2.htm
URL relativa:	/otrodir/pagina2.htm

4. Otras consideraciones relativas al servidor web empleado

Al incluir enlaces en nuestras páginas web conviene tener en cuenta que deberemos evitar el uso de mayúsculas y caracteres no-planos en las URL. Supongamos que tenemos una página cuyo nombre contiene alguna letra mayúscula. Si un cliente teclea en el navegador la URL correspondiente a esa página en minúsculas es posible que el servidor no sea capaz de localizar y servir la página al cliente. En general los servidores para Windows (como Internet Information Services) no distinguen entre mayúsculas y minúsculas. Sin embargo, otros servidores vinculados a sistemas Unix, como Apache si diferencian entre mayúsculas y minúsculas.

Otra limitación que puede venir impuesta por la tecnología de servidor utilizada es la longitud máxima de la URL, que en la mayoría de casos no debe superar los 240 caracteres.

Una consideración a tener en cuenta a la hora de trabajar con URLs es el nombre de página o páginas por defecto definidas en el servidor web. La mayoría de servidores pueden configurarse para buscar una página por defecto en cada directorio, de manera que si el cliente teclea en el navegador una URL correspondiente a un directorio (no a una página web), el servidor tratará de buscar la página predeterminada en ese directorio. La mayoría de servidores web asumen como página por defecto cualquiera con los nombres 'index.htm' o 'index.html'.

Por último, la mayoría de servidores web permiten definir **directorios virtuales**. Un directorio virtual se comporta como un enlace simbólico que apunta a un fichero o directorio. Esta posibilidad permite simplificar enormemente las URL en sitios web con una estructura compleja.

5. Enlaces en (X)HTML

Los enlaces en (X)HTML se crean mediante la etiqueta <a>. Esta etiqueta es de tipo in-line y admite los siguientes atributos:

- ***href = "****<URL>****"*** → permite especificar la URL del recurso que se quiere enlazar.
- ***name = "****<nombre_de_ancla>"* → permite especificar un nombre para un punto de anclaje dentro de la propia página web.
- ***hreflang = "<***código_de_idioma>***"*** → indica el idioma en el que se encuentra el recurso enlazado. Estos son algunos ejemplos de valores admitidos por este atributo:

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

La lista completa de códigos de idioma está definida en el estándar ISO 639.

- ***charset = "***<codificación>***"*** → Indica la codificación utilizada por el recurso enlazado. Los valores que se pueden utilizar también están estandarizados y las codificaciones en los países occidentales son UTF-8 y ISO-8859-1. Todas ellas están estandarizadas por el organismo IANA (http://www.iana.org/assignments/character-sets)

- ***type = "***<tipo_de_contenido>***"*** → Indica al navegador el tipo de contenido con el que se enlaza para que pueda tomar acciones adecuadas en caso de no ser capaz de manejar ese tipo de contenido. Los valores más utilizados para este atributo son:

    - "text/html" → páginas web.
    - "image/png" → imagen con formato png.
    - "image/gif" → imagen con formato gif.
    - "image/jpg" → imagen con formato jpg.
    - "text/css" → hoja de estilo CSS.
    - "application/rss+xml" → archivo RSS.

  La lista complete de los valores admitidos por este atributo está definida en los estándares RFC 2045 y RFC 2046.

- ***rel = "***<tipo_de_relación>***"*** → Describe la relación de la página actual con el recurso enlazado.

- ***rev = "***<tipo_de_relación>***"*** → Describe la relación que tiene el recurso enlazado con la página actual.

  Los tipos de relación que se definen para los anteriores atributos son:

    - ***alternate*** → indica que el enlace es una versión alternativa del documento actual (por ejemplo, en otro idioma o una versión preparada para otro tipo de dispositivo como una tablet o un Smartphone).

    - ***stylesheet*** → indica que el enlace es una hoja de estilos.

    - ***start*** → indica que se trata del primer documento de una colección de documentos.

    - ***next*** → indica que es el documento que sigue al actual dentro de una colección de documentos.

    - ***prev*** → indica que es el documento que precede al actual dentro de una colección de documentos.

    - ***contents*** → indica que el documento enlazado contiene el índice o tabla de contenidos dentro de una colección de documentos.

    - ***bookmark*** → establece el enlace como un marcador o favorito.

  La especificación completa de tipos de relación viene recogida en http://www.w3.org/TR/1999/REC-html401-19991224/types.html#type-links

Los atributos más utilizados son *href* y *name*. Cuando el usuario hace clic sobre un enlace, el navegador se dirige a la URL del recurso indicado mediante el atributo href. Las URL de los enlaces pueden ser absolutas o relativas y además pueden ser internas o externas:
Un enlace que hace que hace que naveguemos hacia un recurso externo al sitio web en el que nos encontramos se dice que es un **enlace externo**. De manera análoga, cuando un enlace nos conduce a un recurso del mismo sitio web en el que estamos navegando, se dice que es un **enlace interno**.

Veamos algunos ejemplos:

Para crear un enlace que apunte a la página principal de Google deberíamos incluir el siguiente enlace en nuestra página web:

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

El texto que se incluye entre la etiqueta de apertura y la de cierre aparecerá por defecto subrayado y, al pasar el ratón sobre él, el cursor cambiará de aspecto para indicar que se trata de un enlace a otro recurso.

El atributo href de la etiqueta <a> puede enlazar con cualquier tipo de recurso, no solamente con páginas web. Por ejemplo, un enlace puede hacer referencia a una imagen o un documento pdf.

La mayoría de sitios web disponen de un enlace que permite acceder a la página principal del sitio web. Esto se consigue de manera sencilla con el siguiente enlace que hace uso de una ruta relativa:

```
<a href="/">Página principal</a>
```

Por otro lado, el atributo "name" permite crear enlaces o puntos de anclaje dentro de la propia página web, de manera que podemos acceder a una zona concreta (punto de anclaje) de la página web. Esto resulta especialmente útil si tenemos una página web muy extensa y queremos navegar a una sección concreta dentro de ella.

Por ejemplo si creamos una página web con puntos de anclaje como la siguiente:

```
[…]

<a name="primera_seccion"/>
<h1>Los empresarios subirán los sueldos en 2013</h1>
[…]

<a name="segunda_seccion"/>
<h2>La crisis ha terminado</h2>
[…]
```

podremos acceder a la página normalmente con el enlace:

```
<a href="http://www.ejemplo.com/noticias.htm">Noticias</a>
```

Y además podremos acceder directamente a los puntos de anclaje con los siguientes enlaces:

```
<a href="http://www.ejemplo.com/noticias.htm#primera_seccion">Titulares</a>
 <a href="http://www.ejemplo.com/noticias.htm#segunda_seccion">Breves</a>
```

Como puede verse, la sintaxis utilizada para estos enlaces es la misma que la de los enlaces normales pero añadiendo el símbolo '#' seguido del nombre del punto de anclaje al que se quiere acceder. Además, también es posible utilizar rutas relativas en vez de rutas absolutas. Por ejemplo, si los enlaces anteriores estuvieran en la misma página en la que se han definido los puntos de anclaje, podríamos haberlos escrito así:

```
<a href="#primera_seccion">Titulares</a>

<a href="#segunda_seccion">Breves</a>
```

De manera alternativa a la creación de puntos de anclaje, es posible definir un identificador para una etiqueta y utilizar dicho identificador como punto de anclaje. Por ejemplo, si creamos la página anterior con el siguiente código:

```
[…]

<h1 id="primera_seccion">Los empresarios subirán los
sueldos en 2013</h1>
[…]

<h2 id="segunda_seccion">La crisis ha terminado</h2>
[…]
```

Podremos utilizar los mismos enlaces a los puntos de anclaje que hemos definido antes.

6. Otros tipos de enlaces

Los enlaces creados con la etiqueta <a> que hemos visto hasta ahora son enlaces bajo demanda. Esto quiere decir que los recursos enlazados son cargados cuando el usuario los activa. En otras palabras, el navegador no carga ningún recurso enlazado con la etiqueta <a> hasta que el usuario hace clic sobre el enlace.

Además de los enlaces creados con la etiqueta <a>, en (X)HTML se pueden crear otros tipos de enlaces que cargan los recursos automáticamente (sin intervención del usuario) cuando se carga el documento que contiene los enlaces. Por ejemplo, si una página utiliza un archivo CSS para aplicar estilos a sus contenidos, será necesario que dicho archivo CSS se cargue automáticamente al visualizar el documento. Del mismo modo, muchas páginas web dinámicas necesitan que el navegador cargue uno o varios archivos de script para funcionar correctamente.

(X)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.

6.1. Scripts

Una página escrita en (X)HTML íntegramente se dice que es **estática**: la página muestra siempre la misma información y el usuario no puede interactuar con ella (excepto con los hiperenlaces). Para introducir, aunque de manera limitada, interacción con la página y aportar la posibilidad de ejecutar instrucciones sencillas, HTML es capaz de coexistir con algunos lenguajes llamados ***lenguajes de script***. Estos lenguajes (JavaScript y VBscript principalmente) permiten desarrollar pequeños subprogramas, llamados ***scripts***, que posibilitan la realización de tareas diversas, como cálculos sencillos, la modificación del aspecto de la página, aplicar movimiento a elementos en la página y el envío de información al servidor. Estos scripts se ejecutan en el cliente, es decir, al incluirlos en una página, el usuario los descarga junto con la página y los ejecuta en su propia máquina. No obstante, no son un lenguaje de programación completo y por tanto tienen una funcionalidad limitada (no se pueden utilizar para conectar a una base de datos o para leer de un fichero, etc…).

Para utilizar un script en una página disponemos de la etiqueta <script>. Los fragmentos de código script se pueden escribir intercalados con el HTML o bien pueden residir en un fichero independiente. Por ello, la etiqueta <script> tiene dos modos de funcionamiento:

- Enlazar un archivo de script externo.
- Insertar un bloque de script en la propia página.

Cada vez que el navegador encuentra una etiqueta <script> enlazando a un archivo de script externo, procede a descargar y procesar dicho archivo. De igual modo, cuando el navegador encuentra una etiqueta <script> con un bloque de código embebido en la página, procesa dicho bloque.

Estos son los atributos que admite la etiqueta <script>:

- ***src =*** <URL_del_recurso> → Indica la URL del archivo externo que contiene el código de script.

- ***type =*** <tipo_de_contenido> → Describe el tipo de código que se incluye. El lenguaje de script "JavaScript" es el más extendido por lo que el valor más habitual de este atributo es ***"text/javascript"***.

- ***defer = "defer"*** → Indica que el código script será **procesado** después de que la página se haya cargado completamente (procesamiento diferido). Esto implica que el script no va a modificar el contenido de la página.

  En una página con scripts, el orden de **descarga** es siempre secuencial, a medida que el navegador encuentra las etiquetas <script>. Sin embargo, el orden de procesamiento de estos scripts depende de la presencia o ausencia del atributo defer y de la localización in-line o externa del bloque de script. El orden de procesamiento que se aplica es el siguiente:

- Scripts no diferidos, en su orden de aparición en el código de la página.
- Scripts diferidos cuyo código está embebido en la página.
- Scripts diferidos cuyo código reside en un archivo externo.

- ***Charset =*** <tipo_de_charset> → Indica la codificación del código enlazado.

Cuando se incluye un enlace a un fichero de script externo, suele incluirse el enlace en la sección <head> de la página web para que se descargue y esté disponible al comenzar la carga de la página (aunque es posible utilizarla en cualquier parte de la página):

```
<head>
	[…]
	<script type="text/javascript" src="scripts/formularios.js"></script>
</head>
```

SI, por el contrario, se incluye el script embebido en el propio código de la página utilizamos la sintaxis siguiente:

```
<head>
	[…]
	<script type="text/javascript">
		//<![CDATA[
			window.onload = function() {alert("página cargada");}
		//]]>
	</script>
</head>
```

Como puede verse en el ejemplo anterior, al incluir código in-line (embebido) en la propia página web es necesario insertarlo dentro de una sección llamada CDATA. 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. Abordaremos el estudio de esta y otras secciones al estudiar XML.

6.2. Enlaces a otros recursos no-documentales

La etiqueta <link> permite enlazar con recursos externos no-documentales (aquéllos que no contienen información útil para el lector de la página). Esta etiqueta se utiliza principalmente para enlazar con archivos de hojas de estilo CSS. Al contrario que la etiqueta <script>, la etiqueta <link> sólo puede usarse dentro de la sección <head> de la página web.

Estos son los atributos que admite:

- ***charset, href, hreflang, type, rel y rev*** → con el mismo significado que en la eitqueta <a>.

- ***media=***<tipo_de_medio> → Indica el medio en el que es válido aplicar el recurso enlazado. Los medios disponibles están estandarizados, siendo los más comunes:

    - screen → para contenidos mostrados por pantalla.
    - print → para contenidos enviados a una impresora.
    - handheld → para dispositivos móviles como palm y smartphones.

Un ejemplo típico de enlace a una hoja de estilos CSS es el siguiente:

```
<head>
       […]
       <script   rel="stylesheet"   type="text/css"   href="/css/estilo1.css"/>
</head>
```

7. Algunos enlaces habituales

- **Enlace a la página principal del sitio web** (suponiendo que se ha definido una página por defecto en el servidor y que dicha página existe):

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

- **Enlace a un email:**

```
<a href="mailto:nombre@direccion.com?subject=Solicitud de info&body=Escribe tu
mensaje" title= "Dirección para solicitar información">

       Solicita más información

</a>
```

Este tipo de enlace plantea un problema: al hacer clic sobre ellos se inicia el cliente de correo electrónico por defecto en la máquina del usuario. Muchos usuarios no tienen un cliente de correo instalado y configurado o bien no saben utilizarlo por lo que este tipo de enlaces pueden suponer que no se logre contactar con el correo del enlace. Además, al incluir una dirección de correo en una página es fácil caer en las redes de spam puesto que existen programas automatizados que buscan direcciones de correo en las páginas para el envío masivo de publicidad. Por todo ello, el uso de este tipo de enlace está desaconsejado. Si se quiere incluir una dirección de correo en una página puede hacerse a través de una imagen o bien indicarla de forma que solamente los usuarios puedan entenderla.

- **Enlace a un FTP:**

```
<a href="ftp://ftp.servidor.com/ruta/archivo.zip">
      Descarga aquí todos los contenidos.
</a>
```

- **Enlazar hojas de estilos 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" />
```

- **Enlazar el favicon:**

El favicon es el icono que muestran las páginas en la barra de título de la ventana del navegador y/o en la pestaña y/o en la barra de direcciones. La mayoría de navegadores soportan imágenes en formato .ico y .png .

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

- **Enlazar un archivo RSS:**

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

- **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>
```