Enlaces en HTML

Tiempo estimado: 15min

Hasta el momento, hemos presentado cómo añadir texto a los documentos HTML. Cómo dividirlo en secciones y cómo proporcionar información adicional sobre el énfasis a dar a distintas palabras y/o frases. Ahora, vamos a presentar los enlaces. A través de ellos, podemos dividir el contenido en distintos documentos HTML así como incrustar imágenes y, tal como veremos más adelante en el curso, audio y video.

Para comenzar, introducimos el concepto de enlace. El medio a través del cual referenciar a otro recurso que puede encontrarse dentro de nuestro sitio web o en otro. A continuación, presentamos los elementos `<a>` y `<nav>` con los que enlazar con recursos a los que el usuario puede navegar o ir. Finalmente, presentamos cómo incrustar imágenes en los documentos HTML. Indispensables en cualquier sitio web hoy en día.

Al finalizar la lección, el estudiante sabrá:

- Qué es un enlace.
- Cómo identificar un recurso interno o externo al sitio web.
- Cómo crear un enlace para que el usuario pueda acceder al recurso referenciado.
- Cómo crear una estructura organizada de enlaces.
- Cómo incrustar imágenes en el documento web.

Introducción

Un enlace (*link*) es un medio a través del cual indicar el acceso a un recurso como otro documento HTML, una imagen, un fichero de audio o video, etc. Recordemos que un recurso (*resource*) no es más que un objeto que puede ser accedido a través de la web. Estos recursos tienen asociados identificadores únicos que actúan a modo de dirección, conocido formalmente como URL. Y que es lo que indicaremos en los enlaces para así poder acceder a ellos.

URL

Un URI (*Uniform Resource Identifier*, identificador de recurso uniforme) es un texto que identifica un recurso de la red de forma unívoca como, por ejemplo, un documento HTML, una imagen, un archivo de audio, etc. Un URL (*Uniform Resource Locator*, localizador de recurso uniforme) es un tipo especial de URI que referencia a un recurso cuyo contenido puede cambiar con el tiempo.

Los URIs y, por tanto, los URLs se dividen en varios componentes: protocolo, servidor, puerto, ruta, cadena de consulta. En el caso de los URLs, la sintaxis es la siguiente:

```
protocolo://servidor[:puerto]/ruta[#ancla][?consulta]
```

Mediante el protocolo (*protocol*), indicamos cómo debe ser accedido el recurso. Por ejemplo, mediante HTTP, FTP, etc. El servidor (*host*) indica el servidor al que debemos remitirnos para acceder al recurso, usando el protocolo indicado. El puerto (*port*) indica el puerto de red en el que escucha la aplicación servidora. La ruta (*path*) indica la ubicación en la que se encuentra el recurso en el servidor. El ancla (*anchor*) a qué sección del recurso acceder. Y finalmente, la cadena de consulta (*query string*) contiene datos adicionales a pasar a la aplicación servidora.

A continuación, se muestra un ejemplo ilustrativo. Es el URL en el que se encuentra la documentación oficial del módulo url de la plataforma Node:

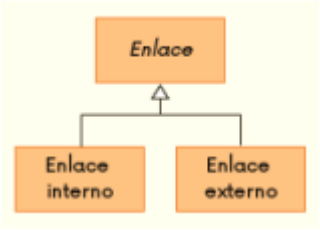
```
http://nodejs.org/dist/latest-v6.x/docs/api/url.html
```

En el ejemplo anterior, se utiliza el protocolo HTTP para acceder al recurso. El servidor es nodejs.org. Como se omite el puerto, se asume el estándar del protocolo, en el caso de HTTP, el puerto es el 80. Finalmente, la ruta del recurso es /dist/latest-v6.x/docs/api/url.html.

Copyright © 2016 nodemy.com. Reservados todos los derechos.

Tipos de enlace

Atendiendo a la ubicación del recurso enlazado, los enlaces se clasifican en internos o externos.

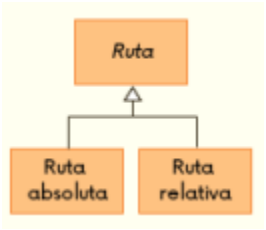


Un enlace interno (*internal link*) es aquel que hace referencia a un recurso que se se encuentra dentro del propio sitio. En cambio, un enlace externo (*external link*) es aquel que se encuentra en otro sitio web.

Enlaces internos

Mediante un enlace interno (*internal link*), indicamos un recurso ubicado o localizado dentro del propio sitio web. Se indica pues sólo la ruta y, si fuera necesario, el ancla y la cadena de consulta.

Ya hemos visto que las rutas indican la ubicación del recurso dentro del servidor. Se estructura en un formato directorio archivo, en la cual la raíz del sitio web se representa mediante el carácter /. Atendiendo al punto desde el que se indica la ruta del recurso enlazado, se distingue entre ruta absoluta o relativa.



Una ruta absoluta (*absolute path*) es aquella que indica la ubicación del recurso comenzando en la raíz del sitio web. Así pues, la ruta comenzará por /, por ejemplo, /mi/doc.html. En cambio, una ruta relativa (*relative path*) es aquella que indica la ubicación del recurso con respecto a la carpeta o directorio en el que se encuentra el documento actual. Por lo tanto, *no* comienza por /. Si es necesario, se puede indicar explícitamente el propio directorio contenedor mediante un punto (.) o el directorio padre mediante dos puntos (..), al igual que hacemos en los sistemas de ficheros. Ejemplos: mi/doc.html, ./mi/doc.html o ../mi/doc.html.

Enlaces externos

Un enlace externo (*external link*) es aquel que referencia a un recurso externo al sitio web. Por esta razón, hay que indicar todo el URL, incluido el protocolo y el sitio web en el que se encuentra.

Aunque se puede indicar un enlace externo para un recurso interno, generalmente, se usa un enlace interno. Se deja los externos para aquellos que no se encuentran dentro del propio sitio web. Por otra parte, generalmente, en los enlaces internos, se prefiere las rutas absolutas a las relativas. Aunque en algunas ocasiones, las relativas son más recomendadas.

Enlace local

Un enlace local (*local link*) es aquel que referencia a un determinado archivo de nuestro disco duro. Este tipo de enlaces se utilizan cuando la aplicación no se sirve desde un servidor web, sino que el navegador accede directamente al disco duro y muestra el documento HTML.

Si tenemos la necesidad de referenciar a archivos del disco local, hay que utilizar el protocolo file, en vez de http. Ejemplo:

```
file:///mi/carpeta/doc.html
```

Copyright © 2016 nodemy.com. Reservados todos los derechos.

Hiperenlace

Un hiperenlace (*hyperlink*), conocido también simplemente como enlace (*link*), es un enlace a otro recurso al que deseamos remitir al usuario. Si el usuario hace clic en el enlace, el navegador debe mostrar el contenido del recurso que está detrás de él.

Los hiperenlaces se representan, en el documento HTML, mediante elementos <a>, un elemento en línea que, cuando el usuario lo sigue, accede a él en la misma ventana o pestaña u otra si fuera necesario.

La sintaxis básica del enlace es la siguiente:

```
<a href="url">texto</a>
```

El texto indicado como contenido del elemento, es decir, entre las etiquetas <a> y , es lo que debe presentar el navegador al usuario. Y lo que utilizará el usuario para acceder al recurso referenciado.

Ejemplo:

```
<p>
  En el sitio web oficial de <a href="http://nodejs.org">Node</a>,
  podemos encontrar instaladores para <i>Windows</i> y <i>Linux</i>.
</p>
```

Atributos de los hiperenlaces

Los principales atributos de los enlaces son los siguientes:

Atributo	Tipo de datos	Descripción
href	texto	URL del recuso referenciado.
hreflang	texto	Idioma en el que se encuentra el documento referenciado como, por ejemplo, en, es, it, etc.
target	texto	Dónde debe abrirlo el navegador: • _self. En el mismo marco, ventana o pestaña. Es el valor predeterminado. • _top. En el cuerpo de la misma página. • _blank. En una nueva pestaña. • _parent. En el marco padre.
download	booleano o texto	Indica que el navegador debe descargar el recurso y no mostrarlo. Si se desea, se puede indicar un nombre de archivo con el que descargarlo.
type	texto	El tipo MIME del recurso como, por ejemplo, text/html, text/xml, application/json, application/pdf, audio/mp4, audio/mpeg, etc.
media	texto	Indica el tipo de dispositivo para el que está optimizado como, por ejemplo, all, print, screen, etc.
rel	texto	Indica qué relación existe entre el documento actual y el recuso referenciado. Los valores más utilizados son los siguientes: • alternate. Le informa al navegador que se trata de una versión alternativa del documento para, por ejemplo, otro dispositivo o idioma. • author. Informa de un documento en el que encontrar información del autor del presente documento. • nofollow. Cuando un motor de búsqueda indexa un documento, sigue los enlaces. Si no deseamos que el enlace sea seguido por el indexador, hay que fijar el valor del atributo rel a nofollow. • noreferrer. Le indica al navegador que no adjunte la cabecera HTTP Referer en la petición del recuso. • prefetch. Le indica al navegador que caché el recurso.

Copyright © 2016 nodemy.com. Reservados todos los derechos.

Estados de los hiperenlaces

Cuando se presenta el contenido de un enlace, el navegador puede mostrarlo en uno de varios estados, donde el estado (*state*) indica si el enlace ha sido accedido por el usuario. Se distingue los siguientes estados:

- No visitado (*unvisited*). El usuario no ha accedido todavía al recurso referenciado por el enlace.
- Visitado (*visited*). El usuario ya ha accedido al recurso.

Los navegadores permiten configurar, principalmente, el color en que presentar el enlace atendiendo a su estado. Para ello, se puede utilizar CSS, concretamente, las seudoclases :link y :visited, que representan los estados no visitado y visitado, respectivamente.

Anclas

Un ancla (*anchor*) es un enlace a un elemento dentro de un documento HTML. Mediante las anclas, se puede crear enlaces a determinadas partes del documento referenciado. Para conseguirlo, primero, hay que identificar cada sección accesible mediante ancla por un identificador. El cual hay que indicar mediante el atributo id. Ejemplo:

```
<section id="cabecera">
  …
</section>
```

Por otra parte, el hiperenlace debe indicar ese identificador en el URL precedido de una almohadilla (#). Ejemplos:

```
<!-- enlace externo con ancla -->
<a href="/mi/doc.html#cabecera">Ir a la cabecera del documento doc.html</a>

<!-- enlace a sección dentro del documento actual -->
<a href="#cabecera">Ir a la cabecera del documento actual</a>
```

Navegación

La navegación (*navegation*) representa un menú horizontal o vertical con enlaces a otras páginas, generalmente, internas del sitio. Se denota mediante el elemento <nav>, el cual se suele ubicar en el encabezamiento, o sea, en un elemento <header>:

```
<nav>menú</nav>
```

He aquí un ejemplo de menú de navegación horizontal:

```
<header>
  <nav>
    <a href="/">Home</a> |
    <a href="/ContactUs.html">Contacto</a>
  </nav>
</header>
```

Ahora otro ejemplo, pero éste utilizando una lista desordenada:

```
<header>
  <nav>
    <ul>
      <li><a href="/">Home</a></li>
      <li><a href="/ContactUs.html">Contacto</a></li>
    </ul>
  </nav>
</header>
```

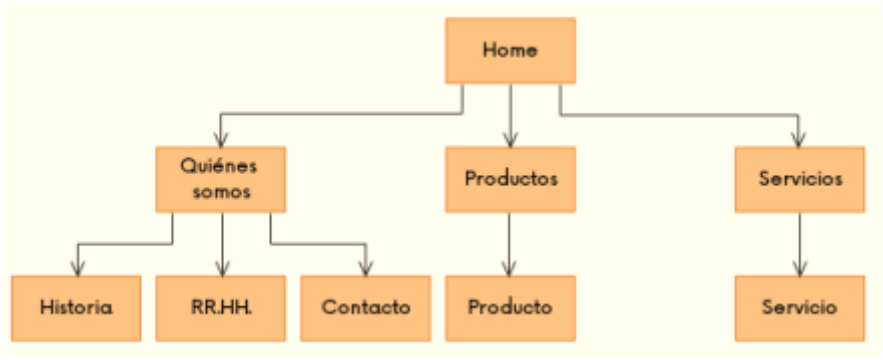
Siempre que defina un conjunto de enlaces relacionados con la estructura organizativa del sitio web, ubíquelos dentro de un <nav>.

Diagrama de navegación

Un diagrama de navegación (*nav diagram*) es una representación gráfica de cómo se encuentran enlazados los documentos HTML de nuestro sitio web. Se recomienda encarecidamente hacerlos, pues ayudan a comprender mejor cómo se encuentran enlazados los documentos.

Veamos un ejemplo:

Copyright © 2016 nodemy.com. Reservados todos los derechos.



Imágenes

El uso de imágenes en un documento HTML está ampliamente extendido.

Los hiperenlaces, <a>, se utilizan principalmente para permitir al usuario que navegue de un recurso a otro del sitio web. El recurso será casi siempre otro documento HTML, pero no hay ningún problema en que el recurso referenciado sea de otro tipo como, por ejemplo, un PDF.

Ahora bien, ¿qué ocurre si deseamos mostrar al usuario una imagen? ¿Lo hacemos también mediante un hiperenlace? La respuesta es: depende. Si no deseamos que se muestre la imagen en el documento, simplemente redirigir al usuario a la imagen y, si hace clic en el hiperenlace, entonces mostrársela de manera separada, usaremos <a>. Pero si deseamos insertar la imagen en un determinado punto del documento, o sea, que se muestre en el propio documento sin que el usuario deba hacer clic en ella, tenemos que usar <figure> o <img>.

Es importante observar que un elemento <a> no incrusta algo en el documento. Simplemente, genera un enlace que al hacer clic en él llevará al recurso asociado. Así pues, recuerde, si para acceder al recurso hay que hacer clic en el enlace, hay que usar <a>. Si en cambio deseamos que el recurso aparezca formando parte del documento, tendremos que usar otro tipo de elemento. En el caso de las imágenes, <figure> o <img>. En el caso de audio y video, tal como veremos en una lección posterior, <audio> y <video>, respectivamente.

Volviendo a las imágenes, es muy importante utilizar el formato adecuado. Cada uno de ellos, se desarrolló con unos objetivos en mente y presenta mejor o peor calidad para cada situación. Así, por ejemplo, GIF sólo soporta 256 colores, lo que reduce su calidad. JPEG se recomienda para fotografías, se diseñó teniéndolas en cuenta. PNG, por su parte, se recomienda para logos e iconos. Y SVG es ideal para imágenes que deben adaptar su tamaño sin pérdida de calidad; lo que se conoce como imágenes adaptables (*responsive images*).

Elemento img

La manera más común de incrustar o insertar una imagen en un documento HTML es mediante el elemento <img>, cuya sintaxis básica es como sigue:

```
<img src="url" alt="descripción de la imagen">
```

Mediante su atributo src indicamos el enlace a la imagen, el cual puede ser interno o externo y, en caso de ser interno, absoluto o relativo. El atributo alt se utiliza para proporcionar una descripción de la imagen, encarecidamente recomendado, pues los motores de búsqueda mejoran el posicionamiento de las páginas que tienen las figuras con título y/o descripción. La información adicional es necesaria para ayudar al motor de búsqueda a saber qué se está mostrando en la figura.

Ejemplo:

```
<img src="/img/logo.png" alt="Logo de mi.sitio.com">
```

Elemento figure

El elemento <figure> también se puede utilizar para insertar imágenes en el documento. Su sintaxis es como sigue:

```
<figure>
  <figcaption>título de la figura</figcaption>
  <img src="url" alt="descripción de la imagen">
</figure>
```

Copyright © 2016 nodemy.com. Reservados todos los derechos.

El elemento <figcaption> es opcional, pero recomendable, proporciona un título a la figura, el cual se suele mostrar encima de la imagen.

Ejemplo:

```
<figure>
  <figcaption>Logo</figcaption>
  <img src="/img/logo.png" alt="Logo de mi.sitio.com">
</figure>
```

Copyright © 2016 nodemy.com. Reservados todos los derechos.