



ELEARNING TOTAL

Diseño Web Responsive HTML5 y CSS3 – Unidad 2

Diseño Web Responsive HTML5 y CSS3

Unidad 2: Introducción a HTML



Indice

Unidad 2: Introducción a HTML

Enlaces

Listas

Imágenes



Objetivos

Que el alumno logre:

- Reconocer e implementar las diferentes etiquetas de HTML.





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.

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.

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.frre.utn.edu.ar/)**: 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 (</home.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.elearning-total.com/campus/course/view.php?id=39#42>

Las cinco partes que forman la URL anterior son:

.: **Protocolo (<http://>)**

.: **Servidor (<http://www.elearning-total.com/>)**

.: **Ruta (</campus/course/view.php>)**

.: **Consulta ([?id=39](#))**: 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](#))**: 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.



Enlaces relativos y absolutos

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

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.

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.

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.

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.

URL absoluta: <http://www.ejemplo.com/ruta1/ruta2/ruta3/ruta4/pagina2.html>

URL relativa: ruta4/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.

URL absoluta: <http://www.ejemplo.com/ruta7/pagina2.html>

URL relativa: /ruta7/pagina2.html



Enlaces básicos

Los enlaces en HTML se crean mediante la etiqueta `<a>` (su nombre viene del inglés “anchor”, literalmente traducido como “ancla”).

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>
```



Listas

En ocasiones, es posible agrupar determinadas palabras o frases en un conjunto de elementos que tienen más significado de forma conjunta.

El lenguaje HTML define tres tipos diferentes de listas para agrupar los elementos: listas no ordenadas (se trata de una colección simple de elementos en la que no importa su orden), listas ordenadas (similar a la anterior, pero los elementos están numerados y por tanto, importa su orden) y listas de definición (un conjunto de términos y definiciones similar a un diccionario).

Listas no ordenadas

Las listas no ordenadas son las más sencillas y las que más se utilizan. Una lista no ordenada es un conjunto de elementos relacionados entre sí pero para los que no se indica un orden o secuencia determinados. La etiqueta `` encierra todos los elementos de la lista y la etiqueta `` cada uno de sus elementos.

El siguiente código HTML muestra un ejemplo sencillo de lista no ordenada:

```
<!DOCTYPE html>
<html>

<head><title>Ejemplo de etiqueta UL</title></head>

<body>
<h1>Menú</h1>

<ul>
<li>Inicio</li>
<li>Noticias</li>
<li>Artículos</li>
<li>Contacto</li>
</ul>

</body>
</html>
```



Listas ordenadas

Las listas ordenadas son casi idénticas a las listas no ordenadas, salvo que en este caso los elementos relacionados se muestran siguiendo un orden determinado. Cuando se crea por ejemplo una lista con las instrucciones de un producto, es importante el orden en el que se realiza cada paso. Cuando se muestra un índice o tabla de contenidos en un libro, es importante el orden de cada elemento del índice.

En todos estos casos, la lista más adecuada es la lista ordenada, que se define mediante la etiqueta ``. Los elementos de la lista se definen mediante la etiqueta ``, la misma que se utiliza en las listas no ordenadas.

```
<!DOCTYPE html>
<html>
<head><title>Ejemplo de etiqueta OL</title></head>
<body>
<h1>Instrucciones</h1>
<ol>
<li>Enchufar correctamente</li>
<li>Comprobar conexiones</li>
<li>Encender el aparato</li>
</ol>
</body>
</html>
```

Listas de definición

Las listas de definición apenas se utilizan en la mayoría de páginas HTML. Su funcionamiento es similar al de un diccionario, ya que cada elemento de la lista está formado por términos y definiciones. La etiqueta `<dl>` crea la lista de definición y las etiquetas `<dt>` y `<dd>` definen respectivamente el término y la descripción de cada elemento de la lista.



Imágenes

Las imágenes son uno de los elementos más importantes de las páginas web. De hecho, prácticamente todas las páginas web contienen alguna imagen y la mayoría incluyen decenas de imágenes. Dentro de las imágenes que se pueden incluir en una página HTML se deben distinguir dos tipos: las imágenes de contenido y las imágenes de adorno.

Las imágenes de contenido son las que proporcionan información y complementan la información textual. Las imágenes de adorno son las que se utilizan para hacer bordes redondeados, para mostrar pequeños iconos en las listas de elementos, para mostrar fondos de página, etc. Las imágenes de contenido se incluyen directamente en el código HTML mediante la etiqueta `` y las imágenes de adorno no se deberían incluir en el código HTML, sino que deberían emplearse hojas de estilos CSS para mostrarlas.

Atributos específicos

- .: `src = "url"` - Indica la URL de la imagen que se muestra
- .: `alt = "texto"` - Descripción corta de la imagen
- .: `longdesc = "url"` - Indica una URL en la que puede encontrarse una descripción más detallada de la imagen
- .: `name = "texto"` - Nombre del elemento imagen
- .: `height = "unidad_de_medida"` - Indica la altura con la que se debe mostrar la imagen (no es obligatorio que coincida con la altura original de la imagen)
- .: `width = "unidad_de_medida"` - Indica la anchura con la que se debe mostrar la imagen (no es obligatorio que coincida con la anchura original de la imagen)

Los dos atributos requeridos son **src** y **alt**. El atributo `src` es similar al atributo `href` de los enlaces, ya que establece la URL de la imagen que se va a mostrar en la página. Las URL indicadas pueden ser absolutas o relativas.

El atributo `alt` permite describir el contenido de la imagen mediante un texto breve. Las descripciones deben tener una longitud inferior a 1024 caracteres y son útiles para las personas y dispositivos discapacitados que no pueden acceder a las imágenes.

Ejemplo sencillo para incluir una imagen:

```

```



Como `` es una etiqueta vacía, no tiene etiqueta de cierre. No obstante, para que la página XHTML sea válida, todas las etiquetas deben estar cerradas. Como ya se explicó anteriormente, para cerrar una etiqueta vacía se incluyen los caracteres `/>` al final de la etiqueta.

HTML no impone ninguna restricción sobre el formato gráfico que se puede utilizar en las imágenes, por lo que en principio la etiqueta `` puede incluir cualquier formato gráfico existente. Sin embargo, si la imagen utiliza un formato poco habitual, todos o algunos navegadores no serán capaces de mostrar esa imagen.

La recomendación es utilizar uno de los tres siguientes formatos gráficos que entienden todos los navegadores modernos: GIF, JPG y PNG. El formato PNG presenta el inconveniente de que los navegadores obsoletos como Internet Explorer 6 no muestran correctamente las imágenes con transparencias de 24 bits.





Resumen

En esta Unidad...

Comenzamos a introducirnos en el lenguaje de creación de sitios web: HTML.

Con las etiquetas propuestas podemos comenzar a plantear la estructura semántica de una página web.

En la próxima Unidad...

En la próxima unidad vamos a comenzar a trabajar con los nuevos elementos incorporados en la versión HTML5.