

Listas

Ahora volvamos nuestra atención hacia las listas. Las listas están en todos los aspectos de nuestra vida: desde la lista de compras a la lista de instrucciones que sigues inconscientemente para llegar a casa todos los días, o las listas de instrucciones que sigues en estos tutoriales. Las listas están en todos lados en la web también y hay tres diferentes tipos de las que nos vamos a ocupar.

Listas no ordenadas

Las listas no ordenadas se usan para marcar listas de artículos cuyo orden no es importante. Tomemos como ejemplo una lista de compras:

```
leche  
huevos  
pan  
hummus
```

Copy to Clipboard

Cada lista desordenada comienza con un elemento [``](#) («*unordered list*») que delimita todos los elementos de la lista:

```
<ul>  
leche  
huevos  
pan  
hummus  
</ul>
```

Copy to Clipboard

El siguiente paso es delimitar cada artículo de la lista con un elemento [``](#) («*list item*»).

```
<ul>  
  <li>leche</li>  
  <li>huevos</li>  
  <li>pan</li>  
  <li>hummus</li>  
</ul>
```

Copy to Clipboard

Aprendizaje activo: marcar una lista no ordenada

Edita el siguiente ejemplo para crear tu propia lista HTML no ordenada.

Listas ordenadas

Las listas ordenadas son aquellas en las que el orden de los elementos *sí* importa. Tomemos como ejemplo una lista de instrucciones para seguir un itinerario:

```
Conduce hasta el final de la calle
Gira a la derecha
Sigue derecho por las dos primeras glorieta
Gira a la izquierda en la tercer glorieta
El colegio está a la derecha, 300 metros más adelante
```

Copy to Clipboard

La estructura de marcado es la misma que para las listas no ordenadas, excepto que debes delimitar los elementos de la lista con una etiqueta `` («ordered list»), en lugar de ``:

```
<ol>
  <li>Conduce hasta el final de la calle</li>
  <li>Gira a la derecha</li>
  <li>Sigue derecho por las dos primeras glorieta</li>
  <li>Gira a la izquierda en la tercera glorieta</li>
  <li>El colegio está a tu derecha, 300 metros más adelante</li>
</ol>
```

Copy to Clipboard

Aprendizaje activo: Marcar una lista ordenada

Edita el siguiente ejemplo para crear tu propia lista ordenada en HTML.

Aprendizaje activo: marcar la página de tu receta

Llegados a este punto del artículo, tienes toda la información necesaria para marcar la página de ejemplo con tu receta. Puedes escoger entre guardar una copia local del archivo inicial [text-start.html](#) y trabajar en él, o hacerlo sobre el ejemplo editable de abajo. Probablemente sea mejor trabajar en tu copia local porque podrás guardar tu progreso, mientras que, si lo haces sobre el ejemplo editable, los cambios se perderán la próxima vez que cargues la página. Ambos métodos tienen pros y contras.

Si encallas, siempre puedes pulsar el botón *Mostrar solución* o comprobar el ejemplo completo [text-complete.html](#) en nuestro repositorio de Github.

Listas anidadas

Es perfectamente correcto anidar una lista dentro de otra. Posiblemente quieras tener subelementos bajo elementos de rango superior. Tomemos la segunda lista de nuestro ejemplo de la receta:

```
<ol>
  <li>Pela el ajo y picarlo en trozos gruesos.</li>
  <li>Retira las semillas y el tallo del pimiento, y cortarlo en trozos gruesos.</li>
  <li>Mete todos los alimentos en un procesador de alimentos.</li>
  <li>Pica todos los ingredientes hasta conseguir una pasta.</li>
  <li>Si deseas un hummus "grueso", procésalo corto tiempo.</li>
  <li>Pica durante más tiempo si se desea obtener un hummus "suave".</li>
</ol>
```

Copy to Clipboard

Puesto que los dos últimos elementos están estrechamente relacionados con el elemento anterior (se leen como subinstrucciones u opciones que encajan bajo ese elemento), puede tener sentido anidarlos dentro de su propia lista no ordenada e introducir esa lista bajo el cuarto elemento. Tendría el siguiente aspecto:

```
<ol>
  <li>Pela el ajo y picarlo en trozos gruesos.</li>
  <li>Retira las semillas y el tallo del pimiento, y cortarlo en trozos gruesos.</li>
  <li>Mete todos los alimentos en un procesador de alimentos.</li>
  <li>Procesa todos los ingredientes hasta conseguir una pasta.
    <ul>
      <li>Si deseas un hummus "grueso", procésalo corto tiempo.</li>
      <li>Pica durante más tiempo si se desea obtener un hummus "suave".</li>
    </ul>
  </li>
</ol>
```

¿Qué es un hipervínculo?

Los hipervínculos son una de las innovaciones más interesantes que ofrece la Web. Han formado parte de esta desde el principio, pero hacen que la web sea *web* : Los hipervínculos nos permiten vincular documentos a otros documentos o recursos, vincular a partes específicas de documentos o hacer que las aplicaciones estén disponibles en una dirección web. Prácticamente cualquier contenido web se puede convertir en un enlace que, al pulsarlo (activarlo), dirija el navegador a la dirección web a la que apunta el enlace (URL).

Nota: Una URL puede apuntar a archivos HTML, archivos de texto, imágenes, documentos de texto, archivos de audio o vídeo, y cualquier otra cosa que se pueda mostrar en la web. Si el navegador no sabe cómo manejar el archivo, te preguntará si lo quieres abrir (en cuyo caso la tarea de abrirlo y manejarlo se transferirá a la aplicación nativa instalada en el dispositivo) o si lo quieres descargar (en cuyo caso podrás ocuparte de él más tarde).

El sitio web de la BBC, por ejemplo, contiene una gran cantidad de enlaces que apuntan a multitud de noticias en diferentes zonas de el sitio (funcionalidad de navegación), zonas de acceso/registro (herramientas de usuario) y otras.

The screenshot shows the BBC homepage. At the top is the navigation bar with the BBC logo, a 'Sign in' button, and links for News, Sport, Weather, iPlayer, TV, Radio, and a 'More' dropdown. A search bar is on the right. Below the navigation bar, a 'Welcome to the BBC' message is displayed for Tuesday, 2 February, with a link to 'Customise your Homepage'. The main content area features three large featured stories: 'How Pep Guardiola gatecrashed deadline day' (Football), 'US election: Decision day in Iowa latest' (US & Canada), and 'The tragedy of US journalist who shot herself on TV' (Entertainment & Arts). Below these are two sections: 'Latest news >' and 'Latest sport >'. The 'Latest news' section includes headlines like 'Ted Cruz wins Iowa Republican vote', 'Ted Cruz victorious in Iowa', and 'EU law 'veto powers' to be unveiled'. The 'Latest sport' section includes 'January spending at five-year high', 'Mayweather offered 'crazy numbers'', and 'Which clubs outspent La Liga?'.

BBC Sign in News Sport Weather iPlayer TV Radio More Search

Welcome to the BBC
Tuesday, 2 February
[Sign in](#) with your BBC ID, or [Register](#) to see Weather, Local News and more

[Customise your Homepage](#)

How Pep Guardiola gatecrashed deadline day
FOOTBALL

US election: Decision day in Iowa latest
US & CANADA

The tragedy of US journalist who shot herself on TV
ENTERTAINMENT & ARTS

Latest news >

Ted Cruz wins Iowa Republican vote
US & CANADA

Ted Cruz victorious in Iowa
US ELECTION 2016

EU law 'veto powers' to be unveiled

Latest sport >

January spending at five-year high
FOOTBALL

Mayweather offered 'crazy numbers'
BOXING

Which clubs outspent La Liga?

Anatomía de un enlace

Un enlace básico se crea incluyendo el texto (o cualquier otro contenido, ve Convertir bloques de contenido en enlaces), que queramos convertir en un enlace usando un elemento ancla `<a>`, dándole un atributo `href` (también conocido como «**Hypertext Reference**», «**target**» u **objetivo**) que contendrá la dirección web hacia dónde queremos que apunte el enlace.

```
<p>Crea un enlace a  
<a href="https://www.mozilla.org/es-ES/">la página de inicio de  
Mozilla</a>.  
</p>
```

Copy to Clipboard

Este código producirá el siguiente resultado:

Crea un enlace a la página de inicio de Mozilla.

Añadir información de asistencia con el atributo `title`

Otro atributo que posiblemente quieras agregar a tus enlaces es `title`. El título contiene información adicional sobre el enlace, como qué tipo de información contiene la página o cosas que debes tener en cuenta en el sitio web.

```
<p>Crea un enlace a  
<a href="https://www.mozilla.org/es-ES/"  
  title="El mejor lugar para encontrar más información acerca de la  
  misión de Mozilla  
  y cómo contribuir">la página de inicio de Mozilla</a>.  
</p>
```

Copy to Clipboard

Este código producirá el siguiente resultado (el título se mostrará al pasar el ratón sobre el texto del enlace):

Crea un enlace a la página de inicio de Mozilla.

Nota: El título de un enlace solo será visible al pasar el ratón por encima, lo cual significa que los usuarios que naveguen usando los controles de sus teclados, o pantallas táctiles, tendrán dificultades para acceder a la información proporcionada por el título. Si la información del título es verdaderamente importante para el uso

de la página, deberemos presentar el título de manera que sea accesible a todos los usuarios, por ejemplo incluyéndola como parte del texto del enlace.

Aprendizaje activo: crea tu propio ejemplo de enlace

Es momento del aprendizaje activo — crea un documento HTML con tu editor de código (nuestra plantilla de aprendizaje te hará la tarea más llevadera).

- En el cuerpo del HTML (`body`), añade uno o más párrafos o cualquier otro tipo de contenido de los que ya conoces.
- Convierte algún texto en enlaces.
- Incluye atributos de título (`title`).

Convertir bloques de contenido en enlaces

Como hemos mencionado anteriormente, puedes convertir cualquier contenido en un enlace, incluso Elementos de bloque y elementos en línea. Si quieres convertir una imagen en un enlace, simplemente usa el elemento `<a>` encerrando el elemento `` entre `<a>` y ``.

```
<a href="https://www.mozilla.org/es-ES/">  
    
</a>
```

Copy to Clipboard

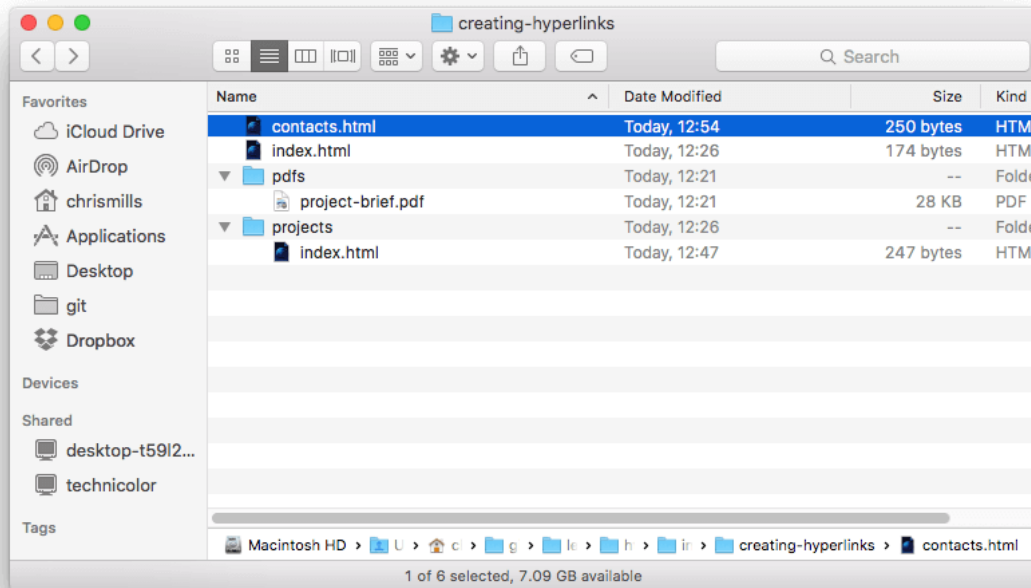
Nota: Encontrarás mucho más sobre el manejo de imágenes en próximos artículos en esta web.

Primer acercamiento a URLs y rutas

Para comprender completamente a dónde apuntan los enlaces, necesitas conocer las URLs y las rutas. En esta sección encontrarás la información que necesitas sobre el tema.

Una localizadora uniforme de recursos (URL, de las iniciales en inglés de «*Uniform Resource Locator*») es simplemente una secuencia de caracteres de texto que definen donde está situado algo en la web. Por ejemplo, la página de Mozilla está ubicada en `https://www.mozilla.org/es-ES/`.

Las URLs utilizan rutas para encontrar los archivos. Las rutas especifican dónde se encuentra el archivo que buscas dentro del sistema de archivos. Veamos un ejemplo de una estructura de directorios (ve el directorio `creating-hyperlinks`).



Al directorio **raíz** de esta estructura de directorios lo hemos llamado `creating-hyperlinks`. Al trabajar en modo local en una web, habrá un directorio que contendrá toda la información. En nuestro ejemplo, dentro de la raíz, encontramos el archivo `index.html` y el archivo `contacts.html`. En una web real, `index.html` es el punto de entrada a la web, lo que se conoce como *página de inicio*.

Observamos también dos directorios dentro de nuestro directorio raíz que son: `pdfs` y `projects`. Cada uno de ellos tiene archivos en su interior — un archivo PDF (`project-brief.pdf`) y un archivo `index.html`, respectivamente. Observa que es posible tener sin problemas dos archivos `index.html` en un proyecto siempre y cuando se encuentren alojados en ubicaciones diferentes de nuestra estructura de archivos — muchos sitios web lo hacen. El segundo `index.html` será la página de inicio para la información relativa a los proyectos.

- **En el mismo directorio:** Si queremos incluir un hipervínculo dentro del archivo `index.html` (el `index.html` del nivel más alto) que apunte al archivo `contacts.html`, simplemente especificaremos el nombre del archivo al que hacemos referencia, porque se encuentra en el mismo directorio en el

que se encuentra el archivo `index.html` desde donde lo queremos llamar.

Por lo tanto, usamos la URL `contacts.html` — veamos el código:

- `<p>¿Quieres contactar con un miembro específico del personal?
Encuentra los detalles en nuestra página de
contactos.</p>`

Copy to Clipboard

- **Bajando por la estructura de subdirectorios:** Si queremos incluir un hipervínculo dentro del archivo `index.html` (el `index.html` de nivel más alto) que apunta a `projects/index.html`, debemos bajar hasta el directorio `projects` antes de indicar al archivo al que queremos enlazar. Para ello especificamos el nombre del directorio y le añadimos una barra inclinada hacia adelante, y a continuación el nombre del archivo. Por lo tanto, utilizaremos la URL `projects/index.html`:

```
<p>Visita mi <a href="projects/index.html">página de inicio del  
proyecto</a>.</p>
```

Copy to Clipboard

- **Subiendo por nuestro sistema de directorios:** Si ahora queremos incluir un hipervínculo dentro del archivo `projects/index.html` que apunte a `pdfs/project-brief.pdf`, hay que subir un nivel en nuestro sistema de directorios, para luego bajar al directorio `pdf`. Para «Subir un nivel» utilizamos los dos puntos — `(. .)` — por lo que usamos la URL `../pdfs/project-brief.pdf`:

```
<p>Un enlace a mi<a href="../pdfs/project-brief.pdf">resumen del  
proyecto</a>.</p>
```

Copy to Clipboard

Nota: Podemos combinar más de una instancia de estas características y generar

URLs más complejas, si es necesario, por

ejemplo: `../../../../ruta/compleja/a/mi/archivo.html`.

Fragmentos de documento

Es posible apuntar hacia una parte concreta de un documento HTML en vez de a todo un documento. Para ello hay que asignar previamente un atributo [id](#) al elemento hacia el que apuntamos. Esto se debe hacer en el encabezado y quedará así:

```
<h2 id="Dirección_de_envío">Dirección de envío</h2>
```

Copy to Clipboard

Posteriormente para hacer referencia a este `id` concreto, lo añadiremos al final de la URL precedido por una almohadilla — veamos el ejemplo:

```
<p>¿Quieres mandarnos una carta? Aquí tienes nuestra <a href="contacts.html#Dirección_de_envío">Dirección de envío</a>.</p>
```

Copy to Clipboard

También podemos usar esta referencia a un fragmento de documento para apuntar hacia otra *parte del mismo documento*:

```
<p>La <a href="#Dirección_de_envío">Dirección de envío de la empresa</a> se encuentra al final de esta página.</p>
```

Copy to Clipboard

URLs absolutas y relativas

Dos términos que encontrarás en la Web son **URL absoluta** y **URL relativa**:

URL absoluta: Hace referencia a una dirección definida por su ubicación absoluta en la web, esta incluye el protocolo y el nombre del dominio. Por ejemplo, si subes una página de inicio `index.html` a un directorio llamado `projects` que se encuentra dentro de la raíz de un servidor web, y el dominio del sitio web es `http://www.example.com`, se podrá acceder a la página desde `http://www.example.com/projects/index.html` (o simplemente `http://www.example.com/projects/`, ya que la mayoría de los servidores web buscan la página de inicio `index.html` para cargarla por omisión si no se les especifica otra en la URL).

Una URL absoluta siempre apuntará a la misma dirección, sin importar desde dónde se utilice.

Una **URL relativa:** Hace referencia a una dirección que depende de la posición del archivo desde donde se utiliza — son las que vimos en la sección anterior. Por ejemplo, si desde un archivo ubicado en `http://www.example.com/projects/index.html` queremos enlazar hacia un

archivo PDF ubicado en el mismo directorio, la URL sería simplemente el nombre del archivo (por ejemplo: `project-brief.pdf`) no necesitamos más información. Si el archivo PDF está situado en un subdirectorio dentro de `projects` llamado `pdfs`, la URL relativa es: `pdfs/project-brief.pdf` (la URL absoluta equivalente sería: `http://www.example.com/projects/pdfs/project-brief.pdf`).

Una URL relativa hará referencia a diferentes direcciones según dónde se encuentre el archivo desde el cual se utiliza — por ejemplo, si movemos nuestro archivo `index.html` del directorio `projects` a la raíz del sitio web (el nivel más alto, no cualquiera de los otros directorios), la URL relativa `pdfs/project-brief.pdf` ahora hará referencia a `http://www.example.com/pdfs/project-brief.pdf`, en lugar de a `http://www.example.com/projects/pdfs/project-brief.pdf`.

Por supuesto, la ubicación del archivo `pdfs/project-brief.pdf` y del directorio `pdfs` no cambian de repente cuando mueves el archivo `index.html`; esto hará que tus enlaces apunten a un sitio equivocado y no funcionarán correctamente al hacer clic en ellos. ¡Por lo tanto debes tener cuidado!

Buenas prácticas en el uso de los enlaces

Hay algunas buenas prácticas que debemos respetar cuando escribimos enlaces. Veamos cuáles son.

Redacción clara del enlace

Es fácil rellenar de enlaces una página, sin más. Pero esto no basta. Hay que lograr que nuestros enlaces sean *accesibles* para todo tipo de lectores, sin importar el contexto o las herramientas que prefieran. Por ejemplo:

- Los usuarios de lectores de pantalla suelen saltar de enlace a enlace en la página y los leen todos sin contexto.
- Los motores de búsqueda utilizan el texto de los enlaces para indizar los archivos buscados, por lo que es una buena idea incluir palabras clave al definir el texto de los enlaces para describir de forma efectiva el sitio al que apuntan.
- Los usuarios visuales suelen echar un vistazo rápido a la página y leen solo lo que les interesa, en lugar de leer todo el texto palabra por palabra, y sus miradas van directamente a las características destacadas de la página, como son los enlaces. Este tipo de usuarios encuentra útiles los textos descriptivos que estos contienen.

Veamos un ejemplo concreto:

Buen texto en un enlace: Descargar Firefox

```
<p><a href="https://firefox.com/">  
  Descargar Firefox  
</a></p>
```

Copy to Clipboard

Mal texto en un enlace: Pulsar aquí para descargar Firefox

```
<p><a href="https://firefox.com/">  
  Haz clic aquí  
</a>  
para descargar Firefox</p>
```

Copy to Clipboard

Otras indicaciones:

- No repetir la URL como parte del texto — Las URLs suenan horrible, y todavía suenan peor si las lee un lector de pantalla, letra por letra.
- No escribir «*link*» o «*link a*» o «enlace» o «enlace a» en el texto del enlace porque es redundante. Los lectores automáticos indican que hay un enlace al encontrarlo. Los usuarios también saben que hay un enlace, porque normalmente se suele cambiar el color del texto y se subraya (no hay que romper esta convención, porque los usuarios están acostumbrados a ella).
- Redacta la etiqueta del enlace de la manera más breve y concisa posible — los textos de enlace largos son especialmente molestos para los usuarios que utilizan lectores automáticos, porque tienen que escuchar todo el texto del enlace.
- Minimiza los casos en los que varias copias del mismo texto están vinculadas a diferentes lugares. Esto puede causar problemas a los usuarios de lectores de pantalla, si hay una lista de enlaces fuera de contexto que están etiquetados como "haz clic aquí", "haz clic aquí", "haz clic aquí".

Utiliza enlaces relativos siempre que sea posible

A partir de las indicaciones anteriores podemos llegar a pensar que es mejor utilizar referencias absolutas en todos los casos; después de todo, estas no se rompen cuando la página se traslada como ocurre con las referencias relativas. Sin embargo, debes utilizar enlaces relativos siempre que sea posible cuando enlaces a otras ubicaciones dentro del *mismo sitio web*. Cuando vinculas a otro sitio web, deberás utilizar un vínculo absoluto.

- Primero, porque el código es mucho más fácil de leer — las URLs relativas suelen ser mucho más cortas que las absolutas, lo que hace que el código sea mucho más fácil de leer.

- Segundo, resulta más eficiente utilizar URLs relativas cuando sea posible. Cuando usas una URL absoluta, el navegador comienza buscando la ubicación real del servidor en el Sistema de nombres de dominio (DNS (en-US)), consulta Cómo funciona la web para obtener más información). Entonces va a ese servidor y busca el archivo solicitado. En cambio, con una URL relativa, el navegador simplemente busca el archivo solicitado en el mismo servidor. Si usas URLs absolutas donde las URLs relativas lo harían, constantemente estás haciendo que tu navegador haga un trabajo adicional, lo cual significa que funcionará de manera menos eficiente.

Indica claramente los recursos no HTML

Cuando damos referencias a recursos para descargarlos (como documentos en formato PDF o Word) o para reproducirlos (como archivos de audio o vídeo) o que tengan un efecto inesperado (una ventana emergente) hay que indicarlo para no confundir al usuario.

Por ejemplo:

- Si tienes una conexión con poco ancho de banda, pulsas un enlace y comienza a descargar un archivo pesado de forma inesperada.
- Si no tienes instalado el reproductor Flash, pulsas un enlace y te conduce a una página que requiere Flash.

Veamos algunos ejemplos, para ver qué texto puede ser aconsejable en estos casos:

```
<p><a href="http://www.example.com/large-report.pdf">
  Descarga el informe de ventas (PDF, 10MB)
</a></p>
```

```
<p><a href="http://www.example.com/video-stream/">
  Reproduce el vídeo (el flujo de datos se abre en una pestaña
  independiente, calidad HD)
</a></p>
```

```
<p><a href="http://www.example.com/car-game">
  Juega al juego del automóvil (requiere Flash)
</a></p>
```

Copy to Clipboard

Utiliza el atributo `download` al enlazar una descarga

Si queremos hacer referencia a una descarga en lugar de a algo que abra el navegador, disponemos del atributo `download` para proporcionar un nombre

predeterminado al archivo a guardar. Veamos un ejemplo con un enlace a la descarga de la versión para Windows de Firefox:

```
<a href="https://download.mozilla.org/?product=firefox-latest-ssl&os=win64&lang=es-MX"
  download="firefox-latest-64bit-installer.exe">
  Descarga la última versión de Firefox para Windows (64 bits) (Español,
es-MX)
</a>
```