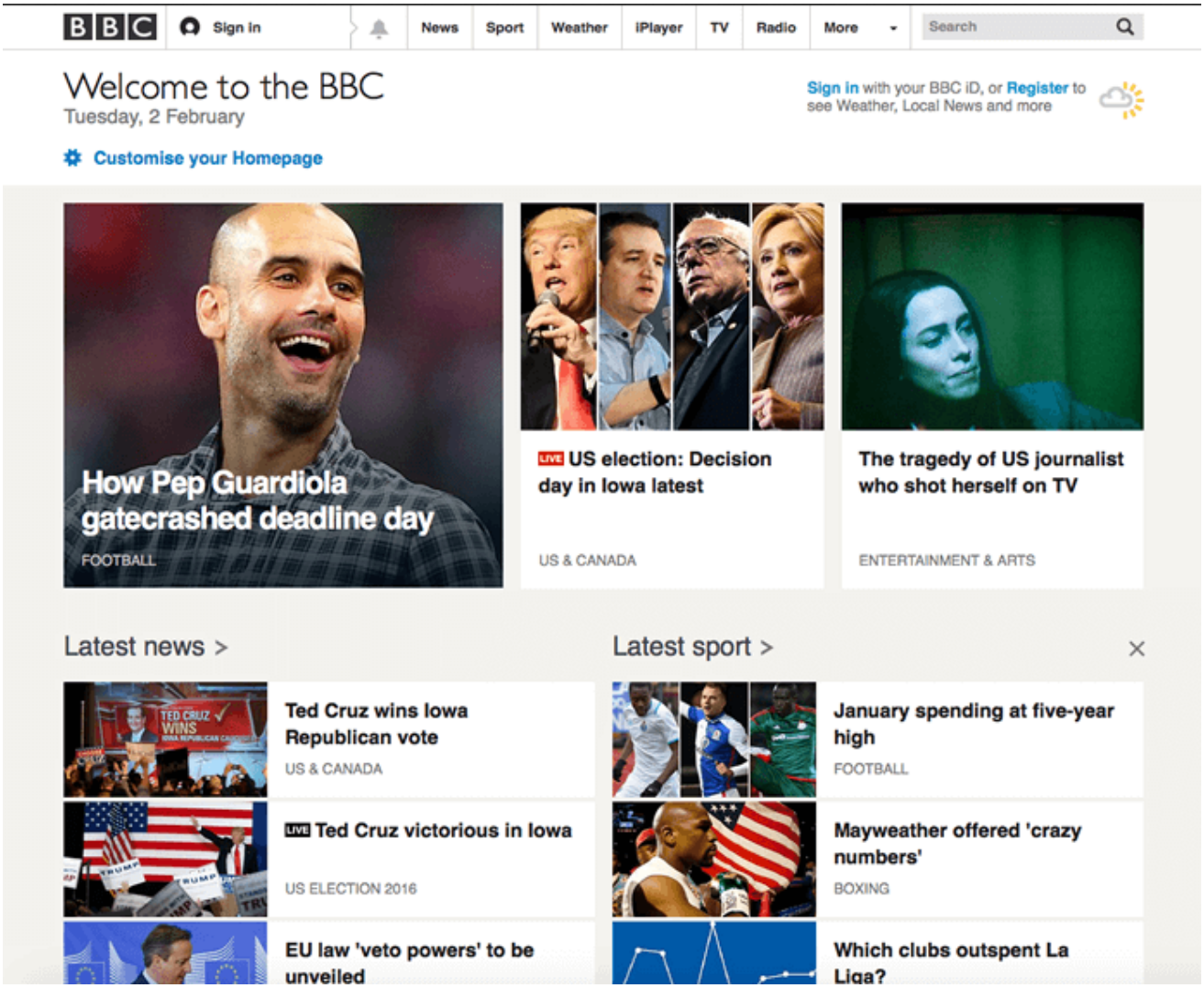
Creando hipervínculos

Los *Hipervínculos* ó *enlaces* son verdaderamente importantes — son los que hacen que la Web sea **Web**. En este artículo aprenderemos la sintaxis requerida para programar un *link* , y un catálogo de buenas prácticas para hacerlo.

¿Qué es un hyperlink?

Los hyperlinks son de las más excitantes novedades que la Web nos ofrece. De acuerdo, han formado parte de la Web desde el principio, pero hacen que la Web sea *Web* — permiten enlazar nuestros documentos a cualquier otro documento (o recurso), hacer referencia a partes específicas de los mismos, podemos hacer las aplicaciones accesibles con una sencilla dirección web (al contrario de las apps nativas que deben ser instaladas con todo lo que esto conlleva). Cualquier contenido web puede ser convertido en un link, para que al pulsarlo (lo activemos) el navegador se dirija a la dirección web a la que apunte el link. (**URL** .)

> **Nota**: Una URL puede apuntar a ficheros HTML, ficheros de texto, imágenes, documentos de texto, archivos de audio o video, y cualquier otra cosa que pueda ser mostrada en la Web. Si el navegador no sabe cómo manejar el archivo, nos preguntará si lo queremos abrir (en cuyo caso la tarea de abrirlo y manejarlo será transferida a la aplicación nativa instalada en el dispositivo) o lo queremos descargar (en cuyo caso puedes ocuparnos de él más tarde). La web de la BBC, por ejemplo, contiene gran cantidad de enlaces apuntando a multitud de noticias, a diferentes zonas de su web (utilidades de navegación), zonas de ingreso/registro (utilidades de usuario) y otras.



Anatomía de un link

Un link básico se crea incluyendo el texto (o cualquier otro contenido **Block level links**) que deseemos convertir en un link dentro de un elemento ancla `<a>`, dándole un atributo `href` (también conocido como **target** - objetivo) que contendrá la dirección web hacia dónde deseemos que apunte el link.

```
<p>I'm creating a link to
<a href="https://www.mozilla.org/en-US/">the Mozilla homepage</a>.
</p>
```

Este código producirá el siguiente resultado:

I'm creating a link to **the Mozilla homepage**.

Añadir información de soporte con el atributo title

Otro atributo que podemos añadir a los links es el título (`title`); este ha sido concebido para proporcionar información útil sobre el destino, como el tipo de información que contiene la página, o cosas a tener en cuenta sobre el mismo. Por ejemplo:

```
<p>I'm creating a link to
<a href="https://www.mozilla.org/en-US/"
   title="The best place to find more information about Mozilla's
          mission and how to contribute">the Mozilla homepage</a>.
</p>
```

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

I'm creating a link to **the Mozilla homepage**.

> **Nota**: El título de un link solo será visible al pasar el ratón por encima, lo que significa que los usuarios que naveguen usando los controles de sus teclados 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 link.

Aprendizaje activo: Crea tu propio ejemplo de link

Es momento del aprendizaje activo: Deberás crear un documento HTML usando tu propio editor de código.

- En el cuerpo del HTML (`body`), intenta añadir uno o más párrafos o cualquier otro tipo de contenidos de los que ya conoces.
- Convierte alguno de los contenidos en links.
- Incluye atributos de título (title).

Convertir bloques de contenido en links

Como hemos mencionado anteriormente, puedes convertir cualquier contenido en un link, incluso los bloques de contenido **block level elements**. Si dispones de una imagen que desees convertirla en un link, simplemente ánclala entre elementos `<a></a>` .

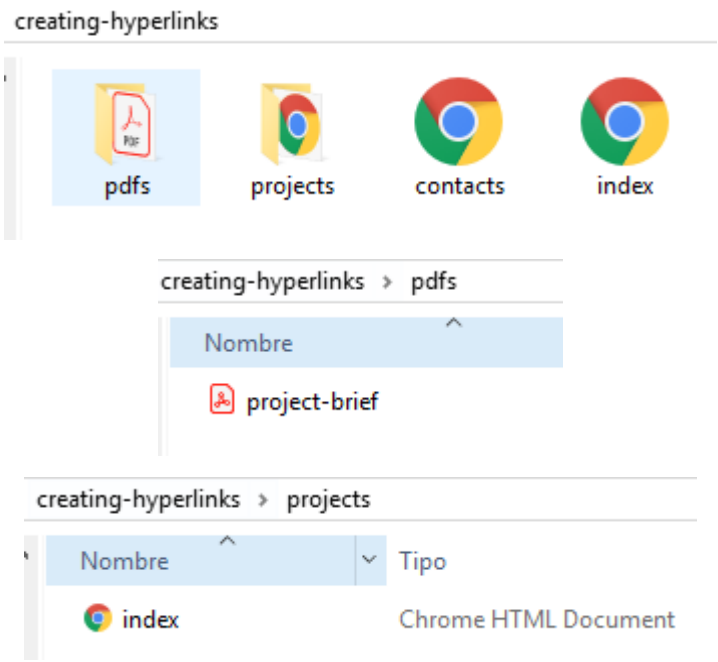
```
<a href="https://www.mozilla.org/en-US/">
  <img src="mozilla-image.png" alt="mozilla logo that links to the mozilla homepage">
</a>
```

Un primer acercamiento a URLs y referencias

Para comprender completamente donde apuntan los links, necesitas conocer URLs y las referencias. En esta sección encontrarás la información que necesitas sobre el tema.

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

Las URLs utilizan las referencias para encontrar los archivos. Las referencias especifican donde se encuentra el archivo que buscas dentro del sistema de archivos. Veamos un ejemplo de una estructura de directorios:



Al directorio **raíz** de esta estructura de directorios la hemos llamado `creating-hyperlinks`. Al trabajar en modo local en una web, tendremos 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`. Estos además tienen un fichero en su interior — un PDF (`project-brief.pdf`) y un fichero `index.html`, respectivamente. Fíjate cómo puedes tranquilamente tener dos archivos `index.html` en un proyecto ya que se encuentran alojados físicamente 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 quisiéramos incluir un hyperlink dentro del archivo `index.html` (el `index.html del nivel más alto`) apuntando al archivo `contacts.html`, simplemente especificaremos el nombre del archivo al que hagamos referencia, pues este se encuentra en el mismo directorio en el que se encuentra el archivo index.html desde donde lo queremos llamar. Por lo tanto, la URL que usaremos será `"contacts.html"`, veamos el código:

```
<p>Want to contact a specific staff member?
Find details on our <a href="contacts.html">contacts page</a>.</p>
```

- **Bajando en la estructura de subdirectorios**: Si quisiéramos incluir un hyperlink dentro del archivo `index.html` (el `index.html de más alto nivel`) apuntando a `projects/index.html`, deberíamos bajar hasta el directorio `projects` antes de indicar al archivo al que queremos enlazar. Esto lo haremos especificando el nombre del directorio añadiéndole una barra inclinada hacia delante, y a continuación el nombre del archivo. Por lo tanto, la URL que utilizaremos será `"projects/index.html"`:

```
<p>Visit my <a href="projects/index.html">project homepage</a>.</p>
```

- **Subiendo en nuestro sistema de directorios**: Si ahora quisiéramos incluir un hyperlink dentro del archivo `projects/index.html` apuntado a `pdfs/project-brief.pdf`, tendríamos que subir un nivel en nuestro sistema de directorios, para luego bajar dentro del directorio `pdf`. Para "Subir un nivel" utilizaremos los dos puntos `..` por lo que la URL que usaremos será `"../pdfs/project-brief.pdf"`:

```
<p>A link to my <a href="../pdfs/project-brief.pdf">project brief</a>.</p>
```

> **Nota**: Podemos combinar multiples instancias de estas características generando URLS más complejas, si se necesitan, por ejemplo. `"../../../complex/path/to/my/file.html"` .

Fragmentos de un Documento

Es posible apuntar hacia una parte concreta de un documento HTML en vez a todo un documento. Para hacerlo deberemos asignar previamente un atributo id `id` al elemento hacia el que queremos apuntar. Esto se deberá hacer en el encabezamiento, quedando como sigue:

```
<h2 id="Mailing_address">Mailing address</h2>
```

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

```
<p>Want to write us a letter? Use our <a href="contacts.html#Mailing_address">mailing
address</a>.</p>
```

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

```
<p>The <a href="#Mailing_address">company mailing address</a> can be found at the bottom of
this page.</p>
```

URLs absolutas versus relativas

Dos nomenclaturas que nos encontramos en la Web son **URL absoluta** y **URL relativa**:

URL absoluta: Hace referencia a una dirección definida por su ubicación absoluta en la web, incluyendo el protocolo **protocol** y el nombre del dominio **domain name**. Por ejemplo, si una página de inicio `index.html` es subida 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` ", la página será accesible desde `"http://www.example.com/projects/index.html` " (o simplemente `"http://www.example.com/projects/` ", ya que la mayoría de los servidores web solo buscan la página de inicio `index.html` para cargarla por defecto si no se les especifica otra en la URL.)

Una URL absoluta siempre apuntará hacia la misma dirección, sin importar desde donde se utilice.

URL relativa: Hace referencia a una dirección que depende de la posición del archivo desde donde se utiliza, son las que hemos visto en la sección anterior. Por ejemplo, si quisiéramos enlazar desde un archivo ubicado en: " **http://www.example.com/projects/index.html** " hacia un archivo PDF ubicado en el mismo directorio, la URL sería simplemente el nombre del archivo — por ejemplo: `project-brief.pdf` — no necesitaríamos más información para referenciarlo. Si el archivo PDF estuviera situado en un subdirectorio dentro de projects llamado pdfs, la URL relativa sería: "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 dependiendo de dónde se encuentre el archivo desde el cual sea utilizada — por ejemplo si movemos nuestro fichero `index.html` del directorio `projects` a la raíz del sitio web (la de nivel más alto, no el de otros directorios), la URL relativa " `pdfs/project-brief.pdf"` referenciará ahora a " `http://www.example.com/pdfs/project-brief.pdf` ", en lugar de a " `http://www.example.com/projects/pdfs/project-brief.pdf"` `como lo hacía anteriormente` .

Buenas prácticas en el uso de los Links

Hay algunas buenas prácticas que debemos observar cuando escribamos links. Veamos cuales son.

Uso de palabras claras como links

Es fácil vomitar links en una página sin más. Esto no es suficiente. Necesitamos hacer que nuestros links sean *accesibles* para todo tipo de lectores, sin importar el contexto o las herramientas que prefieran. Por ejemplo:

- Los lectores autmáticos van saltando de link en link en la página y los leen todos de forma consecutiva.
- Los motores de búsqueda utilizan los links de texto para indexar los archivos buscados, por lo que es buena idea incluir palabras clave al definir los links de texto para describir de forma efectiva el sitio al que apuntan.
- Los usuarios echan un vistazo rápido a la página leyendo solo aquello 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 links. Este tipo de usuarios encuentran útiles los textos descriptivos de los mismos.

Veamos un ejemplo concreto:

Buen texto en un link: ***Download Firefox***

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

Mal texto en un link: ***Click here*** *to download Firefox*

```
<p><a href="https://firefox.com/">
  Click here
</a>
to download Firefox</p>
```

Otras indicaciones:

- No se debe repetir la URL como parte del texto — las URLs suenan horrible, y todavía suenan peor si el que las lee lo hace letra a letra.
- No escribir "link" o "link a" o "enlace" o "enlace a" en el texto del link — esto es redundante. Los lectores automáticos indican que hay un link al encontrarlo. Los usuarios también saben que hay un link, ya que normalmente se suele cambiar el color del texto y se suele subrayar (esta convención no debe ser rota, pues los usuarios están acostumbrados a ella).
- Redactar la etiqueta del link de la manera más breve y concisa posible — los textos de link largos son especialmente molestos para los usuarios que utilizan lectores automáticos, ya que tienen que escuchar todo el texto de la página.
- Reducir las ocurrencias de copias exactas del mismo texto que apunten a lugares diferentes, ya que los lectores automáticos los leen como una lista descontextualizada — varios links etiquetados como "click here", "click here", "click here" pueden resultar confusos para los usuarios.

Utilizar referencias relativas siempre que sea posible

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, se deben usar referencias relativas siempre y cuando estemos haciendo referencia a direcciones dentro de una misma página web (cuando hagamos referencia a páginas web externas, utilizaremos las referencias absolutas):

- Primero, es mucho más fácil leer la codificación — las URLs relativas son por lo general mucho más cortas que las absolutas, lo que hace que el código sea mucho más fácil de leer.
- Segundo, es mucho más eficiente utilizar URLs relativas cuando sea posible. Al utilizar una URL absoluta, el navegador comienza por la dirección real del servidor haciendo un requerimiento al DNS del Dominio **DNS**, a continuación se dirige a ese servidor y busca el archivo requerido. En cambio, con una URL relativa, el navegador solo busca el fichero requerido dentro del mismo servidor. Por lo tanto si utilizamos URLs absolutas en lugar de relativas, estamos haciendo que el navegador realice constantemente trabajos extra, haciendo que el rendimiento sea menos eficiente.

Enlaces a recursos no-HTML — señalizar con claridad

Cuando referenciamos a recursos para ser descargados (como PDFs o documentos Word) o reproducidos (como audio o video) o que tengan un efecto inesperado (ventana emergente, o reproducción Flash) deberemos señalarlo para no confundir al usuario. Resulta bastante molesto por ejemplo:

- Si tienes una conexión con bajo ancho de banda, pulsas un link y de repente comienza a descargarse un archivo pesado de forma inesperada.
- Si no tienes instalado el reproductor Flash, pulsas un link y eres conducido a una página que requiere Flash.

Veamos algunos ejemplos, para ver el texto aconsejable en estos casos:

```
<p><a href="http://www.example.com/large-report.pdf">
  Download the sales report (PDF, 10MB)
</a></p>

<p><a href="http://www.example.com/video-stream/">
  Watch the video (stream opens in separate tab, HD quality)
</a></p>

<p><a href="http://www.example.com/car-game">
  Play the car game (requires Flash)
</a></p>
```

Utilizar el atributo de descarga (download) cuando se referencia una descarga

Si queremos hacer referencia a una descarga en lugar de a algo abierto por el navegador, disponemos del atributo `download` para proporcionar un nombre al archivo guardado. Veamos un ejemplo del enlace a la descarga de la versión para Windows de Firefox 39:

```
<a href="https://download.mozilla.org/?product=firefox-39.0-SSL&os=win&lang=en-US"
   download="firefox-39-installer.exe">
  Download Firefox 39 for Windows
</a>
```

Enlace a e-mail

Es posible crear enlaces o botones que, cuando se pulsen, abran un nuevo correo saliente en lugar de enlazar a un recurso o página. Esto se consigue con el elemento ancla `<a>` y el elemento `mailto:` URL esquema.

En su forma más básica, un enlace `mailto:` simplemente contiene la dirección email de los destinatarios. Por ejemplo:

```
<a href="mailto:nowhere@mozilla.org">Send email to nowhere</a>
```

De hecho, incluso el atributo dirección email es opcional. Si lo quitamos (`href` dejando simplemente "mailto:"), aparecerá una ventana de nuevo correo saliente en el gestor de correo que no incluirá pre-escrita la dirección del destinatario. Esto es útil cuando queremos compartir enlaces que los usuarios puedan pulsar para enviar un email y elegir un destinatario posteriormente.

Especificando otros detalles

Además de la dirección de email, se puede incluir otra información. De hecho, se puede incluir cualquier campo estándar contenido en el encabezamiento de cualquier mensaje en la URL `mailto` que se proporcione. Los más utilizados son el "subject" (asunto), "cc" (copia a) o "bcc" (copia oculta), y "body" (cuerpo del mensaje, que no es realmente un campo de cabecera, pero permite especificar un breve mensaje para el nuevo email). Cada campo y su valor se especifica como un argumento del requerimiento.

Veamos un ejemplo que incluye estos campos:

```
<a href="mailto:nowhere@mozilla.org?
cc=name2@rapidtables.com&bcc=name3@rapidtables.com&amp;subject=The%20subject%20of%20the%20email
&amp;body=The%20body%20of%20the%20email">
  Send mail with cc, bcc, subject and body
</a>
```

Nota: Los valores de cada campo deber tener codificación URL (es decir, sin caracteres no imprimibles, espacios o porcentajes **percent-escaped**). También observamos el uso del ampersand (&) para separar cada campo dentro del enlace `mailto:` . Esta notación es estándar de los requerimientos URL.

A continuación otros ejemplos de utilización de enlaces `mailto` :

- <mailto:>
- **mailto:nowhere@mozilla.org**
- **mailto:nowhere@mozilla.org,nobody@mozilla.org**
- **mailto:nowhere@mozilla.org?cc=nobody@mozilla.org**
- **mailto:nowhere@mozilla.org?cc=nobody@mozilla.org&subject=This%20is%20the%20subject**