

### 1. ¿QUÉ ES UNA URL?

En Internet, la forma de identificar los recursos o servicios disponibles es a través de las **URL** (Uniform Resource Locator), que no son más que una forma estándar de hacer referencia exactamente a un determinado recurso.

Una URL se construye de la siguiente forma:

**tipoURL://Ordenador/Ubicación\_del\_recurso**

Ya conoces el tipo de URL que más se utiliza y que permite acceder a las páginas web de la World Wide Web. Así una dirección del tipo: **http://www.empresa.com/productos/index.htm** es una URL.

Utilizando otros protocolos, podrás acceder a distintos recursos, pero la sintaxis será la misma:

```
ftp://ftp.microsoft.com
telnet://futunet.fi
gopher://gopher.rediris.es
mailto:info@gmail.com
```

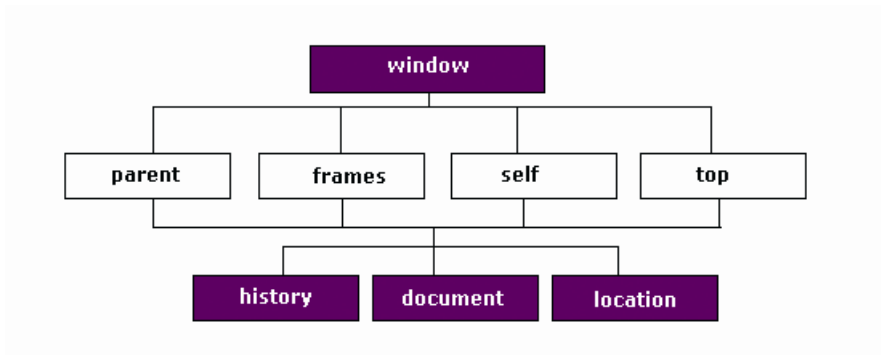
Como puedes ver, se distingue la parte del protocolo, el ordenador al que deseamos acceder y la ubicación, dentro de dicho ordenador, del recurso deseado.

Solo en el caso del protocolo **mailto**, que permite acceder al servicio de correo electrónico, esta estructura no se cumple ya que solo necesita la dirección de correo electrónico.

Pues bien, utilizando el objeto **location**, podrás acceder a la dirección URL de la página web que está cargada. Esto puede servir para cargar páginas de forma directa, sin la necesidad de introducirlas en el campo dirección del navegador o través de un hipervínculo.

### 2. EL OBJETO LOCATION

El objeto **location** representa una URL completa. Si recuerdas la jerarquía de objetos, sabrás que es una propiedad del objeto **window**. Esto quiere decir que puedes utilizar la sintaxis **window.location** o simplemente **location**, siempre que no exista más de un objeto **window**.

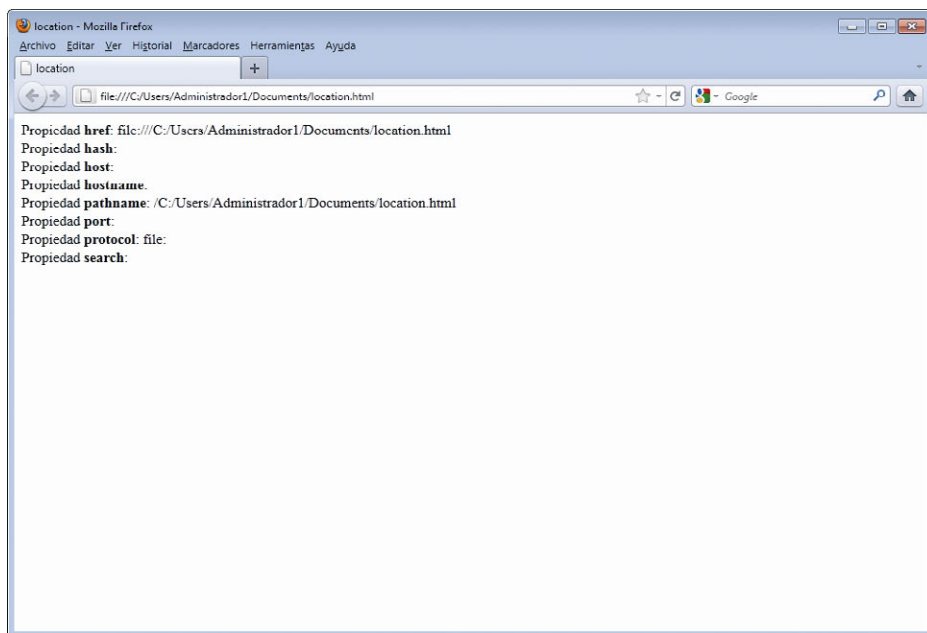


El objeto **location** pertenece a la ventana que contiene el código JavaScript cuando no se utilizan frames. Sin embargo, en una página dividida en frames, cada uno de estos tiene su propio objeto **location**.

Este objeto presenta ocho propiedades y tres métodos, los cuales no se suelen utilizar demasiado. En el siguiente script lo que hacemos es imprimir en la página web el valor de esas propiedades.

```
<!DOCTYPE HTML>
<html>
<head>
<title>location</title>
</head>
<body>
<script type="text/javascript">
<!--
    document.write("Propiedad <strong>href</strong>: "
+ location.href + "<br>")
    document.write("Propiedad <strong>hash</strong>: "
+ location.hash + "<br>")
    document.write("Propiedad <strong>host</strong>: "
+ location.host + "<br>")
    document.write("Propiedad <strong>hostname</strong>: "
+ location.hostname + "<br>")
    document.write("Propiedad <strong>pathname</strong>: "
+ location.pathname + "<br>")
    document.write("Propiedad <strong>port</strong>: "
+ location.port + "<br>")
    document.write("Propiedad <strong>protocol</strong>: "
+ location.protocol + "<br>")
    document.write("Propiedad <strong>search</strong>: "
+ location.search + "<br>")
//-->
</script>
</body>
</html>
```

Veamos el resultado, con lo que la explicación de cada propiedad será más sencilla. Para ello, en esta lección utilizaremos el navegador **Firefox**.



Vayamos por partes:



Si asignas la propiedad **href**, lo que estás haciendo es modificar la dirección de la página web que se carga en el navegador.

- La propiedad **href** es la más importante. Esta propiedad mantiene una cadena de texto que incluye la URL entera de la página. Esta propiedad puede bien obtenerse o bien modificarse.

En este caso se nos indica que la URL de la página es un lugar del disco duro del equipo.

- La propiedad **hash** indica la parte de la URL que representa un marcador. En la URL aparece siempre tras el símbolo #.

En este caso, la URL de la página no contiene esta propiedad.

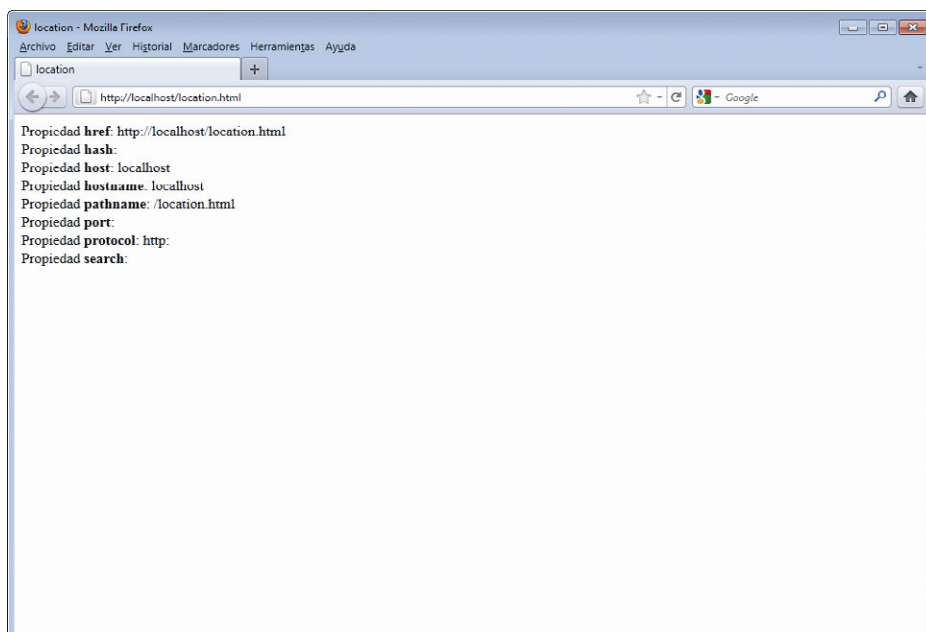
- Las propiedades **host** y **hostname** son similares y representan el ordenador al que se accede. La propiedad **host** incluye, además, el puerto utilizado.
- La propiedad **pathname** representa la parte de la URL que indica la estructura de carpetas donde está ubicado el recurso.
- La propiedad **port** representa el puerto utilizado por el equipo que mantiene la página. No suele utilizarse.
- La propiedad **protocol** representa qué protocolo se está utilizando. Observa cómo, en este caso, es el protocolo **file** ya que la página es un fichero situado en el ordenador local.

- La propiedad **search** se utiliza cuando la URL incluye algún criterio de búsqueda. Esto se utiliza en los buscadores cuando has incluido una o más palabras sobre las que deseas información.

Si te fijas en estos casos, la dirección (URL) de la página donde se muestran los resultados incluye una parte que va tras el símbolo **?**, que es la parte de la URL correspondiente a esta propiedad.

Como la página se encuentra en el disco duro de nuestro equipo y no en un servidor web, los detalles del objeto **location** son poco significativos.

Vamos a acceder a la misma página, pero publicada en un servidor web, en lugar de cargarla como un archivo del disco duro. Fíjate en la URL de la página web de la figura siguiente.



Ahora las propiedades del objeto **location** de esta página han cambiado.

Observa cómo el protocolo es **http** y además podemos obtener el nombre del ordenador en el que reside esta página a través de las propiedades **host** o **hostname** (el nombre es *localhost* en este caso).

Es importante entender la diferencia entre una página publicada y otra que no lo ha sido. Al publicar la página, estás creando un recurso **http**, mientras que un archivo de tu ordenador es un recurso **file**.

### 3. ACTUALIZANDO LA DIRECCIÓN

Ahora que conocemos el objeto **location**, vamos a utilizarlo en algún caso práctico.

Imagina esta situación: tu sitio web ha cambiado de ubicación en Internet, de forma que ahora la dirección que todos conocían ya no es válida.

¿Cuál puede ser una solución para no perder a tus visitantes? Quizás puedas seguir manteniendo una página en la ubicación antigua y acceder automáticamente a la nueva dirección. Al menos, durante un tiempo...

Podemos utilizar la propiedad **href** del objeto **location** para actualizar la página, de forma que se acceda a otra URL.



Al utilizar el manejador de evento **onload** de la etiqueta **<body>** lo que realmente estamos haciendo es escribir código para el evento **load** del objeto **window**.

```
<!DOCTYPE HTML>
<html>
<head>
<title>Este sitio web se ha trasladado</title>
</head>
<body onload="location.href='http://es.yahoo.com';">
<h1>Este sitio web se ha trasladado</h1>
<h1>Accediendo al nuevo sitio web...</h1>
</body>
</html>
```

Es decir, vamos a actualizar la propiedad **href** en el momento en que se cargue la página web (evento **onload** de la etiqueta **body**). En este caso, accederemos a la página del portal *Yahoo*.

Observa cómo ha sido necesario incluir la cadena entre comillas simples porque el valor del manejador de evento **onload** es en sí una cadena.

De esta forma no ha sido necesaria ninguna intervención del usuario para acceder a la nueva ubicación. Esto es muy conveniente ya que debemos evitar todas las molestias posibles a nuestros visitantes para que no caigan en la tentación de ir a otro sitio de la Web.

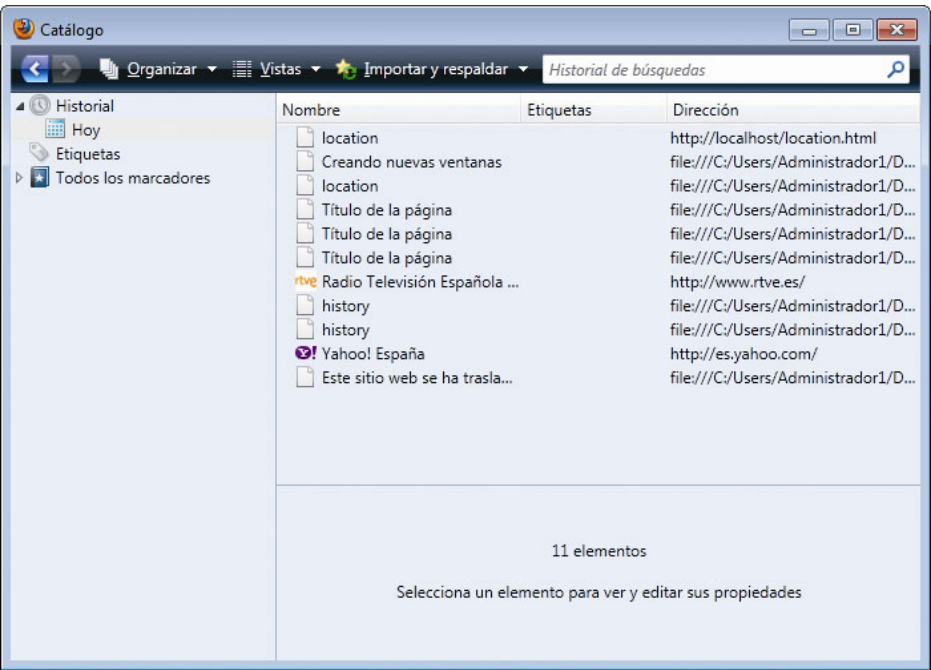
### 4. EL OBJETO HISTORY

El objeto **history** permite acceder a la lista historial del navegador. En esta lista se van situando aquellas direcciones web (URL) que se visitan en una determinada sesión.

Lo importante a la hora de estudiar este objeto es entender que se puede escribir código para moverse a través de esta lista, pero no se puede acceder a las direcciones que están en la misma.

Esto se debe a motivos de seguridad: nadie tiene por qué conocer qué lugares has visitado.

En la figura siguiente puedes ver la lista Historial de Firefox.



La única propiedad del objeto **history** es **length**, que facilita el número de lugares visitados durante la sesión actual, es decir, desde que se inició el navegador.

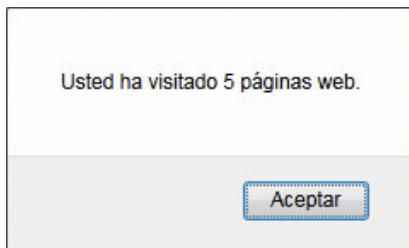
No existen propiedades que permitan acceder a las direcciones (URL) visitadas. Esto es importante recordarlo.

```
<!DOCTYPE HTML>
<html>
<head>
<title>history</title>
</head>
<body>
<script type="text/javascript">
<!--
alert("Usted ha visitado " + history.length + " páginas
web.");
/-->
</script>
</body>
</html>
```

Observa cómo el *script* mostrará el número de páginas visitadas en un cuadro de diálogo **alert**. Para ello, utiliza la propiedad **length** del objeto **history**.

Ahora el navegador nos informa de cuántos lugares has visitado gracias a la propiedad **length**.

Además, el objeto **history** asigna un índice a los lugares que se han ido visitando.



La página actual tiene el valor 0, mientras que los lugares anteriores tienen los valores -1, -2, etc. Los documentos que se han cargado con posterioridad tienen los índices 1, 2, ...

Estos últimos solo aparecen cuando se ha utilizado el botón *Atrás* del navegador o el método **back**.

En este otro ejemplo vamos a utilizar los métodos del objeto **history** para navegar hacia atrás y hacia delante. Para ello, existen tres métodos.

```
<!DOCTYPE HTML>
<html>
<head>
<title>history</title>
</head>
<body>
<form>
<input type="button" value="Atrás"
onclick="history.back();return true;">
<input type="button" value="Adelante"
onclick="history.forward();return true;">
</form>
<a href="http://www.rtve.es/">Últimas noticias</a>
</body>
</html>
```

- **back**: realiza la misma tarea que el botón *Atrás* del navegador. Es útil en aquellas ventanas en las que hemos indicado expresamente que no aparezca la barra de herramientas. En este caso, un botón que realice esta función podría ser adecuado.
- **forward**: realiza la misma tarea que el botón *Adelante* del navegador. Solo tiene sentido cuando se ha utilizado el método **back** con anterioridad o el botón *Atrás* del navegador.

- **go(número | cadena)**: este método permite ir a una URL determinada del historial. Se puede indicar una posición relativa, como en **go(-2)**; o una cadena de texto, indicando que se vaya a la página cuyo título incluya la misma, como en **go("google")**.

Debes recordar que estos métodos no devuelven ningún valor, de forma que no se pueda obtener ninguna información adicional: realizan su labor y nada más.

Por ejemplo, piensa qué ocurriría si el método **go** devolviera el valor *verdadero* o *falso* según se haya podido acceder a una dirección incluida en el historial. Entonces el *script* podría conocer si has accedido a una determinada dirección.

Como puedes observar, utilizando los métodos **back** y **forward**, estamos obteniendo el mismo resultado que si pulsamos en los botones equivalentes del navegador.

Realmente estamos haciendo lo mismo, ya que estos botones se implementan de la misma forma. Piensa en la potencia que te ofrece JavaScript al poder utilizar los objetos del navegador.