Acceso a enlaces

- **document.links** : Array que contiene todos los enlaces de la página.
    - ruta = document.links[0] : Acceso a los enlaces según el orden de aparición. en el ejemplo el primero. Devuelve la ruta del enlace. Solo lectura.
    - ruta = document.links["valor_de_name"] : Acceso a los enlaces según el valor del atributo name.
    - document.links[0].href = "nueva_ruta" : escribir o cambiar la ruta de un enlace.
    - texto = document.links[0].innerHTML : leer texto del enlace.
    - document.links[0].innerHTML = "nuevo_texto" : escribir o cambiar el texto del enlace.

Acceso a imágenes

- **document.images** : Array que contiene todas las imágenes de la página.
    - document.images[0] : acceso a la imagen según su orden de aparición en la página.
    - document.images["valor_de_name"] : acceso a la imagen según su valor del atributo name.
    - Lo que nos interesa realmente es poder leer o escribir los atributos de la imagen, ruta, tamaño, etc.
    - document.images[0].src : acceso a la ruta de la imagen tanto de lectura como de escritura.
    - document.images[0].width : acceso al tamaño (horizontal) de la imagen tanto para lectura como para escritura.

Para los demás atributos de la etiqueta de imagen se procederá de la misma manera.

Acceso a formularios

Formas de acceder a un formulario:

- **document.form** : Array que contiene todos los formularios de la página.

Podemos acceder de cualquiera de estas formas:

- `document.form[0]` : Acceso con Array según el orden de aparición del formulario
- `document.form["nameForm"]` : Acceso con Array según el valor de la etiqueta `name`
- `document.nameForm` : Acceso escribiendo el valor de la etiqueta `name`como una propiedad del objeto `document`.

Formas de acceder a los elementos.

Dentro de cada formulario el DOM crea un array con los elementos. el array **elements**

Accedemos a `elements` como una propiedad del formulario. Desde cualquiera de las formas anteriores de acceder al formulario accedemos a la pñropiedad de cualquiera de estas maneras:

- `document.nameForm.elements[0]` : según el orden de aparición en el formulario.
- `document.nameForm.elements["nameElem"]` : según el valor de la etiqueta `name` del elemento.
- `document.nameForm.nameElem` : Acceso escribiendo el valor de la etiqueta `name` del elemento como una propiedad del formulario.

Propiedades de los elementos

Una vez accedemos a los elementos estos tienen una serie de propiedades comunes:

Propiedades de los elementos

Propiedad	Código y Explicación	
type	**Código**	`tipo = document.nameForm.nameElem.type`
	Explicación	Indica el tipo de elemento. Sólo lectura: ▪ Elementos tipo `input`: valor del atributo `type` ▪ Elementos `select`: valor =`select-one` ó `se multiple` ▪ Elementos `textarea`: valor =`textarea`
name	**Código**	`nombre = document.nameForm.nameElem.name`
	Explicación	Obtiene el valor del atributo `name`. Esta propiedad es de sólo lectura.
value	**Código**	`valor = document.nameForm.nameElem.value` `document.nameForm.nameElem.value = valor`
	Explicación	Permite leer y modificar el valor del atributo `value`. ▪ En campos de texto y `textarea`: se obtiene el escrito por el usuario. ▪ En botones `reset`, `submit` o`button`: Texto muestra el botón ▪ En botones `radio` o `checkbox`: valor atributo `value`.

Botones radio

Los botones `type=radio` que están relacionados tienen todos el mismo atributo `name`. Al acceder a un boton radio se crea un array con todos los botones con el mismo atributo `name`.

Propiedad checked

Una vez hemos accedido a los botones radio, la propiedad `checked` nos indica si el botón está seleccionado:

```
comprobar = document.nameForm.nameElem[0].checked;
```

Este código comprueba si está seleccionado el primer elemento. el resultado es `true` si está seleccionado, y `false` si no lo está.

Mediante un bucle podemos recorrer los elementos del array. Dentro del bucle con una condicional obtenemos el elemento que está seleccionado:

```
var pulsado = document.nameForm.nameElem;
var elegido = "ninguno"
for (i=0; i<pulsado.length; i++) {
    valor = pulsado[i].checked;
    if (valor == true) {
       elegido = pulsado[i].value;
       }
    }
```

En el código anterior obtenemos el valor del atributo "value" del elemento seleccionado.

Botones checkbox

Los botones `checkbox` son independientes unos de otros. La propiedad**checked** nos indica aquí también si están o no seleccionados. Esta propiedad devuelve `true` si está seleccionado, y `false` si no lo está.

```
seleccion1 = document.nameForm.nameElem1.checked;
```

El resultado lo obtenemos en la variable `seleccion1`: para el elemento checkbox con `name="nameElem1"` en el formulario con `name="nameForm"`sera `true` si el elemento está marcado, o `false` si no está marcado.

Listas desplegables

Debemos buscar el elemento o elementos seleccionados, Para una lista en la que sólo se puede elegir un elemento seguiremos los siguientes pasos:

Acceso a elementos select

Paso	Código y Explicación	
1º	Código	`lista = document.nameForm.nameElem`
	Explicación	Accedemos al elemento select del DOM y lo guardamos en una variable.
2º	Código	`opciones = lista.options`
	Explicación	La propiedad `options` crea un array con todas las opciones
3º	Código	`num = lista.selectedIndex`
	Explicación	La propiedad `selectedIndex` indica el número que ocupa en el array el primer elemento `option` seleccionado.
4º	Código	`valor = opciones[num].value;` `texto = opciones[num].text;`
	Explicación	Buscamos el elemento seleccionado en el array y después

		tenemos dos opciones: ▪ Propiedad `value`: obtenemos el valor del atributo del elemento seleccionado. ▪ Propiedad `text`: obtenemos el texto mostrado pantalla del elemento seleccionado.

Podemos reducir todo el código del paso 2 al 4 de la siguiente manera:

```
valor = lista.options[lista.selectedIndex].value;
texto = lista.options[lista.selectedIndex].text;
```

Listas de selección múltiple

La propiedad `selectedIndex` no sirve para acceder a los elementos en una selección múltiple, ya que sólo nos devuelve el primero.

Utilizamos la propiedad **select** que se aplica al array `option`. Para buscar los botones seleccionados lo normal es crear un bucle que recorra todas las opciones y nos diga mediante una instrucción if que valores están seleccionados.

```
opcion0 = lista.option[0].selected
```

En este ejemplo se comprueba si el primer elemento de la lista está seleccionado. el resultado es `true` si está seleccionado o `false` si no lo está.

Ventanas

Algunos métodos del DOM permiten abrir nuevas ventanas o pestañas en el navegador. Estos dependen directamente del objeto `window`, por lo que se escriben directamente.

Ventanas de alerta

alert : Ventana que muestra un mensaje.

```
alert("muestra mensaje");
```

prompt : muestra un mensaje que espera respuesta por parte del usuario.

```
nombre = prompt("¿Cómo te llamas?");
```

confirm : muestra un mensaje cuya respuesta por parte del usuario consiste en aceptar o cancelar.

```
elige = confirm("Acepta o Cancela");
```

Historial de páginas

El método `history.go()` permite el acceso a las páginas visitadas de la misma manera que los botones adelante y atrás del navegador.

Para acceder a la página visitada anteriormente:

```
history.go(-1)
```

Para volver a la página que estábamos antes de retroceder:

```
history.go(1)
```

Ventanas emergentes

Para abrir otras páginas en ventanas emergentes utilizaremos el método `open()`, el cual tiene tres parámetros:

`open("URL","nombre","propiedades")`

`"URL"` : Ruta en la que se encuentra el archivo de la página que queremos abrir.

`"nombre"` : Un nombre para identificar la ventana.

`"propiedades"` : Son una serie de propiedades que indican cómo debe aparecer la ventana, Se escribirán dentro de las comillas separadas por comas. Estas son:

- **toolbar[=yes|no]:** Mostrar o no la barra de herramientas.
- **statusbar[=yes|no]:** Mostrar o no la barra de estado.
- **titlebar[=yes|no]:** Mostrar o no la barra de título.
- **menubar[=yes|no]:** Mostrar o no la barra de menús
- **scrollbars[=yes|no]:** Mostrar o no las barras de desplazamiento.
- **resizable[=yes|no]:** Establece si se puede redimensionar el tamañó de la ventana. Sólo funciona en Internet Explorer.
- **width=pixels:** Ancho de la ventana en pixels
- **height=pixels:** Alto de la ventana en pixels
- **top=pixels:** Distancia en pixels desde el borde superior de la pantalla al borde superior de la ventana.
- **left=pixels:** Distancia en pixels desde el borde izquierdo de la pantalla al borde izquierdo de la ventana.

Veamos un ejemplo:

```
open("http://www.google.es" , "Google" ,
"toolbar=yes,statusbar=yes,width=600,height=400");
```

NOTA: *el código anterior debemos escribirlo en una sola línea.*

Otros métodos y propiedades del DOM

Otras propiedades de la página que pueden sernos útiles son:

- **document.bgColor:** acceso o modificación del color de fondo de la página. Éste puede anotarse en notación de nombres o en hexadecimal.
- **document.fgColor:** acceso o modificación del color del texto de la página. Éste puede anotarse en notación de nombres o en hexadecimal.
- **location:** accede a la URL o dirección de la página. Como depende directamente del objeto window, se escribe simplemente así.
- **document.title:** Acceso o modificación del título de la página (el que aparece en la pestaña del navegador).
- **document.lastModified:** Acceso a la fecha de la última modificación de la página.
- **navigator.userAgent:** Acceso a la información sobre el navegador y el sistema operativo que se está usando.
- **screen.width:** Acceso al ancho de resolución de pantalla que se está usando.
- **screen.height:** Acceso al alto de resolución de pantalla que se está usando.

El método `eval()`

El método `eval` convierte el contenido de una cadena de texto en código javascript.

```
texto = "alert('hola mundo')";
eval(texto);
```

En el ejemplo el texto de la cadena `texto` se convierte en código javascript y se ejecuta.

La propiedad Location Esta propiedad accede a la dirección de la página, además posee unas propiedades que permiten obtener más información sobre la página. Estas propiedades sonlas siguientes y nos dan la siguiente información:

- **location.hash** : nombre del enlace, dentro de la URL.
- **location.host** : nombre del servidor y el número del puerto, dentro de la URL.
- **location.hostname** : nombre de dominio del servidor (o la dirección IP), dentro de la URL.
- **location.href** : Cadena que contiene la URL completa.
- **location.pathname** : Cadena que contiene el camino al recurso, dentro de la URL.
- **location.port** : Cadena que contiene el número de puerto del servidor, dentro de la URL.
- **location.protocol** : Cadena que contiene el protocolo utilizado (incluyendo los dos puntos), dentro de la URL.
- **location.search** : Cadena que contiene la información pasada en una llamada a un script, dentro de la URL.

Además la propiedad `location` posee los métodos **reload()** y**replace()**

El método **reload()** recarga la página. Puede también cargar una nueva página si antes se especifica con la propiedad `href`:

```
location.href = "http://www.yahoo.es":
cambio = location.href;
cambio.reload();
```

El método **replace()** carga una nueva página cuya ruta se especificará en el parámetro.

```
location.replace("http://www.yahoo.es")
```