

getElementsByName()

La función `getElementsByName(nombreEtiqueta)` obtiene todos los elementos del documento HTML cuya etiqueta sea igual que el parámetro que se le pasa a la función:

```
var parrafos = document.getElementsByName("p");
```

El valor que se indica delante del nombre de la función es el nodo a partir del cual se realiza la búsqueda de los elementos. En el ejemplo anterior es `document` pero si se indica otro elemento de la página (`div`, `p`, `ul`, ...), la función devolverá todos los elementos contenidos en él.

```
var cabecera = document.getElementById("cabecera");
var parrafosCabecera = cabecera.getElementsByName("p");
```

El valor que devuelve la función es un array con todos los nodos que cumplen la condición de que su etiqueta coincide con el parámetro proporcionado. Es un array de nodos DOM, no un array de cadenas de texto. De este modo, se puede obtener el primer párrafo de la página o la cantidad de párrafos encontrados, de la siguiente manera:

```
var primerParrafo = parrafos[0];
var cantidadParrafos = parrafos.length;
```

CREACIÓN Y ELIMINACIÓN DE NODOS

- Creación de un Elemento, hay que realizar 4 operaciones

```
var parrafo = document.createElement("p"); //Crear nodo de tipo Element
var contenido = document.createTextNode("Hola Mundo!"); //Crear nodo con el texto
parrafo.appendChild(contenido); //Añadir el nodo Text al nodo Element
document.body.appendChild(parrafo); //Añadir el nodo Element a body u otro elemento
```

- Eliminar un elemento, la función `removeChild()` requiere como parámetro el nodo que se va a eliminar y debe ser invocada desde el elemento padre de ese nodo que se quiere eliminar (`nodoHijo.parentNode`)

```
var parrafo = document.getElementById("nombreparrafo");
parrafo.parentNode.removeChild(parrafo);
```

Cuando se elimina un nodo, también se eliminan automáticamente todos los nodos hijos que tenga, por lo que no es necesario borrar manualmente cada nodo hijo.

- Creación de un nodo atributo, para añadir un atributo a un elemento hay que realizar 3 operaciones

```
var miatributo = document.createAttribute("onclick"); //Crear nodo de tipo atributo
miatributo.value="comprobar()"; //Asignar el valor al atributo
boton.setAttributeNode(miatributo); //Asignar el atributo al elemento
```

Deshabilitar y ocultar un elemento

Para deshabilitar un elemento cuando se produzca algún evento, podemos usar la propiedad

```
elemento.disabled=true
```

Para ocultar un elemento cuando se produzca algún evento, podemos usar la propiedad

```
elemento.style.display=none;
```

Evento onkeypress y onkeyup

Con cada pulsación de tecla al escribir sobre una caja de texto o textarea se producen los siguientes eventos:

`onkeyup` se produce justo después de que el carácter haya sido escrito en una caja de texto o textarea. Es útil para controlar lo que se está escribiendo o para actualizar información en otro elemento.

`onkeypress` se produce cuando se ha pulsado la tecla, pero todavía no ha sido escrita. Podemos saber que tecla se ha pulsado gracias a la propiedad `keyCode` del evento generado y anular el efecto de la tecla pulsada. Anular el efecto de la tecla enter si en el evento `onkeypress` ejecutamos:

```
if (event.keyCode==13) {  
    event.returnValue = false; //esto anula la tecla pulsada  
}
```

El keycode correspondiente a algunas teclas es (se pueden consultar todas en internet)

backspace	8
tab	9
enter	13
shift	16
ctrl	17
alt	18
caps lock	20
escape	27
page up	33
Space	32
page down	34
end	35
home	36
arrow left	37
arrow up	38
arrow right	39
arrow down	40

Ejecutar una función cada cierto tiempo

Para llamar una función cada 1000 milisegundos deberíamos ejecutar la siguiente instrucción

```
var variable=setInterval(muestraReloj, 1000);
```

Y para que no vuelva a llamarse más habría que ejecutar

```
clearInterval(variable);
```

Ejecutar una función pasado un cierto tiempo

Muestra un cuadro de alerta después de 3 segundos (3000 milisegundos):

```
setTimeout(function(){ alert("Hello"); }, 3000);
```

Para llamar a una función no hay que poner los paréntesis como hacemos en los eventos, para llamar a una función llamada comprobar() a los 3 segundos:

```
setTimeout(comprobar, 3000);
```