

4. OBJETOS de FORMULARIOS

Desarrollo Web en entorno cliente

IES Clara del Rey

El objeto **form** depende en la jerarquía de objetos del objeto *document*.

De un formulario dependen todos los elementos que hay dentro: textos, selección, checkbox, etc

Acceso al formulario

- A partir de su nombre, asignado con el atributo **name** de HTML
- Mediante la matriz de formularios del objeto *document*, con su índice

Acceso a los objetos/elementos del formulario

- A partir del nombre del objeto asignado con el atributo **NAME** de HTML
- Mediante la matriz de elementos del objeto *form*, con el índice del elemento

Ejemplos de acceso a los elementos de formulario

```
<FORM name="f1">  
<INPUT type=text name=campo1>  
<INPUT type=text name=campo2>  
</FORM>
```

```
<script>  
// El formulario  
document.f1  
document.forms[0]  
  
// Los elementos  
document.f1.campo1  
document.f1.campo2  
document.f1.elements[1]  
document.forms[0].campo2  
document.forms[0].elements[1]  
</script>
```

El objeto form

Propiedades

- action
- elements array
- encoding
- length
- method
- name
- target

Métodos

- submit()
- reset()

<INPUT type="text">

Propiedades

- defaultValue: lo que contiene el atributo value
- form
- name
- type
- value: lo que hay escrito

Métodos

- blur()
- focus()
- select()

<INPUT type=hidden>

<INPUT type=password>

Funcionan del mismo modo, sólo que en *password* salen asteriscos.

Son como campos de texto que están ocultos para el usuario. Muy útiles en el desarrollo de webs para pasar variables en los formularios a las que no debe tener acceso el usuario.

Propiedades: value, form, name, type

<TEXTAREA>

Permite escribir texto, igual que los campos text, sólo que podemos escribir varias líneas a la vez.

Los atributos *cols* y *rows* sirven para indicar la anchura y altura del campo textarea en caracteres. *cols* indica el número de columnas y *rows* el de filas. No se puede acceder a ellos con Javascript.

Propiedades

- defaultValue
- value

Métodos

- blur()
- focus()
- select()

`<INPUT type="checkbox">`

Cajas que permiten marcarlas o no para verificar alguna cosa en un formulario.

Atributos:

- **name:** nombre para poder acceder desde código
- **checked:** campo chequeado por defecto. Booleano. Si no está aparece desmarcado.

Propiedades

- checked
- defaultChecked
- value
- form, name, type

Métodos

- `click()`
- `blur()`
- `focus()`

Checkbox: ejemplo de uso

Mediante nombres

```
<form name="miFormulario" action="/next.php"
  enctype="text/plain">
  <input type="checkbox" name="miCheck">
  <input type="button" value="Click" onclick="metodoClick()">
</form>

<script>
function metodoClick(){
  alert(document.miFormulario.miCheck.checked);
  alert(document.miFormulario.miCheck.value);
  document.miFormulario.miCheck.click();
}
</script>
```

Checkbox: ejemplo de uso

Mediante el id

```
<input type="checkbox" id="miCheckbox">
<div id="msg"></div>

<script>
  var miCheckbox = document.getElementById('miCheckbox');
  var msg = document.getElementById('msg');

  alert(`Valor del checkbox: ${miCheckbox.checked}`);
  miCheckbox.addEventListener('click', function() {
    if(miCheckbox.checked) {
      msg.innerText = 'El elemento está marcado';
    } else {
      msg.innerText = 'Ahora está desmarcado';
    }
  });
</script>
```

`<input type=radio>`

Permite seleccionar una opción y sólo una, sobre un conjunto de posibilidades.

Atributos:

- **name:** nombre que permite agrupar los radio button. Al tener el mismo name se consigue que sólo se pueda elegir una opción entre varias, dentro de ese grupo de botones de radio.
- **checked:** cuál de los radio buttons tiene que estar seleccionado por defecto
- **value:** el valor de cada uno de los radio buttons

Radio Buttons

Propiedades

- checked
- defaultChecked
- value
- **length**
- form, name, type

Métodos

- click()
- blur()
- focus()

Radio Buttons: métodos de acceso

Método **querySelector**: permite acceder a un objeto del DOM a través de un selector de los que usamos en CSS.

```
<input type="radio" name="status" value="si" id="yes">
<label for="yes">SI</label>
<br>
<input type="radio" name="status" value="ns-nc" id="ns">
<label for="ns">NS/NC</label>
<br>
<input type="radio" name="status" value="no" id="no">
<label for="no">No</label>
```

valorActivo =

document.querySelector('input[name="status"]:checked').value;

Problema: Si un elemento no está activo, value es **null**. Habría que comprobar.

Método **querySelectorAll()**: hay que recorrer el conjunto de los botones de radio para encontrar el que nos interesa y entonces chequearlo.

Función de ejemplo de uso

```
function setRadio(name, value) {  
    document.querySelectorAll(  
        `input[name="${name}"]`).forEach(element => {  
        if(element.value === value) {  
            element.checked = true;  
        }  
    });  
}
```

Radio Buttons: Uso tradicional basado en form

```
<form name=fcolores>
<input type="Radio" name="colorin" value="ffffff" checked>
<input type="Radio" name="colorin" value="ff0000"> Rojo
...
<input type="Button" value="Cambia Color" onclick="cambia()"
</form>
<script>
function cambia() {
    var i
    for (i = 0; i < document.fcolores.colorin.length; i++) {
        if (document.fcolores.colorin[i].checked) {
            break;
        }
    }
    document.bgColor = document.fcolores.colorin[i].value
}
</script>
```


<SELECT> <OPTION>

Listas desplegables que nos permiten seleccionar un elemento.

A la etiqueta Select se le añade el atributo NAME para luego acceder a ese campo mediante Javascript. Para cada opción del campo select se utiliza la etiqueta , con el atributo VALUE para expresar su valor. El texto tras la etiqueta es el texto que ve el usuario asociado a esa opción.

Propiedades de Select

- **length**: cantidad de etiquetas *option*
- **Option**: objeto que referencia cada una de las opciones
- **options**: array de las opciones
- **selectedIndex**: índice de la opción seleccionada

Métodos de select

- blur()
- focus()

Objeto Option

Propiedades:

- defaultSelected: indica si es la opción por defecto (se coloca *selected* al *option*)
- index
- selected
- text
- value (atributo *value* de *option*)

Select: Ejemplo de acceso

```
<select name="miSelect">
<option value="10">Muy bien
<option value="5" selected>Regular
<option value="0">Muy mal
</select>
<input type=button value="Dime propiedades"
      onclick="dimePropiedades()">

function dimePropiedades(){
    let cantidad = document.formul.miSelect.length;
    let indice = document.formul.miSelect.selectedIndex;
    let valor =
document.formul.miSelect.options[indice].value;
    let textoOpcion =
document.formul.miSelect.options[indice].text;
    alert(`${cantidad},${indice},${valor},${textoOpcion}`)
}
```

Crea tu formulario

Datos de Instructor

RUN	<input type="text"/>	Nombre Completo	<input type="text"/>
Fecha Nacimiento	<input type="text" value="dd - mm - aaaa"/>	Email	<input type="text" value="@"/>
Teléfono	<input type="text" value="12"/>	Dirección	<input type="text"/>
Ciudad	<input type="text" value="Seleccione una opción ..."/>	Comuna	<input type="text" value="Seleccione una opción ..."/>
Auto	<input type="text" value="XXXX-15"/>		

Añadir

Cancelar