

JavaScript

Programa del curso

- ◆ Clase 1: Introducción y sintaxis
- ◆ Clase 2: Funciones, arrays y objetos
- ◆ Clase 3: Javascript integrado a HTML
- ◆ Clase 4: DOM, selectores y elementos
- ◆ Clase 5: Eventos
- ◆ **Clase 6: Formularios**
- ◆ Clase 7: Ajax
- ◆ Clase 8: Ejercicio integrador

1. Formularios

Formularios

Una de las principales razones por las que se inventó el lenguaje de programación JavaScript fue la necesidad de validar los datos de los formularios directamente en el navegador del usuario.

Por medio de JavaScript podemos manejar los valores y eventos del formulario como también los de sus elementos.

Podemos seleccionar un formulario utilizando los métodos `getElementById()`, `getElementsByTagName()` o `querySelector()`.

```
var form = document.getElementById(idFormulario);
```


Formularios – document.forms

La propiedad **forms** del objeto document devuelve una colección de objetos tipo **array** conteniendo todos los formularios que existan en el documento HTML, con índice 0 para el primer formulario por orden de aparición, índice 1 para el siguiente y así sucesivamente.

```
var formulario = document.forms[0];
```

Formularios – document.forms.elements

La propiedad **elements** de cada objeto **form** obtenido mediante **document.forms** devuelve una colección de objetos tipo array conteniendo todos los campos de formulario que existan en el formulario concreto al que hacemos alusión, con índice 0 para el primer elemento por orden de aparición, índice 1 para el siguiente y así sucesivamente.

```
// retorna todos los elementos del form
```

```
var elementos = form.elements;
```

```
// retorna el elemento del formulario que está en el índice indicado
```

```
var elemento = form.elements[indice];
```

Formularios – Atributos

El objeto formulario tiene atributos como **action**, **target**, **encoding** y **method**. Al ser un objeto de Javascript podemos acceder a todos ellos de la misma forma que lo hacemos con cualquier otro objeto.

Action: Establece la URL del documento que va a procesar la información enviada por el formulario. **Ejemplo:** `form.action;`

Target: Establece el nombre o palabra reservada de donde se debe mostrar la respuesta luego de enviar el formulario. **Ejemplo:** `form.target;`

Formularios – Atributos

Encoding: Establece el tipo MIME con el que se van a encriptar los datos. Ejemplo: `form.encoding;`

Method: Establece cual es el método de HTTP que se va a utilizar para enviar los datos. Puede ser get o post. Ejemplo: `form.method;`

Name: Establece el nombre del formulario. Ejemplo: `form.name;`

Formularios – Elementos – Value

Los elementos del formulario tienen un atributo llamado value que nos permite establecer u obtener el valor de un elemento.

```
// obtenemos el valor del elemento  
elemento.value;
```

```
// establecemos el nuevo valor del elemento.  
elemento.value = nuevoValor;
```

Formularios – Eventos

Por medio del **evento submit del formulario** podemos **mandar los datos** que tiene cargado el formulario al documento que tiene establecido en el atributo del formulario **action**.

```
form.onsubmit = function() {  
    // se envió el formulario  
}
```

Formularios – Eventos

Con JavaScript podemos manejar el evento **onsubmit()** o **submit**. Al igual que con el resto de los eventos, podemos controlar que **no se ejecute el evento por default** con el método del evento **preventDefault()**

```
form.onsubmit = function(evento) {  
  evento.preventDefault();  
}
```

Formularios – Elementos – Input

Los elementos **input type=text** tienen una **propiedad** llamada **value** que nos permite **obtener o establecer el valor** que tiene un campo de texto.

```
// retorna el valor del elemento  
Elemento.value;
```

```
// establece un nuevo valor  
elemento.value = valor;
```


Formularios – Elementos – Select

Por medio de los **elementos select** podemos **obtener el valor elegido**. El objeto select tiene el **atributo selectedIndex** que nos retorna el **índice numérico** de la **opción seleccionada**.

Otro de los atributos que tiene el objeto select es **options** que nos retorna la **colección de elementos options**.

Formularios – Elementos – Select / options

Combinando estos dos atributos podemos obtener el valor del option seleccionado en el elemento select.

Al igual que el resto de los elementos HTML del formulario, el objeto option tiene un atributo value que nos da el valor del mismo.

Formularios – Elementos – Select / options

`select.selectedIndex;` // retorna el índice del valor seleccionado

`select.options;` // retorna la colección de elementos options

`Select.options[indice];` // retorna el option seleccionado

`select.options[indice].value;` // retorna el valor del elemento seleccionado.

Formularios – Elementos – Radio

El objeto formulario también tiene objetos radio button (type=radio). Por lo general este elemento se utiliza para **seleccionar entre varios valores** donde **sólo uno puede estar seleccionado**. Para lograr esto se utiliza el mismo atributo name.

Por medio del selector querySelector() podemos **obtener el elemento radio button que está activo**. Para esto debemos utilizar un selector de **css :checked**.

```
var radioButton = document.querySelector(selector :checked);  
radioButton.value; // obtengo el valor del elemento
```


Formularios – Elementos – Checkbox

Otro de los elementos del formulario es el elemento **checkbox**. Este objeto tiene un **atributo checked** que retorna un **valor booleano** (true/false) por el cual **podemos saber si fue seleccionado**.

El atributo checked se puede utilizar tanto para obtener su valor como también para establecer un nuevo valor.

```
checkbox.checked; // retorna el valor del objeto checked
```

```
checkbox.checked = true/false; // establece un nuevo valor al elemento.
```

Formularios – Elementos – Hidden

Otro de los elementos del formulario es el **input type=hidden**. Este elemento nos permite tener un **valor oculto** dentro del formulario. El usuario no va a ver dicho elemento pero al ser un elemento del formulario y estar dentro del mismo será enviado con el resto de los datos.

Al igual que el resto de los elementos del formulario el input type=hidden tiene un atributo **value** que nos **retorna** o nos permite **establecer** el valor del campo oculto.

```
elementoOculto.value;
```

```
elementoOculto.value = valorOcultoComoString;
```

Formularios – Elementos – Textarea

Otro de los elementos del formulario es el elemento **textarea**. Este elemento nos permite **escribir múltiples líneas**.

Con Javascript podemos **obtener** y **establecer** el **valor** de un objeto textarea por medio del atributo **value**.

```
textarea.value; // obtenemos el valor del textarea
```

```
textarea.value = nuevoValor; // establecemos el nuevo valor del  
textarea
```


Formularios – Elementos – Eventos

Los elementos del formulario pueden manejar eventos por medio de los métodos: **onfocus**, **onblur**, **onchange**, **oninput**. También se pueden escribir utilizando el método **addEventListener(callback)**.

Focus: Este evento se dispara al establecer el foco en un elemento

Blur: Este evento se dispara al remover el foco sobre un elemento

Change: Este evento se dispara cuando cambia el valor de un elemento

Input: Este evento se dispara al ingresar datos a un elemento

Formularios – Elementos – Eventos

```
elemento.onfocus = function() {  
  // código que maneja el focus del elemento  
}
```

```
elemento.onblur = function() {  
  // código que maneja el blur del elemento  
}
```

```
elemento.onchange = function() {  
  // código que maneja el cambio de valor  
}
```

```
elemento.oninput = function() {  
  // código que maneja el ingreso de datos a un elemento  
}
```

Formularios – Elementos – Eventos

Por medio del evento **change()** podemos manejar el **cambio de selección de un elemento select**.

```
elementoSelect.onchange = function() {  
  
var index = elementoSelect.selectedIndex;  
  
var option = elementoSelect.options[index].value;  
  
}
```

Formularios – Validación

Por medio del atributo **value** podemos obtener el valor de un elemento.

Al mismo tiempo podemos utilizar el atributo **length** del objeto string para saber su **longitud**. De esta forma podemos **validar si el elemento está vacío**.

```
if (elemento.length === 0) {  
    // elemento vacio  
}
```

Formularios - Regex

Javascript tiene expresiones regulares como tipo de dato y nos permite encontrar patrones de texto en un string.

Por lo general se pueden utilizar para validar los campos de textos.

Las expresiones regulares o regex tienen un método `test()` que nos permiten validar un string para saber si cumple con la expresión regular.

Formularios - Regex

```
// valida que el string tenga números y letras
```

```
var regexCampoVacio = /^[a-z0-9]+$/i;
```

```
// valida una estructura de mail. Hay muchas opciones para hacer esto
```

```
var regexMail = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}$/;
```

```
// retorna un valor boolean dependiendo si cumple o no con la expresión regular.
```

```
regexCampoVacio.test(textoParaBuscar);
```

```
regexMail.test(textoParaBuscar);
```



A practicar!

Práctica 6 - Formularios



Gracias!

Preguntas?
