



Nuestro compromiso es con el *futuro*.

Introducción al Front End

Clase 3

Formularios

Los **formularios** son un método muy común de interacción con una página web. Un **formulario** es el principal componente de la página de inicio de Google, y la mayoría de nosotros utilizamos **formularios** todos los días para acceder a nuestros sitios favoritos. En esta clase veremos cómo se pueden utilizar los formularios para interactuar con un programa **JavaScript**.

Los **formularios** se componen de un elemento **<form>** (etiqueta HTML) que contiene controles de **formulario** como campos de entrada, menús de selección y botones. Estos campos de entrada pueden rellenarse con

información que se procesa una vez que el formulario ha sido enviado.

Tradicionalmente, cuando se enviaba un **formulario**, se enviaba a un **servidor** en el que la información se procesaba mediante un sistema de gestión. La información se procesaba utilizando un lenguaje de fondo como PHP o Ruby.

Es posible, y cada vez más común, procesar la información en un **formulario** en el **front-end** antes de que se envíe al servidor usando JavaScript, que es en lo que nos centraremos en esta clase.

Formularios

Cada control de **formulario** tiene un valor inicial que puede ser especificado en el código **HTML**.

Este valor puede ser cambiado por un usuario que introduzca información o interactúe con la interfaz del interfaz del **formulario** (como el uso de un deslizador para aumentar o disminuir un valor). El valor también puede cambiarse dinámicamente mediante **JavaScript**.

Cuando se trata de formularios, hay muchas consideraciones de **usabilidad** y **accesibilidad** que hay que tener en cuenta, como usar un marcado correcto y **semántico** hacer que los **formularios** sean accesibles con el teclado y utilizar etiquetas WAI-ARIA.

Pueden investigar mas sobre **accesibilidad** en <https://www.w3.org/WAI/standards-guidelines/aria/> y recomendaciones sobre experiencia de usuario (**UX**) en formularios en este link: (pero hay muchísimos más en la web 😊)

Formularios

Un **formulario** puede ser enviado manualmente por el usuario empleando un botón o elemento de entrada con un atributo type de submit, o incluso un elemento input con un atributo type de imagen:

```
<button type='submit'>Submit</button>
<input type='submit' value='Submit'>
<input type='image' src='button.png'>
```

... y sus valores iniciales

especificados en el HTML. Un botón con un atributo **type of reset** también puede ser utilizado para hacer esto sin la

necesidad de scripts adicionales:

```
<button type='reset'>Reset</button>
```



Tip: Los “reset buttons” no suelen brindar buena usabilidad ya que pueden ser accionados sin querer y borrar toda la data ingresada en el formulario, por lo cual se recomienda ser cuidadosos con su uso.

Eventos en formularios

Los **formularios** desencadenan una serie de eventos como los que ya hemos estudiado, pero algunos son exclusivos de los formularios.

El evento **focus**, por ejemplo, cuando se enfoca un elemento. Por ejemplo en el caso de un `<input>` esto sucede cuando el cursor se sitúa en el interior del elemento, ya sea clickeando, tocando o navegando a través del teclado.

Podemos ver el siguiente ejemplo en el código

```
const input = form.elements.searchInput;

input.addEventListener('focus', () => alert('focused'),
  ↪ false);
```

Cuando abramos el formulario en el browser y coloquemos el cursor en el input veremos el cuadro con el alerta

Eventos en formularios

Envío de un formulario

Posiblemente el evento más importante del formulario es el evento **submit** que ocurre cuando el se envía el **formulario**. Normalmente esto enviará el contenido del formulario al servidor para procesar, pero podemos usar **JavaScript** para interceptar el formulario antes de que se envíe añadiendo un receptor de eventos de envío.

Probemos el siguiente código:

```
const form = document.forms['search'];
form.addEventListener ('submit', search, false);

function search() {
  alert(' Form Submitted');
}
```


Eventos en formularios

Volveremos a cargar la página y haremos clic en el botón Enviar. Debería ver un diálogo de alerta que dice "Formulario enviado". Después de hacer clic en Aceptar, el navegador intenta cargar una página inexistente (la URL debe terminar en algo similar a `.../search?searchInput=hello`). Esto se debe a que cuando el evento se disparó, nuestra función `search()` fue invocada, mostrando el diálogo de alerta. Luego el **formulario** se envió a la URL proporcionada en el atributo de **action** para su procesamiento, pero en este caso, la URL no es una URL real, por lo que no va a ninguna parte (por ahora ya que solo estamos cubriendo el front end).

Podemos evitar que el formulario se envíe a esa URL utilizando usando el método **`preventDefault()`**.

```
function search(event) {
  alert('Form Submitted');
  event.preventDefault();
}
```

Ahora recarguemos la web e intentemos enviar el formulario. Veremos el diálogo de alerta pero después de hacer clic en OK, el formulario no intenta enviarse a sí mismo a la URL falsa.

Recuperación de valores de un formulario

Los objetos de elementos de entrada de texto tienen una propiedad `value` que puede utilizarse para recuperar el texto dentro del campo.

Podemos usar esto para informar de lo que el usuario ha buscado en un campo de búsqueda, por ejemplo:

```
function search(event) {  
  alert(`You Searched for: ${input.value}`);  
  event.preventDefault();  
}
```

El `<select>` en el formulario

Seleccionar lista desplegable

Las listas desplegables de selección pueden utilizarse para seleccionar una o más opciones de una lista de valores. El atributo `multiple` es necesario si se debe seleccionar más de una opción.

Tenga en cuenta que el atributo **selected** puede utilizarse para establecer el valor inicial en el **HTML**.

En este ejemplo, la opción en blanco que proporciona el mensaje instructivo 'Elige tu ciudad' tiene este atributo, por lo que se muestra cuando se carga la página.

El atributo **name** del elemento `select` se utiliza para acceder a él en **JavaScript** como una propiedad del objeto formulario:

`form.ciudad;`

Si sólo se ha seleccionado un elemento, esto devolverá una referencia a esa selección, en caso contrario, se devolverá una colección que contiene cada selección.

Cada objeto de selección tiene una propiedad `value` que es igual al atributo **value** de la etiqueta `<option>` que fue seleccionada. **Veamos el ejemplo en el código**

Validaciones en formularios

Podemos realizar dos tipos de validaciones básicas para los formularios y poder enviarlos con la información completa.

Primero veremos la más básica pero que quizás no ofrece la mejor experiencia al usuario. Es la validación OnSubmit. Es decir una vez que se ingresaron los datos, al tratar de enviar el formulario, en ese momento se le informará al usuario si hay algún error o falta un campo por completar lo cual no es recomendable pero sin embargo suele usarse aún y nos servirá para poder practicar las validaciones básicas.

 ¡Vamos al código!

Validaciones en formularios

Podemos mejorar aún más la usabilidad del formulario dando una respuesta instantánea **onTime** en lugar de esperar a que se envíe el formulario. Esto se puede conseguir añadiendo el evento listener directamente al campo de entrada que se disparará cuando el usuario presione una tecla(utilizando el evento keyup). La retroalimentación puede entonces ser insertada dentro del elemento label del campo de entrada, junto con una clase de error para una respuesta más directa.

✌ ¡Vamos al código!

Validaciones en formularios

Otra técnica útil que puede ayudar a la **usabilidad** es deshabilitar el botón **Enviar** si hay errores en el **formulario**. Si el botón Enviar está **deshabilitado**, no se hará ninguna acción cuando se hace clic. La mayoría de los navegadores también lo mostrarán en un color más claro para indicar que no se puede hacer clic en él. Esto evita que los usuarios envíen un **formulario** que contenga algún error.

Un botón de envío se puede deshabilitar agregando el atributo **disabled** al **<input>**

👐 ¡Vamos al código!

Muchas gracias!



ICARO Asociación Civil
CUIT 30716564815
info@icaro.org.ar
www.icaro.org.ar