INSTITUTO KHIPU

Semestre IV

Sesión 18

PROGRAMA DE ESTUDIOS

DESARROLLOS DE SISTEMAS DE INFORMACIÓN

DESARROLLO FRONTEND DE SISTEMAS DE WEB

Tema

FORMULARIOS Y VALIDACIÓN CON JAVASCRIPT

Formularios y Validación con JavaScript

A continuación, se presenta un formulario básico sin validación y se demostrará qué sucede cuando el usuario ingresa datos incorrectos o deja los campos vacíos.

📌 **Ejemplo de formulario sin validación:**

```
html


<form>
    <label>Nombre:</label>
    <input                    type="text"                    id="nombre">
    <button                                type="submit">Enviar</button>
</form>
```

🔷 **Problema:** Si el usuario deja el campo vacío, el formulario se enviará de todos modos, lo que puede generar errores en la aplicación.

"¿Cómo podemos asegurarnos de que el usuario ingrese información válida antes de enviar un formulario?"

2. Presentación teórica

2.1 Estructura de un formulario en HTML

Un formulario en HTML se define con la etiqueta `<form>`, que contiene diversos elementos como:

- **`<input>`**: Campos de texto, contraseñas, correos electrónicos.
- **`<textarea>`**: Áreas de texto para comentarios largos.
- **`<select>` y `<option>`**: Listas desplegables.
- **`<button>` o `<input type="submit">`**: Botón de envío.

📌 **Ejemplo de formulario estructurado:**

```
html
<form                                                    id="registro">
    <label>Nombre:</label>
```

```
    <input type="text" id="nombre">

    <label>Email:</label>
    <input type="email" id="email">

    <label>Contraseña:</label>
    <input type="password" id="password">

    <button type="submit">Registrar</button>
</form>
```

2.2 Eventos clave en formularios

JavaScript permite capturar eventos en los formularios para validarlos antes del envío:

Evento	Descripción
`submit`	Se activa cuando el usuario envía el formulario.
`change`	Se dispara cuando el usuario cambia el valor de un campo.
`input`	Se activa mientras el usuario escribe en un campo.
`focus`	Se dispara cuando un campo recibe el foco.
`blur`	Se activa cuando un campo pierde el foco.

📌 **Ejemplo:**

```
javascript

document.getElementById("nombre").addEventListener("input", function()
{
    console.log("El usuario está escribiendo...");
});
```

2.3 Métodos para acceder a los elementos del formulario

Método	Descripción
`document.getElementById("id")`	Accede a un elemento por su `id`.

`document.querySelector("selector")`	Accede a un elemento usando selectores CSS.
`document.forms["nombreFormulario"]`	Accede a un formulario por su name.

📌 **Ejemplo:**

```
javascript
let inputNombre = document.getElementById("nombre");
let inputEmail = document.querySelector("#email");
let formulario = document.forms["registro"];
```

3. Validación básica con JavaScript

3.1 Verificación de campos vacíos

Una validación parte de verificar si un campo está vacío, si esto es así entonces se debe mostrar un mensaje de error.

📌 **Ejemplo:**

```
javascript


document.getElementById("registro").addEventListener("submit",
function(event) {
    let nombre = document.getElementById("nombre").value;

    if (nombre.trim() === "") {
        alert("El campo nombre es obligatorio.");
        event.preventDefault(); // Evita el envío del formulario
    }
});
```

3.2 Restricciones en la longitud de caracteres

En muchas ocasiones es necesario restringir la cantidad de caracteres para ello se puede limitar la cantidad de caracteres que el usuario puede ingresar.

📌 **Ejemplo:**

```
javascript
document.getElementById("password").addEventListener("blur",
function()                                                    {
    if          (this.value.length          <          6)          {
        alert("La  contraseña  debe  tener  al  menos  6  caracteres.");
    }
});
```

3.3 Validación de tipos de datos

Dato	Método de validación
Email	`inputEmail.includes("@")`
Número	`isNaN(valor) === false`
Contraseña	`length >= 6`

📌 **Ejemplo:** Validación de email y contraseña

```
javascript
document.getElementById("registro").addEventListener("submit",
function(event)                                               {
    let      email      =      document.getElementById("email").value;
    let    password    =    document.getElementById("password").value;

    if                    (!email.includes("@"))                    {
        alert("Ingrese          un          email          válido.");
        event.preventDefault();
    }

    if          (password.length          <          6)          {
        alert("La  contraseña  debe  tener  al  menos  6  caracteres.");
        event.preventDefault();
    }
});
```

4. Demostración práctica

4.1 Implementación de validaciones en un formulario

📌 **Formulario de registro con validación en tiempo real:**

html

```
<form id="registro">
    <label>Nombre:</label>
    <input type="text" id="nombre">
    <label>Email:</label>
    <input type="email" id="email">

    <label>Contraseña:</label>
    <input type="password" id="password">

    <button type="submit">Registrar</button>
</form>
```

📌 **Validación con JavaScript:**

Javascript

```
document.getElementById("registro").addEventListener("submit",
function(event) {
    let nombre = document.getElementById("nombre").value.trim();
    let email = document.getElementById("email").value.trim();
    let password = document.getElementById("password").value.trim();
    let valido = true;

    if (nombre === "") {
        alert("El nombre no puede estar vacío.");
        valido = false;
    }

    if (!email.includes("@")) {
```

```
        alert("Ingrese un email válido.");
        valido = false;
    }

    if (password.length < 6) {
        alert("La contraseña debe tener al menos 6 caracteres.");
        valido = false;
    }
    if (!valido) {
        event.preventDefault(); // Detiene el envío del formulario si
hay errores
    }
});
```

4.2 Uso de `preventDefault()` para evitar envíos incorrectos

El método `event.preventDefault()` impide que el formulario se envíe si la validación falla.

4.3 Uso de estilos CSS dinámicos

Podemos cambiar el color de los bordes si un campo es inválido.

📌 **Ejemplo:**

```
javascript

document.getElementById("nombre").addEventListener("blur",  function()
{
    if          (this.value.trim()          ===          "")          {
        this.style.border       =       "2px       solid       red";
    }                               else                                {
        this.style.border       =       "2px       solid       green";
    }
});
```

Este contenido permitirá comprender y aplicar validaciones en formularios utilizando JavaScript, mejorando la experiencia del usuario y asegurando la calidad de los datos enviados.

INSTITUTO

KHIPU