Evidencia de Aprendizaje 3 (EA3). Primeros pasos con JavaScript.

Ingeniería web I

Sergio Rivera Urrea

CC: 1.192.759.768

[sergio.rivera@est.iudigital.edu.co](mailto:sergio.rivera@est.iudigital.edu.co)

Docente:

**Antonio Jesús Valderrama**

Mayo de 2024

Jardín, Antioquia

VALIDACION CON JAVASCRIPT

- La función validateForm es un controlador de eventos que se ejecuta

cuando se envía un formulario.

- La línea event.preventDefault() detiene el comportamiento predeterminado

de envío del formulario, lo que evita que la página se recargue.

- Se obtienen las referencias a los campos de entrada del formulario

utilizando document.querySelector('input[name="name"]') y

document.querySelector('input[name="price"]').

- Los valores de los campos de entrada se almacenan en las variables name

y price, después de eliminar los espacios en blanco con trim().

- Se obtienen las referencias a los elementos de mensaje de error utilizando

document.getElementById('name-error') y

document.getElementById('price-error').

- Los mensajes de error anteriores se borran asignando una cadena vacía a

nameError.textContent y priceError.textContent.

- La variable isValid se inicializa como true, y se establecerá en false si se

encuentra algún error en la validación.

- Se realizan las validaciones: si el campo name está vacío, se asigna un

mensaje de error a nameError.textContent y isValid se establece en

false. Lo mismo se aplica para el campo price.

- Si isValid es true, significa que no se encontraron errores y se ejecuta el

bloque de código dentro del condicional.

- Se obtiene la referencia al elemento de mensaje de éxito utilizando

document.getElementById('success-message'), y se le asigna el

mensaje correspondiente.

- Se restablecen los valores de los campos de entrada asignando una

cadena vacía a nameInput.value y priceInput.value.

VALIDACION CON JAVASCRIPT

- La función validateForm es un controlador de eventos que se ejecuta

cuando se envía un formulario.

- La línea event.preventDefault() detiene el comportamiento predeterminado

de envío del formulario, lo que evita que la página se recargue.

- Se obtienen las referencias a los campos de entrada del formulario

utilizando document.querySelector('input[name="name"]') y

document.querySelector('input[name="price"]').

- Los valores de los campos de entrada se almacenan en las variables name

y price, después de eliminar los espacios en blanco con trim().

- Se obtienen las referencias a los elementos de mensaje de error utilizando

document.getElementById('name-error') y

document.getElementById('price-error').

- Los mensajes de error anteriores se borran asignando una cadena vacía a

nameError.textContent y priceError.textContent.

- La variable isValid se inicializa como true, y se establecerá en false si se

encuentra algún error en la validación.

- Se realizan las validaciones: si el campo name está vacío, se asigna un

mensaje de error a nameError.textContent y isValid se establece en

false. Lo mismo se aplica para el campo price.

- Si isValid es true, significa que no se encontraron errores y se ejecuta el

bloque de código dentro del condicional.

- Se obtiene la referencia al elemento de mensaje de éxito utilizando

document.getElementById('success-message'), y se le asigna el

mensaje correspondiente.

- Se restablecen los valores de los campos de entrada asignando una

cadena vacía a nameInput.value y priceInput.value.

VALIDACION CON JAVASCRIPT

- La función validateForm es un controlador de eventos que se ejecuta

cuando se envía un formulario.

- La línea event.preventDefault() detiene el comportamiento predeterminado

de envío del formulario, lo que evita que la página se recargue.

- Se obtienen las referencias a los campos de entrada del formulario

utilizando document.querySelector('input[name="name"]') y

document.querySelector('input[name="price"]').

- Los valores de los campos de entrada se almacenan en las variables name

y price, después de eliminar los espacios en blanco con trim().

- Se obtienen las referencias a los elementos de mensaje de error utilizando

document.getElementById('name-error') y

document.getElementById('price-error').

- Los mensajes de error anteriores se borran asignando una cadena vacía a

nameError.textContent y priceError.textContent.

- La variable isValid se inicializa como true, y se establecerá en false si se

encuentra algún error en la validación.

- Se realizan las validaciones: si el campo name está vacío, se asigna un

mensaje de error a nameError.textContent y isValid se establece en

false. Lo mismo se aplica para el campo price.

- Si isValid es true, significa que no se encontraron errores y se ejecuta el

bloque de código dentro del condicional.

- Se obtiene la referencia al elemento de mensaje de éxito utilizando

document.getElementById('success-message'), y se le asigna el

mensaje correspondiente.

- Se restablecen los valores de los campos de entrada asignando una

cadena vacía a nameInput.value y priceInput.value.

a función validateForm es un controlador de eventos que se ejecuta

cuando se envía un formulario.

- La línea event.preventDefault() detiene el comportamiento predeterminado

de envío del formulario, lo que evita que la página se recargue.

- Se obtienen las referencias a los campos de entrada del formulario

utilizando document.querySelector('input[name="name"]') y

document.querySelector('input[name="price"]').

- Los valores de los campos de entrada se almacenan en las variables name

y price, después de eliminar los espacios en blanco con trim().

- Se obtienen las referencias a los elementos de mensaje de error utilizando

document.getElementById('name-error') y

document.getElementById('price-error').

- Los mensajes de error anteriores se borran asignando una cadena vacía a

nameError.textContent y priceError.textContent.

- La variable isValid se inicializa como true, y se establecerá en false si se

encuentra algún error en la validación.

- Se realizan las validaciones: si el campo name está vacío, se asigna un

mensaje de error a nameError.textContent y isValid se establece en

false. Lo mismo se aplica para el campo price.

- Si isValid es true, significa que no se encontraron errores y se ejecuta el

bloque de código dentro del condicional.

- Se obtiene la referencia al elemento de mensaje de éxito utilizando

document.getElementById('success-message'), y se le asigna el

mensaje correspondiente.

- Se restablecen los valores de los campos de entrada asignando una

cadena vacía a nameInput.value y priceInput.value.

1. **Variables y selección de elementos del DOM**:
   * formulario: Obtiene la referencia al elemento del formulario con el id “formulario”.
   * inputs: Obtiene una lista de todos los elementos input dentro del formulario.
   * submit: Obtiene una lista de todos los elementos con la etiqueta “submit” (esto parece ser un error, ya que no se utiliza más adelante).
2. **Expresiones regulares**:
   * expresiones: Define tres expresiones regulares para validar los campos del formulario:
     + nombre: Permite letras mayúsculas, minúsculas, espacios y caracteres acentuados, con una longitud máxima de 40 caracteres.
     + inventario: Acepta cualquier cadena de hasta 3 caracteres.
     + precio: Acepta cualquier cadena de hasta 10 caracteres.
3. **Objeto campos**:
   * Crea un objeto llamado campos con tres propiedades (nombre, inventario y precio) inicializadas en false.
4. **Función validarFormulario**:
   * Esta función se ejecuta cuando se produce un evento keyup o blur en un campo de entrada.
   * Según el nombre del campo, llama a la función validarCampo con la expresión regular correspondiente.
5. **Función validarCampo**:
   * Valida el valor del campo de entrada según la expresión regular proporcionada.
   * Actualiza las clases CSS y los íconos en el DOM para indicar si el campo es válido o no.
   * Actualiza la propiedad correspondiente en el objeto campos.
6. **Evento submit del formulario**:
   * Evita que el formulario se envíe.
   * Si todos los campos son válidos (según las expresiones regulares), se reinicia el formulario y muestra un mensaje de éxito durante 5 segundos.
   * Si algún campo no es válido, muestra un mensaje de error.

**Conclusión**: Este código implementa la validación de formularios en una página web utilizando expresiones regulares. Verifica que los campos de nombre, inventario y precio cumplan con ciertos patrones antes de enviar el formulario. Si todos los campos son válidos, muestra un mensaje de éxito; de lo contrario, muestra un mensaje de error. Es importante asegurarse de que los nombres de los campos y las clases CSS coincidan con los elementos en el HTML para que funcione correctamente.