Nombre completo del estudiante

Juan Pablo Jiménez Muñoz

Matrícula

PREPROF2201PC-TDS0252

Fecha de elaboración

10/06/2022

Nombre del módulo

Unidad 3. JavaScript

Nombre de la evidencia de aprendizaje

S35 - Unidad 3. JavaScript - Evidencia de Aprendizaje (EA3) de la Actividad 3: Primeros pasos con JavaScript

Nombre del docente

Yaneth Mejía Rendon

**Introducción**

En este documento se realizará la explicación de todos los elementos creados en el archivo de JavaScript llamado index.js y su uso, también se explicará los cambios a algunos elementos del archivo newProduct.html más específicamente a su barra de navegación cuando se visualiza en ciertas pantallas y a su formulario.

**Desarrollo**

En el proyecto de the-store que hemos venido trabajando se creara un nuevo archivo llamado index.js, el cual nos sirve para realizar validaciones a nuestro formulario en el archivo newProduct.html en JavaScript.

En el index.js se crea una variable de tipo inmutable: const bar = document.querySelector(".fa-bars"); se especifica donde se implementará en el documento en este caso con el .querySelector le decimos al programa que busque la clase llamada fa-bars, el cual nos permite agregar una barra de navegación cuando la pantalla llega a un mínimo de pixeles, lo anterior realizado con CSS en el archivo style.css.

Se creará una nueva variable de tipo inmutable: const menu = document.querySelector(".menu"); en donde se espicifica que se implentara en la clase llamada .menu, el cual permite mostrar un menú desplegable en las barras de navegación antes creadas.

Se implementa una función para que el menú se muestre cuando se da click en la barra de navegación y se oculte cuando se oprime nuevamente: bar.addEventListener("click", () => {menu.classList.toggle("show-menu").

Para el formulario se realizan los siguientes cambios en el archivo newProduct: En la etiqueta <form> se le añaden los atributos de: id=”form”, name=”contactForm”, onsubmit="return validateForm()", action="newProduct.html" method="post"> </form>, los cuales nos permite seleccionarlo buscando por nombre e id, especificarle la función que va a retornar desde el archivo de index.js y la acción de recargar la página de new producto cuando se cumplen las condiciones, el método post para ocultar datos.

Dentro del formulario debajo de cada una de las etiquetas <input>, se crean etiquetas <div> con los atributos class=”error” con id diferentes para realizar leyendas que se muestran al usuario cuando no cumplen una condición, próximamente en el archivo JavaScript index.js

Para las validaciones del formulario del archivo newProduct.html en el index.js se crea una función llamada printError la cual permite mostrar al usuario las leyendas correspondientes, estas se especificarán más adelante: function printError(elemId, hintMsg) {document.getElementById(elemId).innerHTML = hintMsg; especifico el elemento por id donde se implementará la función el cual muestra un texto por pantalla.

Se crea una función llamada validateForm() para realizar las comprobaciones a los inputs en el archivo newProduct.html, dentro de esta función se crean variables para cada uno de los inputs dentro del formulario con los nombres de: names, price e inventory, en estas variables les especifico donde se encuentran localizadas en el proyecto y su valor:

var names = document.contactForm.names.value;

var price = document.contactForm.price.value;

var inventory = document.contactForm.inventory.value;

creo una nueva variable con los nombres que se les dio en las etiquetas <div> anteriormente en el archivo newProduct.html igualo las variables y les doy el valor de true.

var namesError = priceError = inventoryError = true;

se crean condicionales para validar si los campos del input name está vacío, si es verdadero se llama a la función printError y se le asigna donde se implementará y el valor que tendrá en este caso se muestra: \* El campo es requerido \*, si el campo no está vacío se llama a la función printError y en este caso no se le asignan valores para indicarle al usuario que el campo fue completado exitosamente y el valor a la etiqueta <div> del input name tendrá el valor de false. if (names == "") {printError("namesError", "\* El campo es requerido \*");} else { printError("namesError", "");namesError = false;}

Para la validación del campo del input price dentro del condicional se siguen los mismos pasos del anterior condicional realizan los respectivos cambios de asignación de valor y donde se implantará, en este caso se agrega un nuevo condicional: si también (else if) para indicar que, si el price es igual o menor a cero, indicarle al usuario que no puede agregar valores dentro de este rango con la función printError asignándole el valor de \* Ingresa valores mayores a cero \*.

if (price == "") {printError("priceError", "\* El campo es requerido \*");} else if (price <= 0){  printError("priceError","\* Ingresa valores mayores a 0 \*");} else {printError("priceError",""); priceError = false;}

Para la validación del campo del input inventory dentro del condicional se siguen los mismos pasos del condicional del price, solo que, en este caso, en el sí también (else if) para indicarla al usuario que solo puede asignar valores positivos a este campo llamando a la función printError y asignando el valor de \* ingresa valores positivos \*.

if (inventory == ""){printError("inventoryError","\* El campo es requerido \*"); }else if (inventory < 0) {printError("inventoryError","Ingresa valores positivos");} else {

printError("inventoryError","");inventoryError = false;}

Por ultimo se crea un nuevo condicional espicificando que si los valores de las etiquetas <div> del archivo newProduct son todos verdaderos (true) devuelva un falso (false), si no (else) se crea una variable llamada dataPreview para mostrar los atributos del producto que se ha creado. Por ultimo se crea una alerta que llama a dataPreview() para mostrar una alerta al usuario del producto agregado. Todos estas funciones, métodos y condicionales están dentro de la función validateForm().

 if ((namesError || priceError || inventoryError) == true){return false;} else {

 var dataPreview = "El producto con los siguientes datos: \n" +"Nombre: " + names +"\n" + "Precio: "+ price + "\n" + "Inventario: "+ inventory + "\n"¡Ha sido creado !";

 alert(dataPreview);}

**Conclusiones**

JavaScript es un lenguaje de programación que nos permite entre muchas otras características hacer validaciones a nuestras paginas web, en este lenguaje no hay una única forma de resolver un problema o realizar una acción, la forma como se programe dependerá del desarrollador del código y la manera mas optima que le parezca a este de resolver una problemática. Para mostrarle al usuario las validaciones de los campos requeridos en este este problema en particular hubo que analizar muy bien el problema y realizar una lógica de programación.