DigitalHouse > Coding School

Desarrollo Web Full Stack Node

Ejercitación - M07C17

Vinculación y manejo DOM

Práctica - D.O.M. + querySelector + style + innerHTML + classList

Rompe hielos

Vamos interactuar con el usuario.

Deberás partir de los archivos HTML, CSS e imágenes ya provistos.

Para empezar deberás crear un archivo de Javascript y vincularlo con el HTML.

1) Mediante un alert mostrar un saludo de bienvenida: "Bienvenidos a mi sitio".

2) Usaremos un confirm para preguntar si está seguro de querer avanzar.

2.1) En el caso que no quiera avanzar debe modificarse la etiqueta <h2> con el mensaje "Lamentamos que no quieras continuar tu visita por este maravilloso sitio". En este caso no debería seguir el flujo de la aplicación. Es decir, ya no deberían ejecutarse el punto 3, 4 y los que siguen...

2.2) Si quiere continuar mostrar en la etiqueta <h2> el mensaje: "Qué alegría que quieras con tu visita por este maravilloso sitio.".

3) Ahora con un prompt deberemos hacer que el usuario ingrese su nombre (tendremos que guardar el nombre en una variable).

4) Modificar la etiqueta <h1> para que diga “Bienvenido *nombre*” donde “*nombre*” debe reemplazarse por el nombre ingresado por el usuario.

5) A través de un prompt vamos a preguntarle la edad.

5.1) Si es mayor a 18 podrá continuar con el proceso.

5.2) Pero en el caso que no sea mayor de edad tendremos que ocultar la etiqueta div con clase “container-general” y mostrar la etiqueta div con clase “contenido-bloqueado”.

6) Vamos a usar un prompt donde pediremos al usuario ingresar hobbies pero cada uno de los hobbies que ingrese deben estar separados por una coma, por ejemplo: “Netflix,Programación,Fútbol,Hockey”.

7) Mediante el uso del método **split*** tendrás que transformar el string en un array (pueden usar la coma para separar los hobbies). A este array lo llamaremos “hobbies”.

> (*) The split() method is used to split a string into an array of substrings, and returns the new array. Para saber más puedes consultar los siguientes sitios web:
>
> https://www.w3schools.com/jsref/jsref_split.asp
>
> https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/String/split

8) Deberemos validar que entre los hobbies estén los strings "programación", "programacion" o "programar" o cada uno de los anteriores empezando con mayúscula.

8.1) En el caso que alguno de los anteriores exista mostraremos un alerta que diga: "Qué bueno que te guste la programación.".

8.2) En el caso que ninguno de los anteriores esté, les mostraremos un alerta que diga: "Qué lástima que no te guste la programación".

9) Anteriormente guardamos los hobbies del usuario. Te pedimos que generes un listado ordenado (<ol>) con los primeros 3 hobbies (en caso de haber) y lo insertes dentro de la etiqueta <article> de clase “hobbies”.

10) Si al usuario no le gusta la programación, reemplazar la imagen de clase “background-img” por una imagen de fondo que de pena (como la imagen gatito.jpeg que está en la carpeta img).

Pero para el caso que le guste reemplazar la imagen de clase background-img por la de un programador (como la imagen programmer.jpeg que está en la carpeta img).

Segunda tanda de ejercicios

11) Mediante un prompt deberán pedirle al usuario que ingrese su color favorito, guardarlo en la variable colorPreferido e imprimirlo en consola.

11.1) Con otro prompt, pedirle al usuario que ingrese su nombre y guardarlo en una variable llamada NOMBRE. Usando esta variable insertarlo en la etiqueta <h1> reemplazando la palabra “usuario”.

11.2) Luego deberán encerrar el nombre del usuario entre las etiquetas para poder modificar esta palabra por separado.

11.3) Ahora vamos a tomar el que acabamos de insertar y deberán agregarle la clase “color-preferido”.

12) Con el array hobbies:

12.1) Previo a insertar el <li> por cada hobby deberán preguntarle al usuario (mediante prompt) la URL del hobby. Por ejemplo, si el hobbie fuera Netflix, podrían relacionar https://www.netflix.com/, o si es fútbol podrían insertar la URL: https://www.ole.com.ar/). Finalmente el <li> debe contener un hipervínculo al lugar indicado. Por ejemplo una estructura final posible sería:

```
<ul>
  <li>  <a href=”http://netflix.com”>Netflix</a> </li>
  <li>  <a href=”http://ole.com”>Fútbol</a>   </li>
</ul>
```

12.2) Crear una función para validar que la longitud de caracteres de cada hobbie sea mayor a 5, pero a su vez que sea menor a 10 caracteres. Es decir que deben cumplir simultáneamente las condiciones. Los hobbies (<li>) que no cumplan esta validación, no deberán mostrarse.

12.3) Deberán centrar los <li>.

12.4) Los <li> que contengan un <a>:

12.4.1) Tendrán que modificar el estilo de la etiqueta para que no tengan el subrayado (text-decoration:none).

12.4.2) También deberán modificar el estilo de esta etiqueta para cambiar el color de fuente.

13) Deberás capturar la etiqueta <img> de clase avatar para cambiar la imagen que se muestra, en su lugar inserta una imagen tuya. Para obtener esa imagen debes consultarle al usuario (mediante prompt) la ruta de la imagen. Para esto no es necesario descargar la imagen, puedes googlear una imagen tuya y enlazarla a la URL donde se encuentra.

¡Y más ejercicios!

A continuación vamos a pedirle al usuario datos los datos de su película favorita para poder mostrarla:

Primero vamos a crear un objeto literal llamado película con las siguientes propiedades:

- nombre (String)
- director (String)
- duracion (Number)
- actor (String)

Luego vamos a pedirle al usuario que ingrese la información. Para esto vamos a usar prompts (uno para cada dato) y guardarlos como propiedades de nuestro objeto peliícula en las propiedades que corresponda.

Para mostrar los datos ingresados vamos utilizar la lista desordenada (<ul>) oculta en el archivo HTML (dentro del).

Ahora cambiemos la lista desordenada por una ordenada (<ol>)

Recuerda desocultar la lista una vez que el usuario haya cargado los datos. Y darle estilo, como por ejemplo centrar los <li>.

Por último agregar al ítem del nombre de la película un link que lleve al título de la misma en la página de IMDB (pedir la URL mediante prompt).

Por ejemplo, si tu pelicula es John Wick: https://www.imdb.com/title/tt2911666/

D.O.M

Este ejercicio vamos a realizarlo dentro del entorno de Node.js.

1. Crear un archivo gastos.ejs y vincularlo con archivo gastos.js. Importante: tener precaución de generar todo nuestro código JS dentro de window.onload.

2. Vamos a generar un reporte de gastos diarios de una familia. Dicho reporte funciona de la siguiente manera: Pide la cantidad de integrantes de la familia, para cada uno de ellos pide nombre y luego la cifra que ésta persona gastó. Al final veremos cada nombre con su correspondiente cifra, la persona que más gastó y la persona que menos gastó. Y el total de gastos de toda la familia.

a. Lo primero que haremos es preguntarle al visitante recién ingresa a la página si quiere iniciar el proceso. Si la respuesta es negativa deberá mostrarse una alerta que diga "Gracias por haber venido" y luego redireccionarlo al sitio web de Netflix.

b. Si la respuesta es positiva vamos a iniciar nuestro proceso. Primero pediremos al visitante que indique la cantidad de integrantes de su familia. Validaremos que el dato ingresado sea un número y que no sea inferior a 3. De NO ser un número deberá alertar que el valor necesario es un número y volverá a preguntar por la cantidad de integrantes. Para este proceso, puede ser de utilidad la siguientes función nativa de JS isNaN(). Puedes buscar qué hace la misma.

c. Una vez con la cantidad de integrantes listos, vamos a pedir para cada uno: nombre y gastos del día. Tener en cuenta de validar que:

- i. El nombre no puede estar vacío.
- ii. Los gastos no puede ser un texto ni estar vacío.

En cualquiera de esos casos alertar del error y volver a pedir el dato. Al final se deberá generar un Array de Objetos Literales cada uno con la propiedad nombre y valor.

Ejemplo:

```
integrantes = [
    {nombre: "Ada", gastos: 300},
    {nombre: "Tim", gastos: 570},
    {nombre: "Vincent", gastos: 80},
    ];
```

d. Mostraremos en consola dicho array.