- Curso Introducción a Javascript
- Profesora: Delgado Enrique

Actividades Clase Numero 17:

¡Hola, chicas! 👩‍💻✨

¡Bienvenidas a la clase 17!
Hoy vamos a explorar temas súper interesantes que te ayudarán a escribir código más limpio y eficiente en JavaScript. 🌟

Temas de hoy:

🔹 **Seguimos Practicando Métodos avanzados en Arrays:**
Exploraremos métodos esenciales que nos permiten trabajar con datos de forma más dinámica:

- **map()**: Ideal para transformar los elementos de un array en uno nuevo.
- **filter()**: Perfecto para seleccionar elementos que cumplen con una condición.
- **find()**: Para localizar el primer elemento que satisface un criterio específico.
- **reduce()**: Una poderosa herramienta para resumir o acumular datos en un único valor.
- **forEach()**: Útil para recorrer elementos sin modificar el array.

¡Prepárate para practicar, experimentar y disfrutar mientras aprendemos juntas! 💻✨

Actividades:

- **Ejercicio 1: Manipulación Completa de un Objeto Literal y Arrays**
  Crea un objeto literal que represente un estudiante. El mismo debe tener las siguientes propiedades: nombre, edad y un array de notas (con 5 notas).
  Luego, escribe la función “procesarEstudiante” que tendrá como objetivo realizar las siguientes tareas:
  1. Agregar una nueva nota al array de notas.
  2. Eliminar la primera nota del array.
  3. Calcular el promedio de las notas restantes.
  4. Convertir el nombre del estudiante a mayúsculas. Para esto investiga sobre el método “*toUpperCase()*” https://www.w3schools.com/jsref/jsref_touppercase.asp
  5. Devuelver un objeto con el nombre en mayúsculas y el promedio de las notas.

- **Ejercicio 2: Análisis y Modificación de Cadenas de Texto**
  Pide al usuario que ingrese una oración. Luego, escribe la función “procesarOracion” que haga lo siguiente:
  1. Quite los espacios en blanco al principio y al final.
  2. Divida la oración en palabras.
  3. Reemplace todas las vocales 'a' por '@'. Tener en cuenta utilizar “/a/g” que indica que se debe buscar todas las apariciones del carácter 'a' en la cadena (la bandera g es para global, lo que significa que reemplazará todas las apariciones, no solo la primera).
  4. Encuentre la posición de la primera aparición de la palabra "javascript". De no aparecer retornar -1
  5. Convierta la oración a una cadena de palabras separadas por guiones.

- **Ejercicio 3: Gestión Compleja de Arrays**

  A partir del siguiente array de productos, escriba la función ‘gestionarProductos’ que realice las siguientes tareas:

  1. Añada un nuevo producto al array.
  2. Elimine el último producto del array.
  3. Encuentre el índice de un producto específico. En este punto pueden usar forEach o investigar el método “*findIndex()”* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/findIndex

  4. Verifique si existe un producto con precio mayor a 50. Para esto investigar el método *“.some()”* https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Array/some

  5. Devuelva una cadena de nombres de productos separados por comas.

- **Ejercicio 4: Transformación y Análisis de Cadenas**

  Pide al usuario que ingrese una lista de nombres separados por comas. Los nombres a ingresar deben ser “Julian”, “Maria”, “Malena”, “Andrea”, “Pablo” y “Pedro”.

  Luego, escribe la función “transformarYAnalizarNombres”que realice las siguientes tareas:

  1. Quite los espacios en blanco alrededor de cada nombre.
  2. Verifique si existe el nombre "Juan".
  3. Reemplace todos los nombres "María" por "Ana".
  4. Encuentre el índice del nombre "Pedro".

5. Devuelva una cadena de nombres en orden alfabético separados por puntos. Investigar el método *".sort()"* https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort

- **Ejercicio 5: Manipulación de Arrays y Cadenas**
  A partir del siguiente array de frases, escribe la función "procesarFrases" que realice las siguientes tareas:
  1. Convierta cada frase a minúsculas. Investigar el método *".toLowerCase()"* para este punto. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/toLowerCase
  2. Divida cada frase en palabras.
  3. Reemplace las palabras "malo" por "bueno".
  4. Combine las palabras de cada frase en una nueva cadena separada por espacios.
  5. Devuelva un nuevo array con las frases modificadas.

```
let frases = [
    "El clima es MALO hoy",
    "Este libro es muy MALO",
    "El servicio aquí es MALO"
];
```

- **Ejercicio 6: Autos y más autos…**

Escribe una función "gestionarAutos" que realice las siguientes tareas con una lista predefinida de marcas de autos:

```
let entrada = "Toyota, Honda, Ford, Chevrolet, Nissan";
```

1. Quite los espacios en blanco alrededor de cada marca.
2. Verifique si existe la marca "Tesla".
3. Reemplace todas las marcas "Ford" por "BMW".
4. Encuentre el índice de la marca "Chevrolet".
5. Devuelva una cadena de marcas en orden alfabético separadas por puntos. Utilizar “.sort()”

- **Ejercicio 7: “La Floreria”**

  Escribe una función "gestionarFloreria" que realice las siguientes tareas con una lista predefinida de nombres de flores:

  ```
  let entrada = "Rosa, Tulipán, Orquídea, Lirio";
  ```

  1. Quite los espacios en blanco alrededor de cada flor.
  2. Verifique si existe la flor "Margarita" y, si está presente, agregue "Azucena" al final de la lista.
  3. Reemplace todas las flores "Orquídea" por "Clavel".
  4. Encuentre el índice de la flor "Girasol" y, si no está presente, agregue "Girasol" al inicio de la lista.

  Devuelva una cadena de flores en orden alfabético separadas por puntos.

Ejercicios - Algoritmos de Busqueda:

- **Ejercicio 8: Lugar y números**

Para practicar la implementación del algoritmo de búsqueda binaria, vamos a empezar con un ejemplo simple. Quiero que se familiaricen con la sintaxis del mismo, por lo que no hare énfasis en los datos, sino en la solución.
Dada la siguiente lista:

```
let list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11];
```

Utilizar el algoritmo de búsqueda binaria para responder los siguientes ítems:

- ¿Cuál es la posición del número 1?
- ¿Cuál es la posición del número 5?
- ¿Cuál es la posición del número 6?
- ¿Cuál es la posición del número 9?
- ¿Cuál es la posición del número 11?

Sabemos que responder estas preguntas es extremadamente fácil, ya que podemos leer el array, y determinar con un cálculo visual la posición de cada elemento, pero, la propuesta que les hago es que codeen un algoritmo de búsqueda binaria, que “busque” ese número, por ejemplo, el 6, y nos indique por consola la posición del mismo.

El objetivo de este ejercicio es que puedan practicar la sintaxis sin añadir complejidad extra.

- **Ejercicio 9: Desafío extra! Orden, lugar y números**

  Al ejercicio anterior vamos a aumentarle la dificultad un poco, y de paso, aplicamos lo aprendido en semanas anteriores. Quiero que hagan lo

mismo del ejercicio anterior (buscar la posición de un número en un array), pero partiendo de esta lista:

```
let list = [12,3,5,7,1,22,47,100];
```

Para aplicar búsqueda binaria, deberán ordenar primero la lista, de menor a mayor, utilizando bubble sort. Luego, respondan las siguientes preguntas:

- ¿Cuál es la posición del número 12?
- ¿Cuál es la posición del número 5?
- ¿Cuál es la posición del número 22?
- ¿Cuál es la posición del número 100?