

VECTORES EN JAVASCRIPT

En JavaScript, los vectores (o arrays) y las funciones de cadena son herramientas clave para la manipulación de datos. Aquí te doy un resumen de cómo funcionan ambos:

1. Vectores (Arrays) en JavaScript

Un vector es una lista de elementos que pueden ser de cualquier tipo de dato: números, cadenas, objetos, etc.

Crear un vector:

```
javascript
```

```
let numeros = [1, 2, 3, 4, 5];  
let frutas = ["manzana", "banana", "naranja"];
```

Métodos útiles para trabajar con arrays:

- **push ()**: Añade un elemento al final del array.

```
javascript
```

```
frutas.push("kiwi");
```

- **pop ()**: Elimina el último elemento del array.

```
javascript
```

```
frutas.pop(); // "naranja" es eliminado
```

- **shift ()**: Elimina el primer elemento del array.

```
javascript
```

```
frutas.shift(); // "manzana" es eliminado
```

- **unshift ()**: Añade un elemento al principio del array.

```
javascript
```

```
frutas.unshift("pera");
```

- **length**: Retorna el número de elementos en el array.

```
javascript
```

```
let longitud = frutas.length;
```

- **map ()**: Aplica una función a cada elemento del array y retorna un nuevo array.

```
javascript
```

```
let numerosCuadrados = numeros.map(num => num * num);
```

- **filter()**: Retorna un nuevo array con los elementos que cumplen una condición.

```
javascript
```

```
let numerosPares = numeros.filter(num => num % 2 === 0);
```

- **reduce()**: Reduce el array a un único valor.

```
javascript
```

```
let suma = numeros.reduce((total, num) => total + num, 0);
```

- **forEach()**: Ejecuta una función para cada elemento del array.

```
javascript
```

```
frutas.forEach(fruta => console.log(fruta));
```

2. Funciones de cadena (String) en JavaScript

Las cadenas de texto en JavaScript son objetos que tienen varios métodos útiles para su manipulación.

Crear una cadena:

```
javascript
```

```
let saludo = "Hola, mundo!";
```

Métodos útiles para trabajar con cadenas:

- **length**: Retorna la longitud de la cadena.

```
javascript
```

```
let longitud = saludo.length;
```

- **toUpperCase()**: Convierte la cadena a mayúsculas.

```
javascript
```

```
let enMayusculas = saludo.toUpperCase(); // "HOLA, MUNDO!"
```

- **toLowerCase()**: Convierte la cadena a minúsculas.

```
javascript
```

```
let enMinusculas = saludo.toLowerCase(); // "hola, mundo!"
```

- **substring()**: Retorna una parte de la cadena.

```
javascript
```

```
let parte = saludo.substring(0, 4); // "Hola"
```

- **split()**: Divide la cadena en un array según un separador.

```
javascript
```

```
let palabras = saludo.split(" "); // ["Hola,", "mundo!"]
```

- **indexOf()**: Retorna la posición de la primera aparición de un valor en la cadena. Si no lo encuentra, devuelve -1.

```
javascript
```

```
let posicion = saludo.indexOf("mundo"); // 6
```

- **includes()**: Verifica si una cadena contiene otra cadena.

```
javascript
```

```
let contiene = saludo.includes("Hola"); // true
```

- **replace()**: Reemplaza una parte de la cadena con otra.

```
javascript
```

```
let nuevoSaludo = saludo.replace("mundo", "JavaScript"); // "Hola, JavaScript!"
```

Ejemplo combinando Arrays y Strings:

```
javascript
```

```
let palabras = ["Hola", "mundo"];  
let frase = palabras.join(" "); // "Hola mundo"
```

Ambos conjuntos de funciones te permiten manipular datos de manera eficiente en JavaScript.

ejemplos de cómo trabajar con elementos de formulario como **radio buttons**, **checkboxes** y **select** en JavaScript. Estos ejemplos incluyen cómo obtener el valor seleccionado, verificar el estado de los checkboxes y cambiar el valor del `select` dinámicamente.

1. Radio Button

El siguiente código te permite seleccionar una opción de un grupo de radios y obtener el valor seleccionado con JavaScript:

html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Ejemplo Radio Buttons</title>
</head>
<body>
  <h3>Selecciona tu color favorito:</h3>
  <label><input type="radio" name="color" value="Rojo">
Rojo</label>
  <label><input type="radio" name="color" value="Azul">
Azul</label>
  <label><input type="radio" name="color" value="Verde">
Verde</label>

  <button onclick="getSelectedRadio()">Mostrar
selección</button>
  <p id="resultado"></p>

  <script>
    function getSelectedRadio() {
      let radios =
document.getElementsByName('color');
      let seleccionado = '';
      for (let i = 0; i < radios.length; i++) {
        if (radios[i].checked) {
          seleccionado = radios[i].value;
          break;
        }
      }
      document.getElementById('resultado').innerText
= "Color seleccionado: " + seleccionado;
    }
  </script>
</body>
</html>
```

2. Checkbox

Este ejemplo muestra cómo verificar qué casillas están seleccionadas y mostrar el valor de cada uno:

```
html

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Ejemplo Checkboxes</title>
</head>
<body>
  <h3>Selecciona tus frutas favoritas:</h3>
  <label><input type="checkbox" name="fruta"
value="Manzana"> Manzana</label>
  <label><input type="checkbox" name="fruta"
value="Banana"> Banana</label>
  <label><input type="checkbox" name="fruta"
value="Naranja"> Naranja</label>

  <button onclick="getSelectedCheckboxes()">Mostrar
selección</button>
  <p id="resultado"></p>

  <script>
    function getSelectedCheckboxes() {
      let checkboxes =
document.getElementsByName('fruta');
      let seleccionadas = [];
      for (let i = 0; i < checkboxes.length; i++) {
        if (checkboxes[i].checked) {

seleccionadas.push(checkboxes[i].value);
        }
      }
      document.getElementById('resultado').innerText
= "Frutas seleccionadas: " + seleccionadas.join(", ");
    }
  </script>
</body>
</html>
```

3. Select

El siguiente código te permite seleccionar una opción de un menú desplegable y obtener el valor seleccionado con JavaScript:

html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Ejemplo Select</title>
</head>
<body>
  <h3>Selecciona un país:</h3>
  <select id="pais">
    <option value="Argentina">Argentina</option>
    <option value="México">México</option>
    <option value="España">España</option>
  </select>

  <button onclick="getSelectedOption()">Mostrar
selección</button>
  <p id="resultado"></p>

  <script>
    function getSelectedOption() {
      let select = document.getElementById('pais');
      let seleccionado =
select.options[select.selectedIndex].value;
      document.getElementById('resultado').innerText
= "País seleccionado: " + seleccionado;
    }
  </script>
</body>
</html>
```

4. Cambiar el valor dinámicamente

En este ejemplo se cambia el valor de un select dinámicamente usando JavaScript:

html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Ejemplo Dinámico</title>
</head>
<body>
  <h3>Selecciona una ciudad:</h3>
  <select id="ciudad">
```

```
        <option value="Buenos Aires">Buenos Aires</option>
        <option value="Ciudad de México">Ciudad de
México</option>
        <option value="Madrid">Madrid</option>
    </select>

    <button onclick="setCity('Barcelona')">Cambiar a
Barcelona</button>
    <p id="resultado"></p>

    <script>
        function setCity(city) {
            let select = document.getElementById('ciudad');
            for (let i = 0; i < select.options.length; i++)
            {
                if (select.options[i].value === city) {
                    select.selectedIndex = i;
                    break;
                }
            }
            document.getElementById('resultado').innerText
= "Ciudad seleccionada: " +
select.options[select.selectedIndex].value;
        }
    </script>
</body>
</html>
```

Estos ejemplos cubren el uso básico de **radio buttons**, **checkboxes**, y **selects** en HTML y cómo interactuar con ellos mediante JavaScript.

1. HTML - Estructura del Slider

Define la estructura del carrusel con contenedores para las imágenes y los botones de navegación:

```
html

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Slider con JavaScript</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="slider-container">
        <div class="slider">
```

```
        
        
        
    </div>
    <button class="prev" onclick="moverSlider(-
1)">&#10094;</button>
    <button class="next"
onclick="moverSlider(1)">&#10095;</button>
    </div>

    <script src="script.js"></script>
</body>
</html>
```

2. CSS - Estilo del Slider

Define el estilo para el slider, asegurando que las imágenes se superpongan entre sí y que los botones de navegación estén posicionados correctamente:

css

```
/* Estilos generales del slider */
.slider-container {
    position: relative;
    max-width: 800px;
    margin: auto;
    overflow: hidden;
}

.slider {
    display: flex;
    width: 100%;
    transition: transform 0.5s ease-in-out;
}

.slide {
    min-width: 100%;
    transition: opacity 0.5s ease-in-out;
}

/* Estilos de los botones de navegación */
.prev, .next {
    position: absolute;
    top: 50%;
    transform: translateY(-50%);
    background-color: rgba(0, 0, 0, 0.5);
    color: white;
    border: none;
```



```
padding: 10px;
cursor: pointer;
font-size: 18px;
}

.prev {
  left: 10px;
}

.next {
  right: 10px;
}

.prev:hover, .next:hover {
  background-color: rgba(0, 0, 0, 0.8);
}
```

3. JavaScript - Funcionalidad del Slider

Aquí viene la parte de JavaScript que permite mover las imágenes del slider hacia adelante y hacia atrás cuando se hace clic en los botones.

```
javascript

let indiceSlide = 0;

function mostrarSlide(indice) {
  let slides = document.getElementsByClassName("slide");

  if (indice >= slides.length) {
    indiceSlide = 0;
  }

  if (indice < 0) {
    indiceSlide = slides.length - 1;
  }

  // Ocultar todos los slides
  for (let i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }

  // Mostrar el slide actual
  slides[indiceSlide].style.display = "block";
}

function moverSlider(direccion) {
  indiceSlide += direccion;
  mostrarSlide(indiceSlide);
}
```

```
// Mostrar el primer slide al cargar la página
mostrarSlide(indiceSlide);

// Desplazar automáticamente cada 3 segundos
setInterval(function () {
    moverSlider(1);
}, 3000);
```

¿Cómo funciona?

1. **HTML:** Define un contenedor `slider-container` que contiene el slider (`slider`) con las imágenes (`slide`) y dos botones para desplazarse entre las imágenes (`prev` y `next`).
2. **CSS:** El `slider` utiliza `display: flex` para alinear las imágenes en una fila. Solo una imagen se muestra a la vez, y los botones están posicionados a la izquierda y derecha de la pantalla.
3. **JavaScript:**
 - La función `mostrarSlide` muestra el slide actual (según el índice) y oculta los demás.
 - La función `moverSlider` cambia el índice del slide (avanzando o retrocediendo) y luego llama a `mostrarSlide`.
 - El slider cambia automáticamente de imagen cada 3 segundos con `setInterval`.

Resultado esperado:

Cuando abras este proyecto en tu navegador, verás un slider de imágenes que puedes controlar con los botones de navegación o dejar que avance automáticamente.

Personalizaciones:

- Puedes ajustar la velocidad del slider cambiando el valor en `setInterval`.