Manual de Arreglos JS

JUAN PABLO SAAVEDRA CHAMBO

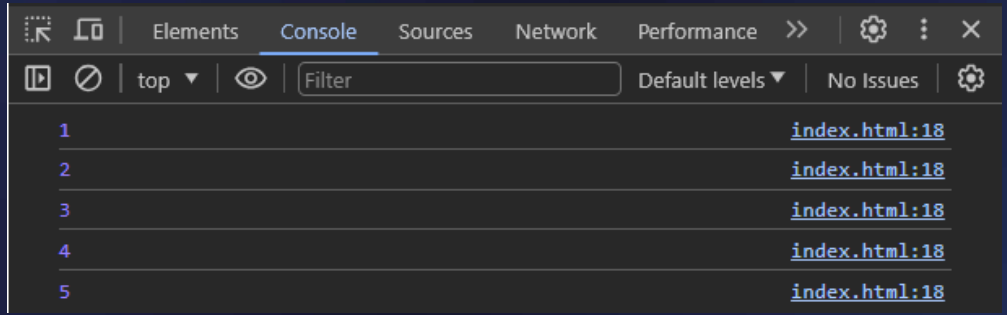
1. Ejercicio práctico de arreglos

Este código define una array de números (numeros) y luego la recorre con iterar para imprimir cada elemento en la consola.

VARIABLE	TIPO DE VARIABLE
numeros	Arreglo (array), Numérico (int)
primerNumero	Numérico (int)
segNumero	Numérico (int)
terNumero	Numérico (int)
cantidad	Numérico (int)
i	Numérico (int)

```
<body>
    <script>
        let numeros = [1, 2, 3, 4, 5];
        let primerNumero = numeros[0];
        let segundoNumero = numeros[2];
        let tercerNumero = numeros[4];
        let cantidad = numeros.length;
        for (let i = 0; i < cantidad; i++) {
           console.log(numeros[i]);
        }
    </script>

</body>
```



2. Contar los números del 1 al 10.

Este código JavaScript crea tres arreglos llamados numeros, par e impar. El arreglo numseros contiene números del 1 al 10, mientras que los arreglos par e impar almacenan los números pares e impares dentro de ese rango respectivamente.

VARIABLE	TIPO DE VARIABLE
numeros	Arreglo (array), Numérico (int)
pares	Arreglo (array), Numérico (int)
impares	Arreglo (array), Numérico (int)
longitud	Numérico (int)

```
Los números del 1 al 10: 1,2,3,4,5,6,7,8,9,10        ParesImpares.html:24
Números pares: 2,4,6,8,10                            ParesImpares.html:25
Números impares: 1,3,5,7,9                           ParesImpares.html:26
```

```
<script>
        let numeros = [];
        let pares = [];
        let impares = [];

        for (let longitud = 1; longitud <= 10; longitud++) {
            numeros.push(longitud);

            if (longitud % 2 === 0) {
                pares.push(longitud);
            } else {
                impares.push(longitud);
            }
        }

        console.log("Los números del 1 al 10: " + numeros);
        console.log("Números pares: " + pares);
        console.log("Números impares: " + impares);
</script>
```

3. Realizar una matriz de 5x5 de las tablas de multiplicar de 5 y el 9, sumar los números pares e impares.
Tabla 5

El código genera y muestra las tablas del 5 y del 9, identificando los números pares e impares en cada una y luego suma los valores de los pares e impares por separado.

VARIABLE	TIPO DE VARIABLE
tablaCinco	Arreglo (array), Numérico (int)
paresTablaCinco	Arreglo (array), Numérico (int)
imparesTablaCinco	Arreglo (array), Numérico (int)
numeroBase	Numérico (int)
sumaParesTablaCinco	Numérico (int)
sumaImparesTablaCinco	Numérico (int)
fila	Numérico (int)
columna	Numérico (int)

3. Realizar una matriz de 5x5 de las tablas de multiplicar de 5 y el 9, sumar los números pares e impares.

```
<script>
        let tablaCinco = [];
        let paresCinco = [];
        let imparesCinco = [];
        let numeroBase = 1;
        let sumaParesCinco;
        let sumaImparesCinco;

        for (let fila = 0; fila < 5; fila++) {
            tablaCinco[fila] = [];
            for (let columna = 0; columna < 5; columna++) {
                tablaCinco[fila][columna] = numeroBase * 5;
                numeroBase++;
                if (tablaCinco[fila][columna] % 2 === 0) {
                    paresCinco.push(tablaCinco[fila][columna]);
                } else {
                    imparesCinco.push(tablaCinco[fila][columna]);
                }
                sumaParesCinco = 0;
                for (let i = 0; i < paresCinco.length; i++) {
                    sumaParesCinco += paresCinco[i];
                }
                sumaImparesCinco = 0;
                for (let i = 0; i < imparesCinco.length; i++) {
                    sumaImparesCinco += imparesCinco[i];
                }
            }
        }
        console.log("Matriz 5: ", tablaCinco);
        console.log("Pares de la tabla del 5: " + paresCinco);
        console.log("Impares de la tabla del 5: " + imparesCinco);
        console.log("La suma total de los pares de la tabla del 5 es " + sumaParesCinco);
        console.log("La suma total de los impares de la tabla del 5 es " + sumaImparesCinco);
```

```
Matriz 5:                                                    TablasImparesPares.html:37
▼ (5) [Array(5), Array(5), Array(5), Array(5), Array(5)] i
  ▶ 0: (5) [5, 10, 15, 20, 25]
  ▶ 1: (5) [30, 35, 40, 45, 50]
  ▶ 2: (5) [55, 60, 65, 70, 75]
  ▶ 3: (5) [80, 85, 90, 95, 100]
  ▶ 4: (5) [105, 110, 115, 120, 125]
    length: 5
  ▶ [[Prototype]]: Array(0)
Pares de la tabla del 5:                                     TablasImparesPares.html:38
10,20,30,40,50,60,70,80,90,100,110,120
Impares de la tabla del 5:                                   TablasImparesPares.html:39
5,15,25,35,45,55,65,75,85,95,105,115,125
La suma total de los pares de la tabla del 5                 TablasImparesPares.html:40
es 780
La suma total de los impares de la tabla del                 TablasImparesPares.html:41
5 es 845
```

3. Realizar una matriz de 5x5 de las tablas de multiplicar de 5 y el 9, sumar los números pares e impares.

Tabla 9

El código genera y muestra las tablas del 5 y del 9, identificando los números pares e impares en cada una y luego suma los valores de los pares e impares por separado.

VARIABLE	TIPO DE VARIABLE
tablaNueve	Arreglo (array), Numérico (int)
paresTablaNueve	Arreglo (array), Numérico (int)
imparesTablaNueve	Arreglo (array), Numérico (int)
Multiplicadores	Numérico (int)
sumaParesTablaNueve	Numérico (int)
sumaImparesTablaNueve	Numérico (int)
fila	Numérico (int)
columna	Numérico (int)

3. Realizar una matriz de 5x5 de las tablas de multiplicar de 5 y el 9, sumar los números pares e impares.

```
let tablaNueve = [];
        let paresNueve = [];
        let imparesNueve = [];
        let multiplicador = 1;
        let sumaParesNueve;
        let sumaImparesNueve;
        for (let fila = 0; fila < 5; fila++) {
            tablaNueve[fila] = [];
            for (let columna = 0; columna < 5; columna++) {
                tablaNueve[fila][columna] = multiplicador * 9;
                multiplicador++;
                if (tablaNueve[fila][columna] % 2 === 0) {
                    paresNueve.push(tablaNueve[fila][columna]);
                } else {
                    imparesNueve.push(tablaNueve[fila][columna]);
                }
                sumaParesNueve = 0;
                for (let i = 0; i < paresNueve.length; i++) {
                    sumaParesNueve += paresNueve[i];
                }
                sumaImparesNueve = 0;
                for (let i = 0; i < imparesNueve.length; i++) {
                    sumaImparesNueve += imparesNueve[i];
                }
            }
        }
        console.log("Matriz 9:", tablaNueve);
        console.log("Pares de la tabla del 9:" + paresNueve);
        console.log("Impares de la tabla del 9:" + imparesNueve);
        console.log("La suma total de los pares de la tabla del 9 es " + sumaParesNueve);
        console.log("La suma total de los impares de la tabla del 9 es " + sumaImparesNueve);
    </script>
```

```
Matriz 9:                                                    index.html:79
▼ (5) [Array(5), Array(5), Array(5), Array(5), Array(5)] i
  ▶ 0: (5) [9, 18, 27, 36, 45]
  ▶ 1: (5) [54, 63, 72, 81, 90]
  ▶ 2: (5) [99, 108, 117, 126, 135]
  ▶ 3: (5) [144, 153, 162, 171, 180]
  ▶ 4: (5) [189, 198, 207, 216, 225]
    length: 5
  ▶ [[Prototype]]: Array(0)
Pares de la tabla del                                        index.html:80
9:18,36,54,72,90,108,126,144,162,180,198,216
Impares de la tabla del                                      index.html:81
9:9,27,45,63,81,99,117,135,153,171,189,207,225
La suma total de los pares de la tabla del 9 es 1404         index.html:82
La suma total de los impares de la tabla del 9 es 1521       index.html:83
```

4. Matriz en X

El código genera una matriz aleatoria de 5x5 y luego la transforma en una matriz en forma de X, donde los elementos en diagonal y anti-diagonal se mantienen, y los demás son espacios en blanco. Finalmente, muestra ambas matrices por consola.

Arreglos: matriz, matrizX1, matrizX2, matrixX3, B, I, N, G, O,

VARIABLE	TIPO DE VARIABLE
$matrizAleatoria	Arreglo (array), Numérico (int)
$filaAleatoria	Numérico (int)
$columnaAleatoria	Numérico (int)
$numeroAleatorio	Numérico (int)
$fila	Numérico (int)
$matrizX	Numérico (int)
$filaX	Numérico (int)
$columnaX	Numérico (int)

4. Matriz en X

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Matriz en forma de X</title>
</head>
<body>
    <script>

        let matrizOriginal = [];

        for (let fila = 0; fila < 5; fila++) {
            matrizOriginal[fila] = [];
            for (let columna = 0; columna < 5; columna++) {
                let numero = Math.floor(Math.random() * 100) + 1;
                matrizOriginal[fila][columna] = numero;
            }
        }
        console.log("Matriz generada:");
        console.log(matrizOriginal);

        // Matriz en forma de X
        let matrizX = "";

        for (let filaX = 0; filaX < 5; filaX++) {
            for (let columnaX = 0; columnaX < 5; columnaX++) {
                if (filaX === columnaX || filaX + columnaX === 4) {
                    matrizX += matrizOriginal[filaX][columnaX] + "\t";
                } else {
                    matrizX += " \t";
                }
            }
            matrizX += "\n";
        }

        console.log("Matriz en forma de X:");
        console.log(matrizX);
    </script>

</body>
</html>
```

```
Matriz generada:                                    MatrizX.html:20
                                                    MatrizX.html:21
▼ (5) [Array(5), Array(5), Array(5), Array(5), Array(5)] i
  ▶ 0: (5) [40, 20, 36, 78, 3]
  ▶ 1: (5) [74, 8, 99, 76, 67]
  ▶ 2: (5) [99, 23, 73, 22, 64]
  ▶ 3: (5) [15, 98, 83, 15, 19]
  ▶ 4: (5) [5, 50, 50, 1, 85]
    length: 5
  ▶ [[Prototype]]: Array(0)
Matriz en forma de X:                               MatrizX.html:37
40              3                                   MatrizX.html:38
    8       76
        73
    98      15
5               85
```

5. Tabla del bingo.

El código genera una tabla de bingo de 5x5 con números, divide los números en grupos correspondientes a las letras B, I, N, G, O, y luego extrae los números en forma de X en tres partes diferentes (X1, X2 y X3). Finalmente, muestra cada parte por consola.

Arreglos: matriz, matrizX1, matrizX2, matrixX3, B, I, N, G, O,

VARIABLE	TIPO DE VARIABLE
tabla	Arreglo (array), Numérico (int)
tablaX1	Arreglo (array), Numérico (int)
tablaX2	Arreglo (array), Numérico (int)
tablaX3	Numérico (int)
grupoB	Arreglo (array), Numérico (int)
grupo	Arreglo (array), Numérico (int)
grupoN	Arreglo (array), Numérico (int)
grupoG	Arreglo (array), Numérico (int)
grupo	Arreglo (array), Numérico (int)
numero	Numérico (int)
fila	Numérico (int)
columna	Numérico (int)

5. Tabla del bingo.

```
<script>
        let tabla = [];
        let tablaX1 = [];
        let tablaX2 = [];
        let tablaX3 = [];
        let grupoB = [];
        let grupoI = [];
        let grupoN = [];
        let grupoG = [];
        let grupoO = [];
        let numero = 1;

        for (let fila = 0; fila < 5; fila++) {
            tabla[fila] = [];
            for (let columna = 0; columna < 5; columna++) {
                tabla[fila][columna] = numero * 2;

                if (columna === 0) grupoB.push(tabla[fila][columna]);
                else if (columna === 1) grupoI.push(tabla[fila][columna]);
                else if (columna === 2) grupoN.push(tabla[fila][columna]);
                else if (columna === 3) grupoG.push(tabla[fila][columna]);
                else if (columna === 4) grupoO.push(tabla[fila][columna]);
                numero++;
            }
        }

        for (let fila = 0; fila < 3; fila++) {
            for (let columna = 0; columna < 3; columna++) {
                if(fila == columna || fila + columna == 2){
                    tablaX1.push(tabla[fila][columna]);
                }
            }
        }

        for (let fila = 0; fila < 5; fila++) {
            for (let columna = 0; columna < 3; columna++) {
                if(fila + columna == (columna + 1) * 2 || fila + columna == 4){
                    tablaX2.push(tabla[fila][columna]);
                }
            }
        }

        for (let fila = 0; fila < 3; fila++) {
            for (let columna = 2; columna < 5; columna++) {
                if(fila + columna == (columna + 1) * 2 || fila + columna == 4){
                    tablaX3.push(tabla[fila][columna]);
                }
            }
        }

        console.log(tabla);
        console.log("B: " + grupoB);
        console.log("I: " + grupoI);
        console.log("N: " + grupoN);
        console.log("G: " + grupoG);
        console.log("O: " + grupoO);
        console.log("X1: " + tablaX1);
        console.log("X2: " + tablaX2);
        console.log("X3: " + tablaX3);
    </script>
```

```
index.html:62
▼ (5) [Array(5), Array(5), Array(5), Array(5), Array(5)] i
  ▶ 0: (5) [2, 4, 6, 8, 10]
  ▶ 1: (5) [12, 14, 16, 18, 20]
  ▶ 2: (5) [22, 24, 26, 28, 30]
  ▶ 3: (5) [32, 34, 36, 38, 40]
  ▶ 4: (5) [42, 44, 46, 48, 50]
    length: 5
  ▶ [[Prototype]]: Array(0)
B: 2,12,22,32,42                    index.html:63
I: 4,14,24,34,44                    index.html:64
N: 6,16,26,36,46                    index.html:65
G: 8,18,28,38,48                    index.html:66
O: 10,20,30,40,50                   index.html:67
X1: 2,6,14,22,26                    index.html:68
X2: 22,26,34,42,46                  index.html:69
X3: 10,18,26                        index.html:70
```

6 Nomina.

Este código en JavaScript calcula la nómina de empleados, aplicando diferentes cálculos relacionados con los salarios, subsidios, retenciones y deducciones, y luego almacena los resultados en un array llamado `listaNomina`.

Arreglos: personas, nomina

VARIABLE	TIPO DE VARIABLE
persona	Arreglo (array), Numérico (int)
nomina	Arreglo (array), Numérico (int)
salarioMin	Arreglo (array), Numérico (int)
numRegistros	Numérico (int)
mostrar	Arreglo (array), Numérico (int)
mostrarRetencion	Arreglo (array), Numérico (int)

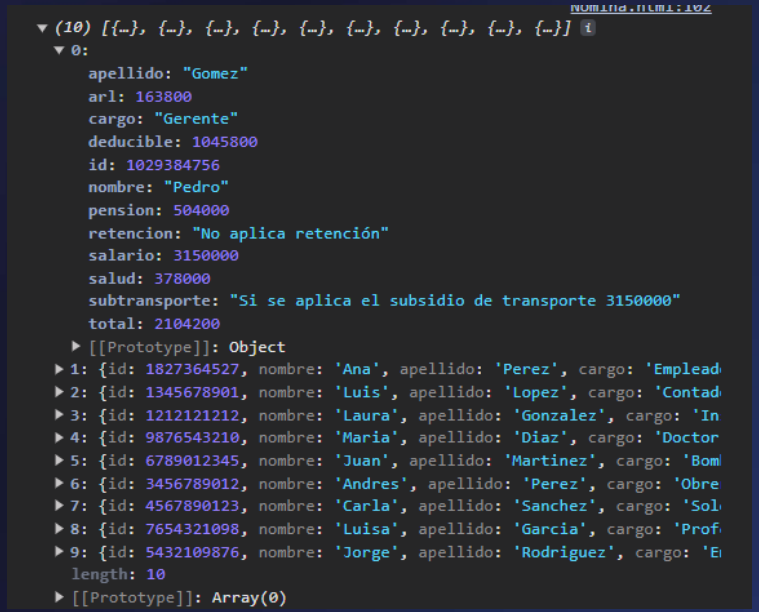
Arreglo: Persona

VARIABLE	TIPO DE VARIABLE
id	Numérico (int)
nombre	string
apellido	string
cargo	Numérico (int)
valorDia	Numérico (int)
diasTrabajados	Numérico (int)

6. Nomina.

Arreglos: nomina

VARIABLE	TIPO DE VARIABLE
Iteracion	Numérico (int)
trabajador	Arreglo (array),
Numérico (int)	Numérico (int)
subTransCalculado	Numérico (int)
retencionCalculada	string
saludCalculada	string
pensionCalculada	Numérico (int)
arlCalculada	Numérico (int)
deducibleCalculada	Numérico (int)
totalCalculado	Numérico (int)



6. Nomina

```
<script>
    let empleados = []; // el array principal
    let listaNomina = []; // el array para almacenar la nómina
    let salarioMinimo = 16000000;
    let cantidadRegistros;
    let mostrar;
    let mostrarRetencion;

    const calcularSalario = function(valorDia, dias) {
        return valorDia * dias;
    };

    const calcularSubsidioTransporte = function(salario) {
        if (salario < salarioMinimo * 2) {
            return "Si se aplica el subsidio de transporte " + salario;
        } else {
            return "No se aplica el subsidio de transporte";
        }
    };

    const calcularSalud = function(salario) {
        return salario * 0.12;
    };

    const calcularPension = function(salario) {
        return salario * 0.16;
    };

    const calcularArl = function(salario) {
        return salario * 0.052;
    };

    const calcularDeducible = function(salario) {
        return calcularSalud(salario) + calcularPension(salario) + calcularArl(salario);
    };

    const calcularRetencion = function(salario) {
        if (salario > salarioMinimo * 12) {
            return "Retencion de 0.08 " + salario * 0.08;
        } else if (salario > salarioMinimo * 8) {
            return "Retencion de 0.04 " + salario * 0.04;
        } else if (salario > salarioMinimo * 6) {
            return "Retencion de 0.02 " + salario * 0.02;
        } else {
            return "No aplica retención";
        }
    };

    const calcularTotal = function(salario) {
        return salario - calcularDeducible(salario);
    };

    empleados = [
    { id: 1029384756, nombre: "Pedro", apellido: 'Gomez', cargo: 'Gerente', valorDia: 75000, diasTrabajado: 42 },
    { id: 1827364527, nombre: "Ana", apellido: "Perez", cargo: 'Empleado', valorDia: 60000, diasTrabajado: 50 },
    { id: 1345678901, nombre: 'Luis', apellido: 'Lopez', cargo: 'Contador', valorDia: 35000, diasTrabajado: 60 },
    { id: 1212121212, nombre: "Laura", apellido: 'Gonzalez', cargo: "Instructor", valorDia: 42000, diasTrabajado: 70 },
    { id: 9876543210, nombre: "Maria", apellido: 'Diaz', cargo: 'Doctor', valorDia: 15000, diasTrabajado: 80 },
    { id: 6789012345, nombre: 'Juan', apellido: 'Martinez', cargo: 'Bombero', valorDia: 55000, diasTrabajado: 85 },
    { id: 3456789012, nombre: "Andres", apellido: 'Perez', cargo: 'Obrero', valorDia: 18000, diasTrabajado: 95 },
    { id: 4567890123, nombre: "Carla", apellido: 'Sanchez', cargo: 'Soldador', valorDia: 31000, diasTrabajado: 15 },
    { id: 7654321098, nombre: 'Luisa', apellido: 'Garcia', cargo: 'Profesor', valorDia: 135000, diasTrabajado: 25 },
    { id: 5432109876, nombre: "Jorge", apellido: "Rodriguez", cargo: "Empleado", valorDia: 320000, diasTrabajado: 40 }
];


    cantidadRegistros = empleados.length;

    for (let iteracion = 0; iteracion < cantidadRegistros; iteracion++) {
        let trabajador = empleados[iteracion];
        let salarioCalculado = calcularSalario(trabajador.valorDia, trabajador.diasTrabajado);
        let subTransCalculada = calcularSubsidioTransporte(salarioCalculado);
        let retencionCalculada = calcularRetencion(salarioCalculado);
        let saludCalculada = calcularSalud(salarioCalculado);
        let pensionCalculada = calcularPension(salarioCalculado);
        let arlCalculada = calcularArl(salarioCalculado);
        let deducibleCalculada = calcularDeducible(salarioCalculado);
        let totalCalculado = calcularTotal(salarioCalculado);
        listaNomina.push({
            id: trabajador.id,
            nombre: trabajador.nombre,
            apellido: trabajador.apellido,
            cargo: trabajador.cargo,
            salario: salarioCalculado,
            subtransporte: subTransCalculada,
            retencion: retencionCalculada,
            salud: saludCalculada,
            pension: pensionCalculada,
            arl: arlCalculada,
            deducible: deducibleCalculada,
            total: totalCalculado
        });
    }
    console.log(listaNomina);
</script>

</body>
</html>
```

7. Ejercicio de tienda

Este código sirve como un sistema básico de gestión de inventario, permitiendo agregar nuevos productos y buscar artículos específicos (frutas y verduras) dentro de la tienda.

Arreglos: tienda y listaBusqueda

VARIABLE	TIPO DE VARIABLE
tienda	Arreglo (array)
nuevoProducto	Arreglo (array)
listaBusqueda	Arreglo (array)

Arreglos: tienda y nuevoProducto

VARIABLE	TIPO DE VARIABLE
producto	string
tipoProducto	string
tipoUnidad	string
cantidad	Numérico (int)
precio	Numérico (int)

7. Ejercicio de tienda

```
<script>
    let tienda = [];
    let nuevoProducto;
    let listaBusqueda = [];

    tienda = [
        { producto: 'Arroz', tipoProducto: 'granos', tipoUnidad:
'gramos', cantidad: 1000, precio: 2450 },
        { producto: "Trucha", tipoProducto: 'carnes', tipoUnidad:
'gramos', cantidad: 1000, precio: 9000 },
        { producto: 'Papa', tipoProducto: 'fruver', tipoUnidad:
'gramos', cantidad: 500, precio: 1000 },
        { producto: 'Mora', tipoProducto: 'fruver', tipoUnidad:
'gramos', cantidad: 500, precio: 1500 },
        { producto: 'Pollo Entero', tipoProducto: 'carnes',
tipoUnidad: 'gramos', cantidad: 1000, precio: 4500 },
        { producto: 'Carne entera', tipoProducto: 'carne',
tipoUnidad: 'gramos', cantidad: 1000, precio: 2450 }
    ];

    nuevoProducto = {producto: 'Res', tipoProducto:'carnes',
tipoUnidad: 'gramos', cantidad:1000, precio:7500};
    tienda.push (nuevoProducto);

    for (let iteracion = 0; iteracion < tienda.length; iteracion++)
{
        console.log(tienda[iteracion]);
        if (tienda[iteracion].tipoProducto === 'fruver') {
            listaBusqueda.push(tienda[iteracion]);
        }
    }
    console.log(listaBusqueda);
</script>
```