**Funciones punto 1**

**abs() - Valor absoluto:** La función abs() devuelve el valor absoluto de un número, es decir, su valor sin el signo negativo.

Ejemplo:

javascript

let num = -5;

let absValue = Math.abs(num); // absValue será 5

**round() - Redondeo al entero más cercano:** La función round() redondea un número al entero más cercano.

Ejemplo:

javascript

let num = 4.6;

let roundedValue = Math.round(num); // roundedValue será 5

**ceil() - Redondeo hacia arriba:** La función ceil() redondea un número al siguiente entero mayor o igual.

Ejemplo:

javascript

let num = 3.2;

let ceilValue = Math.ceil(num); // ceilValue será 4

**floor() - Redondeo hacia abajo:** La función floor() redondea un número al siguiente entero menor o igual.

Ejemplo:

javascript

let num = 6.8;

let floorValue = Math.floor(num); // floorValue será 6

**exp() - Exponencial:** La función exp() devuelve el valor de *e* elevado a la potencia del argumento.

Ejemplo:

javascript

let exponent = 2;

let expValue = Math.exp(exponent); // expValue será aproximadamente 7.389

**log() - Logaritmo natural:** La función log() devuelve el logaritmo natural (base *e*) de un número.

Ejemplo:

javascript

let num = 10;

let logValue = Math.log(num); // logValue será aproximadamente 2.302

**random() - Número aleatorio:** La función random() devuelve un número decimal pseudoaleatorio en el rango [0, 1).

Ejemplo:

javascript

let randomValue = Math.random();

PUNTO 2:

**push() - Agregar elementos al final:** El método push() se utiliza para agregar uno o más elementos al final de un arreglo. Estos elementos se añaden en el orden en que se proporcionan.

Ejemplo:

javascript

let fruits = ["apple", "banana"];

fruits.push("orange"); // Ahora el arreglo fruits contiene ["apple", "banana", "orange"]

También puedes usar push() para agregar múltiples elementos al final del arreglo:

javascript

 fruits.push("grape", "kiwi"); // Ahora el arreglo fruits contiene ["apple", "banana", "orange", "grape", "kiwi"]

 **pop() - Quitar elemento del final:** El método pop() se utiliza para eliminar y devolver el último elemento de un arreglo. El tamaño del arreglo disminuirá en uno después de la operación.

Ejemplo:

javascript

1. let fruits = ["apple", "banana", "orange"];
2. let removedFruit = fruits.pop(); // removedFruit contendrá "orange" y fruits será ["apple", "banana"]
3. pop() es útil para eliminar elementos de la pila (LIFO - Last In, First Out), como en la implementación de estructuras de datos.

Ambos métodos son formas efectivas de manipular arreglos y son ampliamente utilizados en programación para agregar o quitar elementos en la estructura de datos. Es importante tener en cuenta que push() y pop() afectan directamente al arreglo original y que pop() devuelve el elemento eliminado, lo que te permite realizar acciones con él después de la operación.

PUNTO 3

En JavaScript, el método reverse() se utiliza en arreglos (matrices) para invertir el orden de los elementos del arreglo. Esto significa que el primer elemento se convierte en el último, el segundo en el penúltimo, y así sucesivamente. El método reverse() modifica el arreglo original y no devuelve un nuevo arreglo invertido; en cambio, altera el arreglo existente.

Aquí tienes un ejemplo de cómo se utiliza el método reverse():

javascript

let numbers = [1, 2, 3, 4, 5];

numbers.reverse();

console.log(numbers); // Resultado: [5, 4, 3, 2, 1]

En este ejemplo, el arreglo numbers original se invierte y se convierte en [5, 4, 3, 2, 1].

El método reverse() es útil cuando se necesita cambiar el orden de los elementos en un arreglo de manera directa y sencilla. Sin embargo, es importante recordar que este método modifica el arreglo original, por lo que si necesitas mantener el arreglo original intacto, es recomendable crear una copia antes de usar reverse().