

Javascript Array Operations

Exercise 1: Array Basics

// Create an empty array called myArray.

```
let myArray = [ ];
```

// Add three different data types to myArray.

```
myArray.push("Hello", 42, true);
```

// Print the length of the array to the console.

```
console.log("Length of myArray:", myArray.length);
```

// Use a loop to log each element in the array to the console.

```
for (let i = 0; i < myArray.length; i++) {  
  console.log("Element", i + 1, ":", myArray[i]);  
}
```

Run the code and observe the output in the console.

Exercise 2: Array Methods

// Create an array fruits with at least five fruit names.

```
let fruits = ["Apple", "Banana", "Orange", "Grapes", "Mango"];
```

// Use push() to add a new fruit to the end.

```
fruits.push("Pineapple");

// Use pop() to remove the last fruit.
fruits.pop();

// Use unshift() to add a new fruit to the beginning.
fruits.unshift("Strawberry");

// Use shift() to remove the first fruit.
fruits.shift();

// Print the modified fruits array.
console.log("Modified fruits array:", fruits);
```

Run the code and observe the modified `fruits` array.

Exercise 3: Array Iteration

```
// Create an array numbers with at least five numbers.
let numbers = [1, 3, 5, 7, 9];

// Use forEach() to print each element to the console.
numbers.forEach(function (number) {
  console.log("Number:", number);
});

// Use map() to create a new array multiplied by 2.
```

```
let doubledNumbers = numbers.map(function (number) {  
  return number * 2;  
});
```

// Use filter() to create a new array with numbers greater than 5.

```
let greaterThanFive = numbers.filter(function (number) {  
  return number > 5;  
});
```

// Print the modified arrays.

```
console.log("Doubled numbers:", doubledNumbers);  
console.log("Numbers greater than 5:", greaterThanFive);
```

Exercise 4: Multi-dimensional Arrays

// Create a 2D array named matrix.

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

// Access and print an element from the second row and third column.

```
console.log("Element at (2, 3):", matrix[1][2]);
```

// Use nested loops to iterate through all elements and print them.

```
for (let row = 0; row < matrix.length; row++) {  
  for (let col = 0; col < matrix[row].length; col++) {  
    console.log("Element at (" + (row + 1) + ", " + (col + 1) +  
      "):", matrix[row][col]);  
  }  
}
```

Exercise 5: Array Challenges

// Solve array challenges.

```
let numbersArray = [2, 4, 6, 8, 10];
```

// Find the sum of all numbers in the array.

```
let sum = numbersArray.reduce(function (acc, num) {  
  return acc + num;  
}, 0);
```

// Identify and print the largest number in the array.

```
let maxNumber = Math.max(...numbersArray);
```

// here the three dots represent the spread syntax in JavaScript. Here it is used to spread the each element of

The array to provide as an argument.

The spread syntax allows you to "spread" the elements of an array into separate values. So, when ...numbersArray is used as an argument for Math.max, it effectively passes each element of the array as a separate argument to the Math.max function

```
// Check if a specific value exists in the array.  
let isValueExist = numbersArray.includes(6);
```

```
// Reverse the order of elements in the array.  
let reversedArray = numbersArray.reverse();
```

```
// Print the results.  
console.log("Sum of numbers:", sum);  
console.log("Largest number:", maxNumber);  
console.log("Does 6 exist in the array?", isValueExist);  
console.log("Reversed array:", reversedArray);
```