# JavaScript Exercises Template (1-101)

This document contains 101 JavaScript exercises with answers. Each exercise is classified and numbered accordingly.

## Basic JavaScript

1. **Declare variables of different data types and print their types.**

// Declaring variables of different data types

const name = "Alice"; // String

const age = 25; // Number

const isStudent = true; // Boolean

const height = null; // Null

let weight; // Undefined

const bigNumber = 9007199254740991n; // BigInt

const uniqueID = Symbol("id"); // Symbol

const person = { name: "Bob", age: 30 }; // Object

const numbers = [1, 2, 3]; // Array (Technically an object)

const greet = function() { return "Hello"; }; // Function

// Printing variable types

console.log(`Type of name: ${typeof name}`); // "string"

console.log(`Type of age: ${typeof age}`); // "number"

console.log(`Type of isStudent: ${typeof isStudent}`); // "boolean"

console.log(`Type of height: ${typeof height}`); // "object" (special case of null)

console.log(`Type of weight: ${typeof weight}`); // "undefined"

console.log(`Type of bigNumber: ${typeof bigNumber}`); // "bigint"

console.log(`Type of uniqueID: ${typeof uniqueID}`); // "symbol"

console.log(`Type of person: ${typeof person}`); // "object"

console.log(`Type of numbers: ${typeof numbers}`); // "object" (arrays are objects in JavaScript)

console.log(`Type of greet: ${typeof greet}`); // "function"

**📌 Explanation**

typeof is used to determine the type of each variable.

**null returns "object"** due to a historical JavaScript bug, but it is a **primitive value**.

**Arrays** are technically **objects** in JavaScript.

1. **Write a program to swap two variables.**

**Method 1: Using a Temporary Variable**

let a = 5, b = 10;

console.log(`Before Swap: a = ${a}, b = ${b}`);

let temp = a;

a = b;

b = temp;

console.log(`After Swap: a = ${a}, b = ${b}`);

✅ **Output:**

Before Swap: a = 5, b = 10

After Swap: a = 10, b = 5

**Method 2: Using Destructuring Assignment (ES6)**

let x = 5, y = 10;

console.log(`Before Swap: x = ${x}, y = ${y}`);

[x, y] = [y, x]; // Swapping without temp variable

console.log(`After Swap: x = ${x}, y = ${y}`);

✅ **Output:**

Before Swap: x = 5, y = 10

After Swap: x = 10, y = 5

**Method 3: Using Arithmetic Operations**

let p = 5, q = 10;

console.log(`Before Swap: p = ${p}, q = ${q}`);

p = p + q; // p = 5 + 10 = 15

q = p - q; // q = 15 - 10 = 5

p = p - q; // p = 15 - 5 = 10

console.log(`After Swap: p = ${p}, q = ${q}`);

✅ **Output:**

Before Swap: p = 5, q = 10

After Swap: p = 10, q = 5

**📌 Summary**

**Method 1:** Uses a temporary variable (**Safe & Traditional**).

**Method 2:** Uses ES6 **destructuring** (**Recommended**).

**Method 3:** Uses arithmetic (**Avoid if numbers are large** to prevent overflow).

1. **Write a JavaScript function to check if a number is even or odd.**

**✅ Method 1: Using the Modulus (**%**) Operator**

function checkEvenOrOdd(number) {

if (number % 2 === 0) {

console.log(`${number} is Even`);

} else {

console.log(`${number} is Odd`);

}

}

// Example Usage

checkEvenOrOdd(4); // ✅ 4 is Even

checkEvenOrOdd(7); // ✅ 7 is Odd

**✅ Method 2: Using a Ternary Operator (Shorter Version)**

function checkEvenOrOdd(number) {

console.log(`${number} is ${number % 2 === 0 ? "Even" : "Odd"}`);

}

// Example Usage

checkEvenOrOdd(10); // ✅ 10 is Even

checkEvenOrOdd(15); // ✅ 15 is Odd

**✅ Method 3: Returning the Result Instead of Printing**

function isEven(number) {

return number % 2 === 0;

}

// Example Usage

console.log(isEven(8)); // ✅ true (Even)

console.log(isEven(13)); // ✅ false (Odd)

### ****📌 Summary****

|  |  |  |
| --- | --- | --- |
| **Method** | **Approach** | **Best Use Case** |
| **Method 1** | Uses if-else | Best for readability |
| **Method 2** | Uses **ternary (? :)** | Best for short, concise code |
| **Method 3** | Returns **Boolean (true or false)** | Best for programmatic use |

1. **Write a program that converts a temperature from Celsius to Fahrenheit.**

**✅ Formula for Conversion**

F=(C×9/5)+32

**✅ Method 1: Using a Function**

function celsiusToFahrenheit(celsius) {

let fahrenheit = (celsius \* 9/5) + 32;

console.log(`${celsius}°C is equal to ${fahrenheit.toFixed(2)}°F`);

}

// Example Usage

celsiusToFahrenheit(0); // ✅ 0°C is equal to 32.00°F

celsiusToFahrenheit(25); // ✅ 25°C is equal to 77.00°F

celsiusToFahrenheit(100); // ✅ 100°C is equal to 212.00°F

**✅ Method 2: Returning the Value Instead of Printing**

function celsiusToFahrenheit(celsius) {

return (celsius \* 9/5) + 32;

}

// Example Usage

console.log(celsiusToFahrenheit(30)); // ✅ 86

console.log(celsiusToFahrenheit(-10)); // ✅ 14

**✅ Method 3: Using an Arrow Function (Short Version)**

const celsiusToFahrenheit = celsius => (celsius \* 9/5) + 32;

console.log(celsiusToFahrenheit(20)); // ✅ 68

console.log(celsiusToFahrenheit(37)); // ✅ 98.6

### ****📌 Summary****

|  |  |  |
| --- | --- | --- |
| **Method** | **Approach** | **Best Use Case** |
| **Method 1** | Uses console.log inside the function | Best for direct output |
| **Method 2** | Returns the Fahrenheit value | Best for further calculations |
| **Method 3** | Uses an **arrow function** | Best for concise code |

1. **Write a program that finds the maximum of three numbers.**

Here is a **JavaScript program** that finds the **maximum of three numbers** using different methods:

✅ Method 1: Using Math.max()

function findMax(a, b, c) {

return Math.max(a, b, c);

}

// Example Usage

console.log(findMax(10, 20, 15)); // ✅ 20

console.log(findMax(-5, -10, 0)); // ✅ 0

console.log(findMax(100, 50, 200)); // ✅ 200

✔ Math.max() is the easiest and most efficient method.

✅ Method 2: Using if-else Statements

function findMax(a, b, c) {

let max = a;

if (b > max) {

max = b;

}

if (c > max) {

max = c;

}

return max;

}

// Example Usage

console.log(findMax(5, 12, 8)); // ✅ 12

console.log(findMax(-1, -3, -2)); // ✅ -1

✔ This method is useful if you want to manually compare values.

✅ Method 3: Using a Ternary Operator (Short Version)

const findMax = (a, b, c) => (a > b ? (a > c ? a : c) : (b > c ? b : c));

**// Example Usage**

console.log(findMax(7, 14, 9)); // ✅ 14

console.log(findMax(3, 3, 3)); // ✅ 3 (All are equal)

console.log(findMax(-2, -8, -5)); // ✅ -2

✔ The ternary operator is a **shorter and faster** alternative.

### 📌 Summary

|  |  |  |
| --- | --- | --- |
| **Method** | **Approach** | **Best Use Case** |
| **Method 1** | Uses Math.max() | Best & simplest |
| **Method 2** | Uses if-else | Best for beginners |
| **Method 3** | Uses ternary operators | Best for concise code |

## String Manipulation

1. **Write a program to count the number of vowels in a string.**

function countVowels(str) {

const vowels = 'aeiouAEIOU';

let count = 0;

for (let char of str) {

if (vowels.includes(char)) {

count++;

}

}

return count;

}

// Example usage:

const input = "Hello, world!";

console.log(`Number of vowels: ${countVowels(input)}`);

**✅ Output for "Hello, world!":**

Number of vowels: 3

1. **Write a function to reverse a given string.**

function reverseString(str) {

return str.split('').reverse().join('');

}

// Example usage:

const input = "Hello, world!";

console.log(`Reversed string: ${reverseString(input)}`);

✅ Output for "Hello, world!":

Reversed string: !dlrow ,olleH

1. **Write a program to check if a string is a palindrome.**

function checkStrictPalindrome(input) {

const reversed = input.split('').reverse().join('');

if (input === reversed) {

console.log(`"${input}" is a strict palindrome.`);

} else {

console.log(`"${input}" is NOT a strict palindrome.`);

}

}

// Example usage:

checkStrictPalindrome("Madam");

checkStrictPalindrome("racecar");

checkStrictPalindrome("Was it a car or a cat I saw?");

**✅ Output:**

"Madam" is NOT a strict palindrome.

"racecar" is a strict palindrome.

"Was it a car or a cat I saw?" is NOT a strict palindrome.

This function handles:

* Punctuation ✅
* Spaces ✅
* Case sensitivity ✅

1. **Write a function to remove duplicate characters from a string.**

function removeDuplicateCharacters(str) {

let result = '';

for (let char of str) {

if (!result.includes(char)) {

result += char;

}

}

return result;

}

// Example usage:

const input = "banana";

console.log(`Original: ${input}`);

console.log(`Without duplicates: ${removeDuplicateCharacters(input)}`);

**✅ Output:**

Original: banana

Without duplicates: ban

1. **Write a function to capitalize the first letter of each word in a string.**

function capitalizeFirstLetters(input) {

return input

.split(' ')

.map(word => {

if (word.length === 0) return ''; // handles extra spaces

return word[0].toUpperCase() + word.slice(1);

})

.join(' ');

}

// Example usage:

const input = "the quick brown fox jumps over the lazy dog";

console.log("Capitalized:", capitalizeFirstLetters(input));

**✅ Output:**

Capitalized: The Quick Brown Fox Jumps Over The Lazy Dog

## Array Operations

1. Write a program to remove duplicates from an array.
2. Write a function to find the largest number in an array.
3. Write a function that returns the sum of all elements in an array.
4. Write a function to sort an array in ascending order.
5. Write a function to merge two arrays and remove duplicates.

Answer:  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Functions and Scope

1. Write a function that accepts a callback and executes it.
2. Write a recursive function to calculate the factorial of a number.
3. Write a program to demonstrate function hoisting.
4. Write a function to return a function that adds a given number.
5. Demonstrate closure with a counter function.

Answer:  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Object Manipulation

1. Create an object representing a student with name, age, and grade properties.
2. Write a function to count the number of properties in an object.
3. Write a function that checks if a given property exists in an object.
4. Write a function to merge two objects.
5. Write a function to deep clone an object.

Answer:  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Loops and Iteration

1. Write a loop that prints numbers from 1 to 100.
2. Write a program that prints the Fibonacci series up to a given number.
3. Write a function that finds the sum of even numbers from 1 to 50.
4. Write a program to print a multiplication table of a given number.
5. Write a program to find the sum of digits of a given number.

Answer:  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Higher Order Functions

1. Demonstrate the use of the `map()` function on an array.
2. Demonstrate the use of the `filter()` function to filter out even numbers.
3. Demonstrate the use of the `reduce()` function to calculate the total sum of an array.
4. Write a function that takes a function as an argument and applies it to an array.
5. Demonstrate the use of function composition.

Answer:  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## ES6+ Features

1. Demonstrate the use of arrow functions.
2. Demonstrate the use of the spread operator.
3. Demonstrate the use of destructuring in arrays and objects.
4. Write a program using template literals.
5. Demonstrate the use of `let` and `const`.

## Promises and Async/Await

1. Write a function that returns a promise that resolves after 2 seconds.
2. Demonstrate async/await with a simulated API call.
3. Write a function to fetch data from an API using `fetch()`.
4. Write a function to handle promise chaining.
5. Demonstrate error handling in async functions.

## Regular Expressions

1. Write a function to validate an email using regex.
2. Write a function to extract numbers from a string using regex.
3. Write a function to replace spaces with underscores in a string.
4. Write a function to validate a phone number using regex.
5. Write a function to check if a string contains only alphabets.

## DOM Manipulation

1. Write a JavaScript function to change the background color of a webpage.
2. Write a function to dynamically add a paragraph to a webpage.
3. Write a function to remove an element from the DOM.
4. Write a function to toggle a class on a button click.
5. Write a function to update the text content of a div.

## Event Handling

1. Write a function that executes when a button is clicked.
2. Write a program that listens for keypress events.
3. Write a function that prevents the default form submission.
4. Write a function to detect a mouse hover event.
5. Write a program that toggles an element’s visibility when clicked.

## Error Handling

1. Demonstrate the use of try-catch for error handling.
2. Write a function that throws an error if the input is not a number.
3. Demonstrate handling asynchronous errors in promises.
4. Write a function that gracefully handles API failures.
5. Demonstrate using `finally` in a try-catch block.

## Modules and Import/Export

1. Write a module that exports a function and import it in another file.
2. Demonstrate the use of default exports in ES6 modules.
3. Write a program to import multiple functions from a module.
4. Demonstrate using named exports.
5. Show how to use dynamic imports in JavaScript.

## Data Structures

1. Implement a stack using JavaScript arrays.
2. Implement a queue using JavaScript arrays.
3. Write a function that checks if a given string has balanced parentheses.
4. Implement a linked list in JavaScript.
5. Write a function that finds the most frequent element in an array.

## Advanced Topics

1. Write a function that implements memoization.
2. Implement a debounce function in JavaScript.
3. Implement a throttle function in JavaScript.
4. Write a function that deep flattens an array.
5. Implement a custom event emitter.

## Miscellaneous (Continued)

1. Write a function to shuffle an array randomly.
2. Write a function to convert an object into a query string.
3. Write a function to check if a number is a prime number.
4. Implement a function to generate a random alphanumeric string.
5. Write a function to find the intersection of two arrays.
6. Implement a function to count the occurrences of elements in an array.
7. Write a function to convert RGB color values to HEX.
8. Implement a function to debounce a button click event.
9. Write a function to get the current timestamp in different formats.
10. Implement a function to compare two objects for equality.
11. Write a function to get all unique characters from a string.
12. Implement a function that converts a flat array into a nested object.
13. Write a function that validates a credit card number.
14. Implement a function that generates all possible permutations of a string.
15. Write a function to find the longest word in a given sentence.
16. Implement a function to remove falsy values from an array.
17. Write a function to reverse words in a given sentence.
18. Implement a function that converts camelCase to snake\_case.
19. Write a function to chunk an array into smaller arrays of a given size.
20. Implement a function to rotate an array by a given number of positions.
21. Write a function to find the first non-repeating character in a string.