Lab Exercise: Introduction to JavaScript Functions

Background

Functions are a core concept in JavaScript, and they are used to perform specific tasks or operations. In this lab exercise, you will practice defining and using functions in JavaScript.

Objectives

In this lab exercise, you will learn how to:

- Define and call functions in JavaScript
- Use parameters and return values in functions
- Use function expressions and arrow functions
- Define and call functions with default parameter values
- Use default parameter values to simplify function code
- Understand how default parameter values interact with other function parameters
- Define and call functions with rest parameters
- Use rest parameters to handle an indefinite number of arguments
- Understand how rest parameters interact with other function parameters
-

Instructions

1. Define a function called greet that takes a name as a parameter and logs a greeting to the console. For example, calling greet('Alice') should log Hello, Alice! to the console.

2. Define a function called sum that takes two numbers as parameters and returns their sum. For example, calling sum(2, 3) should return 5.

3. Define a function called isEven that takes a number as a parameter and returns true if the number is even, and false if it is odd. You can use the modulo operator (%) to determine if a number is even or odd.

4. Define a function expression called double that takes a number as a parameter and returns its double value. For example, calling double(3) should return 6.

5. Define an arrow function called triple that takes a number as a parameter and returns its triple value. For example, calling triple(3) should return 9.

6. Test your functions by calling them with different values and verifying that they return the expected output. You can log the output to the console or display it in the HTML of the page.

7. Use the sum, double, and triple functions to perform calculations on user input collected from an HTML form. You can use the prompt function to collect input from the user in a pop-up window. For example, you can prompt the user for two numbers and display the sum, double, and triple of the sum of these numbers using your functions.

8. Define an anonymous function that takes a name as a parameter and logs a greeting to the console. Call the function with different names and verify that it logs the expected greeting to the console.

9. Define an arrow function that takes two numbers as parameters and returns the product of the two numbers. Call the function with different values and verify that it returns the expected output.

10. Define a higher-order function called calculator that takes two numbers and a function as parameters. The calculator function should use the provided function to perform a calculation on the two numbers and return the result. For example, calling calculator(2, 3, add) should return 5, while calling calculator(2, 3, (a, b) => a * b) should return 6.

11. Define a higher-order function called repeater that takes a string and a number as parameters, and a function that takes a string as a parameter. The repeater function should call the provided function with the string parameter the specified number of times and return the combined result. For example, calling repeater('Hello', 3, (str) => str.toUpperCase()) should return 'HELLOHELLOHELLO'.

12. Define a function called sum that takes any number of parameters and returns the sum of all the parameters. Use the rest parameter syntax to define the function. For example, calling sum(1, 2, 3) should return 6, while calling sum(4, 5, 6, 7) should return 22.

13. Define a function called max that takes any number of parameters and returns the largest of all the parameters. Use the rest parameter syntax to define the function. For example, calling max(1, 2, 3) should return 3, while calling max(4, 5, 6, 7) should return 7.

14. Define a function called concat that takes any number of string parameters and returns the concatenated string of all the parameters. Use the rest parameter syntax to define the function. For example, calling concat('Hello', ' ', 'World') should return 'Hello World', while calling concat('The', ' ', 'quick', ' ', 'brown', ' ', 'fox') should return 'The quick brown fox'.

15. Define a function called avg that takes any number of parameters and returns the average of all the parameters. Use the rest parameter syntax to define the function. For example, calling avg(1, 2, 3) should return 2, while calling avg(4, 5, 6, 7) should return 5.5.