1. The Word Reverser

Create a function called reverseWord that:

- Takes 1 argument: a word (string).

- Reverses the input word.

- Outputs the reversed word to the screen.

Call the function three times with different words.

2. The Palindrome Checker

Write a function named isPalindrome that:

- Takes 1 argument: a word or phrase (string).

- Checks if the input is a palindrome (reads the same forwards and backwards, ignoring spaces and capitalization).

- Outputs either "The word/phrase is a palindrome!" or "The word/phrase is not a palindrome!" accordingly.

Test the function with at least three different phrases or words.

3. The Even or Odd Number Identifier

Create a function named evenOrOdd that:

- Takes 1 argument: a number.

- Determines if the number is even or odd.

- Outputs "The number is even" or "The number is odd" based on the input.

Test the function with at least three different numbers.

4. The Factorial Calculator

Write a function called calculateFactorial that:

- Takes 1 argument: a positive integer.

- Calculates the factorial of that number.

- Outputs the factorial value.

Call the function with different integers.

5. The Prime Number Checker

Create a function named isPrime that:

- Takes 1 argument: a number.

- Checks if the number is a prime number.

- Outputs "The number is a prime number" or "The number is not a prime number" based on the input.

Test the function with different numbers to check for primality.

### Writing Functions

6. The Sum Calculator

Create a function called calculateSum that:

- Takes any number of arguments (numbers).

- Calculates the sum of all the provided numbers.

- Outputs the total sum.

Test the function with different numbers of arguments.

7. The String Concatenator

Write a function named concatStrings that:

- Takes any number of arguments (strings).

- Concatenates all the input strings into a single string.

- Outputs the concatenated string.

Test the function with different numbers of string arguments.

### Documenting Functions

8. Function Documentation

Write a function called greet that:

- Takes 1 argument: a person's name (string).

- Greets the person with a personalized message.

- Includes proper documentation using comments to explain what the function does and how to use it.

Document the function greet properly.

### Passing Parameters

9. The Shopping Cart Calculator

Create a function named calculateTotal that:

- Takes an array of numbers as an argument (representing prices of items in a shopping cart).

- Calculates the total cost by adding all the prices.

- Outputs the total cost.

Test the function with different arrays of prices.

10. The Sentence Builder

Write a function called buildSentence that:

- Takes an array of strings as an argument (representing words).

- Constructs a sentence by joining all the words in the array with proper spacing.

- Outputs the constructed sentence.

Test the function with various arrays of words.

### Returning Values

11. The Power Calculator

Create a function named calculatePower that:

- Takes 2 arguments: a base number and an exponent.

- Calculates the result of raising the base to the power of the exponent.

- Returns the result.

Call the function with different base and exponent values.

12. The Absolute Difference Finder

Write a function called findAbsoluteDifference that:

- Takes 2 numbers as arguments.

- Calculates the absolute difference between the two numbers.

- Returns the absolute difference.

Test the function with different pairs of numbers.

### Additional Exercises

13. The Array Reverser

Create a function named reverseArray that:

- Takes an array as an argument.

- Reverses the elements of the array.

- Returns the reversed array without modifying the original.

Test the function with different arrays of elements.

14. The Array Sum Calculator

Write a function named sumArray that:

- Takes an array of numbers as an argument.

- Calculates the sum of all the numbers in the array.

- Returns the total sum.

Test the function with various arrays of numbers.

15. The Middle Character Extractor

Create a function named extractMiddle that:

- Takes a string as an argument.

- Determines the middle character(s) of the string.

- Returns the middle character(s).

Test the function with different strings of various lengths.

working with the this keyword in JavaScript is crucial for understanding context within functions and objects. Here are some exercises related to the this keyword:

### Object Method Utilizing this

1. The Person Object

Create an object called person with the following properties:

- name

- age

- greet method that outputs a greeting using the name property.

Use this within the greet method to access the name property.

2. The Counter Object

Create an object called counter with the following properties:

- count starting at 0

- increment method that increments the count property by 1

- decrement method that decrements the count property by 1

Utilize this within the increment and decrement methods to access the count property.

### Constructor Functions and this

3. The Car Constructor

Create a constructor function called Car that:

- Takes parameters for make, model, and year

- Sets these parameters as properties of the object created using this

- Includes a method displayInfo that outputs a string displaying all the car's details using the properties.

Use this to reference the properties within the displayInfo method.

### Event Handlers and this

4. The Button Click Handler

Create an HTML button and attach a click event handler.

Inside the event handler function, use this to refer to the button element and change its text content or style.

Add functionality to the button using this within the event handler.

### Prototype and this

5. The Animal Prototype

Create a constructor function Animal with properties name and sound.

Use the prototype to add a method makeSound that outputs the sound of the animal along with its name.

Use this within the makeSound method to access the name and sound properties.

6. The Person Prototype

Create a constructor function Person with properties name and age.

Use the prototype to add a method introduce that outputs a sentence introducing the person with their name and age.

Utilize this within the introduce method to access the name and age properties.

### Context and Arrow Functions

7. \*The Arrow Function and this\*

Create an object with a regular function method and an arrow function method.

Inside both methods, try to access a property of the object using this.

Observe and note the differences in how this behaves within regular functions and arrow functions.

Feel free to explore and experiment with these exercises to gain a better understanding of how this works in different contexts within JavaScript functions and objects!