Tutorial 3

You may use repl.it (Node.JS template), Visual Studio Code or simply use your browser to work on your exercises. Place completed answers into a word document or zip up your project folder and submit.

Reference:

- Learn about Array
- Learn about Functions
- Learn about Type Coercion

Exercise 1: Add Elements to an Array

Objective

Learn how to add elements to an array using various methods.

Instructions:

- Create an array named fruits and initialize it with some fruit names.
- Add "mango" to the end of the array using the push() method.
- Add "strawberry" to the beginning of the array using the unshift() method.
- Print the final array to the console.

Exercise 2: Remove Elements from an Array

Objective

Practice removing elements from both ends of an array.

Instructions

- Initialise an array named colors with several colors.
- Remove the last element using the pop() method and the first element using the shift() method.
- Print the modified array and the elements removed.

Common Mistakes

- Incorrect variable names when printing removed items.
- Forgetting to store the return value of pop() and shift(), which are the elements removed.

Exercise 3: Reverse Array

Objective

Create a function that is allows one to reverse an array.

Do not use prebuilt reverse functions.

Hint: Think about using loops differently

Exercise 4:Comparing Numbers & Strings

Objective: Identify the difference between == and === when comparing numbers and strings.

Instructions:

- Compare a number and a string with the same value using == and ===.
- Print both results to observe the difference.

Thinking hats

- why == returns true (type coercion occurs, converting the string to a number).
- why === returns false (no type coercion, types are different).

References

- https://developer.mozilla.org/en-US/docs/Glossary/Type_coercion

Exercise 5: Basic Function to Calculate Area

Objective

Write a function that calculates the area of a rectangle.

Instructions

Define a function named calculateArea that takes two parameters: width and height.
Inside the function, compute the area as width * height and return the result.
Call the function with different sets of values to test it and print the results.

Exercise 6: Check Age Group

Objective: Write a function to categorize a person's age into child, teen, or adult.

Instructions

- Define a function named checkAge that takes one parameter: age.
- Use conditional statements to categorize the age and return the category.
- Print the result based on the age.

Exercise 7: Temperature Conversion Function

Objective

Create a function that converts temperatures from Celsius to Fahrenheit.

Instructions

- Define a function named celsiusToFahrenheit that takes one parameter: celsius.
- Convert the Celsius temperature to Fahrenheit using the formula (celsius * 9/5) + 32.
- Return the Fahrenheit temperature.
- Call the function and print the output for several Celsius values.

Exercise 8: Simple Interest Calculator

Objective

Create a function to calculate simple interest given principal, rate, and time.

Instructions

- Define a function named calculateInterest that takes three parameters: principal, rate, and time.
- Calculate the interest using the formula: Interest = (Principal × Rate × Time)/ 100
- Return the interest and print it.

Exercise 9: Odd or Even Checker

Objective

Write a function that checks if a number is odd or even.

Instructions

- Define a function named isOddOrEven that takes one parameter: number.
- Inside the function, use a conditional statement to check if the number is odd or even.
- Return "Odd" or "Even".
- Test the function with different numbers and print the results.

Exercise 10: Calculator Function

Objective

Implement a function that performs basic arithmetic operations.

Instructions

- Define a function named calculator that takes three parameters: operand1, operand2, and operator.
- Inside the function, use a switch or if-else statement to perform the operation based on the operator (+, -, *, /).
- Return the result of the operation.
- Test the function with different operands and operators.

Exercise 11: Convert Minutes into Seconds

Objective

Write a function that converts minutes into seconds.

Instructions

- Define a function named minutesToSeconds that takes one parameter: minutes.
- Calculate the number of seconds in the given minutes and return that value.
- Test the function with a few sample inputs.

Exercise 12: Multiplication Table Generator

Objective

Create a function that prints the multiplication table for a given number up to 10.

Instructions

- Define a function named printTable that takes one parameter: number.
- Use a loop inside the function to generate and print the multiplication table for the number.
- Call the function to display the multiplication table for any number.

Exercise 13: Calculate Circle Area

Objective

Create a function to calculate the area of a circle given the radius.

Instructions

- Define a function named calculateCircleArea that takes the radius as a parameter.
- Use the formula: Area = π × radius^2 to calculate the area.
- Return the area and print the result.

Exercise 14: Validate User Age

Objective

Write a function that checks if a user is eligible to vote (age must be 21 or older).

Instructions

- Define a function named canVote that takes age as a parameter.
- If the age is 21 or older, return true; otherwise, return false.
- Test the function with various ages.

Exercise 15: Find the Largest Number

Objective

Develop a function that returns the largest number from two given numbers.

Instructions

- Define a function named findLargest that takes two parameters, num1 and num2.
- Compare the two numbers and return the larger one.
- Print the result for several pairs of numbers.

Exercise 16: Debugging Challenge

Task

- Identify and fix issues so the function correctly adds two numbers and returns the sum.
- Ensure the function handles different data types appropriately.

```
function addNumbers(a, b) {
    var sum = a + b;
    console.log(sum);
}

addNumbers(10, '20');  // Incorrectly prints "1020" instead of 30
```