MERN Stack Training

Tasks

Task 2:

1. Objects and methods, "this":

Task 1: Create an object named book with properties: title, author, and yearPublished. Add a method named getSummary that returns a string summarizing the book.

Task 2: Modify the book object to include a method named age that calculates how old the book is based on its publication year.

Task 3: Create an object calculator with methods add(), subtract(), multiply(), and divide(). Use the this keyword to refer to the object's properties within these methods.

Task 4: Extend the calculator object to include a history array. Each time a calculation is made, store the operation and result in this array.

Task 5: Create an object representing a person. Add methods to increase and decrease their age property. Ensure the age doesn't go below 0.

Task 6: For the person object, add a method named greet that uses the this keyword to greet with the person's name.

Task 7: Design a circle object with properties radius and a method to calculate its area using this.radius.

Task 8: Extend the circle object with methods to calculate its diameter and circumference.

Task 9: Create an object account with properties: name, balance and methods: deposit, withdraw. Use the this keyword appropriately.

Task 10: For the account object, ensure that the balance can't go negative using the this keyword.

2. Arrays, Array methods (push, pop, shift, unshift):

Task 1: Initialize an array of your favorite fruits. Add "Mango" to the end of the array using push().

Task 2: Remove the last fruit from the array using pop().

Task 3: Add "Strawberry" to the beginning of the fruits array using unshift().

Task 4: Remove the first fruit from the array using shift().

Task 5: Create a function that accepts an array of numbers and uses push() to add the number 7 to it.

Task 6: Write a function that accepts a string. Convert the string to an array of words and remove the last word using pop().

Task 7: Create an array of days of the week. Using shift() and unshift(), move Sunday to the end of the array.

Task 8: Given an array of letters, write a function that adds a letter 'Z' at the beginning and end of the array.

Task 9: Initialize an array with five movie names. Ask the user for another movie name and add it to the end of the array.

Task 10: Remove the third item from the movie array.

3. Additional methods: map(), filter(), reduce(), slice(), splice()

1. map()

The map() method processes each element of the original array using a function you provide, and returns a new array of the same length with the processed results.

Example:
Convert an array of temperatures from Celsius to Fahrenheit:

```
const celsius = [0, 10, 20, 30];
```

```
const toFahrenheit = temp => (temp * 9/5) + 32;

const fahrenheit = celsius.map(toFahrenheit);

console.log(fahrenheit); // [32, 50, 68, 86]
```

---

2. filter()

The filter() method returns a new array that contains only the elements for which the provided function returns true.

Example:
Filter out odd numbers from an array:

```
const numbers = [1, 2, 3, 4, 5];

const evenNumbers = numbers.filter(num => num % 2 === 0);

console.log(evenNumbers); // [2, 4]
```

---

3. reduce()

The reduce() method accumulates the values in an array into a single value based on a function you provide. It essentially “reduces” the array to a single value.

Example:
Find the sum of numbers in an array:

```
const numbers = [1, 2, 3, 4, 5];

const sum = numbers.reduce((accumulator, currentValue) => accumulator + currentValue, 0);

console.log(sum); // 15
```

---

4. slice()

The slice() method returns a shallow copy of a portion of an array without modifying the original array. You can specify a starting index and an end index (not included).

Example:
Get the first three elements of an array:

```
const fruits = ["apple", "banana", "cherry", "date", "fig"];

const someFruits = fruits.slice(0, 3);

console.log(someFruits); // ["apple", "banana", "cherry"]
```

---

5. splice()

The splice() method can add, replace, or remove one or more elements in the original array. It modifies the original array and returns a new array containing the deleted or replaced elements.

Example:
Remove the second and third elements and add two new elements in their place:

```
const colors = ["red", "green", "blue", "yellow"];

colors.splice(1, 2, "cyan", "magenta");

console.log(colors); // ["red", "cyan", "magenta", "yellow"]
```

Task 1: Create an array of numbers. Use map() to create a new array with each number squared.

Task 2: Use filter() on an array of numbers to get a new array with only even numbers.

Task 3: Create an array of product prices. Use reduce() to find the total price.

Task 4: For an array of strings, use map() to create a new array that contains the length of each string.

Task 5: Use splice() to remove the third item of an array and replace it with the string "replaced!".

Task 6: For an array of integers, use slice() to get a new array containing the 2nd, 3rd, and 4th elements.

Task 7: Create an array of names. Use filter() to produce a new array that contains names starting with the letter 'A'.

Task 8: For an array of scores (out of 100), use map() to grade each score (e.g., 90-100 = 'A', 80-89 = 'B').

Task 9: Given an array of ages, use reduce() to find the average age.

Task 10: Use splice() to insert two new fruits after the second fruit in an array of fruits.

Mini Project: "Personal Library"

Objective:
To create a digital representation of a personal book library where users can manage their book collection and perform various operations on it.

---

1. Define the Book Object with methods:

- Properties: title, author, yearPublished, readStatus
- Methods:
    - getSummary: Returns a string summary of the book.
    - toggleReadStatus: Toggles the readStatus of the book using this.

2. Books Collection (Array)

Initialize an empty array library to store book objects.

3. Array Methods:

- Create functions:
    - addBook(book): Adds a new book to the library using push().
    - removeLastBook(): Removes the last book from the library using pop().
    - addBookToFront(book): Adds a book to the beginning of the library using unshift().
    - removeFirstBook(): Removes the first book from the library using shift().

4. Additional Methods:

- Create functions:
    - getAllTitles(): Uses map() to return an array of all book titles in the library.
    - getBooksByAuthor(author): Uses filter() to return books written by a specific author.
    - getTotalBooksPublishedBefore(year): Uses filter() and reduce() to count how many books were published before a given year.
    - removeBookByTitle(title): Uses splice() to remove a book with a specific title from the library.
    - getBooksByReadStatus(status): Uses filter() to return books based on their read status (read or unread).
    - getSubLibrary(start, end): Uses slice() to get a portion of the library.

Project Workflow:

1. Start by defining the book object with its properties and methods.

2. Initialize the library array.
3. Implement the array methods for managing the library.
4. Implement the additional methods to perform various operations on the library.

Bonus: You can implement user interaction using prompt and alert to allow users to manually add, remove, or search books in their library.