Content

1. Basics of JavaScript

- **Start with Syntax**: Learn how to write JavaScript code.
  - Variables: `var`, `let`, `const`.
  - Data types: Numbers, Strings, Booleans, Objects, Arrays.
  - Operators: Arithmetic, Comparison, Logical.
- **Example Task**: Create a script to display a welcome alert when a page loads.

2. DOM Manipulation

- Learn how to select and manipulate HTML elements.
  - Methods: `getElementById`, `querySelector`.
  - Events: `onclick`, `onmouseover`, `onkeyup`.
- **Example Task**: Create a button that changes the background color of the page when clicked.

3. JavaScript Functions

- Understand function declarations and expressions.
- Learn about parameters and return values.
- **Example Task**: Write a function to calculate the area of a rectangle based on user input.

4. Loops and Conditional Statements

- `if`, `else`, `else if`, and `switch`.
- Loops: `for`, `while`, `do...while`.
- **Example Task**: Display numbers from 1 to 10 using a loop.

5. Working with Arrays and Objects

- Array methods: `push`, `pop`, `map`, `filter`.
- Object basics: keys, values, and methods.
- **Example Task**: Create a to-do list app where tasks can be added and removed.

6. Asynchronous JavaScript

- Understand `setTimeout` and `setInterval`.
- Basics of Promises and `async/await`.
- **Example Task**: Simulate fetching data from a server using `setTimeout`.

7. Modern JavaScript Features

- Learn ES6+ concepts: Arrow functions, Template literals, Destructuring, Spread/rest operators.
- **Example Task**: Rewrite a traditional function using arrow syntax.

8. Projects

- Start with small projects to practice:
    - Calculator.
    - Weather App (using an API like OpenWeather).
    - Quiz Game.
- Use frameworks like Bootstrap to style your apps.

Practice Set I

Section 1: Basics

1. **Variables and Data Types**
   - Declare a variable to store your name and log it to the console.
   - Create a variable to hold your age. Increase it by 1 and log the new value.
2. **Operators**
   - Write a program that calculates the area of a rectangle (length × width).
   - Write a program to swap two variables without using a third variable.

Section 2: Control Flow

3. **Conditionals**
   - Write a program that checks if a number is odd or even.
   - Create a program to check if a person is eligible to vote (age >= 18).
4. **Loops**
   - Print numbers from 1 to 10 using a `for` loop.
   - Write a program to calculate the sum of numbers from 1 to 50.

Section 3: Functions

5. **Basic Functions**
   - Write a function `greet(name)` that returns a greeting message for a given name.
   - Create a function to calculate the square of a number.
6. **Arrow Functions**
   - Convert the `greet(name)` function into an arrow function.
   - Write an arrow function to check if a number is positive, negative, or zero.

Section 4: Arrays

7. **Array Operations**
   - Create an array of your five favorite colors and log them.
   - Write a program to find the largest number in an array of integers.
8. **Array Methods**
   - Use the `.map()` method to create a new array with the square of each number from `[1, 2, 3, 4, 5]`.
   - Use the `.filter()` method to extract all numbers greater than 10 from `[5, 8, 15, 22, 1, 3]`.

Section 5: Objects

9. **Object Basics**
    - Create an object to represent a car with properties like `brand`, `model`, and `year`. Log the car's details.
    - Write a program to update the `year` property of the car object.
10. **Object Methods**
    - Add a method `start()` to the car object that logs "The car has started!" when called.
    - Create a method to calculate the car's age based on the current year.

Section 6: DOM Manipulation

11. **Selecting and Modifying Elements**
    - Write a script to change the text of an HTML element with an ID of `#greeting` to "Hello, World!".
    - Change the background color of a `<div>` element when a button is clicked.
12. **Event Listeners**
    - Write a program to display an alert when a button is clicked.
    - Add an event listener to an input field that logs its value every time it changes.

Section 7: Mini Projects

13. **Simple Calculator**
    - Create a basic calculator with buttons for addition, subtraction, multiplication, and division.
14. **To-Do List**
    - Build a to-do list that allows users to add, remove, and mark tasks as completed.
15. **Guess the Number Game**
    - Write a program that generates a random number between 1 and 100. The user has to guess the number, with feedback provided for "higher" or "lower" guesses.

Practice Set II

Section 1: Basics

1. **String Operations**
    - Write a program to reverse a given string (e.g., `"hello"` becomes `"olleh"`).
    - Count the number of vowels in a given string.
2. **Math Operations**
    - Write a program to find the greatest of three numbers.
    - Create a program to calculate the factorial of a number using a loop.

Section 2: Control Flow

3. **Nested Loops**
    - Write a program to print the following pattern:

    - Create a program to display a multiplication table for a number input by the user.
4. **Switch Statements**
    - Write a program to take a number (1-7) as input and display the corresponding day of the week.
    - Create a calculator using `switch` to perform addition, subtraction, multiplication, or division based on the user's choice.

Section 3: Functions

5. **Intermediate Functions**
    - Write a function `isPalindrome(string)` to check if a string is a palindrome (reads the same forwards and backward).
    - Create a function that converts Celsius to Fahrenheit and vice versa.
6. **Default Parameters**

- Write a function that takes two numbers and returns their product. If only one number is provided, multiply it by 10 (use default parameters).

Section 4: Arrays

7. **Array Challenges**
   - Write a program to merge two arrays and remove duplicates.
   - Sort an array of numbers in ascending and descending order.
8. **Array Iteration**
   - Use the `.reduce()` method to calculate the sum of numbers in an array.
   - Write a program to find the index of the first occurrence of a given number in an array.

Section 5: Objects

9. **Object Challenges**
   - Write a program to create an object representing a student with properties like `name`, `age`, and `grades`. Add a method to calculate the average grade.
   - Write a function that takes an array of objects (e.g., `[{name: "A", age: 25}, {name: "B", age: 20}]`) and returns the name of the oldest person.
10. **Object Destructuring**
    - Given the object `{a: 10, b: 20, c: 30}`, use destructuring to extract its properties into variables.
    - Use destructuring in a function parameter to extract specific properties from an object.

Section 6: DOM Manipulation

11. **Dynamic Content**
    - Write a program to add five new list items to an unordered list on a webpage.
    - Create a script that hides or shows an element when a button is clicked.
12. **Form Validation**
    - Write a script to validate a form that requires a user's name and email. Display an error message if the fields are empty.
    - Ensure an input field only accepts numbers between 1 and 100.

Section 7: Mini Projects

13. **Random Quote Generator**
    - Create a program that displays a random quote from an array of quotes when a button is clicked.

14. **Stopwatch**
    - Build a simple stopwatch with buttons to start, stop, and reset the timer.
15. **Rock-Paper-Scissors**
    - Write a program to let the user play Rock-Paper-Scissors against the computer. The computer's choice should be random.