Js-opg loops, arrays, objects, functions

Phase 1: The Basics (1–10)

Goal: Master syntax and accessing data.

1. **Array Access:** Create an array `fruits` containing: `"Apple", "Banana", "Cherry", "Date", "Elderberry"`.

- **Task:** Log the 1st and 5th items.
- **Expected Output:** `Apple`, `Elderberry`

2. **Object Properties:** Create an object `book` with: `title: "The Hobbit"`, `author: "J.R.R. Tolkien"`, and `pages: 310`.

- **Task:** Log a sentence: "[Title] was written by [Author]."
- **Expected Output:** `The Hobbit was written by J.R.R. Tolkien.`

3. **Array Growth:** Create an empty array `colors`.

- **Task:** Use `.push()` to add `"Red"`, then `"Green"`, then `"Blue"`. Log the final array.
- **Expected Output:** `['Red', 'Green', 'Blue']`

4. **Property Update:** Use the `book` object from Exercise 2.

- **Task:** Change `pages` to `320` and add a new boolean property `isRead: true`. Log the object.
- **Expected Output:** `{ title: 'The Hobbit', author: 'J.R.R. Tolkien', pages: 320, isRead: true }`

5. **Basic Function:** Write a function `multiply(a, b)` that returns the product.

- **Task:** Log the result of `multiply(6, 7)`.
- **Expected Output:** `42`

6. **The `for` Loop:** Create an array `numbers` with: `10, 20, 30, 40, 50`.

- **Task:** Loop through and log each number plus 5.
- **Expected Output:** `15, 25, 35, 45, 55` (on separate lines)

7. **Object Methods:** Create an object `calculator`. Add a method `add(a, b)` that returns the sum.

- **Task:** Log `calculator.add(15, 25)`.
- **Expected Output:** `40`

8. **Array Length:** Create an array `guests` with 4 names.

- **Task:** Write a function `checkGuestCount(arr)` that returns the array length.
- **Expected Output:** `4`

9. **Modulo Logic:** Write a function `isEven(num)`.

- **Task:** Log the result of `isEven(8)` and `isEven(7)`.
- **Expected Output:** `true`, `false`

10. **Removing Elements:** Create an array `tasks` with `"Email", "Call", "Code"`.

- **Task:** Remove the last item. Log the array.
- **Expected Output:** `['Email', 'Call']`

Phase 2: Intermediate (11–20)

Goal: Combining loops with objects and basic logic.

11. **Array of Objects:** Create an array `users` with 3 objects. Each has `name` (string) and `age` (number).
12. **Filtering Loop:** Use the `users` array from #11.

- **Task:** Loop through and log only the names of users where `age > 18`.

13. **Mass Update:** Create an array `prices` with `[100, 200, 300]`.

- **Task:** Use a `for` loop to create a *new* array `discounted` where each price is 10% off. (Do not use `.map()`).
- **Expected Output:** `[90, 180, 270]`

14. **Find the Max:** Create an array `scores` with `[45, 82, 94, 32, 78]`.

- **Task:** Use a loop to find the highest score.
- **Expected Output:** `94`

15. **The Shopping Cart:** Create an array `cart` with 3 objects: `{item: "Bread", price: 20}`, `{item: "Milk", price: 15}`, `{item: "Cheese", price: 50}`.

- **Task:** Calculate the total sum of all prices using a loop.
- **Expected Output:** `85`

16. **Nested Access:** Create an object `classroom` with `students: ["Alice", "Bob", "Charlie"]` and `teacher: "Mr. Jones"`.

- **Task:** Log the second student's name.
- **Expected Output:** `Bob`

17. **String Manipulation:** Write a function `shoutAll(arr)` that takes an array of 3 strings and returns a new array with all strings in UPPERCASE.
18. **Search Function:** Create an array `inventory` with `"Screwdriver", "Hammer", "Drill"`.

- **Task:** Write a function `hasItem(arr, item)` that returns `true` if the item exists.

19. **Object Keys:** Create an object `stats` with `strength: 10, agility: 15, intelligence: 20`.

- **Task:** Use `Object.keys()` and a loop to log each key name.

20. **Conditional Array:** Write a function that takes an array of numbers `[1, -2, 3, -4]` and returns a new array with only the positive numbers.

- **Expected Output:** `[1, 3]`

Phase 3: Advanced (21–30)

Goal: Nested arrays, complex data, and transformation.

21. **2D Array (Matrix):** Create a 3x3 matrix: `[[1,2,3], [4,5,6], [7,8,9]]`.

- **Task:** Log the number `5` by accessing the correct indices.

22. **Matrix Total:** Use nested `for` loops to sum all numbers in the 3x3 matrix from #21.

- **Expected Output:** `45`

23. **Object Transformation:** You have an array `players`: `[{id: 1, score: 10}, {id: 2, score: 20}]`.

- **Task:** Create a new array where each object also has `active: true`.

24. **Category Filter:** Create an array `products` (4 items) with `name` and `category` (either "Fruit" or "Veggie").

- **Task:** Create a loop that builds a new array containing only the "Fruit" objects.

25. **The Dictionary:** Create an object `translate` with `apple: "manzana"`, `hello: "hola"`.

- **Task:** Write a function that takes a string and returns the translation if it exists, or "Not found".

26. **Deep Nesting:** Create an object `user` with `profile: { settings: { theme: "Dark" } }`.

- **Task:** Log the theme and then update it to "Light".

27. **Array to Object:** Take `['id_101', 'id_102', 'id_103']`.

- **Task:** Turn this into an object where each ID is a key and the value is `null`.
- **Expected Output:** `{ id_101: null, id_102: null, id_103: null }`

28. **Frequency Counter:** Create an array `votes`: `['Yes', 'No', 'Yes', 'Yes', 'No']`.

- **Task:** Use an object to count the results.
- **Expected Output:** `{ Yes: 3, No: 2 }`

29. **Sorting Logic (Manual):** Create an array `unsorted = [5, 2, 9, 1]`.

- **Task:** Use `.sort()` to order them ascending. Then do the same for an array of objects by a `price` property.

30. **The Grand Finale:** Create an array `company` containing 2 `department` objects. Each department has a `name` and an array of `employees`. Each employee is an object with `name` and `salary`.

- **Task:** Use nested loops to calculate the **total salary** of the entire company.