Java script practice Questions

To become a truly skilled developer, don't rely on ChatGPT; instead, build your critical thinking muscle.

1. Filter Even Numbers – Fitness App

Scenario:

You are building a reporting feature for a fitness app that tracks the number of steps a user walks each day over a week. The app stores this weekly data in an array of numbers, where each number represents the steps taken on a given day.

Task:

Use a loop to filter and display only the days where the user walked an even number of steps.

Input Example:

```
let steps = [3450, 4220, 3891, 5000, 6190, 5800, 7051];
```

Expected Output:

```
4220
5000
6190
5800
```

---

2. Count Vowels – Text Analysis Tool

Scenario:

You are working on a text analysis tool for a writing platform. When a user submits a paragraph, the system should analyze how many vowels are used — which can be a proxy for readability.

Task:

Declare a string. Loop through it and count how many vowels (`a, e, i, o, u`) are present.

Input Example:

```
let text = "Education is the passport to the future.";
```

Expected Output:

```
Number of vowels: 14
```

---

3. Positive Numbers – Finance Report

Scenario:

You are working on a personal finance dashboard. The app receives an array of transaction amounts, where negative values represent expenses and positive values represent income. You need to filter out all the positive amounts (income).

Task:

Loop through the transaction list and print only the income values.

Input:

```
let transactions = [-500, 2000, -150, 7000, -300, 400];
```

Expected Output:

```
2000
7000
400
```

---

4. Count Characters – Username Validator

Scenario:

Your app allows usernames with up to 15 characters. Write code to count and print how many characters are in the given username.

Input:

```
let username = "theCodeWarrior";
```

Expected Output:

```
Character count: 14
```

---

5. Replace All Dashes – Clean Up Input

Scenario:

You receive user input in a URL format that uses dashes (`-`) instead of spaces. Your job is to clean it up for display by replacing dashes with spaces.

Input:

```javascript
let urlTitle = "learn-javascript-by-building-projects";
```

Expected Output:

```
learn javascript by building projects
```

---

6. Extract Digits – Chat App

Scenario:

A chat moderation tool needs to extract numbers from messages to detect potential phone numbers or OTPs.

Task:

Print only the digits from a string.

Input:

```javascript
let message = "Hey my number is 987654 and OTP is 1234";
```

Expected Output:

```
9876541234
```

---

7. Count Words – Essay Scorer

Scenario:

In an online exam portal, you want to count how many words a student wrote in an essay.

Task:

Split the string by space and count the number of words.

Input:

```
let essay = "JavaScript is fun and powerful.";
```

Expected Output:

```
Word count: 5
```

---

8. Filter Scores > 80 – Student Result Analyzer

Scenario:

A teacher wants to identify students who scored above 80 in a test.

Task:

Print only the scores that are greater than 80.

Input:

```
let scores = [45, 67, 82, 90, 88, 76];
```

Expected Output:

```
82
90
88
```

---

9. Find Longest Word – Text Utility

Scenario:

You are building a blogging platform. Highlight the longest word in the title to suggest better SEO keywords.

Task:

Find and print the longest word.

Input:

```
let title = "Build modern fullstack web apps";
```

Expected Output:

```
fullstack
```

---

10. Count Uppercase Letters – Email Formatter

Scenario:

Emails should not be written in all caps. Your formatter warns users if there are too many capital letters.

Task:

Count how many uppercase letters are present.

Input:

```
let email = "HELLO This is A Test EMAIL";
```

Expected Output:

```
Uppercase count: 10
```

---

11. Pattern – Increasing Stars

```
// Input: 5
// Output:
*
**
***
****
*****
```

12. Pattern – Decreasing Stars

```
// Input: 5
// Output:
*****
****
***
**
*
```

13. Pattern – Right Aligned Triangle

```
// Input: 4
// Output:
   *
  **
 ***
****
```

14. Pattern – Number Triangle

```
// Input: 4
// Output:
1
12
123
1234
```

15. Pattern – Reverse Number Triangle

```
// Input: 4
// Output:
1234
123
12
1
```

16. Pattern – Alphabet Triangle

```
// Input: 4
// Output:
A
AB
ABC
ABCD
```

17. Pattern – Square of Stars

```
// Input: 3
// Output:
***
***
***
```

18. Pattern – Hollow Square

```
// Input: 4
// Output:
****
*  *
*  *
****
```

19. Pattern – Pyramid

```
// Input: 3
// Output:
  *
 ***
*****
```

20. Pattern – Inverted Pyramid

```
// Input: 3
// Output:
```

```
*****
 ***
  *
```

21. Pattern – Diamond

```
// Input: 3
// Output:
  *
 ***
*****
 ***
  *
```

22. Pattern – Left Half Diamond

```
// Input: 3
// Output:
*
**
***
**
*
```

23. Pattern – Pascal Triangle (basic format)

```
// Input: 5
// Output:
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

24. Pattern – Floyd's Triangle

```
// Input: 4
// Output:
1
2 3
4 5 6
7 8 9 10
```

25. Pattern – Binary Triangle

```
// Input: 4
// Output:
1
0 1
1 0 1
0 1 0 1
```

26. Pattern – Hollow Right Triangle

```
// Input: 5
// Output:
*
**
* *
*  *
*****
```

27. Pattern – Hollow Pyramid

```
// Input: 4
// Output:
   *
  * *
 *   *
*******
```

28. Pattern – X Pattern

```
// Input: 5
// Output:
*   *
 * *
  *
 * *
*   *
```

29. Pattern – Z Pattern

```
// Input: 5
// Output:
*****
   *
  *
 *
*****
```

30. Pattern – Number Pyramid

```
// Input: 3
// Output:
  1
 121
12321
```