

JavaScript Practice Questions

If Else

Beginner

- What is the purpose of the if statement in JavaScript?
- Write a simple if statement that checks if a variable age is greater than 18.
- How do you execute code when an if condition is false? Provide an example.
- Explain the difference between if and else in JavaScript.

Intermediate

- Compare and contrast the use of if versus else if in JavaScript decision-making.
- Give an example of nested if statements in JavaScript.
- How can you combine multiple conditions using logical operators (&&, ||) in an if statement?
- Discuss the concept of truthy and falsy values in JavaScript conditions.

Hard

- Explain short-circuit evaluation in the context of if statements. Provide an example.
- Describe scenarios where you might use the ternary operator (condition ? expr1 : expr2) instead of a regular if-else statement.

If Else If Else

Beginner

- What is the syntax for an if-else if-else statement in JavaScript?
- Write an example of an if-else if-else statement that checks the value of a variable grade.
- How many conditions can you chain together using else if in JavaScript?
- What happens if none of the conditions in an if-else if-else statement are true?

Intermediate

- Explain the concept of 'fall-through' in if-else if-else statements. How can you prevent it?
- Compare the performance implications of using multiple else if versus a switch statement.
- Discuss best practices for structuring complex if-else if-else logic.

Hard

- Describe the difference between using if-else if-else versus a switch statement. When would you choose one over the other?
- How does JavaScript handle ambiguities or conflicts in conditions within an if-else if-else chain?
- Implement a scenario where you would prefer using a chain of if-else if-else statements over a switch statement.

Switch

Beginner

- What is the purpose of the switch statement in JavaScript?
- Write a simple switch statement that checks the value of a variable day and logs a message based on whether it's a weekday or weekend.
- How do you handle default cases in a switch statement?
- Explain the difference between using break and return inside a switch statement.

Intermediate

- Discuss scenarios where you might prefer using a switch statement over multiple if-else statements.
- Can you have multiple case values in a switch statement? If so, how?
- What are some common pitfalls or mistakes to avoid when using switch statements in JavaScript?

Hard

- Explain how fall-through works in a switch statement. Provide an example.
- Implement a complex scenario where a switch statement is the most efficient choice compared to other decision-making structures.
- Discuss the efficiency considerations when using switch statements with large numbers of cases.

for Loops

Beginner

- What is the syntax for a for loop in JavaScript?
- Write a simple for loop that counts from 1 to 5 and logs each number to the console.
- How do you iterate over an array using a for loop?
- Explain the purpose of the initialization, condition, and iteration expressions in a for loop.

Intermediate

- Compare the for loop with the for...in loop and for...of loop in JavaScript.
- Discuss the concept of looping through objects using for...in versus for...of loops.
- How can you optimize a for loop for performance when iterating over a large array?

Hard

- Describe the concept of 'loop unrolling' and its potential benefits in JavaScript for loops.
- Implement a scenario where you would choose a for loop over forEach or map methods for iterating over an array.
- Discuss asynchronous operations and their implications when using for loops in JavaScript.

while Loops

Beginner

- What is the syntax for a while loop in JavaScript?

- Write a simple while loop that counts from 1 to 5 and logs each number to the console.
- How do you ensure that a while loop terminates? Provide an example.
- Explain the difference between while and do-while loops.

Intermediate

- Discuss scenarios where you would choose a while loop over a for loop in JavaScript.
- How can you avoid infinite loops when using while loops? What precautions should you take?
- Explain how you can use a while loop to simulate asynchronous behavior in JavaScript.

Hard

- Implement a scenario where a while loop is more suitable than a for loop or recursion.
- Discuss performance considerations when using while loops for iterative operations compared to other looping constructs.
- Describe the role of condition evaluation in optimizing while loop performance.

do-while Loops

Beginner

- What is the syntax for a do-while loop in JavaScript?
- Write a simple do-while loop that prompts the user for input until they enter a valid number.
- How does a do-while loop differ from a while loop in terms of execution order?
- When should you use a do-while loop instead of other types of loops?

Intermediate

- Discuss scenarios where the use of a do-while loop enhances code readability and maintainability.
- How can you refactor a do-while loop into a more efficient construct, if possible?
- Compare and contrast error handling strategies between do-while and while loops.

Hard

- Implement a scenario where the exit condition of a do-while loop depends on external state changes.
- Describe potential performance implications of using do-while loops versus other looping techniques.
- Discuss the use of do-while loops in asynchronous JavaScript programming.

Functions

Beginner

- What is a function in JavaScript, and why is it useful?
- Write a basic function named greet that takes a name parameter and logs 'Hello, {name}!' to the console.
- How do you call a function in JavaScript? Provide an example.
- Explain the difference between parameters and arguments in the context of JavaScript functions.

Intermediate

- Discuss the concept of function scope in JavaScript. How do var, let, and const affect function scope?
- Compare named functions and anonymous functions in JavaScript. When would you use each?
- How can you return a value from a function in JavaScript? Provide examples of using return.

Hard

- Explain the concept of higher-order functions in JavaScript. Provide an example.
- Discuss the role of closures in JavaScript functions. How do they contribute to maintaining state?
- Implement a scenario where you would use recursion instead of iteration within a function.