



LEARN JAVASCRIPT

 JavaScript Enthusiasts, It's

Quiz Time! 

JavaScript Control Structures

JavaScript Control Structures	1
JavaScript Control Structures	2
1. Conditional Statements	2
Example: Using if, else if, else	2
2. Looping Structures	3
Example: Using a for loop	3
Example: Using a while loop	3
3. Error Handling	4
Example: Using try...catch	4
Quiz Questions and Answers	4
Q1: What is the purpose of the else if clause in JavaScript?	4
Q2: How does a while loop start?	4
Q3: In a try...catch statement, what is the role of the catch block?	5
Q4: What will the following code output?	5
Q5: Which statement is true about the break keyword in loops?	6

Whether you're a seasoned developer or just starting out, this quiz is a fantastic way to test your knowledge on JavaScript's control structures, a crucial aspect of

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

coding. Are you ready to assess your understanding of if-else statements, loops, and error handling in JavaScript? 🤔

🔍 Here's a glimpse of what to expect:

How well do you know your loops – for, while, and do...while?

Can you handle the intricacies of conditional statements?

Are you a master at error handling with try...catch?

🌟 No code snippets, just pure logic, and understanding. Dive into the fundamentals and see how much you really know about controlling the flow of your JavaScript programs. 🌊

JavaScript Control Structures

JavaScript control structures are fundamental for directing the flow of a program. They include conditional statements like if, else if, else, loops like for, while, and do...while, and error handling with try...catch.

1. Conditional Statements

Example: Using if, else if, else

```
let number = 5;
if (number > 10) {
  console.log("Number is greater than 10");
} else if (number < 10) {
  console.log("Number is less than 10");
} else {
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
    console.log("Number is exactly 10");  
}
```

Explanation: This code checks a number against different conditions and executes the corresponding code block.

2. Looping Structures

Example: Using a for loop

```
for (let i = 0; i < 5; i++) {  
    console.log(i);  
}
```

Explanation: This for loop prints numbers from 0 to 4. It initializes a variable *i*, checks a condition, and increments *i* in each iteration.

Example: Using a while loop

```
let i = 0;  
while (i < 5) {  
    console.log(i);  
    i++;  
}
```

Explanation: This loop does the same as the for loop but using a while structure. The code inside the loop runs as long as the condition is true.

3. Error Handling

Example: Using try...catch

```
try {  
  let result = someFunctionThatMightFail();  
  console.log(result);  
} catch (error) {  
  console.log("An error occurred:", error.message);  
}
```

Explanation: This structure allows handling exceptions gracefully. If an error occurs in the try block, it is caught in the catch block.

Quiz Questions and Answers

Q1: What is the purpose of the else if clause in JavaScript?

- A) To specify a new condition if the first condition is false
- B) To end an if statement
- C) To repeat a block of code multiple times

Answer: A) To specify a new condition if the first condition is false

Q2: How does a while loop start?

A) while (condition) { ... }

B) while { ... } (condition);

C) while execute { ... }

Answer: A) while (condition) { ... }

Q3: In a try...catch statement, what is the role of the catch block?

A) To execute a block of code if an error occurs in the try block

B) To test a block of code for errors

C) To display an error message regardless of an exception

Answer: A) To execute a block of code if an error occurs in the try block

Q4: What will the following code output?

```
for (let i = 0; i <= 5; i++) {  
  console.log(i);  
}
```

A) It will print numbers from 0 to 4

B) It will print numbers from 1 to 5

C) It will print numbers from 0 to 5

Answer: C) It will print numbers from 0 to 5

Q5: Which statement is true about the break keyword in loops?

- A) It terminates the current loop and resumes execution at the next statement
- B) It skips the current iteration of the loop
- C) It restarts the loop from the beginning

Answer: A) It terminates the current loop and resumes execution at the next statement

Understanding these control structures in JavaScript is essential for creating programs that make decisions, repeat operations, and handle errors effectively.