



# LEARN JAVASCRIPT






Dive into the World of JavaScript!




## Basic JavaScript Syntax and Fundamentals

<b>Basic JavaScript Syntax and Fundamentals</b>	<b>2</b>
<b>Quiz Questions and Answers</b>	<b>5</b>
<b>More Quiz Questions and Answers</b>	<b>6</b>

 We explored various aspects of JavaScript, from declaring variables with `let` and `const` to the nuances of the `this` keyword. Understanding these elements is crucial for any web developer's toolkit. 

 Did you know that the way you declare a variable can impact its scope and reusability? Or that the `this` keyword in JavaScript can change its meaning based on the context in which it's used? These are just a few of the intriguing facets we discussed.

 JavaScript isn't just about writing code; it's about understanding the principles that make your code efficient, readable, and adaptable. It's these fundamentals that empower developers to build more robust and interactive web applications.

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

💡 Whether you're debugging a tricky piece of code or optimizing a web application, a solid grasp of JavaScript basics can make a world of difference.

📖 Continuous learning is the key to staying ahead in the ever-evolving world of technology. So, keep exploring, keep experimenting, and most importantly, keep learning!

## Basic JavaScript Syntax and Fundamentals

### 1. Declaring a JavaScript Variable that Cannot be Re-assigned

- Example: `const PI = 3.14;`
- Explanation: The `const` keyword declares a read-only named constant. Once a constant is declared and assigned, its value cannot be changed.

### 2. Declaring a JavaScript Variable that Can Change in Value

- Example: `let age = 25;`
- Explanation: The `let` keyword declares a block-scoped variable, optionally initializing it to a value. `let` variables can be reassigned.

### 3. Declaring a Constant in JavaScript

- Example: `const MAX_USERS = 100;`
- Explanation: Similar to the first example, this declares a constant whose value cannot be changed.

### 4. Declaring a JavaScript Variable

- Example: `var userName = 'Alice';`
- Explanation: The `var` keyword declares a function-scoped or globally-scoped variable, optionally initializing it to a value.

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

## 5. Adding a Comment in JavaScript

- Example: `// This is a single line comment`
- Explanation: Using `//` will comment out the rest of the line in JavaScript, making it ignored by the interpreter.

## 6. Writing an if Statement in JavaScript

```
if (age > 18) {  
  console.log('You are an adult.');
```

Explanation: This if statement checks if age is greater than 18. If true, it executes the code block within the curly braces.

## 7. Starting a For Loop

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

Explanation: This loop starts with `i` at 0, and as long as `i` is less than 5, it will execute the code block and then increment `i`.

## 8. Writing a Loop that Repeats 5 Times

- Example: Similar to the for loop example above.

## 9. Creating a Function in JavaScript

```
function greet(name) {  
  return `Hello, ${name}!`;  
}
```

Explanation: This function `greet` takes a parameter `name` and returns a greeting string.

## 10. Defining an Anonymous Function

```
let greet = function(name) {  
  return `Hello, ${name}!`;  
};
```

Explanation: This is an anonymous function (no name) assigned to the variable `greet`. It behaves like the named function in the previous example.

## 11. Correct Way to Write an Array in JavaScript

- Example: `let fruits = ['apple', 'banana', 'cherry'];`
- Explanation: This creates an array `fruits` containing three elements.

## 12. Creating an Object in JavaScript

```
let person = {  
  name: 'Alice',  
  age: 25  
};
```

Explanation: This is an object `person` with two properties: `name` and `age`.

## 13. The `this` Keyword in JavaScript

```
let person = {  
  name: 'Alice',  
  greet: function() {  
    return `Hello, I am ${this.name}`;  
  }  
};
```

Explanation: Here, `this` refers to the `person` object. `this.name` is the `name` property of the `person`.

## Quiz Questions and Answers

Q1: What will happen if you try to change the value of a constant in JavaScript?

- A) The value will change
- B) An error will be thrown
- C) The value will become undefined

Answer: B) An error will be thrown

Q2: How do you write a loop that runs 3 times in JavaScript?

- A) for (i < 3; i++) { ... }
- B) for (let i = 0; i < 3; i++) { ... }
- C) for (let i = 1; i <= 3; i++) { ... }

Answer: B) for (let i = 0; i < 3; i++) { ... }

Q3: Which of the following is not a valid way to declare a variable in JavaScript?

- A) let x = 1;
- B) const y = 2;
- C) integer z = 3;

Answer: C) integer z = 3;

Q4: What is the output of this code?

```
let fruits = ['apple', 'banana', 'cherry'];
```

```
console.log(fruits[1]);
```

- A) apple
- B) banana
- C) cherry

Answer: B) banana

Q5: What does the this keyword refer to inside an object method in JavaScript?

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

- A) The global window object
- B) The method itself
- C) The object the method is a part of

Answer: C) The object the method is a part of

Explanation (continued): In JavaScript, this generally refers to the object it belongs to. In the context of a method, this refers to the owner object. However, its value can change based on how a function is called (regular call, method call, constructor call, etc.).

## More Quiz Questions and Answers

Q6: If you declare a variable with `let` inside a block (like a loop), where is it accessible?

- A) Anywhere in the function containing the block
- B) Only within the block it was declared in
- C) Globally throughout the entire script

Answer: B) Only within the block it was declared in

Q7: What does the following function return when called?

```
function checkNumber(num) {  
  if (num > 10) {  
    return "Greater than 10";  
  }  
  return "10 or Less";  
}
```

- A) "Greater than 10" for all numbers
- B) "10 or Less" for all numbers
- C) "Greater than 10" for numbers greater than 10, and "10 or Less" for others

Answer: C) "Greater than 10" for numbers greater than 10, and "10 or Less" for others

Q8: What is the correct syntax for an anonymous function that adds two numbers?

- A) function add(a, b) { return a + b; }
- B) let add = function(a, b) { return a + b; };
- C) let add(a, b) = { return a + b; };

Answer: B) let add = function(a, b) { return a + b; };

Q9: How do you access the second element of an array named colors?

- A) colors[1];
- B) colors[2];
- C) colors.get(1);

Answer: A) colors[1];

Q10: What does the following code snippet do?

```
let person = {  
  name: 'Alice',  
  greet: function() {  
    return `Hello, I am ${this.name}`;  
  }  
};
```

```
console.log(person.greet());
```

- A) Prints "Hello, I am Alice"
- B) Returns a function
- C) Throws an error

Answer: A) Prints "Hello, I am Alice"