

# JavaScript

---

## JavaScript Basics

### 1. What is JavaScript?

JavaScript is a scripting language used to create dynamic web applications and automate browser actions in testing.

### 2. Difference between var, let, and const?

var is function-scoped, let and const are block-scoped. const cannot be reassigned.

### 3. What are data types in JavaScript?

Primitive: string, number, boolean, null, undefined, symbol, bigint.

Non-primitive: object, array, function.

### 4. What is undefined vs null?

undefined means a variable is declared but not assigned.

null is an deliberate empty value.

### 5. What is typeof operator?

It returns the data type of a variable.

---

## Functions & Scope

### 6. What is a function in JavaScript?

A reusable block of code used to perform tasks.

### 7. What is arrow function?

A shorter syntax for writing functions using =>.

### 8. What is scope?

Scope defines where variables are accessible (global, function, block).

### 9. What is closure?

A closure is when a function remembers variables from its outer scope.

### 10. What is hoisting?

JavaScript moves variable and function declarations to the top of their scope during execution.

---

## Objects & Arrays

### 11. What is an object?

A collection of key-value pairs used to store structured data.

### 12. How do you create an array?

Using [] or new Array().

### **13. Difference between map and forEach?**

map() returns a new array; forEach() does not.

### **14. What is push() and pop()?**

push() adds an element; pop() removes the last element.

### **15. What is JSON?**

A lightweight format used to exchange data in API testing.

---

## **Async JavaScript**

### **16. What is asynchronous JavaScript?**

Code that runs without blocking execution (e.g., API calls).

### **17. What is a Promise?**

An object representing success or failure of an async operation.

### **18. What are async/await?**

Syntax to handle promises in a synchronous-looking way.

### **19. What is callback function?**

A function passed into another function to run later.

### **20. What is fetch()?**

A method used to make HTTP requests.

---

## **DOM & Browser**

### **21. What is DOM?**

Document Object Model represents HTML elements as objects.

### **22. How do you select an element?**

Using querySelector() or getElementById().

### **23. What is event handling?**

Responding to user actions like clicks or typing.

### **24. What is event bubbling?**

Events propagate from child to parent elements.

### **25. What is localStorage?**

Browser storage to save data locally.

---

## **ES6 Features**

### **26. What are template literals?**

Strings written using backticks with \${} interpolation.

**27. What is destructuring?**

Extracting values from arrays/objects into variables.

**28. What is spread operator?**

... expands arrays/objects.

**29. What are modules?**

Reusable JS files using import/export.

**30. What is default parameter?**

Function parameters with default values.

---

**Error Handling****31. What is try-catch?**

Used to handle runtime errors.

**32. What is throw?**

Manually triggers an error.

---

**Testing & Automation Focus****33. Why is JavaScript important in test automation?**

Used in tools like Playwright, Cypress, and k6.

**34. What is event loop?**

Handles async tasks in JavaScript execution.

**35. What is synchronous vs asynchronous code?**

Sync runs step-by-step; async runs in background.

**36. What is setTimeout()?**

Executes code after a delay.

**37. What is setInterval()?**

Repeats execution at intervals.

**38. What is JSON.parse() and JSON.stringify()?**

Convert JSON string ↔ object.

**39. What is strict mode?**

Enforces safer coding rules.

**40. What is NaN?**

Represents invalid numbers.

---

**Advanced Basics (Interview Favorite)**

**41. What is == vs ===?**

== compares value; === compares value and type.

**42. What is truthy and falsy?**

Falsy: false, 0, "", null, undefined, NaN.

**43. What is prototype?**

Mechanism for inheritance in JS.

**44. What is this keyword?**

Refers to the current object context.

**45. What is call(), apply(), bind()?**

Methods to control function context.

**46. What is IIFE?**

Immediately Invoked Function Expression.

**47. What is shallow vs deep copy?**

Shallow copies reference; deep copies duplicate.

**48. What is debounce?**

Limits how often a function runs.

**49. What is throttle?**

Controls execution rate.

**50. What is garbage collection?**

Automatic memory cleanup in JavaScript.

---