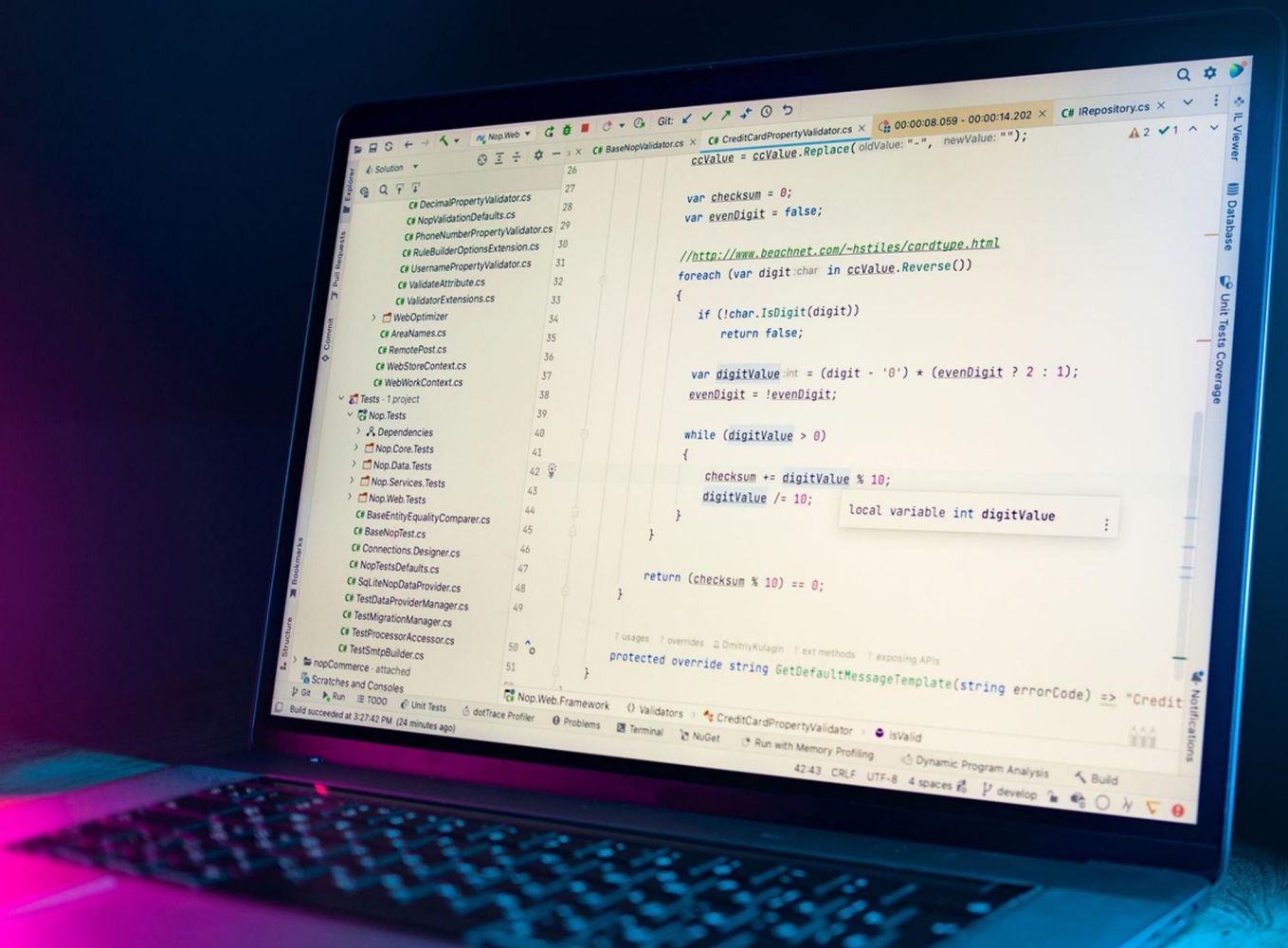


JAVASCRIPT CERTIFICATION PRACTICE TEST (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR
THE FULL VERSION STUDY GUIDE

<https://javascriptcert.examzify.com>



Copyright © 2026 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

Notice: Examzify makes every reasonable effort to obtain accurate, complete, and timely information about this product from reliable sources.

SAMPLE

Table of Contents

Copyright	1
Table of Contents	2
Introduction	3
How to Use This Guide	4
Questions	5
Answers	8
Explanations	10
Next Steps	16

SAMPLE

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

Questions

SAMPLE

1. What method is used to change the current URL without reloading the page?
 - A. history.push()
 - B. history.replace()
 - C. history.pushState()
 - D. history.setState()
2. Which of these methods can be used to select an HTML element by its ID in JavaScript?
 - A. getElementById
 - B. querySelector
 - C. selectById
 - D. findElementById
3. What do `map()` and `forEach()` methods return respectively?
 - A. Both return the same values
 - B. `map()` returns an array, `forEach()` returns undefined
 - C. `map()` returns undefined, `forEach()` returns an array
 - D. Both return the original array
4. What is JSON in the context of JavaScript?
 - A. A method for styling web pages
 - B. A lightweight data-interchange format
 - C. A type of JavaScript variable
 - D. A debugging tool for JavaScript code
5. What is the primary advantage of using AJAX for web applications?
 - A. It eliminates the need for JavaScript
 - B. It allows seamless page refreshing
 - C. It can update part of a page without reloading the whole page
 - D. It ensures faster server responses

6. Which of the following loops should you use if a loop must run at least once even if the condition is false?
- A. for
 - B. while
 - C. do while
 - D. foreach
7. Which of the following statements is not true of the variables declared using the let keyword?
- A. They can be updated
 - B. They are block-scoped
 - C. Variables declared using let are hoisted
 - D. They cannot be redeclared in the same scope
8. What can be a side effect of modifying an object directly in JavaScript?
- A. Creating a new object
 - B. Reducing memory usage
 - C. Potentially impacting other references to that object
 - D. A void return value
9. What is the term for a JavaScript function that is used to create a custom object?
- A. Method
 - B. Constructor
 - C. Prototype
 - D. Initializer
10. In JavaScript, which of the following is used to declare a block-scoped variable?
- A. var
 - B. let
 - C. const
 - D. Both let and const

Answers

SAMPLE

1. C
2. A
3. B
4. B
5. C
6. C
7. C
8. C
9. B
10. D

SAMPLE

Explanations

SAMPLE

1. What method is used to change the current URL without reloading the page?

- A. `history.push()`
- B. `history.replace()`
- C. `history.pushState()`
- D. `history.setState()`

The method used to change the current URL without reloading the page is `history.pushState()`. This method is part of the History API in JavaScript, which allows developers to manipulate the browser history and manage the state of web applications. When you use `history.pushState()`, you can add a new entry to the browser's session history stack. This enables you to change the URL displayed in the browser's address bar, allowing users to navigate back to the previous state without triggering a reload of the page. This is particularly useful for single-page applications where you want to maintain a dynamic experience while still providing meaningful URLs. The arguments you typically pass to `history.pushState()` include an object representing the state of the application, a title (which is generally ignored by most browsers), and the new URL you want to set. This helps enhance user experience by ensuring that the displayed URL reflects the current view of the application, even as various parts of the application change without full-page refreshes. Other methods like `history.push()` and `history.replace()` do not exist within the API. The `replace` method does exist, but it functions differently by replacing the current entry in the session history instead of adding a new one, which does not suit the purpose of changing URLs while maintaining the user's

2. Which of these methods can be used to select an HTML element by its ID in JavaScript?

- A. `getElementById`
- B. `querySelector`
- C. `selectById`
- D. `findElementById`

The method `getElementById` is specifically designed to select a single HTML element by its ID attribute in the document. This method is part of the Document Object Model (DOM) API and allows you to retrieve an element efficiently by providing the exact ID of the element you want to access. It returns the first element with the specified ID, which is particularly useful when you know that IDs must be unique within an HTML document. While `querySelector` is also capable of selecting elements, it uses CSS selectors to select elements and can be used for a broader range of selections. However, the most direct and efficient way to select an element by its ID is by using `getElementById`. The options `selectById` and `findElementById` do not exist in the JavaScript API and therefore cannot be used to access elements, reinforcing that the correct choice is indeed `getElementById`.

3. What do `map()` and `forEach()` methods return respectively?

- A. Both return the same values
- B. `map()` returns an array, `forEach()` returns undefined**
- C. `map()` returns undefined, `forEach()` returns an array
- D. Both return the original array

The `map()` and `forEach()` methods serve different purposes in JavaScript, and their return values reflect their functions accordingly. The `map()` method is designed to create a new array by applying a provided function to each element of the original array. The key characteristic of `map()` is that it returns this new array, containing the results of each function call. This return value is essential for functional programming patterns where transformations of arrays are common. On the other hand, the `forEach()` method is used to execute a provided function once for each array element. Its main purpose is to perform side effects for each element without creating a new array. Because its role is to iterate through the elements and execute side effects, `forEach()` does not return any value; thus, it returns `undefined`. This distinction is critical in using these methods effectively in JavaScript. Choosing between them depends on whether you need a transformed array (in which case you would use `map()`) or just need to perform operations on each item without caring about the result (in which case `forEach()` is appropriate).

4. What is JSON in the context of JavaScript?

- A. A method for styling web pages
- B. A lightweight data-interchange format**
- C. A type of JavaScript variable
- D. A debugging tool for JavaScript code

JSON stands for JavaScript Object Notation, and it is a lightweight data-interchange format that is easy for humans to read and write as well as easy for machines to parse and generate. Its syntax is a subset of the JavaScript programming language, making it naturally compatible with JavaScript, but it is language-independent, meaning it can be used with many programming languages. The use of JSON is particularly prevalent in web applications where data exchange between a client and server is necessary. For example, when sending data from a web application to a server or receiving data from a server, JSON serves as an effective format because it organizes data into key-value pairs, facilitating straightforward data manipulation. In contrast to the other options, styling web pages pertains to CSS, not JSON. There is no specific type of JavaScript variable classified as JSON; rather, JSON is a format for structuring data. Similarly, although debugging tools are essential for identifying errors in code, they do not relate to the concept or functionality of JSON within JavaScript. Thus, the definition of JSON as a lightweight data-interchange format is both accurate and critical for understanding how data is handled in JavaScript applications.

5. What is the primary advantage of using AJAX for web applications?

- A. It eliminates the need for JavaScript
- B. It allows seamless page refreshing
- C. It can update part of a page without reloading the whole page**
- D. It ensures faster server responses

The primary advantage of using AJAX is that it allows web applications to update parts of a page without needing to reload the entire page. This is achieved through asynchronous communication with the server, which allows data to be sent and received in the background. As a result, only the specific content that needs updating is refreshed, leading to a smoother user experience. This method reduces loading times and minimizes the amount of data that needs to be transmitted, as only relevant information is processed rather than the whole web page. This capability significantly enhances responsiveness and interactivity within web applications, allowing users to continue interacting with the page while new data is being fetched. For instance, in a real-time chat application, messages can appear without the user needing to refresh the browser, creating a fluid and dynamic experience.

6. Which of the following loops should you use if a loop must run at least once even if the condition is false?

- A. for
- B. while
- C. do while**
- D. foreach

The do while loop is the appropriate choice when you need a loop that guarantees execution at least once, regardless of whether the condition evaluates to true or false. In a do while loop, the block of code is executed first, and then the condition is checked after the execution. This behavior ensures that even if the condition evaluates to false on the first check, the statements inside the loop will still run once before the condition is evaluated. This contrasts with other types of loops such as the for, while, and forEach loops, which check the condition before executing the loop body. If the condition is false at the outset in those cases, the code within the loop will not execute at all. For example, if you have a task that must perform an initial setup or operation before checking a repeat condition (like prompting a user for input), a do while loop is ideal for ensuring that the operation happens at least once.

7. Which of the following statements is not true of the variables declared using the let keyword?

- A. They can be updated
- B. They are block-scoped
- C. Variables declared using let are hoisted**
- D. They cannot be redeclared in the same scope

Variables declared using the `let` keyword have some distinct characteristics in JavaScript. When it comes to hoisting, variables declared with `let` are indeed hoisted to the top of their enclosing block; however, they cannot be accessed until the line of code where they are defined is executed. This leads to the concept of the "temporal dead zone," where the variable exists in a hoisted state but is not initialized until its definition is evaluated. As a result, trying to access it before this line will throw a `ReferenceError`. This understanding clarifies that, although `let` variables are hoisted, they behave differently from variables declared with `var`. In summary, while `let` variables are hoisted, their access before declaration is restricted, making the statement regarding their hoisting not entirely reflective of their behavior in practical use.

8. What can be a side effect of modifying an object directly in JavaScript?

- A. Creating a new object
- B. Reducing memory usage
- C. Potentially impacting other references to that object**
- D. A void return value

Modifying an object directly in JavaScript can lead to potentially impacting other references to that object because objects in JavaScript are reference types. When you assign an object to a variable, you are actually storing a reference to that object, not the object itself. Therefore, if you have multiple variables referencing the same object and you modify that object through one variable, all other references will see the changes since they all point to the same underlying data in memory. For example, if you have an object assigned to two different variables, and you modify a property using one variable, the change will be reflected when you access that property through the other variable. This can lead to unintended consequences, especially in larger codebases where the same object may be modified by different parts of the code, potentially causing bugs and unexpected behavior. The other options are less relevant as side effects of directly modifying an object. Creating a new object is not a side effect of modification; instead, it typically results from creating a copy or a new instance. Reducing memory usage is usually not a direct consequence of modifying an object, and in fact, modification may lead to increased memory usage if it results in additional references or properties. A void return value isn't a side effect but rather a characteristic of

9. What is the term for a JavaScript function that is used to create a custom object?

- A. Method
- B. Constructor**
- C. Prototype
- D. Initializer

The term for a JavaScript function that is used to create a custom object is "Constructor." In JavaScript, a constructor function is typically defined using a regular function, but it is conventionally named with an initial uppercase letter to signify that it is intended to be used with the `new` keyword. When the `new` keyword is used with a constructor function, it creates a new object that is an instance of the constructor's prototype. In the body of the constructor, properties can be assigned to the `this` context, which refers to the newly created object. This allows the constructor to initialize the object's properties based on parameters passed to it. When developers want to create multiple instances of an object with similar characteristics, they use constructors to encapsulate the initialization logic in a reusable manner. The other terms mentioned relate to different aspects of JavaScript but do not specifically describe the function that creates an object. A method refers to a function that is a property of an object, a prototype is an object from which other objects inherit properties, and an initializer typically refers to the process of assigning a value to a variable or property at the time of declaration rather than a function for creating objects.

10. In JavaScript, which of the following is used to declare a block-scoped variable?

A. var

B. let

C. const

D. Both let and const

Both `let` and `const` are used to declare block-scoped variables in JavaScript, which is why this answer is correct. When you declare a variable with `let`, it is only accessible within the nearest enclosing block (such as a loop or an `if` statement), making it useful for situations where you want to limit the variable's scope to prevent accidental modification from outside the block. Similarly, `const` also defines a block-scoped variable, but with a key distinction: while it protects the variable from being reassigned to a new value, it does allow for mutation of objects or arrays that are assigned to it. This means you cannot reassign a `const` variable, but you can still change properties or elements within an object or array declared as `const`. In contrast, `var` declares variables that are function-scoped or globally-scoped and does not respect block scope. This means that a variable declared with `var` could be accessed outside of the block in which it was defined, leading to potential conflicts and unintended behaviors in code. Thus, the use of `let` and `const` for block-scoped variable declarations helps in maintaining better control over variable accessibility and integrity within a specific block

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

<https://javascriptcert.examzify.com>

We wish you the very best on your exam journey. You've got this!

SAMPLE