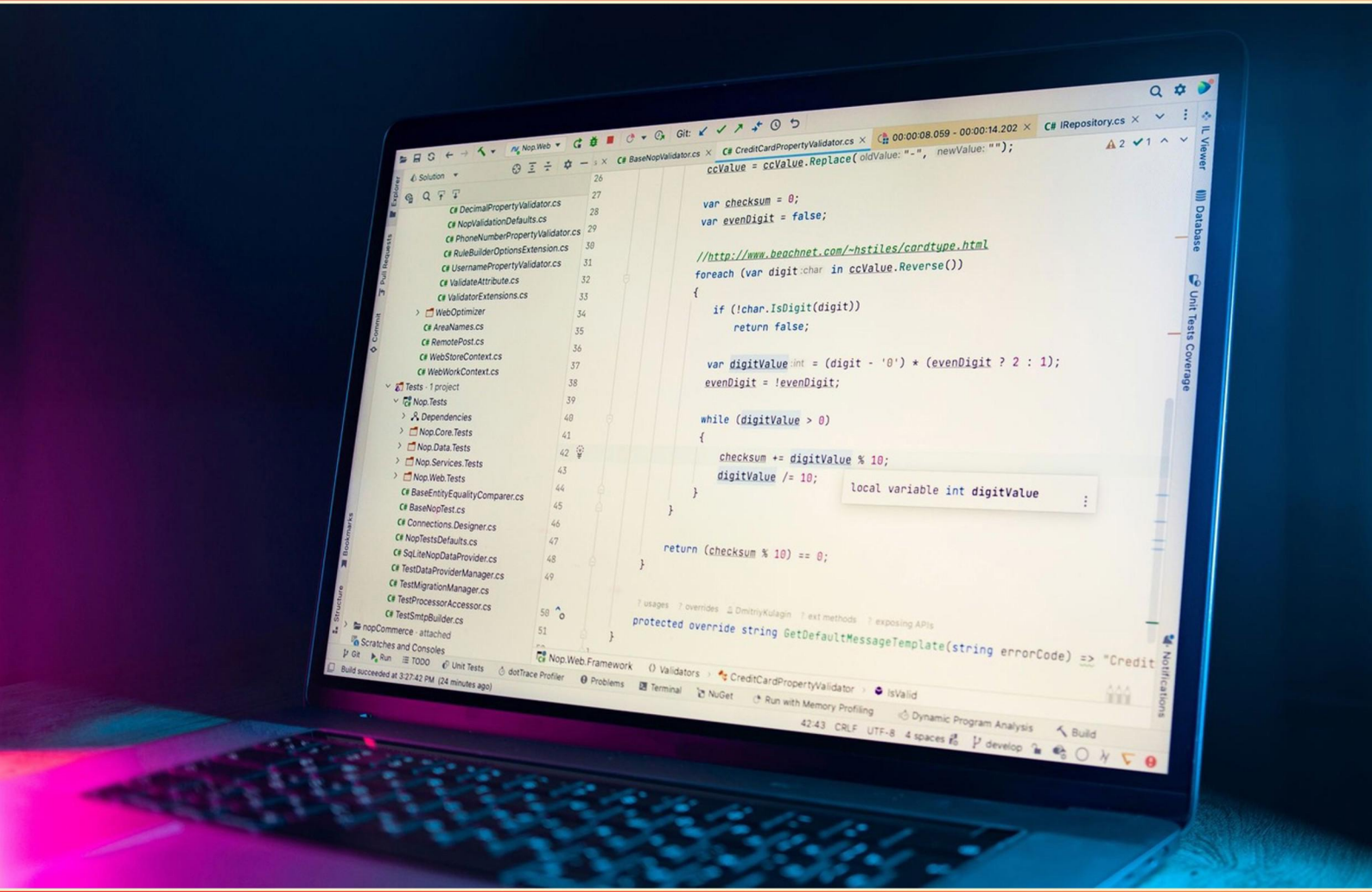
JAVASCRIPT CERTIFICATION PRACTICE TEST (SAMPLE)

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.

SAMPLE

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!

SAMPLE

Questions

SAMPLE

1. What does the `concat()` method do in JavaScript?

A. It merges two or more arrays and returns a new array

B. It changes the order of elements in an array

C. It adds an element to the end of an array

D. It removes the last element from an array

2. What do "truthy" values represent in JavaScript?

A. Values that evaluate to false in a boolean context

B. Values that always return true when evaluated

C. Values that evaluate to true in a boolean context

D. Any non-zero number

3. Which method of the history object is used to create a new history entry?

A. replaceState()

B. pushState()

C. addState()

D. setState()

4. 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

5. What type of value is returned by the `every()` method?

A. An array of boolean values

B. A single boolean value

C. An array of the original elements

D. Undefined

SAMPLE

6. Which property of a document object returns the URL of the current document?

A. location

B. URL

C. href

D. src

7. Which property is used to add CSS styles to elements in a web page?

A. HTMLElement.style

B. Element.styleSheet

C. Document.body.style

D. Window.style

8. What programming technique involves changing variables from one data type to another?

A. Conversion

B. Casting

C. Parsing

D. Implicit typing

9. Which method correctly retrieves the value of an HTML element's attribute within the DOM?

A. querySelector

B. getElementById

C. getAttribute(AttributeName)

D. getElementsByClassName

10. Which form object will you use to mask the value entered by a user?

A. text

B. email

C. password

D. hidden

SAMPLE

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. What does the `concat()` method do in JavaScript?

A. It merges two or more arrays and returns a new array

B. It changes the order of elements in an array

C. It adds an element to the end of an array

D. It removes the last element from an array

The `concat()` method in JavaScript is specifically designed to merge two or more arrays and return a new array containing the elements of the original arrays combined together. When you use `concat()`, it does not alter the existing arrays but instead creates and returns a new array that includes the elements from both the original arrays and any additional arrays you may provide as arguments. This method can be particularly useful when you want to combine data from multiple sources or when you need to create a unified array without modifying the original data structures. For example, if you have one array of numbers and another array of more numbers, employing `concat()` allows you to seamlessly produce a single array with all those values. By contrast, some of the other methods mentioned relate to manipulating the order or contents of an array, such as changing the order of elements, adding elements to the end, or removing elements from the end, none of which describe the functionality provided by `concat()`. This focus on merging arrays is the key feature that distinguishes it in JavaScript array manipulation.

2. What do "truthy" values represent in JavaScript?

A. Values that evaluate to false in a boolean context

B. Values that always return true when evaluated

C. Values that evaluate to true in a boolean context

D. Any non-zero number

In JavaScript, "truthy" values are those that evaluate to true when used in a boolean context, such as in conditional statements or logical operations. This means that if a value is considered truthy, it will pass checks that expect a true condition. Examples of truthy values include non-zero numbers, non-empty strings, objects, and arrays. The concept of truthy is particularly important when you deal with logical statements, conditions in if statements, and loops where you may be testing the validity of expressions or variables. Any value that is not one of the defined "falsy" values — such as false, 0, "", null, undefined, and NaN — is considered truthy. Thus, the identification of truthy values allows developers to understand how JavaScript interprets various data types and values in different contexts, which is essential for writing effective and bug-free code.

SAMPLE

3. Which method of the history object is used to create a new history entry?

A. replaceState()

B. pushState()

C. addState()

D. setState()

The method used to create a new history entry in the history object is `pushState()`. This method allows developers to add a new entry to the browser's session history stack. By using `pushState()`, a new state object is created and can be associated with a particular URL without reloading the page. This is particularly useful in single-page applications (SPAs), where updates to the browser's history and URL are handled dynamically, enhancing user experience and allowing for deeper navigation options without the need for full page reloads. The parameters passed to `pushState()` include the state object (which can be any data you want to associate with that history entry), the title (which is mostly ignored by browsers), and the URL to be displayed in the address bar. Each time `pushState()` is called, a new entry is added to the stack, allowing the user to navigate back and forth through their browsing history. The other methods mentioned are not used for creating new history entries. For example, `replaceState()` modifies the current history entry, while terms like `addState()` and `setState()` do not exist in the context of the history API. This distinction is important for understanding how to effectively manipulate the browser history within a Java

4. 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`.

5. What type of value is returned by the `every()` method?

A. An array of boolean values

B. A single boolean value

C. An array of the original elements

D. Undefined

The `every()` method is designed to evaluate whether all elements in an array satisfy a specific condition defined by a provided testing function. When invoked on an array, it executes that function for each element, and if all elements pass the test (i.e., the function returns `true` for every element), the method returns a single boolean value of `true`. However, if at least one element fails the test (where the function returns `false`), the method will return `false`. This behavior makes `every()` especially useful for validating criteria across a set of items in an array, where you want to ensure that all items conform to a set condition. Since it returns a singular boolean result indicating the overall outcome of the evaluation—either all elements meet the condition or they don't—it correctly returns a single boolean value. In contrast, the other options suggest varying forms of return types that don't align with the actual functionality of `every()`, such as returning an array or being undefined.

SAMPLE

6. Which property of a document object returns the URL of the current document?

A. location

B. URL

C. href

D. src

The property of the document object that returns the URL of the current document is indeed the URL property. The URL property is a straightforward way to access the complete URL of a document as a string. It is part of the Window interface in the browser's JavaScript environment, which means it provides precise information about the context in which the script is running. When a script is executed in a web page, using the URL property allows the developer to get the entire address of the page, including the protocol (like HTTP or HTTPS), domain, path, and any query parameters that might be present. This can be particularly useful for scripts that require context about the page to modify behavior based on the URL. In contrast, the location property is referenced from the window object and provides a more complex representation, as it contains various components such as protocol, host, pathname, and more, while the href property refers specifically to the URL contained in that location object and is used more broadly within anchor tags. The src attribute is typically used for elements like images or script tags to denote the source URL, but it does not directly relate to fetching the URL of the document itself. Thus, the URL property succinctly serves the purpose of returning the current document's URL, making

7. Which property is used to add CSS styles to elements in a web page?

A. HTMLElement.style

B. Element.styleSheet

C. Document.body.style

D. Window.style

The property used to add CSS styles to elements in a web page is the HTMLElement.style property. This property allows you to access and manipulate the inline styles of an HTML element through JavaScript. When you use this property, you can set or retrieve the style attributes directly for that specific element. For example, if you have an element in your HTML, you can use JavaScript to change its styles dynamically by referencing its `style` property. This can include modifications such as changing the `backgroundColor`, `fontSize`, `display`, and many other CSS properties, all of which can be manipulated as properties of the style object. While other choices mention properties that seem related to style manipulation, they do not provide the correct mechanism for directly styling individual elements. For instance, Element.styleSheet is not a valid property used for adding styles directly to an element but rather refers to managing styles on a stylesheet level, which is not the focus of the question. Similarly, Document.body.style pertains specifically to the body element and does not facilitate styling of any arbitrary HTML element. The Window.style option does not exist, as the Window object is related to the global context and does not have CSS properties associated with it. Thus, the HTMLElement.style property stands out

SAMPLE

8. What programming technique involves changing variables from one data type to another?

A. Conversion

B. Casting

C. Parsing

D. Implicit typing

The programming technique that involves changing variables from one data type to another is known as casting. Casting explicitly converts a variable from one data type to another, allowing developers to define how the conversion should occur. For instance, you might cast a string to a number when performing arithmetic operations, which enables the correct calculations to take place. In programming languages like JavaScript, casting can be explicit, where the developer uses specific functions or constructs to convert between types, such as `Number("123")` to convert a string into a number. This technique is essential to ensure type compatibility, especially in situations where operations expect certain types of data. While conversion is a broader term that can encompass various processes of changing data types, casting specifically refers to the explicit act of changing a variable's type. Parsing, on the other hand, generally refers to the process of analyzing a string or data structure to extract meaningful information rather than directly changing types. Implicit typing refers to the behavior of some programming languages that automatically assign data types based on the variable's value, rather than being defined by the programmer, and is not directly related to manually changing types.

9. Which method correctly retrieves the value of an HTML element's attribute within the DOM?

A. querySelector

B. getElementById

C. getAttribute(AttributeName)

D. getElementsByClassName

The method that correctly retrieves the value of an HTML element's attribute within the DOM is the getAttribute(AttributeName). This method is specifically designed to access the attributes of an HTML element by passing the name of the attribute as a parameter. When you use getAttribute, it returns the current value of the specified attribute as a string. For example, if you have an element like `<input type="text" id="myInput" value="Hello">`, calling `element.getAttribute('value')` would return "Hello". This functionality allows developers to interact with attributes directly, which is crucial when manipulating or retrieving data from the DOM. While other methods listed can be useful for selecting or manipulating elements, they do not serve the purpose of directly retrieving attribute values. For instance, querySelector and getElementById are primarily used to select elements based on CSS selectors or IDs, respectively, but they do not retrieve attribute values by themselves. Similarly, getElementsByClassName returns a collection of elements that have the specified class, and it requires additional steps to access an element's attributes.

SAMPLE

10. Which form object will you use to mask the value entered by a user?

A. text

B. email

C. password

D. hidden

The password form object is specifically designed to mask user input, ensuring that the characters typed by the user are not visible on the screen. When a user enters a value into a password input field, the browser typically replaces the characters with dots or asterisks. This feature is crucial for protecting sensitive information, such as user passwords, during data entry. Using a password input type provides an additional layer of security compared to a standard text input, where all characters are displayed as they are typed. Although options like the hidden input type do not display any characters, they are not appropriate for user input as they are intended for storing data that should not be visible to the user at all. Therefore, for the purpose of user interaction while ensuring privacy, the password input type is the ideal choice.

SAMPLE

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