

Free Salesforce Certified JavaScript Developer - Multiple Choice vce dumps & latest JS-Dev-101 examcollection dumps



2026 Latest TestValid JS-Dev-101 PDF Dumps and JS-Dev-101 Exam Engine Free Share: https://drive.google.com/open?id=1fOZT3Yugm-G6vg_xjuCPwgZGS4XDLXjc

From the experience of our former customers, you can finish practicing all the contents in our JS-Dev-101 training materials within 20 to 30 hours, which is enough for you to pass the JS-Dev-101 exam as well as get the related certification. That is to say, you can pass the JS-Dev-101 Exam as well as getting the related certification only with the minimum of time and efforts under the guidance of our JS-Dev-101 training materials. And the pass rate of our JS-Dev-101 learning guide is as high as more than 98%.

Salesforce JS-Dev-101 Exam Syllabus Topics:

Topic	Details
Topic 1	<ul style="list-style-type: none"> • Debugging and Error Handling: Covers proper error handling techniques and the use of the console and breakpoints to debug code.
Topic 2	<ul style="list-style-type: none"> • Testing: Covers evaluating unit test effectiveness against a block of code and modifying tests to improve their coverage and reliability.
Topic 3	<ul style="list-style-type: none"> • Asynchronous Programming: Covers asynchronous programming concepts and understanding how the event loop controls execution flow and determines outcomes.
Topic 4	<ul style="list-style-type: none"> • Objects, Functions, and Classes: Covers function, object, and class implementations to meet business requirements, along with the use of modules, decorators, variable scope, and execution flow.
Topic 5	<ul style="list-style-type: none"> • Variables, Types, and Collections: Covers declaring and initializing variables, working with strings, numbers, dates, arrays, and JSON, along with understanding type coercion and truthy • falsy evaluations.
Topic 6	<ul style="list-style-type: none"> • Server Side JavaScript: Covers Node.js implementations, CLI commands, core modules, and package management solutions for given scenarios.

>> JS-Dev-101 Instant Access <<

Scrutinize Quality With The Salesforce JS-Dev-101 Exam Questions Demo

Once you have decided to purchase our JS-Dev-101 study materials, you can add it to your cart. Then just click to buy and pay for

the certain money. When the interface displays that you have successfully paid for our JS-Dev-101 study materials, our specific online sales workers will soon deal with your orders. You will receive the JS-Dev-101 study materials no later than ten minutes. You need to ensure that you have written down the correct email address. Please check it carefully. If you need the invoice, please contact our online workers. They will send you an electronic invoice, which is convenient. You can download the electronic invoice of the JS-Dev-101 Study Materials and reserve it.

Salesforce Certified JavaScript Developer - Multiple Choice Sample Questions (Q10-Q15):

NEW QUESTION # 10

Refer to the code below:

```
01 async function functionUnderTest(isOK) {
02 if (isOK) return 'OK';
03 throw new Error('not OK');
04 }
```

Which assertion accurately tests the above code?

- A. `console.assert(await (functionUnderTest(true), 'not OK'))`
- B. `console.assert(functionUnderTest(true), 'OK')`
- C. `console.assert(await functionUnderTest(true), 'not OK')`
- **D. `console.assert(await functionUnderTest(true), 'OK')`**

Answer: D

Explanation:

The function:

```
async function functionUnderTest(isOK) {
  if (isOK) return 'OK';
  throw new Error('not OK');
}
```

Behavior:

If isOK is true:

The function resolves (fulfills the promise) with value 'OK'.

If isOK is false:

The function throws, which in an async function becomes a rejected promise with `Error('not OK')`.

We want an assertion that accurately tests the successful path for `isOK === true`.

Key points about `console.assert`:

Signature: `console.assert(condition, message?)`

If condition is falsy, it logs message as an assertion failure.

If condition is truthy, nothing is logged.

We also must understand `await`:

`await functionUnderTest(true)` will resolve to the string 'OK'.

Now evaluate options.

Option A:

```
console.assert(await functionUnderTest(true), 'OK');
```

```
await functionUnderTest(true) → 'OK' (truthy).
```

```
console.assert('OK', 'OK');
```

Condition is 'OK' (truthy), so assertion passes.

The message 'OK' is only shown if the condition is falsy, which it is not.

This correctly verifies that the promise resolved (i.e., did not reject). Among the given options, this is the only one that both:

Uses `await` properly on the async function, and

Associates the message with the expected success "OK".

Option B:

```
console.assert(await (functionUnderTest(true), 'not OK'));
```

Inside the parentheses, the comma operator (a, b) evaluates a, discards it, and returns b.

```
So (functionUnderTest(true), 'not OK'):
```

Calls `functionUnderTest(true)` (returns a promise, but its result is discarded), The expression evaluates to the string 'not OK'.

Then `await 'not OK'` just resolves to 'not OK' immediately (not a promise).

```
console.assert('not OK');
```

Condition is 'not OK' (truthy), so the assertion passes regardless of what `functionUnderTest` actually does.

This does not meaningfully test the function and is misleading.

Option C:

```
console.assert(functionUnderTest(true), 'OK');
```

functionUnderTest(true) returns a Promise, not the string 'OK' directly.

A Promise object is always truthy.

So console.assert(promise, 'OK') will pass, even if the promise later rejects.

Also, there is no await, so we are not actually waiting for the async result.

This is not a correct way to assert on an async function's resolved value.

Option D:

```
console.assert(await functionUnderTest(true), 'not OK');
```

await functionUnderTest(true) still resolves to 'OK' (truthy).

```
console.assert('OK', 'not OK');
```

passes.

However, the message 'not OK' is the opposite of what we expect for the success case and is only used when the condition fails.

This makes the assertion logically inconsistent with the function behavior (it would show "not OK" if the assertion failed).

Therefore, the only assertion that:

Properly waits on the async function, and

Has expectation text that matches the successful behavior ('OK')

is:

Answer: A

Study Guide / Concept Reference (no links):

async and await behavior in JavaScript

Promises resolving vs rejecting in async functions

console.assert(condition, message) usage

Truthy and falsy values in JavaScript

Comma operator (a, b) semantics

NEW QUESTION # 11

Which three actions can the code execute in the browser console?

- A. View and change the DOM of the page.
- B. View, change, and debug the JavaScript code of the page.
- C. Run code that is not related to the page.
- D. View and change security cookies.
- E. Display a report showing the performance of a page.

Answer: A,B,C

Explanation:

Comprehensive and Detailed Explanation From Exact Extract JavaScript Knowledge The browser console (Developer Tools → Console) allows:

Running code unrelated to the page

Developers can type arbitrary JavaScript not tied to the page's logic.

This is allowed → A is correct.

Viewing and modifying JavaScript code execution

Using the Sources tab and console, developers can:

Inspect variables

Modify code at runtime

Set breakpoints

Therefore → D is correct.

Viewing and manipulating the DOM

From the console, developers can:

Query DOM nodes using selectors

Modify element properties and attributes

Insert or remove elements

Therefore → E is correct.

Why the other options are incorrect:

B (security cookies)

Many cookies have flags such as HttpOnly and Secure.

Cookies with the HttpOnly flag cannot be viewed or modified via JavaScript or console.

Therefore this is not guaranteed → incorrect.

C (performance report)

The console itself does not provide a performance report.

The Performance or Lighthouse tabs provide this, not the console.

Therefore incorrect.

JavaScript Knowledge Reference (text-only)

Developer console can execute arbitrary JavaScript.

Console allows DOM inspection and manipulation.

Security cookies (HttpOnly) cannot be accessed via JavaScript.

NEW QUESTION # 12

Given the following code:

What is the output of line 02?

- A. "x-
- B. "undefined" 0
- C. 'object'
- D. "null"

Answer: C

NEW QUESTION # 13

Refer to the code below:

```
01 let country = {  
02   get capital() {  
03     let city = Number("London");  
04  
05     return {  
06       cityString: city.toString(),  
07     }  
08   }  
09 }
```

Which value can a developer expect when referencing `country.capital.cityString`?

- A. An error
- B. NaN
- C. undefined
- D. 'London'

Answer: B

Explanation:

In the getter:

```
let city = Number("London");
```

The `Number()` constructor attempts to convert the string "London" into a numeric value.

"London" is not a valid numeric string.

When JavaScript attempts numeric conversion of a non-numeric string:

```
Number("London") → NaN
```

Next line:

```
city.toString()
```

When `city` is `NaN`, calling `.toString()` yields:

```
NaN.toString() → "NaN"
```

The getter returns:

```
{  
  cityString: "NaN"  
}
```

The question asks:

Which value can a developer expect?

The multiple-choice options include `NaN`, but not `"NaN"`.

Given the available choices, the intended correct answer is B (NaN) because the root cause is the Number("London") conversion returning NaN.

JavaScript Knowledge Reference (text-only)

Number(nonNumericString) returns NaN.

NaN.toString() produces "NaN".

Getters compute and return their values each time the property is accessed.

NEW QUESTION # 14

A developer implements a function that adds a few values.

```
Function sum(num) {  
  If(num == undefined) {  
    Num=0;  
  }  
  Return function( num2, num3){  
    If(num3 ==undefined) {  
      Num3 =0 ;  
    }  
    Return num + num2 + num3;  
  }  
}
```

Which three options can the developer invoke for this function to get a return value of 10 ?

Choose 3 answers

- A. sum() (5, 5)
- B. sum(10) ()
- C. Sum (5, 5) ()
- D. Sum () (20)
- E. sum(5)(5)

Answer: A,E

NEW QUESTION # 15

.....

TestValid JS-Dev-101 exam dumps have been developed with a conscious effort to abridge information into fewer questions and answers that any candidate can learn easily. Now you don't need to go through the hassle of studying lengthy manuals for JS-Dev-101 Exam Questions preparation. What you actually required is packed into easy to grasp content. Fix your attention on these JS-Dev-101 questions and answers and your success is guaranteed.

Test JS-Dev-101 Centres: <https://www.testvalid.com/JS-Dev-101-exam-collection.html>

- 2026 Salesforce JS-Dev-101: Salesforce Certified JavaScript Developer - Multiple Choice Instant Access Search for **【 JS-Dev-101 】** and download it for free immediately on \Rightarrow www.vce4dumps.com \Leftarrow Valid Braindumps JS-Dev-101 Files
- JS-Dev-101 Exam Dumps.zip JS-Dev-101 Latest Exam Discount Valid JS-Dev-101 Test Forum Enter 「 www.pdfvce.com 」 and search for 「 JS-Dev-101 」 to download for free JS-Dev-101 Latest Test Cram
- JS-Dev-101 Latest Test Cram Test JS-Dev-101 Score Report JS-Dev-101 Study Group Open **【 www.torrentvce.com 】** enter { JS-Dev-101 } and obtain a free download Dumps JS-Dev-101 Torrent
- Realistic JS-Dev-101 Instant Access - Salesforce Certified JavaScript Developer - Multiple Choice 100% Pass Quiz The page for free download of \blacktriangleright JS-Dev-101 on \ll www.pdfvce.com \gg will open immediately JS-Dev-101 Exam Quick Prep
- 100% Pass Salesforce - JS-Dev-101 -High-quality Instant Access Open \blacktriangleright www.dumpsquestion.com and search for \Rightarrow JS-Dev-101 \Leftarrow to download exam materials for free Reliable JS-Dev-101 Test Question
- JS-Dev-101 Latest Exam Discount Dumps JS-Dev-101 Torrent JS-Dev-101 Latest Test Cram Open website www.pdfvce.com and search for \blacktriangleright JS-Dev-101 for free download JS-Dev-101 Detailed Answers
- Realistic JS-Dev-101 Instant Access - Salesforce Certified JavaScript Developer - Multiple Choice 100% Pass Quiz Easily obtain free download of \ll JS-Dev-101 \gg by searching on **【 www.testkingpass.com 】** \blacklozenge Test JS-Dev-101 Score Report

- 100% Pass Quiz 2026 JS-Dev-101: Professional Salesforce Certified JavaScript Developer - Multiple Choice Instant Access □ Search for ⇒ JS-Dev-101 ⇐ and easily obtain a free download on ▷ www.pdfvce.com ◁ □ JS-Dev-101 Study Group
- JS-Dev-101 Exam Dumps.zip □ Exam JS-Dev-101 Dump □ JS-Dev-101 Latest Test Online □ Search for ► JS-Dev-101 □ on 【 www.dumpsquestion.com 】 immediately to obtain a free download □ JS-Dev-101 Latest Test Online
- Exam JS-Dev-101 Dump □ JS-Dev-101 Latest Test Cram □ Exam JS-Dev-101 Dump □ Search for ✨ JS-Dev-101 □ ✨ □ and download exam materials for free through [www.pdfvce.com] □ JS-Dev-101 Online Training
- JS-Dev-101 Test Preparation: Salesforce Developers - JS-Dev-101 Best Questions □ Download 「 JS-Dev-101 」 for free by simply searching on 《 www.troytecdumps.com 》 □ JS-Dev-101 Exam Quick Prep
- johsocial.com, donnaabraf618063.wikiinside.com, bookmarkahref.com, fanniedzn1495695.blogrelation.com, socialrus.com, heathudxe363548.plpwiki.com, rebeccavlyb603798.ssnblog.com, deannagbeb463855.buyoutblog.com, sairawxnc837308.get-blogging.com, georgiapdjr248992.blogsvila.com, Disposable vapes

BONUS!!! Download part of TestValid JS-Dev-101 dumps for free: https://drive.google.com/open?id=1fOZT3Yugn-G6vg_xjuCPwgZGS4XDLXjc