JS-Dev-101 Examcollection, Exam JS-Dev-101 Objectives

P.S. Free 2026 Salesforce JS-Dev-101 dumps are available on Google Drive shared by Itbraindumps: https://drive.google.com/open?id=1SALKsrYyUrQOhhvC7hYb3027O0goZsE0

Maybe you are still having trouble with the Salesforce JS-Dev-101 exam; maybe you still don't know how to choose the JS-Dev-101 exam materials; maybe you are still hesitant. But now, your search is ended as you have got to the right place where you can catch the finest JS-Dev-101 exam materials. Here you can answer your doubts; you can easily pass the exam on your first attempt. All applicants who are working on the JS-Dev-101 exam are expected to achieve their goals, but there are many ways to prepare for exam. Everyone may have their own way to discover. Some candidates may like to accept the help of their friends or mentors, and some candidates may only rely on some JS-Dev-101 books. But none of these ways are more effective than our JS-Dev-101 exam material. In summary, choose our exam materials will be the best method to defeat the exam.

Salesforce JS-Dev-101 Exam Syllabus Topics:

Topic	Details
Topic 1	• Testing: Covers evaluating unit test effectiveness against a block of code and modifying tests to improve their coverage and reliability.
Topic 2	• Server Side JavaScript: Covers Node.js implementations, CLI commands, core modules, and package management solutions for given scenarios.
Topic 3	• 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 4	• Debugging and Error Handling: Covers proper error handling techniques and the use of the console and breakpoints to debug code.
Topic 5	• Asynchronous Programming: Covers asynchronous programming concepts and understanding how the event loop controls execution flow and determines outcomes.

Topic 6	• 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.

>> JS-Dev-101 Examcollection <<

Fantastic JS-Dev-101 Examcollection Provide Prefect Assistance in JS-Dev-101 Preparation

Most of the JS-Dev-101 exam dumps on the platform are out of reach for most users due to their high price. Visit the Salesforce JS-Dev-101 exam dumps if you want to buy real Salesforce JS-Dev-101 Exam Questions at a good price. Start your Salesforce JS-Dev-101 exam preparation with our exam practice questions.

Salesforce Certified JavaScript Developer - Multiple Choice Sample Questions (Q90-Q95):

NEW QUESTION # 90
Refer to the code below:
01 <html lang="en">
02 <table onclick="console.log('Table log');">
03 <tr id="row1">
04 <td>Click me!</td>
05 </tr>
06 </table>
07 <script>
08 function printMessage(event) {
09 console.log('Row log');
10 event.stopPropagation();
11 }
12
13 let elem = document.getElementById('row1');
14 elem.addEventListener('click', printMessage, false);
15 </script>
16 </html>
Which code change should be done for the console to log the following when "Click me!" is clicked?
Row log
Table log

- A. Remove lines 13 and 14
- B. Remove line 10
- C. Change line 10 to event.stopPropagation(false);
- D. Change line 14 to elem.addEventListener('click', printMessage, true);

Answer: B

Explanation:
Current behavior:
Clicking <td> triggers the click event on row1, then bubbles up to <table>.
printMessage runs, logs "Row log", then event.stopPropagation() stops the event from bubbling to the table.
So "Table log" never appears.
To allow the table's inline onclick to run after the row handler:
Remove the propagation stop:
function printMessage(event) {
console.log('Row log');
// event.stopPropagation(); // remove this
}
Now the event bubbles:

printMessage logs "Row log".
The table's onclick runs, logging "Table log".
Option A only changes capture/bubble phase but still stops propagation. C does nothing meaningful (stopPropagation takes no arguments). B removes the row handler entirely.

NEW QUESTION # 91
A developer copied a JavaScript object:
01 function Person() {
02 this.firstName = "John";
03 this.lastName = "Doe";
04 this.name = () => `${this.firstName},${this.lastName}`;
05 }
06
07 const john = new Person();
08 const dan = Object.assign({}, john);
09 dan.firstName = 'Dan';
How does the developer access dan's firstName, lastName?

- A. dan.firstName + dan.lastName
- B. dan.firstName() + dan.lastName()
- C. dan.name()
- D. dan.name

Answer: C

Explanation:
Person instances have:
firstName and lastName as string properties.
A name method that returns a combined string: `${this.firstName},${this.lastName}`.
Object.assign({}, john) creates a shallow copy of john into a new object, dan.
After:
dan.firstName = 'Dan';
dan.name() returns "Dan,Doe".
Analysis of options:
A: dan.firstName() and dan.lastName() are function calls, but firstName/lastName are strings, not functions → TypeError.
B: Calls the defined method and uses both names correctly.
C: dan.name is a function reference; you'd still need to call it: dan.name().
D: dan.firstName + dan.lastName is "DanDoe", no separator. It accesses the properties but not in the method the developer defined.
The intended way, using the provided API, is dan.name().

NEW QUESTION # 92
Refer to the code below (corrected to use a template literal on line 08):
01 let car1 = new Promise((_, reject) =>
02 setTimeout(reject, 2000, "Car 1 crashed in")
03);
04 let car2 = new Promise(resolve =>
05 setTimeout(resolve, 1500, "Car 2 completed")
06);
07 let car3 = new Promise(resolve =>
08 setTimeout(resolve, 3000, "Car 3 completed")
09);
10
11 Promise.race([car1, car2, car3])
12 .then(value => {
13 let result = `${value} the race.`;
14 })

15 .catch(err => {
16 console.log("Race is cancelled.", err);
17 });
What is the value of result when Promise.race executes?

- A. Car 3 completed the race.
- B. Race is cancelled.
- C. Car 1 crashed in the race.
- D. Car 2 completed the race.

Answer: D

Explanation:
Comprehensive and Detailed Explanation From Exact Extract JavaScript knowledge:
Understand the three promises:
car1:
let car1 = new Promise((_, reject) =>
setTimeout(reject, 2000, "Car 1 crashed in")
);
Rejects after 2000 ms (2 seconds) with message "Car 1 crashed in".
car2:
let car2 = new Promise(resolve =>
setTimeout(resolve, 1500, "Car 2 completed")
);
Resolves after 1500 ms (1.5 seconds) with message "Car 2 completed".
car3:
let car3 = new Promise(resolve =>
setTimeout(resolve, 3000, "Car 3 completed")
);
Resolves after 3000 ms (3 seconds) with message "Car 3 completed".
Promise.race:
Promise.race([car1, car2, car3])
.then(value => {
let result = `${value} the race.`;
})
.catch(err => {
console.log("Race is cancelled.", err);
});
Behavior of Promise.race:
It settles (resolves or rejects) as soon as any of the given promises settles.
It uses the value or reason from the first settled promise.
Timing:
car2 resolves in 1500 ms.
car1 rejects in 2000 ms.
car3 resolves in 3000 ms.
The first to settle is car2 at 1500 ms, with value "Car 2 completed".
Therefore:
Promise.race resolves (not rejects) with value = "Car 2 completed".
The .then handler runs; .catch is ignored because there is no rejection.
Inside .then:
let result = `${value} the race.`;
Substitute value:
let result = "Car 2 completed the race.";
So, result becomes:
Car 2 completed the race.
Compare to options:
A . Car 3 completed the race.
This would be correct if car3 were the first to resolve, which it is not (it resolves last).
B . Car 2 completed the race.
Exactly matches the first-resolving promise and the constructed message.
C . Race is cancelled.

This is the prefix of the string logged in the .catch handler, but .catch never runs because the race resolves, it does not reject first.
D . Car 1 crashed in the race.
car1 is the first rejection, but since a resolution from car2 happens earlier, the race is already settled successfully before car1 rejects.
Thus the correct value of result as set in the .then block is:
Answe r: B
Study Guide / Concept Reference (no links):
Promise.race(iterable) semantics (first settled promise wins)
setTimeout and timing interactions with Promises
Resolve vs reject paths and .then / .catch
Template literals and string interpolation for building result messages

NEW QUESTION # 93
Refer to the code:
01 console.log('Start');
02 Promise.resolve('Success').then(function(value) {
03 console.log('Success');
04 });
05 console.log('End');
What is the output after the code executes successfully?

- A. End
  Start
  Success
- B. Start
  Success
  End
- C. Success
  Start
  End
- D. Start
  End
  Success

Answer: D

Explanation:
_____
Comprehensive and Detailed Explanation From Exact Extract JavaScript Knowledge console.log('Start') runs immediately (synchronous).
Promise.resolve().then(...) places the callback in the microtask queue. The .then handler does not run immediately.
console.log('End') runs next (still synchronous).
After the synchronous script finishes, the microtask queue runs, logging "Success".
Execution order:
Start
End
Success
This matches option B.
_____
JavaScript Knowledge Reference (text-only)
Promise .then() callbacks execute after the current call stack finishes (microtask queue).
Synchronous logs run immediately, before promise callbacks.

NEW QUESTION # 94
Refer to the code declarations below:
Which three expressions return the string JavaScript?
Choose 3 answers

- A. Str1.concat (str2);

- B. Concat (str1, str2);
- C. Str1.join (str2);
- D. $(str1) $ (str2} ';
- E. Str1 + str2;

Answer: A,D,E

NEW QUESTION # 95

.....

If you don't prepare with real Salesforce JS-Dev-101 questions, you fail, lose time and money. Itbraindumps product is specially designed to help you pass the exam on the first try. The study material is easy to use. You can choose from 3 different formats available according to your needs. The 3 formats are Salesforce JS-Dev-101 desktop practice test software, browser based practice exam, and PDF.

Exam JS-Dev-101 Objectives: https://www.itbraindumps.com/JS-Dev-101_exam.html

- Quiz 2026 Reliable JS-Dev-101: Salesforce Certified JavaScript Developer - Multiple Choice Examcollection □ Enter { www.prepawaypdf.com } and search for □ JS-Dev-101 □ to download for free □Latest JS-Dev-101 Test Answers
- JS-Dev-101 Reliable Test Labs □ Reliable JS-Dev-101 Test Cost □ Certification JS-Dev-101 Training □ Search for ➥ JS-Dev-101 □ and easily obtain a free download on □ www.pdfvce.com □ □JS-Dev-101 Dump Torrent
- Test JS-Dev-101 Valid □ JS-Dev-101 Reliable Test Labs □ New JS-Dev-101 Exam Book □ Download 《 JS-Dev-101 》 for free by simply entering 「 www.testkingpass.com 」 website □New JS-Dev-101 Exam Book
- 100% Pass Quiz Latest JS-Dev-101 - Salesforce Certified JavaScript Developer - Multiple Choice Examcollection □ Search for □ JS-Dev-101 □ and download it for free immediately on （ www.pdfvce.com ） □Valid JS-Dev-101 Exam Cram
- Dump JS-Dev-101 Collection □ JS-Dev-101 Valid Test Blueprint □ Dump JS-Dev-101 Collection □ Download " JS-Dev-101 " for free by simply searching on □ www.prepawayexam.com □ □Latest JS-Dev-101 Exam Labs
- Pass Guaranteed Quiz JS-Dev-101 - Salesforce Certified JavaScript Developer - Multiple Choice –The Best Examcollection □ Easily obtain 《 JS-Dev-101 》 for free download through ➠ www.pdfvce.com □ □Exam JS-Dev-101 Success
- JS-Dev-101 practice questions - JS-Dev-101 latest torrent - JS-Dev-101 training material □ Go to website [www.practicevce.com] open and search for ☀ JS-Dev-101 □☀□ to download for free □Test JS-Dev-101 Valid
- Well-Prepared JS-Dev-101 Examcollection - Pass-Sure Exam JS-Dev-101 Objectives - Reliable Salesforce Salesforce Certified JavaScript Developer - Multiple Choice □ Search for ▹ JS-Dev-101 ◃ and download it for free immediately on ⇛ www.pdfvce.com ⇚ □JS-Dev-101 Valid Test Blueprint
- Trusting Reliable JS-Dev-101 Examcollection Is The Quickest Way to Pass Salesforce Certified JavaScript Developer - Multiple Choice □ Open website ➡ www.pdfdumps.com □□□ and search for ➡ JS-Dev-101 □□□ for free download □Certification JS-Dev-101 Training
- Pass Guaranteed Quiz JS-Dev-101 - Salesforce Certified JavaScript Developer - Multiple Choice –The Best Examcollection ✌ Enter ➡ www.pdfvce.com □□□ and search for ➤ JS-Dev-101 □ to download for free □Latest JS-Dev-101 Test Answers
- Dumps JS-Dev-101 Discount □ Test JS-Dev-101 Valid □ Test JS-Dev-101 Valid □ Open ✔ www.testkingpass.com □✔□ enter ☀ JS-Dev-101 □☀□ and obtain a free download □Exam Dumps JS-Dev-101 Provider
- miriamkpkk200684.activablog.com, mayaiooe283168.activablog.com, myportal.utt.edu.tt, aronsjtv949533.wikiinside.com, weixiuguan.com, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, adreaidaf255628.empirewiki.com, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, Disposable vapes

What's more, part of that Itbraindumps JS-Dev-101 dumps now are free: https://drive.google.com/open?id=1SALKsrYyUrQOhhvC7hYb3027O0goZsE0