Test Salesforce JS-Dev-101 Tutorials - Braindumps JS-Dev-101 Downloads

This relieves any sort of anxiety in the candidate mind before the purchase of Salesforce JS-Dev-101 exam preparation material. This JS-Dev-101 exam study material is offered to you at a very low price. We also offer up to 365 days of free updates on Salesforce JS-Dev-101 Dumps after the date of purchase.

VCETorrent has been designing and offering real Salesforce Salesforce Certified JavaScript Developer - Multiple Choice exam dumps for many years. We regularly update our valid Salesforce JS-Dev-101 certification test preparation material to keep them in line with the current Salesforce Certified JavaScript Developer - Multiple Choice (JS-Dev-101) exam content and industry standards. Professionals from different countries give us their valuable feedback to refine JS-Dev-101 actual dumps even more.

>> Test Salesforce JS-Dev-101 Tutorials <<

To Become a Certified Holder Prepare With Actual Salesforce JS-Dev-101 Questions

With the Software version of our JS-Dev-101 exam questions, you will find that there are no limits for the amount of the computers when download and installation and the users. You can use our JS-Dev-101 study materials to stimulate the exam to adjust yourself to the atmosphere of the real exam and adjust your speed to answer the questions. The other two versions also boost the strenght and applicable method and you could learn our JS-Dev-101 training quiz by choosing the most suitable version to according to your practical situation.

Salesforce JS-Dev-101 Exam Syllabus Topics:

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

Salesforce Certified JavaScript Developer - Multiple Choice Sample Questions (Q104-Q109):

NEW QUESTION # 104
Refer to the code below:
01 const addBy = ?
02 const addByEight = addBy(8);
03 const sum = addByEight(50);
Which two functions can replace line 01 and return 58 to sum?

- A. const addBy = function(num1) {
  return num1 * num2;
  }
- B. const addBy = function(num1) {
  return function(num2) {
  return num1 + num2;
  }
  }
- C. (Corrected for typing errors)
  const addBy = (num1) => {
  return function(num2) {
  return num1 + num2;
  }
- D. const addBy = (num1) => num1 + num2;

Answer: B,C

Explanation:
}
Explanation:
We want:
const addByEight = addBy(8);
const sum = addByEight(50);
And we need sum to be 58.
That means:
addBy(8) must return a function.
That returned function, when called with 50, must compute 8 + 50.
So addBy must be a higher-order function that returns another function capturing num1 and later using it with num2 (a closure).
Option A
const addBy = function(num1) {
return function(num2) {
return num1 + num2;

}
}
Step-by-step:
Call addBy(8):
num1 is 8.
The function returns an inner function: function(num2) { return num1 + num2; }.
So addByEight becomes this inner function, with num1 closed over as 8.
Then call addByEight(50):
num2 is 50.
The body computes num1 + num2, i.e., 8 + 50 = 58.
Therefore, with Option A in place:
const addByEight = addBy(8); // returns inner function
const sum = addByEight(50); // 58
So sum is 58. Option A is correct.
Option D (corrected)
const addBy = (num1) => {
return function(num2) {
return num1 + num2;
}
}
This is essentially the same logic expressed with an arrow function for the outer function:
Call addBy(8):
num1 is 8.
Returns the inner function function(num2) { return num1 + num2; }.
Call addByEight(50):
num2 is 50.
Computes num1 + num2 → 8 + 50 = 58.
So again:
const addByEight = addBy(8); // inner function with num1 = 8
const sum = addByEight(50); // 58
Option D is also correct.
Thus, A and D are the two functions that satisfy the requirement.
Why B and C are incorrect
Option B:
const addBy = function(num1) {
return num1 * num2;
}
This does not return a function; it returns a value.
addBy(8) returns 8 * num2, but num2 is not defined in this scope, which would cause a ReferenceError.
Also, addByEight would be a number (if num2 existed), not a function, so addByEight(50) would fail.
Option C:
const addBy = (num1) => num1 + num2;
Again, addBy returns a value, not a function.
addBy(8) returns 8 + num2, but num2 is not defined, so this is also invalid due to ReferenceError.
addByEight would be a number (or error), not a function.
Neither B nor C creates the required closure nor returns a function to be called later.
Reference / Study Guide concepts (no links):
Higher-order functions in JavaScript
Closures: inner functions capturing outer variables (num1)
Arrow functions vs function expressions
Returning functions from functions (function factories / currying)
Scope and ReferenceError when a variable is not defined

NEW QUESTION # 105
Refer to the following code:
Let sampleText = 'The quick brown fox jumps';
A developer needs to determine if a certainsubstring is part of a string.
Which three expressions return true for the given substring ?
Choose 3 answers

- A. sampleText.includes(' Fox ', 3)
- B. sampleText.includes(' quick ') !== -1;
- C. sampleText.includes(' fox ');
- D. sampleText.includes(' quick ', 4);
- E. sampleText.includes('fox');

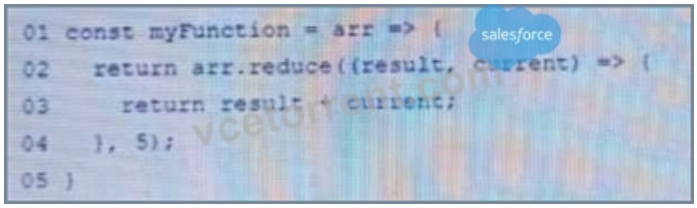
Answer: B,C,D

NEW QUESTION # 106
A developer has two ways to write a function:
Option A:
function Monster() {
This.growl = () => {
Console.log ("Grr!");
}
}
Option B:
function Monster() {};
Monster.prototype.growl =() => {
console.log("Grr!");
}
After deciding on an option, the developer creates 1000 monster objects.
How many growl methods are created with Option A Option B?

- A. 1000 growl method is created for Option A. 1 growl methods are created for Option
- B. 1 growl method is created regardless of whichoption is used.
- C. 1 growl method is created for Option 1000 growl methods are created for Option B.
- D. 1000 growl methods are created regardless of which option is used.

Answer: A

NEW QUESTION # 107
Refer to the code below:



What is the output of this function when called with an empty array?

- A. Return Infinity
- B. Return 0
- C. Return NaN
- D. Return 5

Answer: D

NEW QUESTION # 108
Refer to the code below:
class Student {
constructor(name) {
this._name = name;
}
displayGrade() {
console.log(`${this._name} got 70% on test.`);
}

```
}
class GraduateStudent extends Student {
constructor(name) {
super(name);
this._name = "Graduate Student " + name;
}
displayGrade() {
console.log(`${this._name} got 100% on test.`);
}
}
let student = new GraduateStudent("Jane");
student.displayGrade();
```
What is the console output?

- A. Better student Jackie got 70% on test.
- B. Jackie got 70% on test.
- C. Uncaught ReferenceError
- D. Graduate Student Jane got 100% on test.

Answer: D

Explanation:
The correct answer is C, after correcting the option text to match the actual code.
The object is created here:
```
let student = new GraduateStudent("Jane");
```
Because GraduateStudent extends Student, its constructor runs:
```
constructor(name) {
super(name);
this._name = "Graduate Student " + name;
}
```
The call to:
```
super(name);
```
runs the parent Student constructor first and temporarily sets:
```
this._name = "Jane";
```
Then this line in the child constructor overwrites that value:
```
this._name = "Graduate Student " + name;
```
So the final value of this._name becomes:
"Graduate Student Jane"
Next, this line executes:
```
student.displayGrade();
```
Since student is an instance of GraduateStudent, JavaScript uses the overridden displayGrade() method from GraduateStudent, not the parent method from Student.
The executed method is:
```
displayGrade() {
console.log(`${this._name} got 100% on test.`);
}
```
Therefore, the console output is:
Graduate Student Jane got 100% on test.
Important correction: the original option C said something like "Better student Jackie got 100% on test.", but the actual code uses "Graduate Student " and the name "Jane". The verified corrected answer remains C.

NEW QUESTION # 109
.....

Braindumps JS-Dev-101 Downloads: https://www.vcetorrent.com/JS-Dev-101-valid-vce-torrent.html

- 100% Pass Quiz 2026 Salesforce JS-Dev-101 Authoritative Test Tutorials □ Simply search for 【 JS-Dev-101 】 for free download on □ www.prepawaypdf.com □ □Exam JS-Dev-101 Guide
- 100% Pass Quiz 2026 Salesforce JS-Dev-101 Authoritative Test Tutorials □ Download ➥ JS-Dev-101 □ for free by simply searching on [www.pdfvce.com] □Valid JS-Dev-101 Test Labs
- JS-Dev-101 Exam Quiz □ JS-Dev-101 Authorized Exam Dumps □ New JS-Dev-101 Practice Materials □ Search for □ JS-Dev-101 □ and download it for free on ✔ www.examcollectionpass.com □✔□ website □Study JS-Dev-101 Materials
- New JS-Dev-101 Practice Materials □ Study JS-Dev-101 Materials □ Exam JS-Dev-101 Guide □ Search for " JS-Dev-101 " and download it for free immediately on [www.pdfvce.com] □Valid JS-Dev-101 Test Labs
- 100% Pass Quiz 2026 Salesforce JS-Dev-101 Authoritative Test Tutorials □ ☀ www.prepawayexam.com □☀□ is best website to obtain □ JS-Dev-101 □ for free download ▸New JS-Dev-101 Test Simulator
- 100% Pass Quiz 2026 Salesforce JS-Dev-101 Unparalleled Test Tutorials □ Search on □ www.pdfvce.com □ for 【 JS-Dev-101 】 to obtain exam materials for free download □JS-Dev-101 Trustworthy Dumps
- Free PDF Quiz JS-Dev-101 - Salesforce Certified JavaScript Developer - Multiple Choice Useful Test Tutorials □ Search for { JS-Dev-101 } and easily obtain a free download on ➠ www.prep4away.com □ □Study JS-Dev-101 Center
- 100% Pass Quiz 2026 Salesforce JS-Dev-101 Authoritative Test Tutorials □ Download □ JS-Dev-101 □ for free by simply entering ➡ www.pdfvce.com □□□ website □JS-Dev-101 Sample Questions Answers
- JS-Dev-101 Valid Exam Papers □ Study JS-Dev-101 Center □ JS-Dev-101 Exam Quiz □ ☀ www.practicevce.com □☀□ is best website to obtain " JS-Dev-101 " for free download □JS-Dev-101 Trustworthy Dumps
- 100% Pass Quiz 2026 Salesforce JS-Dev-101 Unparalleled Test Tutorials □ The page for free download of ✔ JS-Dev-101 □✔□ on 【 www.pdfvce.com 】 will open immediately □New JS-Dev-101 Test Simulator
- JS-Dev-101 Actual Exam □ JS-Dev-101 Exam Actual Questions □□ Exam Dumps JS-Dev-101 Free □ Search for ➡ JS-Dev-101 □□□ and download exam materials for free through ☀ www.vce4dumps.com □☀□ □Exam JS-Dev-101 Guide
- gretayedx204210.blog-ezine.com, graysoncgoh544539.tusblogos.com, adamjunc441384.kylieblog.com, poppiergbs164011.blogdosaga.com, thegreatbookmark.com, francesuwdg885049.bloggazza.com, aoifeylbp532472.blogacep.com, phoebebafr043797.wikiap.com, gregoryrqrw051551.wikikarts.com, socialmediatotal.com, Disposable vapes