

Aktuelle Salesforce JS-Dev-101 Prüfung pdf Torrent für JS-Dev-101 Examen Erfolg prep

Salesforce CRT-101 Certification Preparation for Administrator Exam 1



CRT-101 Test Dumps, CRT-101 VCE Engine Ausbildung, CRT-101 aktuelle Prüfung

Die Salesforce CRT-101 Zertifizierungsunterlagen von ExamFragen sind unbedingt die Unterlagen für Salesforce CRT-101 Zertifizierungsprüfung, an der Sie glauben können. Falls Sie nicht glauben, probieren Sie bitte persönlich, dann können Sie diese Tatsachen wissen. Klicken Sie bitte die Demo von ExamFragen Website. PDF-Versionen und Software-Versionen sind beide vorhanden. Probieren Sie bitte zuerst. Sie können persönlich die Qualität der Salesforce CRT-101 Dumps überprüfen.

Salesforce CRT-101 Prüfungsplan:

Thema	Einzelheiten
Thema 1	- Identify the capabilities and implications of the sales process - Describe capabilities and use cases for the approval process
Thema 2	- Describe the capabilities of the Salesforce Mobile App - Describe the capabilities of lead automation tools and campaign management
Thema 3	- Distinguish between the various UI features that an administrator controls - Describe the capabilities of the Salesforce Communities
Thema 4	- Describe the capabilities of custom report types - Describe the capabilities of case management

>> CRT-101 Prüfungsfrage <<

CRT-101 Prüfungsfragen Prüfungsvorbereitungen, CRT-101

CRT-101 Test Dumps, CRT-101 VCE Engine Ausbildung, CRT-101 aktuelle Prüfung

P.S. Kostenlose 2026 Salesforce JS-Dev-101 Prüfungsfragen sind auf Google Drive freigegeben von PrüfungFrage verfügbar:
<https://drive.google.com/open?id=1ZfZIIKexsJiizc0uZEvPJGs0aNlbWHFy>

Die Zertifikat der Salesforce JS-Dev-101 ist international anerkannt. Sie zu erwerben bedeutet, dass Sie den Schlüssel zur höheren Stelle besitzen. Die Salesforce JS-Dev-101 Prüfungsunterlagen von PrüfungFrage werden von erfahrenen IT-Profis herstellt und immer wieder aktualisiert. Jetzt können Sie mit günstigem Preis die verlässliche Salesforce JS-Dev-101 Prüfungsunterlagen genießen. Nachdem Sie die Zertifizierung erworbt haben, können Sie leicht eine höhere Arbeitsposition oder Gehalten bekommen.

Salesforce JS-Dev-101 Prüfungsplan:

Thema	Einzelheiten
Thema 1	Asynchronous Programming: Covers asynchronous programming concepts and understanding how the event loop controls execution flow and determines outcomes.
Thema 2	Server Side JavaScript: Covers Node.js implementations, CLI commands, core modules, and package management solutions for given scenarios.
Thema 3	- 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.

Thema 4	• 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.
---------	---

>> JS-Dev-101 Fragenpool <<

Salesforce JS-Dev-101 Fragen und Antworten, Salesforce Certified JavaScript Developer - Multiple Choice Prüfungsfragen

Wir alle wissen, dass einige IT-Zertifikate zu bekommen ist in der heutigen konkurrenzfähigen Gesellschaft ganz notwendig ist. Das IT-Zertifikat ist der beste Beweis für Ihre Fachkenntnisse. Die Salesforce JS-Dev-101 Zertifizierungsprüfung ist eine wichtige Zertifizierungsprüfung. Aber es ist schwer, die Prüfung zu bestehen. Es ist doch wert, Geld für ein Ausbildungsinstitut auszugeben, um im Beruf befördert zu werden. PrüfungFrage hat die zielgerichteten Schulungsunterlagen zur Salesforce JS-Dev-101 Zertifizierungsprüfung, deren Ähnlichkeit mit den echten Prüfungen 95% beträgt. Wenn Sie an der Ausbildung von PrüfungFrage teilnehmen, können Sie dann 100% die Prüfung bestehen. Sonst geben wir Ihnen eine Rückerstattung.

Salesforce Certified JavaScript Developer - Multiple Choice JS-Dev-101 Prüfungsfragen mit Lösungen (Q51-Q56):

51. Frage

Refer to the code below?

Let searchString = ' look for this ';

Which two options remove the whitespace from the beginning of searchString?

Choose 2 answers

- A. searchString.replace(/*\s*/, "");
- B. trimStart(searchString);
- C. searchString.trimStart();
- D. searchString.trimEnd();

Antwort: A,C

52. Frage

Refer to the code below:

```
new Promise((resolve, reject) => {
  const fraction = Math.random();
  if (fraction > 0.5) reject("fraction > 0.5, " + fraction);
  resolve(fraction);
})
.then(() => console.log("resolved"))
.catch((error) => console.error(error))
.finally(() => console.log("when am I called?"));
```

```

new Promise((resolve, reject) => {
  const fraction = Math.random();
  if( fraction > 0.5) reject("fraction > 0.5, " + fraction);
  reject(fraction);
})
.then(() => console.log("resolved"))
.catch((error) => console.error(error))
.finally(() => console.log(" when am I called?"));

```

0.024493713608408196 VM1560:7

when am I called? VM1560:8

Promise {<fulfilled>: undefined}

```

new Promise((resolve, reject) => {
  const fraction = Math.random();
  if( fraction > 0.5) reject("fraction > 0.5, " + fraction);
  resolve(fraction);
})
.then(() => console.log("resolved"))
.catch((error) => console.error(error))
.finally(() => console.log(" when am I called?"));

```

resolved VM1349:6

when am I called? VM1349:8

Promise {<fulfilled>: undefined}

When does Promise.finally on line 08 get called?

- A. When rejected
- B. When resolved
- C. When resolved or rejected
- D. When resolved and settled

Antwort: C

53. Frage

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

Antwort: A

Begründung:

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.

54. Frage

A developer has a `fizzbuzz` function that, when passed in a number, returns the following:

'fizz' if the number is divisible by 3.

'buzz' if the number is divisible by 5.

'fizzbuzz' if the number is divisible by both 3 and 5.

Empty string "" if the number is divisible by neither 3 nor 5.

Which two test cases properly test scenarios for the `fizzbuzz` function?

- A. `let res = fizzbuzz(3);`
`console.assert(res === "");`
- B. `let res = fizzbuzz(5);`
`console.assert(res === 'fizz');`
- C. `let res = fizzbuzz(true);`
`console.assert(res === "");`
- D. `let res = fizzbuzz(15);`
`console.assert(res === 'fizzbuzz');`

Antwort: C,D

Begründung:

:

First, recall the `fizzbuzz` rules:

If `n` divisible by 3 and not by 5 → return 'fizz'.

If `n` divisible by 5 and not by 3 → return 'buzz'.

If `n` divisible by both 3 and 5 → return 'fizzbuzz'.

Otherwise → return "" (empty string).

Evaluate each test:

Option A:

```
let res = fizzbuzz(true);  
console.assert(res === "");
```

In JavaScript, when using arithmetic operations with `true`, it is coerced to the number 1. A typical implementation of `fizzbuzz` would treat the argument as a number and compute:

`true % 3 → 1 % 3 → 1`

`true % 5 → 1 % 5 → 1`

So `true` is effectively treated as `1`, which is divisible by neither `3` nor `5`. The function should return `"`. The assertion `res === ""` will pass.

This test covers the scenario "neither divisible by `3` nor `5`".

Option B:

```
let res = fizzbuzz(3);
```

```
console.assert(res === "");
```

`3` is divisible by `3` but not by `5`.

According to `fizzbuzz` rules, `fizzbuzz(3)` should return `'fizz'`.

This test expects `"`, so it is incorrect; the assertion would fail in a correct implementation.

Option C:

```
let res = fizzbuzz(5);
```

```
console.assert(res === 'fizz');
```

`5` is divisible by `5` but not by `3`.

According to `fizzbuzz` rules, `fizzbuzz(5)` should return `'buzz'`.

This test expects `'fizz'`, so it is incorrect.

Option D:

```
let res = fizzbuzz(15);
```

```
console.assert(res === 'fizzbuzz');
```

`15` is divisible by both `3` and `5`.

According to `fizzbuzz` rules, `fizzbuzz(15)` should return `'fizzbuzz'`.

This assertion correctly matches the expected result, so it is a proper test case.

Thus, the two test cases that correctly test valid `fizzbuzz` behavior are:

Answer: A, D

Study Guide / Concept Reference (no links):

Modulo operator (`%`) for divisibility checks

Truthy/number coercion of boolean `true` in arithmetic (`Number(true) === 1`) Designing unit tests with `console.assert` Typical `fizzbuzz` logic structure and expected outputs

55. Frage

Given the code below:

```

01 setTimeout(() => {
02   console.log(1);
03 }, 1100);
04 console.log(2);
05 new Promise((resolve, reject) => {
06   setTimeout(() => {
07     reject(console.log(3));
08   }, 1000);
09 }).catch(() => {
10   console.log(4);
11 });
12 console.log(5);

```

What is logged to the console?

- A. 1 2 3 4 5
- B. 1 2 5 3 4
- C. 2 5 3 4 1
- D. 2 5 1 3 4

Antwort: C

56. Frage

.....

JS-Dev-101 ist eine Salesforce Zertifizierungsprüfung. So ist JS-Dev-101 Zertifizierung der erste Schritt zur Salesforce Zertifizierung. Deswegen ist die JS-Dev-101 Zertifizierungsprüfung kürzlich immer beliebter geworden. Immer mehr Leute haben sich an der Salesforce JS-Dev-101 Zertifizierungsprüfung beteiligt. Aber die Erfolgsquote in der Prüfung ist nicht so hoch. Wählen Sie auch einschlägige Prüfungskurse, wenn Sie JS-Dev-101 Prüfung ablegen möchten?

JS-Dev-101 Probesfragen: <https://www.pruefungfrage.de/JS-Dev-101-dumps-deutsch.html>

- Neueste JS-Dev-101 Pass Guide - neue Prüfung JS-Dev-101 braindumps - 100% Erfolgsquote ☐ Öffnen Sie die Webseite $\Rightarrow$ www.zertpruefung.ch $\Leftarrow$ und suchen Sie nach kostenloser Download von $\Rightarrow$ JS-Dev-101 ☐ ☐ ☐ JS-Dev-101 Prüfungsübungen
- JS-Dev-101 Prüfungsinformationen ☐ JS-Dev-101 Prüfungsunterlagen ☐ JS-Dev-101 Originale Fragen ☐ Suchen Sie einfach auf $\triangleright$ www.itzert.com ☐ nach kostenloser Download von $\Rightarrow$ JS-Dev-101 ☐ ☐ ☐ JS-Dev-101 Schulungsunterlagen
- JS-Dev-101 Musterprüfungsfragen ☐ JS-Dev-101 Musterprüfungsfragen ☐ JS-Dev-101 Schulungsangebot ☐ Suchen Sie einfach auf ☐ de.fast2test.com ☐ nach kostenloser Download von ☐ JS-Dev-101 ☐ ☐ JS-Dev-101 Buch
- Seit Neuem aktualisierte JS-Dev-101 Examfragen für Salesforce JS-Dev-101 Prüfung ☐ Öffnen Sie ☐ www.itzert.com ☐ geben Sie (JS-Dev-101) ein und erhalten Sie den kostenlosen Download ☐ JS-Dev-101 Ausbildungsressourcen
- JS-Dev-101 Mit Hilfe von uns können Sie bedeutendes Zertifikat der JS-Dev-101 einfach erhalten! ☐ Suchen Sie auf ☐ www.zertfragen.com ☐ nach kostenlosem Download von ☐ JS-Dev-101 ☐ JS-Dev-101 Zertifikatsfragen

- Übrigens, Sie können die vollständige Version der PrüfungFrage JS-Dev-101 Prüfungsfragen aus dem Cloud-Speicher herunterladen: <https://drive.google.com/open?id=1ZfZILKesXJiuz0uZEvPJG0aNIbWHFy>