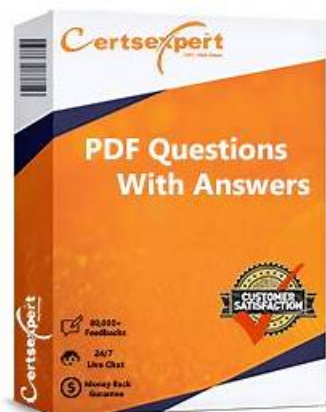


# Marvelous New JS-Dev-101 Exam Name & Leader in Qualification Exams & 100% Pass-Rate JS-Dev-101: Salesforce Certified JavaScript Developer - Multiple Choice



P.S. Free 2026 Salesforce JS-Dev-101 dumps are available on Google Drive shared by PassCollection:  
<https://drive.google.com/open?id=1nGzgbwgEcmdXtuvPbFto6HU13jcHH4K7>

Free demo is available if you purchase JS-Dev-101 exam dumps from us, so that you can have a better understanding of what you are going to buy. If you are satisfied with the free demo and want to buying JS-Dev-101 exam dumps from us, you just need to add to cart and pay for it. You can receive the download link and password within ten minutes for JS-Dev-101 Exam Materials, so that you can start your practicing as quickly as possible. In addition, in order to build up your confidence for the JS-Dev-101 exam dumps, we are pass guarantee and money back guarantee. If you fail to pass the exam, we will give you full refund.

## Salesforce JS-Dev-101 Exam Syllabus Topics:

Topic	Details
Topic 1	<ul style="list-style-type: none"><li>• Debugging and Error Handling: Covers proper error handling techniques and the use of the console and breakpoints to debug code.</li></ul>
Topic 2	<ul style="list-style-type: none"><li>• 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.</li></ul>
Topic 3	<ul style="list-style-type: none"><li>• Browser and Events: Covers DOM manipulation, event handling and propagation, browser-specific APIs, and using Browser Developer Tools to inspect code behavior.</li></ul>

Topic 4	<ul style="list-style-type: none"> <li>• <b>Testing:</b> Covers evaluating unit test effectiveness against a block of code and modifying tests to improve their coverage and reliability.</li> </ul>
Topic 5	<ul style="list-style-type: none"> <li>• <b>Server Side JavaScript:</b> Covers Node.js implementations, CLI commands, core modules, and package management solutions for given scenarios.</li> </ul>
Topic 6	<ul style="list-style-type: none"> <li>• <b>Asynchronous Programming:</b> Covers asynchronous programming concepts and understanding how the event loop controls execution flow and determines outcomes.</li> </ul>

>> New JS-Dev-101 Exam Name <<

## Pass Guaranteed 2026 Salesforce High Hit-Rate New JS-Dev-101 Exam Name

Where there is life, there is hope. Never abandon yourself. You still have many opportunities to counterattack. If you are lack of knowledge and skills, our JS-Dev-101 guide questions are willing to offer you some help. Actually, we are glad that our JS-Dev-101 Study Materials are able to become you top choice. Just look at the warm feedbacks from our JS-Dev-101 learning braindumps, we are very popular in the whole market. And our JS-Dev-101 exam guide won't let you down.

### Salesforce Certified JavaScript Developer - Multiple Choice Sample Questions (Q68-Q73):

#### NEW QUESTION # 68

Refer to following code block:

```
Let array = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,];
```

```
Let output =0;
```

```
For (let num of array){
```

```
if (output >0){
```

```
Break;
```

```
}
```

```
if(num % 2 == 0){
```

```
Continue;
```

```
}
```

```
Output +=num;
```

What is the value of output after the code executes?

- **A. 0**
- B. 1
- C. 2
- D. 3

**Answer: A**

#### NEW QUESTION # 69

A developer wants to create an object from a function in the browser using the code below:

```
Function Monster() { this.name = 'hello' };
```

```
Const z = Monster();
```

What happens due to lack of the new keyword on line 02?

- A. The z variable is assigned the correct object but this.name remains undefined.
- B. The z variable is assigned the correct object.
- C. Window.m is assigned the correct object.
- **D. Window.name is assigned to 'hello' and the variable z remains undefined.**

**Answer: D**

### NEW QUESTION # 70

Code:

```
01 const sayHello = (name) => {  
02 console.log('Hello ', name);  
03 };  
04  
05 const world = () => {  
06 return 'World';  
07 };  
08  
09 sayHello(world);
```

This does not print "Hello World".

What change is needed?

- A. Change line 2 to `console.log('Hello', name());`;
- B. Change line 7 to `}()`;
- C. Change line 9 to `sayHello(world)()`;
- D. Change line 5 to `function world() {`

**Answer: A**

Explanation:

Comprehensive and Detailed Explanation From Exact Extract JavaScript Knowledge:

Currently:

`sayHello` expects a value `name` and prints it.

`world` is a function that returns 'World'.

`sayHello(world)`; passes the function object itself, not the result of calling it.

So `name` inside `sayHello` is a function, not the string "World".

To keep the call `sayHello(world)` yet get "World", `sayHello` must call the function parameter:

```
const sayHello = (name) => {  
  console.log('Hello', name());  
};
```

Now:

`sayHello(world)` passes the function.

Inside `sayHello`, `name()` calls `world()`, which returns "World".

The console logs "Hello World".

Why the others are wrong:

B: Changing line 7 to `}()`; would attempt to IIFE the function definition and break the declaration.

C: `sayHello(world)()` would try to call the return value of `sayHello`, which is undefined, causing an error.

D: Changing `world` to a function declaration does not change the fact that it is passed as a function reference; `sayHello` still prints the function object, not 'World'.

---

### NEW QUESTION # 71

Refer to the code below:

Which replacement for the conditional statement on line 02 allows a developer to correctly determine that a specific element, `myElement` on the page had been clicked?

**Answer:**

Explanation:

```
event.target.id === 'myElement'
```

### NEW QUESTION # 72

A developer wants to use a `try...catch` statement to catch any error that `countSheep ()` may throw and pass it to a `handleError ()` function.

What is the correct implementation of the `try...catch`?

- A. `try {`

