

JS-Dev-101資格認定、JS-Dev-101トレーニング



BONUS!!! JPNTTest JS-Dev-101ダンプの一部を無料でダウンロード: <https://drive.google.com/open?id=1EGXqxT2xK5wFmNde4ej3gBpKk74ouZ2c>

誰もが私たちの人生の貴重を認識する必要があります。時間を無駄にすることはできないので、目標をまっすぐに達成するための良い方法が必要です。もちろん、最新のJS-Dev-101試験トレントが最適です。JS-Dev-101試験の質問から、認定試験の知識だけでなく、質問に迅速かつ正確に回答する方法を学ぶことができることをお約束します。今、JS-Dev-101テストトレントのデモを無料でダウンロードして、素晴らしい品質を確認できます。

JS-Dev-101試験クイズを購入する前に、より快適な体験をお約束するために、JPNTTest体験版サービスを提供しています。JS-Dev-101学習教材の購入を決定したら、終日サービスも提供します。ご質問がある場合は、当社Salesforceのスペシャリストにお問い合わせください。思いやりのあるサービスを提供します。また、JS-Dev-101トレーニングガイドでJS-Dev-101試験に合格することをお勧めします。信頼できるサービスにより、当社のJS-Dev-101のSalesforce Certified JavaScript Developer - Multiple Choice学習教材は決して失望させません。

>> JS-Dev-101資格認定 <<

試験の準備方法-実地的なJS-Dev-101資格認定試験-正確的なJS-Dev-101トレーニング

業界のリーダーとなっているために、我々は確かに独自のリソースを拡大し続ける必要があります。我々JPNTTestは常に試験問題集とソフトウェアの内容を更新します。だから、あなたの使用しているSalesforceのJS-Dev-101試験のソフトウェアは、最新かつ最も全面的な問題集を確認することができます。あなたのSalesforceのJS-Dev-101試験準備のどの段階にあっても、当社のソフトウェアは、あなたの最高のヘルパープロフォーマになることができます。我々はSalesforceのJS-Dev-101試験のデータを整理したり、分析したりするため、経験豊富なエリートチームにそれを完了させます。

Salesforce JS-Dev-101 認定試験の出題範囲:

トピック	出題範囲
トピック 1	<ul style="list-style-type: none"> 非同期プログラミング: 非同期プログラミングの概念と、イベントループが実行フローを制御し、結果を決定する仕組みについて解説します。
トピック 2	<ul style="list-style-type: none"> テスト: コードブロックに対する単体テストの有効性を評価し、テストを修正してカバレッジと信頼性を向上させることを含みます。
トピック 3	<ul style="list-style-type: none"> 変数、型、コレクション: 変数の宣言と初期化、文字列、数値、日付、配列、JSONの操作、型変換と真偽判定の理解について解説します。
トピック 4	<ul style="list-style-type: none"> デバッグとエラー処理: 適切なエラー処理手法、およびコンソールとブレークポイントを使用したコードのデバッグ方法について説明します。
トピック 5	<ul style="list-style-type: none"> オブジェクト、関数、クラス: ビジネス要件を満たすための関数、オブジェクト、クラスの実装に加え、モジュール、デコレータ、変数スコープ、実行フローの使用方法について解説します。

Salesforce Certified JavaScript Developer - Multiple Choice 認定 JS-Dev-101 試験問題 (Q115-Q120):

質問 # 115

Refer to the code:

```
01 function execute() {
02 return new Promise((resolve, reject) => reject());
03 }
04 let promise = execute();
05
06 promise
07 .then() => console.log('Resolved1')
08 .then() => console.log('Resolved2')
09 .then() => console.log('Resolved3')
10 .catch() => console.log('Rejected')
11 .then() => console.log('Resolved4');
```

What is the result when the Promise in the execute function is rejected?

- A. Resolved1 Resolved2 Resolved3 Rejected Resolved4
- **B. Rejected Resolved4**
- C. Rejected
- D. Resolved1 Resolved2 Resolved3 Resolved4

正解: **B**

解説:

Comprehensive and Detailed Explanation From Exact Extract JavaScript Knowledge:

execute() returns a Promise that immediately calls reject().

So promise starts in a rejected state.

When a Promise is rejected and you chain .then() calls without rejection handlers, all those .then() callbacks are skipped until a .catch() is encountered:

```
promise
.then(...) // skipped
.then(...) // skipped
.then(...) // skipped
.catch(...) // executed
.then(...); // executed after catch
```

Execution:

```
.then() => console.log('Resolved1') is skipped.
.then() => console.log('Resolved2') is skipped.
```

.then() => console.log('Resolved3')) is skipped.
.catch() => console.log('Rejected')) runs and logs Rejected.
The .catch() returns a resolved Promise (no explicit return, so undefined), so the next .then() runs:
.then() => console.log('Resolved4')) logs Resolved4.
Final output:
Rejected
Resolved4
This matches option D.

質問 # 116

Which three actions can be using the JavaScript browser console?
Choose 3 answers:

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

正解: A、C、E

質問 # 117

What is the result of the code block?

- A. An error is thrown.
- B. The console logs 'flag' and then an error is thrown.
- C. The console logs only 'flag'.
- D. The console logs 'flag' and another flag.

正解: C

質問 # 118

In which situation should a developer include a try .. catch block around their function call ?

- A. The function contains scheduled code.
- B. The function results in an out of memory issue.
- C. The function has an error that should not be silenced.
- D. The function might raise a runtime error that needs to be handled.

正解: D

質問 # 119

A developer wants to set up a secure web server with Node.js. The developer creates a directory locally called app-server, and the first file is app-server/index.js Without using any third-party libraries, what should the developer add to index.js to create the secure web server?

- A. const tls = require('tls');
- B. const http = require('http');
- C. const https = require('https');
- D. const server = require('secure-server');

正解: C

