

# TestInsides



## Quality and Value

TestInsides Practice Exams are written to the highest standards of technical accuracy, using only certified subject matter experts and published authors for development - no all vce.



## Tested and Approved

We are committed to the process of vendor and third party approvals. We believe professionals and executives alike deserve the confidence of quality coverage these authorizations provide.



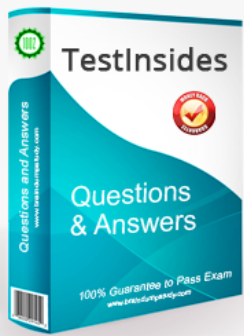
## Easy to Pass

If you prepare for the exams using our TestInsides testing engine, It is easy to succeed for all certifications in the first attempt. You don't have to deal with all dumps or any free torrent / rapidshare all stuff.



## Try Before Buy

TestInsides offers free demo of each product. You can check out the interface, question quality and usability of our practice exams before you decide to buy.



## Try before you buy

Download a free sample of any of our exam questions and answers

- ✓ 24/7 customer support, Secure shopping site
- ✓ Free One year updates to match real exam scenarios
- ✓ If you failed your exam after buying our products we will refund the full amount back to you.

## Choose an exam to sample

Select a vendor... ▼

Select an exam... ▼

Your email address

Download Now

## About Full Refund

Candidates have own unqualified certificate scanned and then sent to our mailbox. After confirmation, we will refund. ([Contact now](#))

## Refund Policy

1. Questions must be purchased before the exam.
2. The transcript's subject must be the same subject when you bought the questions.

<http://www.testinsides.top/>

Testinsides offers the best valid test dumps and test questions

**Exam** : **CRT-600J**

**Title** : Salesforce Certified  
JavaScript Developer I  
(CRT-600日本語版)

**Vendor** : Salesforce

**Version** : DEMO

**QUESTION NO: 1**

開発者は、以下のコードを使用して、ブラウザの関数からオブジェクトを作成したいと考えています。

```
01 function Monster(){ this.name = 'hello' };  
02 const m = Monster();
```

行 02 に mm キーワードがないために何が起こりますか？

- A. window.name は 'hello' に割り当てられ、変数 = は未定義のままです。
- B. window.m には正しいオブジェクトが割り当てられています。
- C. m 変数には正しいオブジェクトが割り当てられていますが、this.name は未定義のままです。
- D. m 変数には正しいオブジェクトが割り当てられます。

**Answer:** A

**QUESTION NO: 2**

以下のコードを参照してください。

```
01 let total = 10;  
02 const interval = setInterval(() => {  
03   total++;  
04   clearInterval(interval);  
05   total++;  
06 }, 0);  
07 total++;  
08 console.log(total);
```

JavaScript がシングルスレッドであることを考えると、コード実行後の 08 行目の出力は何ですか？

- A. 10
- B. 11
- C. 12
- D. 13

**Answer:** B

**QUESTION NO: 3**

以下のコードを参照してください。  
productSKU = '8675309' とします。

開発者には、「sku」で始まり、ゼロが埋め込まれた、常に 19 文字の長さの SKU 番号を生成する必要があります。

値 sku000000008675309 を割り当ててるステートメントはどれですか？

- A. `productSKU = productSKU .padStart (19. '0').padstart('sku');`
- B. `productSKU = productSKU .padEnd (16. '0').padstart('sku');`
- C. `productSKU = productSKU .padEnd (16. '0').padstart(19, 'sku');`
- D. `productSKU = productSKU .padStart (16. '0').padstart(19, 'sku');`

**Answer:** D

#### QUESTION NO: 4

式 var1 と var2 が与えられた場合、2

つの式の連結を返し、それが文字列であることを確認する 2 つの有効な方法は何ですか？

2つの答えを選択してください

- A. `var1 + var2`
- B. `var1.toString ( ) var2.toString ( )`
- C. 文字列 `(var1) .concat (var2)`
- D. `string.concat (var1 +var2)`

**Answer:** B,D

#### QUESTION NO: 5

開発者は、universalContainersLib

という名前のモジュールを使用したいと考えており、そこから関数を呼び出します。

開発者はモジュールからすべての関数をインポートし、関数 foo と bar

をどのように呼び出す必要がありますか？

- A. `import * as lib from '/path/universalContainersLib.js';`  
`lib.foo();`  
`lib.bar();`
- B. `import (foo, bar) from '/path/universalContainersLib.js';`  
`foo();`  
`bar();`
- C. `import all from '/path/universalContainerLib.js';`  
`universalContainersLib.foo();`  
`universalContainersLib.bar();`
- D. `import * from '/path/universalContainerLib.js';`  
`universalContainersLib.foo();`  
`universalContainersLib.bar();`

**Answer:** A

#### QUESTION NO: 6

開発者は、入力フィールドを持つ単純な Web

ページを作成します。ユーザーが入力フィールドにテキストを入力してボタンをクリックすると、フィールドの実際の値がコンソールに表示される必要があります。

HTML ファイルの内容は次のとおりです。

```
<input type="text" value="Hello" name="input">
```

<button type="button" >表示</button>

開発者は以下の JavaScript コードを記述しました。

```
Const button = document.querySelector('button');
```

```
button.addEventListener('click', () => (
```

```
Const input = document.querySelector('input');
```

```
console.log(input.getAttribute('value'));
```

ユーザーがボタンをクリックすると、出力は常に「Hello」になります。

このコードが期待どおりに機能するには、何をする必要がありますか？

A. 行 04 を console.log(input .value); に置き換えます。

B. 行 03 を const input = document.getElementById('input'); に置き換えます。

C. 02 行目を button.addEventListener("onclick", function() {

D. 02 行目を button.addCallback("click", function() {

**Answer: A**

### QUESTION NO: 7

開発者は、try...catch ステートメントを使用して、countSheep ()

がスローする可能性のあるエラーをキャッチし、それを handleError ()

関数に渡したいと考えています。

try...catch の正しい実装は何ですか？

A)

```
try {  
    setTimeout(function() {  
        countSheep();  
    }, 1000);  
} catch (e) {  
    handleError(e);  
}
```

B)

```
try {  
    countSheep();  
} finally {  
    handleError(e);  
}
```

ハ)

```
setTimeout(function() {  
  try {  
    countSheep();  
  } catch (e) {  
    handleError(e);  
  }  
}, 1000);
```

D)

```
try {  
  countSheep();  
} handleError (e){  
  catch(e);
```

- A. オプション
- B. オプション
- C. オプション
- D. オプション

**Answer:** A

#### QUESTION NO: 8

開発者がページのロード間でデータを保持するために使用できるブラウザ固有の API を 3 つ選択してください。

3つの答えを選択してください

- A. IIFE
- B. indexedDB
- C. グローバル変数
- D. クッキー
- E. ローカル ストレージ。

**Answer:** A,B,E

#### QUESTION NO: 9

Universal Containers では、すべてのチームが JavaScript オブジェクトをコピーする独自の方法を持っています。コード スニペットは、1 つのチームの実装を示しています。

```
01 function Person() {
02   this.firstName = "John";
03   this.lastName = "Doe";
04   this.name = () => {
05     console.log(`Hello ${this.firstName} ${this.lastName}`);
06   }
07 }
08
09 const john = new Person();
10 const dan = JSON.stringify(JSON.parse(john));
11 dan.firstName = 'Dan';
12 dan.name();
```

コード実行の出力は何ですか？

- A. こんにちは John Doe
- B. こんにちはダン
- C. こんにちは、Dan Doe
- D. SyntaxError: JSON に予期しないトークンがあります

**Answer:** D

#### QUESTION NO: 10

コア Node.js モジュールのオプションはどれですか？

- A. パス
- B. イオス
- C. メモリ
- D. ロケート

**Answer:** A

#### QUESTION NO: 11

開発者は、universalContainersLib

という名前のモジュールを使用して、そこから関数を呼び出したいと考えています。

開発者はどのようにモジュールからすべての関数をインポートし、関数 foo と bar を呼び出す必要がありますか？

- A. `import * from '/path/universalContainersLib.js';`  
`universalContainersLib.foo ()`  
`universalContainersLib.bar ();`
- B. `import {foo,bar} from '/path/universalCcontainersLib.js';`  
`foo();`  
`bar();`
- C. `import all from '/path/universalContainersLib.js';`  
`universalContainersLib.foo();`

```
universalContainersLib.bar ();
```

```
D. import * as lib from '/path/universalContainersLib.js';
```

```
lib.foo();
```

```
lib. bar ();
```

**Answer:** D

### QUESTION NO: 12

次のコードを参照してください。

```
sampleText = 'ザ クイック ブラウン フォックス ジャンプ';
```

開発者は、特定の部分文字列が文字列の一部であるかどうかを判断する必要があります。

指定された部分文字列に対して true を返す 3 つの式はどれですか？

3つの答えを選択してください

A. sampleText.includes('fox');

B. sampleText.includes(' quick ', 4);

C. sampleText.includes(' Fox ', 3)

D. sampleText.includes(' fox ');

E. sampleText.includes(' quick ') !== -1;

**Answer:** B,D,E

### QUESTION NO: 13

以下の文字列を参照してください。

```
Const str='Salesforce';
```

「販売」という単語になる 2 つのステートメントはどれですか？

**Answer:**

```
Str.substring(0,5);
```

```
Str.substr(0,5);
```

### QUESTION NO: 14

次のコードがあるとします。

```
x = null;
```

```
console.log(typeof x);
```

出力は何ですか？

A. "object"

B. "undefined"

C. "null"

D. "x"

**Answer:** A

### QUESTION NO: 15

以下のコードを参照してください。

```
new Promise((解決、拒否) => {
```

```
const fraction = Math.random();
```

```
if( 分数 >0.5) reject("分数 > 0.5, " + 分数);
```

```
解決(分数);
```

```

)))
.then(() => console.log("解決済み"))
.catch((エラー) => console.error(エラー))
.finally(() => console.log("いつ呼び出されますか?"));

```

```

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?"));

```

▶ 0.024493713600408196 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}

08 行目の Promise.finally はいつ呼び出されますか？

- A. 拒否された場合
- B. 解決・ 決済時
- C. 解決時
- D. 解決または却下された場合

**Answer:** D

#### QUESTION NO: 16

以下のコードを参照してください。

```

const イベント = 新しい CustomEvent(
//コードがありません
);

```

```

obj.dispatchEvent(イベント);

```

開発者は update というカスタム イベントをディスパッチして、recordId に関する情報を送信する必要があります。

これを実現するために開発者が 02 行目のプレースホルダーに挿入できる 2

つのオプションはどれですか？  
2つの答えを選択してください

- A. 'Update', (  
recordId : '123abc'  
(
- B. 'Update', '123abc'
- C. { type : 'update', recordId : '123abc' }
- D. 'Update', {  
Details : {  
recordId : '123abc'  
}  
}

**Answer:** A,D

}

#### QUESTION NO: 17

以下のコードを参照してください。

```
01 function Person() {  
02   this.firstName = 'John';  
03 }  
04  
05 Person.prototype = {  
06   job: x => 'Developer'  
07 };  
08  
09 const myFather = new Person();  
10 const result = myFather.firstName + ' ' + myFather.job();
```

10 行目を実行した後の結果の値は？

- A. エラー: myFather.job は関数ではありません
- B. ジョン開発者
- C. 未定義の開発者
- D. ジョン未定義

**Answer:** B

#### QUESTION NO: 18

以下のコードを参照してください。

```
01 const サーバー = require('サーバー');  
02 /*ここにコードを挿入*/
```

開発者は、Web

サーバーを作成するライブラリをインポートします。インポートされたライブラリは、イベントとコールバックを使用してサーバーを起動します イベントを設定して Webサーバーを起動するには、行 03 にどのコードを挿入する必要がありますか？

- A. `Server.start ();`
- B. `server.on(' connect ', ( port) => { console.log('Listening on ', port);})`
- C. `server()`
- D. `serve(( port) => (`
- E. `console.log( 'Listening on ', port);`

**Answer:** B

### QUESTION NO: 19

商品または売上の現在の価格を返す `currPrice` メソッドを開発者が実装するための新しい要件があります。

```

01 let regItem = new Item('Scarf', 50); // Name, Price
02 let saleItem = new SaleItem('Shirt', 80, .1); // Name, Price, Discount
03 Item.prototype.currPrice = function() { return this.price; }
04 console.log(regItem.currPrice());
05 console.log(saleItem.currPrice());
06
07 SaleItem.prototype.currPrice = function() { return this.price - (this.price * this.discount); }
08 console.log(regItem.currPrice());
09 console.log(saleItem.currPrice());

```

上記のコードを実行したときの出力は何ですか

- A. 50  
キャッチされていない `TypeError: saleItem、description は関数ではありません`  
50  
80
- B. 50  
80  
50  
72
- C. 50  
80  
72
- D. 50  
80  
不明な参照エラー: この割引は未定義です  
72

**Answer:** B

### QUESTION NO: 20

通常関数定義と比較して、太い矢印で定義された関数の 2 つのユニークな特徴は何ですか？

2つの答えを選択してください

**A.** 関数は独自の this

を生成し、関数のスコープをそれを囲むスコープから分離するのに役立ちます。

**B.** この関数は、parentThis

と呼ばれる常にスコープ内にある引数を受け取ります。これは、囲んでいるレキシカルスコープです。

**C.** 関数の本体に単一の式がある場合、その式は評価され、暗黙的に返されます。

**D.** 関数は、外側のスコープから this を使用します。

**Answer:** A,B