



## **Salesforce**

### **Exam Questions JavaScript-Developer-I**

Salesforce Certified JavaScript Developer I

### NEW QUESTION 1

A developer is wondering whether to use, Promise.then or Promise.catch, especially when a Promise throws an error? Which two promises are rejected? Which 2 are correct?

- A. Promise.reject('cool error here').then(error => console.error(error));
- B. Promise.reject('cool error here').catch(error => console.error(error));
- C. New Promise((resolve, reject) => (throw 'cool error here')).catch(error => console.error(error)) ;
- D. New Promise(() => (throw 'cool error here')).then(null, error => console.error(error));

**Answer:** BC

### NEW QUESTION 2

Refer to the code below:

```

01 let o = {
02   get js() {
03     let city1 = String('St.Louis');
04     let city2 = String('New York');
05
06     return {
07       firstCity: city1.toLowerCase(),
08       secondCity: city2.toLowerCase(),
09     }
10   }
11 }

```

What value can a developer expect when referencing o.js.secondCity?

- A. Undefined
- B. ' new york '
- C. ' New York '
- D. An error

**Answer:** B

### NEW QUESTION 3

Refer to the code below: Let str = 'javascript'; Str[0] = 'J'; Str[4] = 'S';

After changing the string index values, the value of str is 'javascript'. What is the reason for this value:

- A. Non-primitive values are mutable.
- B. Non-primitive values are immutable.
- C. Primitive values are mutable.
- D. Primitive values are immutable.

**Answer:** D

### NEW QUESTION 4

A team that works on a big project uses npm to deal with projects dependencies. A developer added a dependency does not get downloaded when they executenpm install.

Which two reasons could be possible explanations for this?

Choose 2 answers

- A. The developer missed the option --add when adding the dependency.
- B. The developer added the dependency as a dev dependency, and NODE\_ENVIs set to production.
- C. The developer missed the option --save when adding the dependency.
- D. The developer added the dependency as a dev dependency, and NODE\_ENV is set to production.

**Answer:** BCD

### NEW QUESTION 5

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

```

B)

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

C)

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

D)

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

- A. Option
- B. Option
- C. Option
- D. Option

**Answer: A**

**NEW QUESTION 6**

A developer is creating a simple webpage with a button. When a user clicks this button for the first time, a message is displayed. The developer wrote the JavaScript code below, but something is missing. The message gets displayed every time a user clicks the button, instead of just the first time.

```
01 function listen(event) {
02
03   alert('Hey! I am John Doe');
04
05 }
06 button.addEventListener('click', listen);
```

Which two code lines make this code work as required? Choose 2 answers

- A. On line 02, use event.first to test if it is the first execution.
- B. On line 04, use event.stopPropagation ( ),
- C. On line 04, use button.removeEventListener(' click" , listen);
- D. On line 06, add an option called once to button.addEventListener().

**Answer: CD**

**NEW QUESTION 7**

A class was written to represent items for purchase in an online store, and a second class Representing items that are on sale at a discounted price. THE constructor sets the name to the first value passed in. The pseudocode is below:

```
class Item {
  constructor(name, price){
    ... // Constructor Implementation
  }
}

class SaleItem extends Item {
  constructor(name, price, discount){
    ... // Constructor Implementation
  }
}
```

There is a new requirement for a developer to implement a description method that will return a brief description for Item and SaleItem.

```

01 let regItem = new Item('Scarf', 55);
02 let saleItem = new SaleItem('Shirt', 80, .1);
03 Item.prototype.description = function() { return 'This is a ' + this.name; }
04 console.log(regItem.description());
05 console.log(saleItem.description());
06
07 SaleItem.prototype.description = function() { return 'This is a discounted ' + this.name; }
08 console.log(regItem.description());
09 console.log(saleItem.description());

```

What is the output when executing the code above ?

- A. This is a ScarfUncaught TypeError:saleItem.description is not a function This is aScarfThis is a discounted Shirt
- B. This is a Scarf This is a Shirt This is a ScarfThis is a discounted Shirt
- C. This is a Scarf This is a ShirtThis is a discounted Scarf This is a discounted Shirt
- D. This is aScarfUncaught TypeError: saleItem.description is not a function This is a ShirtThis is a did counted Shirt

**Answer: B**

#### NEW QUESTION 8

A developer is setting up a new Node.js server with a client library that is built using events and callbacks. The library:

\* Will establish a web socket connection and handle receipt of messages to the server

\* Will be imported with require, and made available with a variable called we.

The developer also wants to add error logging if a connection fails.

Given this info, which code segment shows the correct way to set up a client with two events that listen at execution time?

- A.
 

```

04 ws.connect(() => {
05   console.log('Connected to client');
06 }).catch((error) => {
07   console.log('ERROR', error);
08 });

```
- B.
 

```

04 ws.on('connect', () => {
05   console.log('Connected to client');
06
07   ws.on('error', (error) => {
08     console.log('ERROR', error);
09   });
10 });

```
- C.
 

```

04 ws.on('connect', () => {
05   console.log('Connected to client');
06 });
07
08 ws.on('error', (error) => {
09   console.log('ERROR', error);
10 });

```
- D.
 

```

04 try {
05   ws.connect(() => {
06     console.log('Connected to client');
07   });
08 } catch(error) {
09   console.log('ERROR', error);
10 }

```

**Answer: C**

#### NEW QUESTION 9

Refer to the code:

```
01 function Vehicle(name, price) {
02   this.name = name;
03   this.price = price;
04 }
05 Vehicle.prototype.priceInfo = function () {
06   return 'Cost of the ${this.name} is ${this.price}$';
07 }
08 var ford = new Vehicle('Ford Fiesta', '20,000');
```

Given the requirement to refactor the code above to JavaScript class format, which class definition is correct?

A)

```
01 class Vehicle {
02   constructor(name, price) {
03     this.name = name;
04     this.price = price;
05   }
06   priceInfo() {
07     return 'Cost of the ${this.name} is ${this.price}$';
08   }
09 }
```

B)

```
01 class Vehicle {
02   vehicle(name, price) {
03     this.name = name;
04     this.price = price;
05   }
06   priceInfo() {
07     return 'Cost of the ${this.name} is ${this.price}$';
08   }
09 }
```

C)

```
01 class Vehicle {
02   constructor(name, price) {
03     name = name;
04     price = price;
05   }
06   priceInfo() {
07     return 'Cost of the ${this.name} is ${this.price}$';
08   }
09 }
```

D)

```
01 class Vehicle {
02   constructor() {
03     this.name = name;
04     this.price = price;
05   }
06   priceInfo() {
07     return 'Cost of the ${this.name} is ${this.price}$';
08   }
09 }
```

A.

Answer: B

#### NEW QUESTION 10

Refer to the code below:

```
Let car1 = new Promise((_, reject) => setTimeout(reject, 2000, "car 1 crashed in" =>
Let car2 = new Promise(resolve => setTimeout(resolve, 1500, "car 2 completed"))
Let car3 = new Promise(resolve => setTimeout(resolve, 3000, "car 3 completed"))
Promise.race(( car1, car2, car3))
.then (value => (
Let result = `${value} the race.`;))
.catch(arr => {
console.log("Race is cancelled.", err);
});
```

What is the value of result when Promise.race executes?

- A. Car 3 completes the race
- B. Car 2 completed the race.
- C. Car 1 crashed in the race.
- D. Race is cancelled.

**Answer: B**

#### NEW QUESTION 10

A developer needs to test this function:

```
01 const sum3 = (arr) => (
02   if (!arr.length) return 0,
03   if (arr.length === 1) return arr[0],
04   if (arr.length === 2) return arr[0] + arr[1],
05   return arr[0] + arr[1] + arr[2],
06 );
```

Which two assert statements are valid tests for the function? Choose 2 answers

- A. console.assert(sum3(1, '2')) == 12);
- B. console.assert(sum3(0)) == 0);
- C. console.assert(sum3(-3, 2)) == -1);
- D. console.assert(sum3('hello', 2, 3, 4)) === NaN);

**Answer: AC**

#### NEW QUESTION 12

A developer wrote the following code to test a sum3 function that takes in an array of numbers and returns the sum of the first three numbers in the array, and the test passes.

A different developer made changes to the behavior of sum3 to instead sum only the first two numbers present in the array.

```
01 let res = sum3([1, 4, 1]);
02 console.assert(res === 6);
03
04 res = sum3([1, 5, 0, 5]);
05 console.assert(res === 6);
```

Which two results occur when running this test on the updated sum3 function? Choose 2 answers

- A. The line 05 assertion passes.
- B. The line 02 assertion passes.
- C. The line 02 assertion fails.
- D. The line 05 assertion fails.

**Answer: BD**

#### NEW QUESTION 16

Refer to the HTML below:

```
<div id="main">
<ul>
<li>Leo</li>
<li>Tony</li>
<li>Tiger</li>
</ul>
</div>
```

Which JavaScript statement results in changing " Tony" to "Mr. T."?

- A. document.querySelectorAll('\$main \$TONY').innerHTML = ' M
- B. ';
- C. document.querySelector('\$main li:second-child').innerHTML = ' M
- D. ';
- E. document.querySelector('\$main li.Tony').innerHTML = ' M
- F. ';
- G. document.querySelector('\$main li:nth-child(2)'),innerHTML = ' M
- H. ';

**Answer: D**

#### NEW QUESTION 20

Refer to the code below:

```
1 let car1 = new Promise((_, reject) =>
2   setTimeout(reject, 2000, "Car 1 crashed in"));
3 let car2 = new Promise(resolve => setTimeout(resolve, 1500, "Car 2 completed"));
4 let car3 = new Promise(resolve => setTimeout(resolve, 3000, "Car 3 Completed"));
5 Promise.race([car1, car2, car3])
6 .then(value => (
7   let result = $(value) the race. `; 08 ))
9 catch( arr => (
10  console.log("Race is cancelled.", err); 11 ));
```

What is the value of result when Promise.race executes?

- A. Car 3 completed the race.
- B. Car 1 crashed in the race.
- C. Car 2 completed the race.
- D. Race is cancelled.

**Answer: C**

**NEW QUESTION 25**

Given the following code: `Let x =null; console.log(typeof x);`  
 What is the output of the line 02?

- A. "Null"
- B. "X"
- C. "Object"
- D. "undefined"

**Answer: C**

**NEW QUESTION 27**

Refer to HTML below:

`<p> The current status of an Order: <span id ="status"> In Progress </span> </p>.`

Which JavaScript statement changes the text 'In Progress' to 'Completed' ?

- A. `document.getElementById("status").Value = 'Completed' ;`
- B. `document.getElementById("#status").innerHTML = 'Completed' ;`
- C. `document.getElementById("status").innerHTML = 'Completed' ;`
- D. `document.getElementById(".status").innerHTML = 'Completed' ;`

**Answer: C**

**NEW QUESTION 28**

Refer to the code below:

```
01 let sayHello = () => {
02   console.log('Hello, World!');
03 };
```

Which code executes sayHello once, two minutes from now?

- A. `setTimeout(sayHello, 12000);`
- B. `setInterval(sayHello, 12000);`
- C. `setTimeout(sayHello(), 12000);`
- D. `delay(sayHello, 12000);`

**Answer: A**

**NEW QUESTION 29**

Refer to code below:

`Const objBook = { Title: 'Javascript',`  
`};`  
`Object.preventExtensions(objBook); ConstnewObjBook = objBook; newObjectBook.author = 'Robert';`  
 What are the values of objBook and newObjBook respectively ?

- A. `[title: "javaScript"] [title: "javaScript"]`
- B. `{author: "Robert", title: "javaScript} Undefined`
- C. `{author: "Robert", title: "javaScript}{author: "Robert", title: "javaScript}`
- D. `{author: "Robert"}{author: "Robert", title: "javaScript}`

**Answer: A**

**NEW QUESTION 30**

A developer wrote the following code: `01 let X = object.value;`  
`02`  
`03 try {`  
`04 handleObjectValue(X); 05 } catch (error) {`  
`06 handleError(error);`  
`07 }`

The developer has a getNextValue function to execute after handleObjectValue(), but does not want to execute getNextValue() if an error occurs.  
 How can the developer change the code to ensure thisbehavior?

- A. `03 try{04 handleObjectValue(x);05 } catch(error){06 handleError(error); 07 } then {08 getNextValue();09 }`
- B. `03 try{04 handleObjectValue(x);05 } catch(error){06 handleError(error); 07 } finally {08 getNextValue();10 }`
- C. `03 try{04handleObjectValue(x); 05 } catch(error){06 handleError(error);07 }08 getNextValue();`
- D. `03 try {04 handleObjectValue(x)05 .....`

**Answer: D**

### NEW QUESTION 35

Given the following code:

```
01 let x = null;
02 console.log(typeof x);
```

is the output of line 02?

- A. "x"
- B. "null"
- C. "object"
- D. "undefined"

**Answer: C**

### NEW QUESTION 40

developer publishes a new version of a package with new features that do not break backward compatibility. The previous version number was 1.1.3. Following semantic versioning format, what should the new package version number be?

- A. 2.0.0
- B. 1.2.3
- C. 1.1.4
- D. 1.2.0

**Answer: D**

### NEW QUESTION 45

A developer wants to leverage a module to print a price in pretty format, and has imported a method as shown below:

Import printPrice from '/path/PricePrettyPrint.js';

Based on the code, what must be true about the printPrice function of the PricePrettyPrint module for this import to work ?

- A. printPrice must be a named export
- B. printPrice must be an all export
- C. printPrice must be the default export
- D. printPrice must be a multi export

**Answer: C**

### NEW QUESTION 48

Universal Containers (UC) notices that its application that allows users to search for accounts makes a network request each time a key is pressed. This results in too many requests for the server to handle.

Address this problem, UC decides to implement a debounce function on string change handler.

What are three key steps to implement this debounce function? Choose 3 answers:

- A. If there is an existing setTimeout and the search string change, allow the existing setTimeout to finish, and do not enqueue a new setTimeout.
- B. When the search string changes, enqueue the request within a setTimeout.
- C. Ensure that the network request has the property debounce set to true.
- D. If there is an existing setTimeout and the search string changes, cancel the existing setTimeout using the persisted timerId and replace it with a new setTimeout.
- E. Store the timerId of the setTimeout last enqueued by the search string change handle.

**Answer: ABC**

### NEW QUESTION 52

Given the following code: Let  $x = ('15' + 10) * 2$ ;

What is the value of a?

- A. 3020
- B. 1520
- C. 50
- D. 35

**Answer: A**

### NEW QUESTION 56

Given HTML below:

```
<div>
  <div id="row-uc">Universal Containers</div>
  <div id="row-as">Applied Shipping</div>
  <div id="row-bt">Burlington Textiles</div>
</div>
```

Which statement adds the priority = account CSS class to the universal Containers row ?

- A. Document.querySelector('#row-uc').classes.push('priority-account');
- B. Document.querySelector('#row-uc').addClass('priority-account');
- C. Document.querySelector('#row-uc').classList.add('priority-account');

D. Document.querySelector('#row-uc').classList.add('priority-account');

**Answer: B**

**NEW QUESTION 57**

Universal Containers recently launched its new landing page to host a crowd-funding campaign. The page uses an external library to display some third-party ads. Once the page is fully loaded, it creates more than 50 new HTML items placed randomly inside the DOM, like the one in the code below:

```
<!-- This is an ad -->
<div class="ad-library-item ad-hidden" onload="myFunction()">
  
</div>
```

All the elements includes the same ad-library-item class, They are hidden by default, and they are randomly displayed while the user navigates through the page.

- A. Use the DOM inspector to prevent the load event to be fired.
- B. Use the browser to execute a script that removes all the element containing the class ad-library-item.
- C. Use the DOM inspector to remove all the elements containing the class ad-library-item.
- D. Use the browser console to execute a script that prevents the load event to be fired.

**Answer: B**

**NEW QUESTION 59**

Refer to the code below: function changeValue(param) { Param =5;

}  
 Let a =10;  
 Let b =5; changeValue(b);  
 Const result = a+ " - "+ b;

What is the value of result when code executes?

- A. 10 -10
- B. 5 -5
- C. 5 - 10
- D. 10 - 5

**Answer: A**

**NEW QUESTION 64**

Refer to the code below:

```
const searchText = 'Yay! Salesforce is amazing!';

let result1 = searchText.search(/sales/i);
let result2 = searchText.search(/sales/);

console.log(result1);
console.log(result2);
```

After running this code, which result is displayed on the console?

- A. > true > false
- B. > 5 >undefined
- C. > 5 > -1
- D. >5 > 0

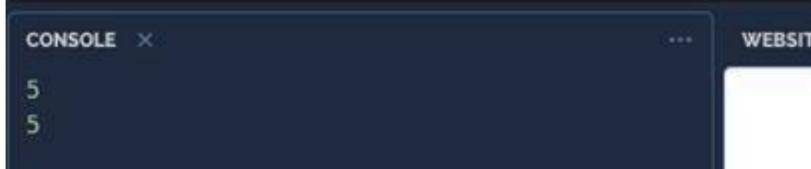
**Answer: B**

**Explanation:**

```

1  import 'bootstrap@4.6.0'
2  import $ from 'jquery'
3
4  const searchText = 'Yay! Salesforce is amazing!';
5
6  let result1 = searchText.search(/sales/i);
7  let result2 = searchText.search(/sales/);
8
9  console.log(result1);
10 console.log(result2);

```



**NEW QUESTION 65**

Given the following code:

```

document.body.addEventListener('click', (event) => { if (/* CODE REPLACEMENT HERE */) {
console.log('button clicked!');
}
});

```

Which replacement for the conditional statement on line 02 allows a developer to correctly determine that a button on page is clicked?

- A. Event.clicked
- B. e.nodeTarget ==this
- C. event.target.nodeName == 'BUTTON'
- D. button.addEventListener('click')

**Answer: C**

**NEW QUESTION 70**

Refer to the code snippet below:

```

Let array = [1, 2, 3, 4,4, 5, 4, 4];
For (let i =0; i < array.length; i++){ if (array[i] === 4) {
array.splice(i, 1);
}
}

```

```

01 let array = [1, 2, 3, 4, 4, 5, 4, 4];
02 for (let i=0; i < array.length; i++) {
03     if (array[i] === 4) {
04         array.splice(i, 1);
05         i--;
06     }
07 }

```

What is the value of the array after the code executes?

- A. [1, 2, 3, 4, 5, 4, 4]
- B. [1, 2, 3, 4, 4, 5, 4]
- C. [1, 2, 3, 4, 5, 4]
- D. [1, 2, 3, 5]

**Answer: C**

**NEW QUESTION 75**

Refer to the code below:

```
01 const myFunction = arr => {
02   return arr.reduce((result, current) => {
03     return result + current;
04   }, 10);
05 }
```

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

- A. Returns 0
- B. Throws an error
- C. Returns 10
- D. Returns NaN

**Answer: C**

#### NEW QUESTION 80

developer creates a new web server that uses Node.js. It imports a server library that uses events and callbacks for handling server functionality. The server library is imported with require and is made available to the code by a variable named server. The developer wants to log any issues that the server has while booting up. Given the code and the information the developer has, which code logs an error at boot with an event?

- A. Server.catch ((server) => { console.log('ERROR', error);});
- B. Server.error ((server) => { console.log('ERROR', error);});
- C. Server.on ('error', (error) => { console.log('ERROR', error);});
- D. Try{server.start();} catch(error) { console.log('ERROR', error);}

**Answer: C**

#### NEW QUESTION 85

developer removes the HTML class attribute from the checkout button, so now it is simply:  
 <button>Checkout</button>.

There is a test to verify the existence of the checkout button, however it looks for a button with class= "blue". The test fails because no such button is found. Which type of test category describes this test?

- A. True positive
- B. True negative
- C. False positive
- D. False negative

**Answer: D**

#### NEW QUESTION 89

Refer to code below: console.log(0);  
 setTimeout(() => ( console.log(1);  
 ));  
 console.log(2); setTimeout(() => { console.log(3);  
 }, 0);  
 console.log(4);  
 In which sequence will the numbers be logged?

- A. 01234
- B. 02431
- C. 02413
- D. 13024

**Answer: C**

#### NEW QUESTION 93

A developer has a formatName function that takes two arguments, firstName and lastName and returns a string. They want to schedule the function to run once after five seconds. What is the correct syntax to schedule this function?

- A. setTimeout (formatName(), 5000, "John", "BDoe");
- B. setTimeout (formatName('John', 'Doe'), 5000);
- C. setTimout(() => { formatName("John", 'Doe') }, 5000);
- D. setTimeout ('formatName', 5000, 'John', "Doe");

**Answer: D**

#### NEW QUESTION 96

A developer has an ErrorHandler module that contains multiple functions. What kind of export be leverages so that multiple functions can be used?

- A. Named
- B. All
- C. Multi

D. Default

**Answer:** A

#### NEW QUESTION 99

Cloud Kicks has a class to represent items for sale in an online store, as shown below: Class Item{ constructor (name, price){ this.name = name; this.price = price; }

formattedPrice(){ return 's' + String(this.price);}}

A new business requirement comes in that requests a ClothingItem class that should have all of the properties and methods of the Item class but will also have properties that are specific to clothes.

Which line of code properly declares the clothingItem class such that it inherits from Item?

- A. Class ClothingItem implements Item{
- B. Class ClothingItem {
- C. Class ClothingItem super Item {
- D. Class ClothingItem extends Item {

**Answer:** D

#### NEW QUESTION 101

Refer to the code below:

```
01 function foo() {
02   const a = 2;
03   function bar() {
04     console.log(a);
05   }
06   return bar;
07 }
```

Why does the function bar have access to variable a ?

- A. Inner function's scope
- B. Hoisting
- C. Outer function's scope
- D. Prototype chain

**Answer:** C

#### NEW QUESTION 103

A developer implements and calls the following code when an application state change occurs: Const onStateChange =innerPageState) => { window.history.pushState(newPageState, ' ', null); }

If the back button is clicked after this method is executed, what can a developer expect?

- A. A navigate event is fired with a state property that details the previous application state.
- B. The page is navigated away from and the previous page in the browser's history is loaded.
- C. The page reloads and all Javascript is reinitialized.
- D. A popstate event is fired with a state property that details the application's last state.

**Answer:** B

#### NEW QUESTION 105

is below:

```
<input type="file" onchange="previewFile()">
<img src="" height="200" alt="Image Preview..." /> The JavaScript portion is:
1 function previewFile(){
2 const preview = document.querySelector('img');
3 const file = document.querySelector('input[type=file]').files[0]; 04 //line 4 code
05 reader.addEventListener("load", () =>{ 06 preview.src = reader.result;
7 },false);
8 //line 8 code
9 }
```

In lines 04 and 08, which code allows the user to select an image from their local computer , and to display the image in the browser?

- A. 04 const reader = new File();08 if (file) URL.createObjectURL(file);
- B. 04 const reader = new FileReader();08 if (file) URL.createObjectURL(file);
- C. 04 const reader = new File();08 if (file) reader.readAsDataURL(file);
- D. 04 const reader = new FileReader();08 if (file) reader.readAsDataURL(file);

**Answer:** D

#### NEW QUESTION 109

Given the code below:

```

01 function GameConsole(name) {
02     this.name = name;
03 }
04
05 GameConsole.prototype.load = function(gamename) {
06     console.log(`${this.name} is loading a game: ${gamename}...`);
07 }
08
09 function Console16bit(name) {
10     GameConsole.call(this, name);
11 }
12
13 Console16bit.prototype = Object.create(GameConsole.prototype);
14
15 //insert code here
16     console.log(`${this.name} is loading a cartridge game: ${gamename}...`);
17 }
18
19 const console16bit = new Console16bit('SNEGeneziz');
20 console16bit.load('Super Monic 3x Force');

```

What should a developer insert at line 15 to output the following message using the method ?  
 > SNEGeneziz is loading a cartridgegame: Super Monic 3x Force . . .

- A. Console16bit.prototype.load(gamename) = function() {
- B. Console16bit.prototype.load = function(gamename) {
- C. Console16bit = Object.create(GameConsole.prototype).load = function (gamename) {
- D. Console16bit.prototype.load(gamename) {

**Answer: B**

#### NEW QUESTION 113

A developer wrote 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. Emptystring if the number is divisible by neither 3 or 5.  
 Which two test cases will properly test scenarios for the fizzbuzz function? Choose 2 answers

- A. let res = fizzbuzz(5); console.assert ( res === ' ' );
- B. let res = fizzbuzz(15);console.assert ( res === ' fizzbuzz ' )
- C. let res = fizzbuzz(Infinity); console.assert ( res === ' ' )
- D. let res = fizzbuzz(3); console.assert ( res === ' buzz ' )

**Answer: BCD**

#### NEW QUESTION 114

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?"));

```

0.024493713600408196 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}

When does Promise.finally on line 08 get called?

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

**Answer: D**

**NEW QUESTION 119**

Why would a developer specify a package.json as a developed forge instead of a dependency ?

- A. It is required by the application in production.
- B. It is only needed for local development and testing.
- C. Other requiredpackages depend on it for development.
- D. It should be bundled when the package is published.

**Answer: B**

**NEW QUESTION 121**

A developer creates a simple webpage with an input field. When a user enters text in the input field and clicks the button, the actual value of the field must be displayed in the console. Here is the HTML file content:

```

<input type =" text" value="Hello" name ="input">
<button type ="button" >Display </button>
The developer wrote the javascript code below:
Const button = document.querySelector('button'); button.addEvenListener('click', () => (
Const input = document.querySelector('input'); console.log(input.getAttribute('value')));

```

When the user clicks the button, the output is always "Hello". What needs to be done make this code work as expected?

- A. Replace line 04 with console.log(input .value);
- B. Replace line 03 with const input = document.getElementByName('input');
- C. Replace line 02 with button.addEventListener("onclick", function() {
- D. Replace line 02 with button.addCallback("click", function() {

**Answer: A**

**NEW QUESTION 125**

Refer to the following code:

```

01 function test(val) {
02   if (val === undefined) {
03     return 'Undefined value!';
04   }
05   if (val === null) {
06     return 'Null value!';
07   }
08   return val;
09 }
10
11 let x;
12
13 test(x);

```

What is returned by the function call on line 13?

- A. Undefined
- B. Line 13 throws an error.
- C. 'Undefined values!'
- D. 'Null value!'

**Answer:** A

#### NEW QUESTION 128

Which two console logs output NaN? Choose 2 answers | |

- A. console.log(10 / Number('5'));
- B. console.log(parseInt('two'));
- C. console.log(10 / 0);
- D. console.log(10 / 'five');

**Answer:** BD

#### NEW QUESTION 133

Which option is a core Node.js module?

- A. Path
- B. los
- C. Memory
- D. locate

**Answer:** A

#### NEW QUESTION 137

Refer to the code below:

```

01 function changeValue(param) {
02   param = 5;
03 }
04 let a = 10;
05 let b = a;
06
07 changeValue(b);
08 const result = a + ' - ' + b;

```

What is the value of result when the code executes?

- A. 10-10
- B. 5-5
- C. 10-5
- D. 5-10

**Answer:** A

#### NEW QUESTION 141

Given a value, which three options can a developer use to detect if the value is NaN? Choose 3 answers !

- A. value == NaN
- B. Object.is(value, NaN)
- C. value === Number.NaN
- D. value !== value
- E. Number.isNaN(value)

**Answer:** AE

#### NEW QUESTION 144

Refer to the following array: Let arr = [ 1,2, 3, 4, 5];  
 Which three options result in x evaluating as [3, 4, 5] ? Choose 3 answers.

- A. Let x= arr.filter (( a) => (a<2));
- B. Let x=arr.splice(2,3);
- C. Let x= arr.slice(2);
- D. Let x= arr.filter((a) => ( return a>2 ));
- E. Let x = arr.slice(2,3);

**Answer:** BCD

#### NEW QUESTION 148

Which two console logs outputs NaN? Choose 2 answers

- A. console.log(10/ Number('5'));
- B. console.log(parseInt('two'));
- C. console.log(10/ "five");
- D. console.log(10/0);

**Answer:** BC

#### NEW QUESTION 152

A developer is debugging a web server that uses Node.js The server hits a runtimeerror every third request to an important endpoint on the web server. The developer added a break point to the start script, that is at index.js at he root of the server's source code. The developer wants to make use of chrome DevTools to debug. Which command can be run to access DevTools and make sure the breakdown is hit ?

- A. node -i index.js
- B. Node --inspect-brk index.js
- C. Node inspect index.js
- D. Node --inspect index.js

**Answer:** D

#### NEW QUESTION 153

Given two expressions var1 and var2. What are two valid ways to return the logical AND of the two expressions and ensure it is data type Boolean ?  
 Choose 2 answers:

- A. Boolean(var1 && var2)
- B. var1 && var2
- C. var1.toBoolean() && var2.toBoolean()
- D. Boolean(var1) && Boolean(var2)

**Answer:** AD

#### NEW QUESTION 155

Given the following code: Counter = 0;  
 const logCounter = () => { console.log(counter);  
 };  
 logCounter(); setTimeout(logCOUNTER, 1100); setInterval(() => {  
 Counter++ logCounter();  
 }, 1000);  
 What is logged by the first four log statements?

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

**Answer:** C

#### NEW QUESTION 158

Given code below: setTimeout (() => ( console.log(1);  
 ). 0);  
 console.log(2);  
 New Promise ((resolve, reject )) => ( setTimeout(() => ( reject(console.log(3));  
 ). 1000);  
 ).catch(() => ( console.log(4);  
 ));  
 console.log(5);  
 What is logged to the console?

- A. 2 1 4 3 5
- B. 2 5 13 4
- C. 1 2 4 3 5
- D. 1 2 5 3 4

**Answer:** B

### NEW QUESTION 163

Refer to the code snippet below: Let array = [1, 2, 3, 4, 4, 5, 4, 4];  
 For (let i = 0; i < array.length; i++) if (array[i] === 4) {  
 array.splice(i, 1);  
 }  
 }

```
let array = [1, 2, 3, 4, 4, 5, 4, 4];
for (let i = 0; i < array.length; i++){
  if (array[i] === 4) {
    array.splice(i, 1);
  }
}
console.log(array)
```

▶ (6) [1, 2, 3, 4, 5, 4] VM1963:7

undefined

What is the value of array after the code executes?

- A. [1, 2, 3, 4, 5, 4, 4]
- B. [1, 2, 3, 4, 4, 5, 4]
- C. [1, 2, 3, 5]
- D. [1, 2, 3, 4, 5, 4]

Answer: B

### NEW QUESTION 168

A developer implements a function that adds a few values.

```
01 function sum(num) {
02   if (num === undefined) {
03     num = 0;
04   }
05   return function(num2, num3) {
06     if (num3 === undefined) {
07       num3 = 0;
08     }
09     return num + num2 + num3;
10   }
11 }
```

Which three options can the developer invoke for this function to get a return value of 10 ? Choose 3 answers

- A. Sum () (20)
- B. Sum (5, 5) ()
- C. sum() (5, 5)
- D. sum(5)(5)
- E. sum(10) ()

Answer: CD

### NEW QUESTION 171

What is the result of the code block?

```
flag();
anotherFlag();

function flag() {
  console.log('flag');
}

const anotherFlag = () => {
  console.log('another flag');
}
```

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

**Answer:** D

#### NEW QUESTION 175

In the browser, the window object is often used to assign variables that require the broadest scope in an application Node.js application does not have access to the window object by default.

Which two methods are used to address this ? Choose 2 answers

- A. Use the document object instead of the window object.
- B. Assign variables to the global object.
- C. Create a new window object in the root file.
- D. Assign variables to module.exports and require them as needed.

**Answer:** B

#### NEW QUESTION 176

Which two code snippets show working examples of a recursive function? Choose 2 answers

- A. `Let countingDown = function(startNumber) { If ( startNumber >0) { console.log(startNumber) ;return countingDown(startNUmber);} else {return startNumber;}};`
- B. `Function factorial ( numVar ) { If (numVar < 0) return;if ( numVar === 0 ) return 1; return numVar -1;`
- C. `Const sumToTen = numVar => { If (numVar < 0)Return;return sumToTen(numVar + 1)};`
- D. `Const factorial =numVar => { If(numVar < 0) return;if ( numVar === 0 ) return 1;return numVar * factorial ( numVar - 1 )};;`

**Answer:** AD

#### NEW QUESTION 181

A developer wants to define a function log to be used a few times on a single-file JavaScript script.

```
01 // Line 1 replacement
02 console.log("LOG:", logInput);
03 }
```

Which two options can correctly replace line 01 and declare the function for use? Choose 2 answers

- A. `function log(logInput) {`
- B. `const log(logInput) {`
- C. `const log = (logInput) => {`
- D. `function log = (logInput) {`

**Answer:** AC

#### NEW QUESTION 184

.....

## Thank You for Trying Our Product

### We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

### JavaScript-Developer-I Practice Exam Features:

- \* JavaScript-Developer-I Questions and Answers Updated Frequently
- \* JavaScript-Developer-I Practice Questions Verified by Expert Senior Certified Staff
- \* JavaScript-Developer-I Most Realistic Questions that Guarantee you a Pass on Your First Try
- \* JavaScript-Developer-I Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

**100% Actual & Verified — Instant Download, Please Click**  
**[Order The JavaScript-Developer-I Practice Test Here](#)**