

Certified JavaScript Developer I.VCEplus.premium.exam.60q

Passing Score: 800
Time Limit: 120 min
File Version: 1.0



Website: <https://vceplus.com> - <https://vceplus.co>
VCE to PDF Converter: <https://vceplus.com/vce-to-pdf/>
Facebook: <https://www.facebook.com/VCE.For.All.VN/>
Twitter : https://twitter.com/VCE_Plus

Certified JavaScript Developer I

Version 1.0



Exam A

QUESTION 1

Which three statements are true about promises? (Choose three.)

- A. A promise has a `.then()` method
- B. A fulfilled or rejected promise will not change states
- C. The executor of a `new Promise` runs automatically
- D. A pending promise can become fulfilled, settled, or rejected
- E. A settled promise can become resolved

Correct Answer: ABD

Section: (none)

Explanation

Explanation/Reference:

QUESTION 2

A developer is trying to convince management that their team will benefit from using Node.js for a backend server that they are going to create. The server will be a web server that handles API requests from a website that the team has already built using HTML, CSS, and JavaScript.

Which three benefits of Node.js can the developer use to persuade their manager? (Choose three.)

- A. Ensures stability with one major release every few years
- B. Uses non-blocking functionality for performant request handling
- C. Installs with its own package manager to install and manage third-party libraries
- D. Performs a static analysis on code before execution to look for runtime errors
- E. Executes server-side JavaScript code to avoid learning a new language



Correct Answer: BCD

Section: (none)

Explanation

Explanation/Reference:

QUESTION 3

A developer creates a generic function to log custom messages in the console. To do this, the function below is implemented.

```
01 function logStatus(status){
02   console.log/*Answer goes here*/('Item status is: %s', status);
03 }
```

Which three console logging methods allow the use of string substitution in line 02? (Choose three.)

- A. `info`
- B. `error`
- C. `message`
- D. `assert`
- E. `log`

Correct Answer: ACD

Section: (none)

Explanation

Explanation/Reference:

QUESTION 4

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

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:



QUESTION 5

Given the code below:

```
const delay = async delay => {  
  return new Promise((resolve, reject) => {  
    setTimeout(resolve, delay);  
  });  
};  
  
const callDelay = async () => {  
  const yup = await delay(1000);  
  console.log(1);  
};  
  
console.log(2);  
callDelay();  
console.log(3);
```

What is logged to the console?

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

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 6

At Universal Containers, every team has its own way of copying JavaScript objects. The code snippet shows as implementation from one team:

```
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.parse(JSON.stringify(john));  
11 dan.firstName = 'Dan';  
12 dan.name();
```

What is the output of the code execution?

- A. Hello John Doe
- B. Hello Dan Doe
- C. TypeError: dan.name is not a function
- D. TypeError: Assignment to constant variable

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

**QUESTION 7**

A test has a dependency on `database.query`. During the test, the dependency is replaced with an object called `database` with the method, `query`, that returns an array. The developer needs to verify how many times the method was called and the arguments used each time.

Which two test approaches describe the requirement? (Choose two.)

- A. Integration
- B. White box
- C. Mocking
- D. Black box

Correct Answer: BD

Section: (none)

Explanation

Explanation/Reference:

QUESTION 8

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 if this function when called with an empty array?

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

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

Reference: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Errors/Reduce_of_empty_array_with_no_initial_value

QUESTION 9 Which statement can a developer apply to increment the browser's navigation history without a page refresh?

- A. `window.history.pushState(newStateObject);`
- B. `window.history.pushState(newStateObject, '', null);`
- C. `window.history.state.push(newStateObject);`
- D. `window.history.replaceState(newStateObject, '', null);`

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://developer.mozilla.org/en-US/docs/Web/API/History/replaceState>

QUESTION 10

Refer to the code below:

```
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();
```



What is the value of `result` after line 10 executes?

- A. John undefined
- B. John Developer
- C. undefined Developer
- D. Error: `myFather.job` is not a function

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 11 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 two.)

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

Correct Answer: AD

Section: (none)

Explanation

Explanation/Reference:

QUESTION 12

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

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://stackoverflow.com/questions/5515310/is-there-a-standard-function-to-check-for-null-undefined-or-blank-variables-in>

QUESTION 13 Which three actions can be done using the JavaScript browser console?
(Choose three.)

- A. View and change security cookies
- B. View and change the DOM of the page
- C. View, change, and debug the JavaScript code of the page

- D. Run code that is not related to the page
- E. Display a report showing the performance of a page

Correct Answer: ABC

Section: (none)

Explanation

Explanation/Reference:

QUESTION 14

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:

```
01 const button = document.querySelector('button');
02 button.addEventListener('click', () => {
03   const input = document.querySelector('input');
04   console.log(input.getAttribute('value'));
05 });
```

When the user clicks the button, the output is always “Hello”.

What needs to be done to make this code work as expected?

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



Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 15

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. The test passes:

```
01 let res = sum3([1, 2, 3]);
02 console.assert(res === 6);
03
04 res=sum3([1, 2, 3, 4]);
05 console.assert(res === 6);
```

A different developer made changes to the behavior of `sum3` to instead sum **all** of the numbers present in the array.

Which two results occur when running the test on the updated `sum3` function? (Choose two.)

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

D. The line 05 assertion fails

Correct Answer: AD

Section: (none)

Explanation

Explanation/Reference:

QUESTION 16 Refer to the string below:

```
const str = 'Salesforce';
```

Which two statements result in the word 'Sales'? (Choose two.)

A. `str.substring(0, 5);`

B. `str.substr(1, 5);`

C. `str.substring(1, 5);`

D. `str.substr(0, 5);`

Correct Answer: AC

Section: (none)

Explanation

Explanation/Reference:

QUESTION 17

A developer is debugging a web server that uses Node.js. The server hits a runtime error 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 the 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 breakpoint is hit?

A. `node -i index.js`

B. `node --inspect index.js`

C. `node inspect index.js`

D. `node --inspect-brk index.js`

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://www.digitalocean.com/community/tutorials/how-to-debug-node-js-with-the-built-in-debugger-and-chrome-devtools>

QUESTION 18 Which statement accurately describes the behavior of the `async/await` keywords?

A. The associated function sometimes returns a promise

B. The associated function will always return a promise

C. The associated class contains some asynchronous functions

D. The associated function can only be called via asynchronous methods

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://marcoamodova.medium.com/lwc-and-async-functions-1d6440080b01>

QUESTION 19 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 `ws`.

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

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

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

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

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

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



B. C.

D.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:**QUESTION 20**

Refer to the following object:

```
01 const cat = {  
02   firstName: 'Fancy',  
03   lastName: 'Whiskers',  
04   get fullName() {  
05     return this.firstName + ' ' + this.lastName;  
06   }  
07 };
```

How can a developer access the `fullName` property for `cat`?

- A. `cat.get.fullName`
- B. `cat.function.fullName()`
- C. `cat.fullName()`
- D. `cat.fullName`

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

Reference: https://www.w3schools.com/js/js_object_methods.asp

QUESTION 21

A developer writes the code below to return a message to a user attempting to register a new username. If the username is available, a variable named `msg` is declared and assigned a value on line 03.

```
01 function getAvailabilityMessage(item) {  
02   if (getAvailability(item)) {  
03     var msg = "Username available";  
04   }  
05   return msg;  
06 }
```

What is the value of `msg` when `getAvailabilityMessage("newUserName")` is executed and `getAvailability("newUserName")` returns true?

- A. `"newUserName"`
- B. `"Username available"`
- C. `undefined`
- D. `"msg is not defined"`

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 22 Which two options are core Node.js modules? (Choose two.)

- A. `http`
- B. `worker`

C.exception
D.iostream

Correct Answer: AB

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://flaviocopes.com/node-core-modules/>

QUESTION 23

Refer to the code:

```
01 function Animal(size, type) {  
02   this.size = size || 'small';  
03   this.type = type || 'Animal';  
04   this.canTalk = false;  
05 }  
06  
07 let Pet = function(size, type, name, owner) {  
08   Animal.call(this, size, type);  
09   this.name = name;  
10   this.owner = owner;  
11 }  
12  
13 Pet.prototype = Object.create(Animal.prototype);  
14 let pet1 = new Pet();
```

Given the code above, which three properties are set for pet1? (Choose three.)

- A. owner
- B. canTalk
- C. type
- D. name
- E. size

Correct Answer: ADE

Section: (none)

Explanation

Explanation/Reference:

QUESTION 24 Which two console logs output NaN? (Choose two.)

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

Correct Answer: AD

Section: (none)

Explanation

Explanation/Reference:

QUESTION 25

Refer to the following code:



```
01 <html lang="en">
02   <div class="outerDiv">
03     <button class="myButton">Click me!</button>
04   </div>
05   <script>
06     function displayInnerMessage(ev) {
07       console.log('Inner message. ');
08     }
09     function displayOuterMessage(ev) {
10       console.log('Outer message. ');
11     }
12
13     window.onload = (event) => {
14       document.querySelector('.outerDiv')
15         .addEventListener('click', displayOuterMessage, true);
16       document.querySelector('.myButton')
17         .addEventListener('click', displayInnerMessage, true);
18     };
19   </script>
20 </html>
```

What will the console show when the button is clicked?

- A. > Inner message
- B. > Inner message
> Outer message
- C. > Outer message
- D. > Outer message
> Inner message

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Explanation:

```
<!DOCTYPE html>

<html lang="en">
  <div class="outerDiv">
    <button class="myButton">Click me!</button>
  </div>
  <script>
    function displayInnerMessage(ev) {
      console.log('Inner message. ');
    }
    function displayOuterMessage(ev) {
      console.log('Outer message. ');
    }

    window.onload = (event) => {
      document.querySelector('.outerDiv')
        .addEventListener('click', displayOuterMessage, true);
      document.querySelector('.myButton')
        .addEventListener('click', displayInnerMessage, true);
    };
  </script>
</html>
```

Click me!

QUESTION 26

`bar.awesome` is a popular JavaScript module. The versions published to `npm` are:

1.2
1.3.1
1.3.5
1.4.0

Teams at Universal Containers use this module in a number of projects. A particular project has the `package.json` definition below.

```
{
  "name": "UC Project Extra",
  "version": "0.0.5",
  "dependencies": {
    "bar.awesome": "~1.3.0."
  }
}
```

A developer runs this command: `npm install`

Which version of `bar.awesome` is installed?

- A. 1.3.5
- B. 1.3.1
- C. 1.4.0
- D. The command fails, because version 1.3.0 is not found

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

**QUESTION 27**

Refer to the code below:

```
01 let first = 'Who';
02 let second = 'What';
03 try {
04   try {
05     throw new Error('Sad trombone');
06   } catch (err) {
07     first = 'Why';
08   } finally {
09     second = 'When';
10   }
11 } catch (err) {
12   second = 'Where';
13 }
```

What are the values for `first` and `second` once the code executes?

- A. `first` is Why and `second` is When
- B. `first` is Why and `second` is Where
- C. `first` is Who and `second` is Where
- D. `first` is Who and `second` is When

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 28

Refer to the code below:

```
01 const exec = (item, delay) =>
02   new Promise(resolve => setTimeout(() => resolve(item), delay));
03
04 async function runParallel() {
05   const [result1, result2, result3] = await Promise.all(
06     [exec('x', '100'), exec('y', '500'), exec('z', '100')]
07   );
08   return 'parallel is done: ${result1}${result2}${result3}';
09 }
```

Which two statements correctly execute the `runParallel()` function? (Choose two.)

- A. `runParallel().then(data);`
- B. `runParallel().done(function(data){ return data; });`
- C. `async runParallel().then(data);`
- D. `runParallel().then(function(data){ return data; });`

Correct Answer: CD

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://www.telerik.com/blogs/what-is-the-point-of-promises>



QUESTION 29

Refer to the code below:

```
flag();
anotherFlag();

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

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

What is result of the code block?

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

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 30

Given the code block 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 load method?

> SNEGeneziz is loading a cartridge game: 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) {

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 31 A developer is trying to handle an error within a function.

Which code segment shows the correct approach to handle an error without propagating it elsewhere?

A.

```
try {
    doSomething();
} catch (error) {
    return error;
}

try {
    doSomething();
} catch (error) {
    return null;
}

try {
    doSomething();
} catch (error) {
    processError (error);
}

try {
    doSomething();
} catch (error) {
    throw new Error('Error found');
}
```

B.

C.

D.



Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://www.valentinog.com/blog/error/>

QUESTION 32

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 this function? (Choose two.)

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

Correct Answer: AC

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://developer.mozilla.org/en-US/docs/Web/API/console/assert>

QUESTION 33

Refer to the code below:

```
01 x = 3.14;
02
03 function myFunction() {
04     'use strict';
05     y = x;
06 }
07
08 z = x;
09 myFunction();
```

Considering the implications of `'use strict'` on line 04, which three statements describe the execution of the code? (Choose three.)

- A. `z` is equal to 3.14
- B. `'use strict'` is hoisted, so it has an effect on all lines
- C. `'use strict'` has an effect only on line 05
- D. `'use strict'` has an effect between line 04 and the end of the file
- E. Line 05 throws an error



Correct Answer: ACD

Section: (none)

Explanation

Explanation/Reference:

QUESTION 34

Given the 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.querySelectorAll('#row-uc').classList.add('priority-account');`
- C. `document.getElementById('row-uc').addClass('priority-account');`
- D. `document.querySelector('#row-uc').classList.add('priority-account');`

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 35

`myArray`, can have one level, two levels, or more levels.

Which statement flattens `myArray` when it can be arbitrarily nested?

- A. `[] .concat (...myArray) ;`
- B. `myArray.join(",").split(",") ;`
- C. `myArray.reduce((prev, curr) => prev.concat(curr), []) ;`
- D. `myArray.flat(Infinity) ;`

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://stackoverflow.com/questions/27266550/how-to-flatten-nested-array-in-javascript>

QUESTION 36

A developer has the function, shown below, that is called when a page loads.

```
function onLoad() {  
  console.log("Page has loaded!");  
}
```



Where can the developer see the log statement after loading the page in the browser?

- A. On the webpage
- B. Browser performance tools
- C. Browser JavaScript console
- D. Terminal running the web server

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 37

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 three.)

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

Correct Answer: ACE

Section: (none)

Explanation

Explanation/Reference:

QUESTION 38

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 Scarf
Uncaught TypeError: saleItem.description is not a function
This is a Scarf

- This is a discounted Shirt
- B. This is a Scarf
This is a Shirt
This is a Scarf
This is a discounted Shirt
- C. This is a ScarfThis is a Shirt
This is a discounted Scarf
This is a discounted Shirt
- D. This is a Scarf
Uncaught TypeError: saleItem.description is not a function
This is a Shirt
This is a discounted Shirt

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 39 Considering type coercion, what does the following expression evaluate to?

true + 3 + '100' + null

- A. '3100null'
- B. 104
- C. 4100
- D. '4100null'

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://www.freecodecamp.org/news/js-type-coercion-explained-27ba3d9a2839/>

QUESTION 40

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?



```
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 }
```

```
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 }
```

```
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 }
```

```
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.

B. C.

D.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 41

Refer to the following code:

```
01 let obj = {  
02   foo: 1,  
03   bar: 2  
04 }  
05 let output = [];  
06  
07 for (let something in obj) {  
08   output.push(something);  
09 }  
10  
11 console.log(output);
```

What is the output of line 11?

- A. ["foo", "bar"]
- B. ["bar", "foo"]
- C. [1, 2]
- D. ["foo:1", "bar:2"]

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:



QUESTION 42

Refer to the following code that imports a module named `Utils`:

```
01 import {foo,bar} from '/path/Utils.js';  
02 foo();  
03 bar();
```

Which two implementations of `Utils.js` export `foo` and `bar` such that the code above runs without error? (Choose two.) A.

```
//FooUtils.js and BarUtils.js exist
import {foo} from '/path/FooUtils.js';
import {bar} from '/path/BarUtils.js';
export {foo, bar}

const foo = () => { return 'foo'; }
const bar = () => { return 'bar'; }
export default foo, bar;

const foo = () => { return 'foo'; }
const bar = () => { return 'bar'; }
export {foo, bar}

export default class {
  foo() { return 'foo'; }
  bar() { return 'bar'; }
}
```

B. C.

D.

Correct Answer: CD

Section: (none)

Explanation

Explanation/Reference:



QUESTION 43

A developer writes the code below to calculate the factorial of a given number.

```
01 function factorial(number) {
02   return number * factorial(number - 1);
03 }
04 factorial(3);
```

What is the result of executing line 04?

- A. 0
- B. RuntimeError
- C. -Infinity
- D. 6

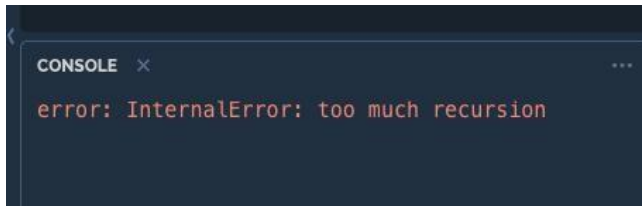
Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

Explanation:

**QUESTION 44**

Refer to the code snippet:

```
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 array the code executes?

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

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:



QUESTION 45 Refer to the 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").innerHTML = 'Completed';
- B. document.getElementById(".status").innerHTML = 'Completed';
- C. document.getElementById("#status").innerHTML = 'Completed';
- D. document.getElementById("status").Value = 'Completed';

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 46 A developer is wondering whether to use, `Promise.then` or `Promise.catch`, especially when a promise throws an error.

Which two promises are rejected? (Choose two.)

- A. `new Promise(() => {throw 'Cool error here'}).then((null, error => console.error(error)));`
- B. `new Promise((resolve, reject) => {throw 'Cool error here'}).catch(error => console.error(error));`

C. `Promise.reject('Cool error here').then(error => console.error(error));`
D. `Promise.reject('Cool error here').catch(error => console.error(error));`

Correct Answer: BD

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://www.javascripttutorial.net/es6/promise-error-handling/>

QUESTION 47

Refer to the code below:

```
01 let country = {  
02   get capital() {  
03     let city = Number("London");  
04  
05     return {  
06       cityString: city.toString(),  
07     }  
08   }  
09 }
```

Which value can a developer expect when referencing `country.capital.cityString`?

- A. 'NaN'
- B. 'London'
- C. An error
- D. undefined

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 48 Which JavaScript methods can be used to serialize an object into a string and deserialize a JSON string into an object, respectively?

- A. `JSON.stringify` and `JSON.parse`
- B. `JSON.encode` and `JSON.decode`
- C. `JSON.serialize` and `JSON.deserialize`
- D. `JSON.parse` and `JSON.deserialize`

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://javascript.info/json>

QUESTION 49

Refer to the code below:



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

Considering that JavaScript is single-threaded, what is the output of line 08 after the code executes?

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

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

Explanation:



```
1  import 'bootstrap@4.6.0'
2  import $ from 'jquery'
3
4  let total = 10;
5  const interval = setInterval (() => {
6    total++;
7    clearInterval(interval);
8    total++;
9  }, 0);
10 total++;
11 console.log(total);
```

CONSOLE x

11



QUESTION 50

Which code statement correctly retrieves and returns an object from localStorage?

```
const retrieveFromLocalStorage = (storageKey) => {  
  return window.localStorage[storageKey];  
}  
  
const retrieveFromLocalStorage = (storageKey) => {  
  return JSON.parse(window.localStorage.getItem(storageKey));  
}  
  
const retrieveFromLocalStorage = (storageKey) => {  
  return window.localStorage.getItem(storageKey);  
}  
  
const retrieveFromLocalStorage = () => {  
  return JSON.stringify(window.localStorage.getItem(storageKey));  
}
```

A.

B. C.

D.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:



QUESTION 51

Refer to the code below:

```
01 const event = new CustomEvent(  
02   //Missing code  
03 );  
04 obj.dispatchEvent(event);
```

A developer needs to dispatch a custom event called `update` to send information about `recordId`.

Which two options could a developer insert at the placeholder in line 02 to achieve this? (Choose two.)

A. `'update', '123abc'`

```
'update', {  
  detail : {  
    recordId : '123abc'  
  }  
}
```

```
'update', {  
  recordId : '123abc'  
}
```

B.

C.

D. {type : 'update', recordId : '123abc'}

Correct Answer: BC

Section: (none)

Explanation

Explanation/Reference:

QUESTION 52

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

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 53

Refer to the following code block:



```
01 class Student {
02   constructor(name) {
03     this.name = name;
04   }
05
06   takeTest() {
07     console.log(`${this.name} got 70% on test.`);
08   }
09 }
10
11 class BetterStudent extends Student {
12   constructor(name) {
13     super(name);
14     this.name = 'Better student' + name;
15   }
16   takeTest() {
17     console.log(`${this.name} got 100% on test.`);
18   }
19 }
20
21 let student = new BetterStudent('Jackie');
22 student.takeTest();
```

What is the console output?

- A.> Better student Jackie got 100% on test
- B.> Jackie got 70% on test
- C.> Better student Jackie got 70% on test
- D.> Uncaught ReferenceError

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:



QUESTION 54

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
- empty string if the number is divisible by neither 3 or 5

Which two test cases will properly test scenarios for the `fizzbuzz` function? (Choose two.)

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

Correct Answer: BC

Section: (none)

Explanation

Explanation/Reference:**QUESTION 55**

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.

To address this problem, UC decides to implement a `debounce` function on the search string change handler.

What are three key steps to implement this `debounce` function? (Choose three.)

- A. If there is an existing `setTimeout` and the search string changes, 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. 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`
- D. Store the `timerId` of the `setTimeout` last enqueued by the search string change handler
- E. Ensure that the network request has the property `debounce` set to `true`

Correct Answer: BCD

Section: (none)

Explanation

Explanation/Reference:**QUESTION 56**

Refer to the code below:

```
01 let a = 'a';
02 let b;
03 // b = a;
04 console.log(b);
```



What is displayed when the code executes?

- A. a
- B. ReferenceError: b is not defined
- C. null
- D. undefined

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

Explanation:

```
1 import 'bootstrap@4.6.0'
2 import $ from 'jquery'
3
4 let a = 'a';
5 let b;
6 // b = a;
7 console.log(b);
```

CONSOLE x
undefined

QUESTION 57

Refer to the code below:

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

Which code executes `sayHello` once, two minutes from now?

- A. `delay(sayHello, 120000);`
- B. `setInterval(sayHello, 120000);`
- C. `setTimeout(sayHello, 120000);`
- D. `setTimeout(sayHello(), 120000);`

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 58

Given the code below:

```
01 function Person(name, email) {
02   this.name = name;
03   this.email = email;
04 }
05
06 const john = new Person('John', 'john@email.com');
07 const jane = new Person('Jane', 'jane@email.com');
08 const emily = new Person('Emily', 'emily@email.com');
09
10 let userList = [john, jane, emily];
```

Which method can be used to provide a visual representation of the list of users and to allow sorting by the name or email attribute?

- A. `console.info(userList);`
- B. `console.group(userList);`
- C. `console.groupCollapsed(userList);`
- D. `console.table(userList);`

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 59

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. `> 5`
`> undefined`
- B. `> 5`
`> 5`
- C. `> 5`
`> -1`
- D. `> true`
`> false`

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

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);
```

CONSOLE x ... WEBSITE

5
5

QUESTION 60 A developer has an `ErrorHandler` module that contains multiple functions.

What kind of export should be leveraged so that multiple functions can be used?

- A. default
- B. multi
- C. all
- D. named

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://codeburst.io/every-thing-you-should-know-about-javascript-export-660cb4e08473>