Salesforce.JavaScript-Developer-I.v2022-07-23.q129

□□□□:	JavaScript-Developer-I
□□□□:	Salesforce Certified JavaScript Developer I Exam
□□□:	Salesforce
□□ □□ □□□:	129
□□:	v2022-07-23
# □□ □:	2639
# □□ □□□:	1290
https://www.krdump.com/Salesforce.JavaScript-Developer-I.v2022-07-23.q129.html	

NEW QUESTION: 1
□□□□ □□□ □□□ □ □□□ □□□□ fizzbuzz □□□ □□□□□□.
* □□□ 3□□ □□□ □□□□ □□ '□□'.
* □□□ 5□ □□□ □□ '□□'.
* □□□ 3□ 5□ □□□ □□□□ □□ 'Fizzbuzz'.
* □□□ 3□□ 5□ □□□ □□□□ □□ □□ □ □□□□□□.
fizzbuzz □□□ □□ □□□□□□ □□□□ □□□□□□ □ □□ □□□ □□□ □□□□□?
2□□ □□□ □□□□□
A. res = fizzbuzz(□□□);
console.assert (res === ' ')
B. res = fizzbuzz(15);
console.assert (res === ' fizzbuzz ')
C. res = fizzbuzz(5);
console.assert (res === ' ');
D. res = fizzbuzz(3);
console.assert (res === ' □□ ')
Answer: A,B,D (LEAVE A REPLY)

NEW QUESTION: 2
□□ □□□ □□□□□.
01 let car1 = □□□ □□((_, □□) =>
02 setTimeout(reject, 2000, "1□ □□□ □□□□□□"));
03 let car2 = new Promise(resolve => setTimeout(resolve, 1500, "□□□ 2
□□□"));
04 let car3 = new Promise(resolve => setTimeout(resolve, 3000, "Car 3
□□□"));
05 Promise.race([□□□1, □□□2, □□□3])

06 .then(□ => (
07 let result = $(value) □□□. `;
08))
09 .catch(arr => (
10 console.log("□□□ □□□□□□□□.", err);
11));
Promise.race□ □□□ □ result□ □□ □□□□□□?
A. □□□ 3□ □□□ □□□□□□□.
B. □□□ 2□ □□□□ □□□□□□□.
C. □□□□ □□□ 1□ □□□□□□□.
D. □□□□ □□□□□□□□.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 3

Universal Containers□ □□ □□□□ □□ □□□□ □□□□ □□ □□□ □□ □□□□ □□□ □□□. □□□□ □□ □□□□□□□ □□□□ □□ □□ □□□ □□□□□. □□□□ □□□ □ □□□ □□ □□□ □□ DOM □□□ □□□□ □□□ 50□ □□□ □ HTML □□□ □□□□□.

□□ □□□ □□□ ad-library-item □□□□ □□□□ □□□□□ □□□ □□□ □□□□ □□□ □ □□□□ □□ □□□□ □□□□□.
A. □□□□ □□□ □□□□ □□ □□□ □□□ □□□□ □□□□□ □□□□□.
B. □□□□□ □□□□ ad-library-item □□□□ □□□□ □□ □□□ □□□□ □□□□□ □□ □□□.
C. DOM □□□□ □□□□ □□ □□□□ □□□□ □□□ □□□.
D. DOM □□□□ □□□□ ad-library-item □□□□ □□□□ □□ □□□ □□□□□.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 4

□□ □□□□ □□□□□□□.
arr1 = [1, 2, 3, 4, 5]□□ □□□.

□□ □ □□ □□□ □ □□ □□□ □□□□□? arr2□ □□□□ arr2□ arr1□ □□ □□□ □□ □□.

A. arr2 = arr1.slice(0, 5)□□ □□□.

B. arr2 = Array.from(arr1);

C. arr2 = arr1.sort();

D. arr2 = rr1□□□ □□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 5

□□ □□□ □□□□□□□.

='a'□□ □□.

□□ b;

// b = □□;

console.log(b);

□□□ □□□ □ □□□□ □□ □□□□□?

A. □

B. ReferenceError: b□ □□□□ □□□□□.

C. □□□□ □□

D. □

Answer: C (LEAVE A REPLY)

NEW QUESTION: 6

□□ □□□ □□□□□.

□□ □□ □ □□□ □□□□□?

```
function Person(firstName, lastName, eyeColor) {
  this.firstName = firstName;
  this.lastName = lastName;
  this.eyeColor = eyeColor;
}
Person.job = 'Developer';

const myFather = new Person('John', 'Doe');
console.log(myFather.job);
```

A. □□□
B. ReferenceError: □□□□ □□ □□ "Person"□ □□
C. □□□□ □□
D. ReferenceError: eyeColor□ □□□□ □□□□□.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 7
□□□□ □□ □□□ □□ □□□ □ □□□□ □□□□. □□□□ □□ □□□ □□□□ □□□ □ □□□ □□□□ □□□ □□ □□ □□□ □□□□□ □□□.
HTML □□ □□□ □□□ □□□□.

```
<input type="text" value="Hello" name="input">
<button type="button">Display</button>
```

□□□□ □□ JavaScript □□□ □□□□□□.

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

□□□□ □□□ □□□□ □□□ □□ "hello"□□□.
□ □□□ □□□□ □□□□□ □□□ □□□ □□ □□□?
A. 04□□ □□, addCallback("□□", function( )(
B. 04□□ button.addEventListener("onclick", □□( )(
C. □ 04□ const □□ = □□□ □□□□. getElementByName('□□') ;
D. 04□□ console.log(□□, □)□ □□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 8
□□ □□□ □□□□□.

```
function changeValue(param) {
  param = 5;
}
let a = 10;
let b = 5;

changeValue(b);
const result = a + ' - ' + b;
```

□□□ □□□ □ □□ □□ □□□□□?
A. 5-5

B. 10-10
C. 5-10
D. 10-5
Answer: B (LEAVE A REPLY)

NEW QUESTION: 9
□□□□ □□□ □□ □□□ □□□ package.jason□ □□□□ □□□ □□□□□?
A. □□□ □□□□ □ □□ □□□□□ □□□.
B. □□□□□□ □□□□□□□ □□□□□.
C. □□ □□ □□□□ □□□ □□ □□□□□.
D. □□ □□ □ □□□□□ □□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 10
□□□□ □□ □□□ □□□□ □□ □□□ □□□□□.

```
01 const date = new Date(2020, 05, 10);
02 const dateDisplayOptions = {
03   year: 'numeric',
04   month: 'long',
05   day: 'numeric'
06 };
07
08 const formattedDate = date.toLocaleDateString('en', dateDisplayOptions);
```

□□ □ formattedDate□ □□ □□□□□?
A. 2020□ 6□ 10□
B. 2020□ 11□ 5□
C. 2020□ 10□ 5□
D. 2020□ 5□ 10□
Answer: D (LEAVE A REPLY)

NEW QUESTION: 11
□□ □□□ □□□□□.
□□□ = 0;
const logCounter = () => {
console.log(□□□);
);
□□ □□□();
setTimeout(□□ □□□, 1100);
setInterval(() => {
□□□++
□□ □□□();

}, 1000);
□□ □ □□ □□ □□ □□□ □□□□□?
A. 0 1 1 2
B. 0 1 2 3
C. 0 1 2 2
D. 0 0 1 2
Answer: (SHOW ANSWER)

NEW QUESTION: 12
□□ □□□ □□□□□.
□□ □□ w 3.1415926;
□□□ □□□ □□□ □□□□□?
A. □□□
B. □□
C. 10□□
D. □□
Answer: B (LEAVE A REPLY)

NEW QUESTION: 13
UC(Universal Container)□ □□□ □□ □□□□ □□□□□ □□□□□
□□□□□ □□□□. □□□□ □ □□□ □□□□ □ □□ □□□ □□□□□. □□ □□□□ □
□,
□□□□ □□ □□□ □□□□ □ □ □□ □□□□□ □□ □□□ □□ □□□ □□□□□ □□
□□□.
□□□□□.
console.time('□□');
□□□AHHeavyFunction();
thisCouldTakeTooLong();
□□□□□ThisOne();
console.endTime('□□');
□□□□ □ □□ □□□ □□□ □□□ □□ □□ □□□ □ □□ □□□ □□□□□?
□□?
A. console.timeStamp()
B. console.trace()
C. console.timeLog()
D. console.getTime()
Answer: C (LEAVE A REPLY)

NEW QUESTION: 14
UC(Universal Container)□ □□□ □□ □□□□ □□□□□ □□□□□ □□□□□ □□□□ □
□□□□. □□□□ □ □□□ □□□□ □ □□ □□□ □□□□□. □□ □□□□ □□ □□□□

□□ □□□ □□□□ □□□ □ □□ □□□□□ □□ □□□ □□□□ □□□ □□□□□ □□ □□□□.

console.time('□□');

□□□AHHeavyFunction();

thisCouldTakeTooLong();

□□□□□ThisOne();

console.endTime('□□');

□□□□ □ □□ □□ □ □□□ □□□ □□□ □□ □□ □□□ □ □□ □□□ □□□□□?

A. console.timeLog()

B. console.trace()

C. console.getTime()

D. console.timeStamp()

Answer: (SHOW ANSWER)

NEW QUESTION: 15

□□□□ universalContainersLib□□ □□□ □□□ □□□□□ □□ □□□□ □□□ □□□□ □.

□□□□ □□□□ □□ □□□ □□□ □□ fuctions foo □ bar □ □□□ □□□□ □□□?

A. '/path/universalContainersLib.js'□□ □□□□(foo, bar);

□();

□□();

B. '/path/universalContaineraLib.js'□□ □□ □□□□;

UniversalContainersLib.foo();

UniversalContainersLib.bar();

C. '/path/universalContaineraLib.js'□□ * □□□□;

UniversalContainersLib.foo();

UniversalContainersLib.bar();

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

lib.foo();

lib.bar();

Answer: D (LEAVE A REPLY)

NEW QUESTION: 16

□□□□ □□□□ □□ □□□□ □□□□ □□□ □□□ □□ □□□ □□ □□□ □□□□ □ □□□ □□□□ □□□□.

□□□ □□□ □□:

□□ □□□ {

// □□□ □□ □□

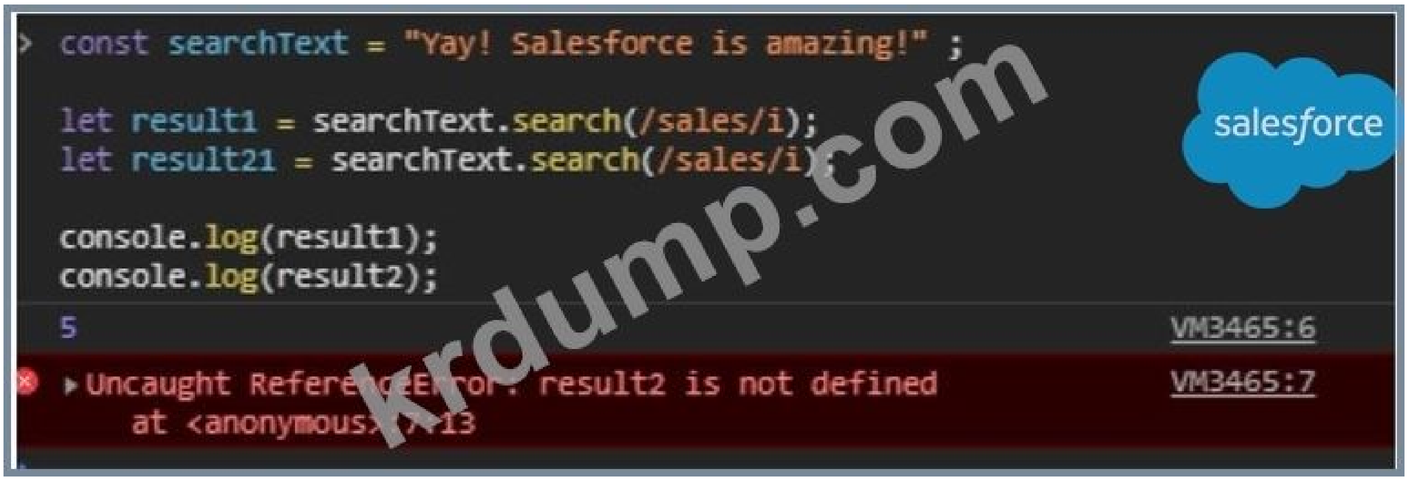
this.body = □□

This.author = □□;

this.viewCount = viewCount;

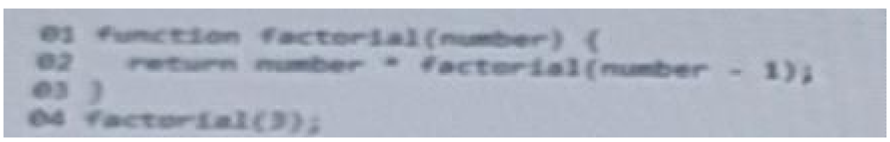
}
}
3□□ □□□ □□□□ □□□ Post□ □ □□□□□ □□□ □□□ □ □□□ □□□ 02□□ □
□ □□□□ □□ □□□□ □□□□ □□□?
A. □□□() {
B. □□(□□, □□□, □□□) {
C. □□ □□□(□□, □□□, □□□) {
D. □□□(□□, □□□, viewCount) {
Answer: (SHOW ANSWER)

NEW QUESTION: 17
□□ □□□ □□□□□.
Const searchTest = '□! Salesforce□ □□□□□!" ;
result1 = searchText.search(/sales/i);
□□ 21 = searchText.search(/sales/i);
console.log(□□1);
console.log(□□2);
□ □□□ □□□ □ □□□ □□ □□□ □□□□□?
A. > □ > □□
B. > 5 >□□□□ □□
C. > 5 > -1
D. > 5 > 0
Answer: B (LEAVE A REPLY)



NEW QUESTION: 18

□□□□ □□□ □□□ □□□ □□□□ □□ □□ □□□ □□□□□.



04□□ □□□ □□□ □□□□□?

A. □□□ □□

B. 0

C. -□□

D. 6

Answer: A (LEAVE A REPLY)

NEW QUESTION: 19

□□ □□□ □□□□□.

□□ □□ = 3.1415326,

□□□ □□□ □□□ □□□□□?

A. □□

B. 10□□

C. □□

D. □□

Answer: C (LEAVE A REPLY)

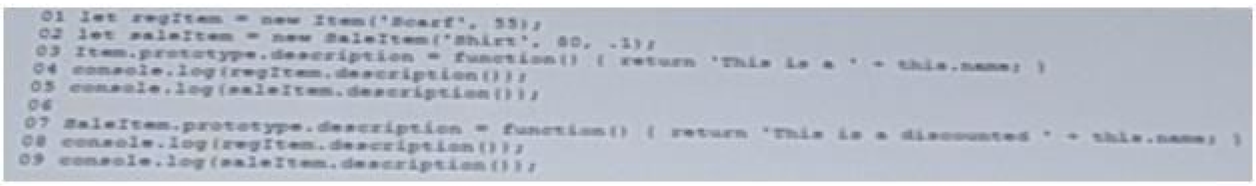
NEW QUESTION: 20

□□□□ □□□ □□□□ □□□ □□□ □□□□ □□ □□□□□□ □ □□ □□□□ □□□ □□□□ □□ □□ □□□ □□□□□. □□□□ □□□ □□□ □ □□ □□□ □□□□□. □□ □□□ □□□ □□□□.


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

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

□□□□ □□ □ □□ □□□ □□ □□□ □□□ □□ □□□□ □□ □□□□ □□□□ □□ □ □□ □□ □□□ □□□□.


```
let regItem = new Item('Scarf', 55);
let saleItem = new SaleItem('Shirt', 80, .1);
Item.prototype.description = function() { return 'This is a ' + this.name; }
console.log(regItem.description());
console.log(saleItem.description());

SaleItem.prototype.description = function() { return 'This is a discounted ' + this.name; }
console.log(regItem.description());
console.log(saleItem.description());
```

□□ □□□ □□□ □ out□ □□□□□?

A. □□□ □□□□□□□.

□□□ □□□□□

□□□ □□□□□□.

□□□ □□□□□

B. □□□ □□□□□□□.

Uncaught TypeError: □□ □□, □□□ □□□ □□□□.

□□□ □□□□□□.

□□□ □□□□□

C. □□□ □□□□□□□.

□□□ □□□□□

□□□ □□□□□

□□□ □□□□□

D. □□□ □□□□□□□.

□□□ □□ TypeError: saleItem, □□□ □□□ □□□□.

□□□ □□□□□

□□□ □□□□□

Answer: A (LEAVE A REPLY)

NEW QUESTION: 21

□□□□ □□ □□□ □□□□□□□.

01 □□ X = object.value;

02

03 □□ {

04 □□□□□(X);

05 } □□(□□) {

06 □□□□(□□);

07 }
□□□□ handleObjectValue() □□ □□□ getNextValue □□□ □□□ □□□ □□□ □□□□
getNextValue()□ □□□□ □□ □□□□.
□□□□ □ □□□ □□□□ □□ □□□ □□□ □□□ □ □□□□?
A. 03 □□{
04 □□□□□(x);
05 } □□(□□){
06 □□□□(□□);
07 }
08 getNextValue();
B. 03 □□{
04 □□□□□(x);
05 } □□(□□){
06 □□□□(□□);
07 } □□□ {
08 getNextValue();
10 }
C. 03 □□{
04 □□□□□(x);
05 } □□(□□){
06 □□□□(□□);
07 } □□ {
08 getNextValue();
09 }
D. 03 □□ {
04 □□□□□(x)
05 ....
Answer: D (LEAVE A REPLY)

NEW QUESTION: 22
Universal Containers□ □□□□ □□□□□ □□ □□□ □□□□ □□ HTML, CSS □
JavaScript□ □□□□ □ □□ □□□□ □□□□. □□□□□ □□□ □□□□(HTML □□□ □
□□ □□ □□) PersonaliseContext□□ □□□□□ □□□□ □□□. □□□ □□ □□□□ □□
□□ □□.
□□ □□□□ □□ □□□ □□ PersonalizeWebsiteContent□ □□□□ □ □□□□ □□ □□
□□□□□?
A. window.addEventListener('onload', PersonalizeWebsiteContext);
B. document.addEventListener(''onDOMContextLoaded', PersonalizeWebsiteContext);
C. Document.addEventListener('''DOMContextLoaded' , PersonalizeWebsiteContext);
D. window.addEventListener('load',personalizeWebsiteContext);
Answer: D (LEAVE A REPLY)

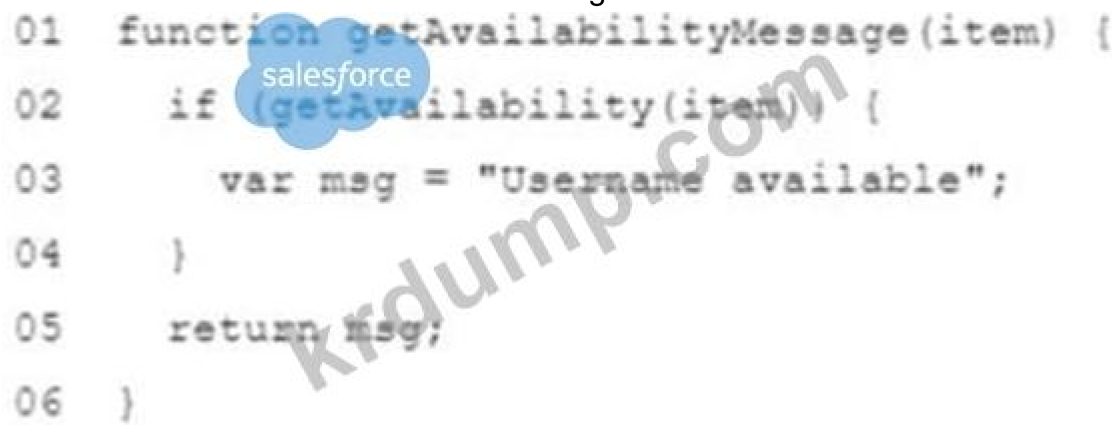
NEW QUESTION: 23
□□□□□ □□□□□□□ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□□.
Const onStateChange =innerPageState) => {
window.history.pushState(newPageState, ' ', null);
}
□ □□□□□ □□□ □ □□ □□□ □□□□ □□□□□ □□□ □□□ □ □□□□?
A. □□□□□□□□ □□□ □□□ □□□ □□□□ state □□□ □□ popstate □□□□ □□□□□.
B. □□□□□ □□ □□□ □□□□ □□□□ □□□ □□ □□□□ □□□□□.
C. □□□□□ □□□□□□□ □□ □□□□□□□□ □□ □□□□□□.
D. □□ □□□□□□□ □□□ □□□ □□□□ state □□□ □□ □□ □□□□ □□□□□.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 24
□□□ □□ □ □□ □□ □ □□ □□?
□ □□□ □□□□ □□□□ □□□□□.
A. □□□ □□□ □□□ □ □□□□.
B. □□□□□□ □□□ □□□ □□□ □□□□ □□□□.
C. □□ □□ □□□ □□, □□ □□ □□□ □ □□□□.
D. □□□□□ . □□ □□ 90 □□.
Answer: (SHOW ANSWER)

NEW QUESTION: 25
□□ □□□ □□□□□□□.
obj ={
□: 1,
□: 2
}
□□ =[],
for(obj{
output.push(□□□);
}
console.log(□□);
□□ □□ 11□ □□□□□□?
A. ["□","□"]
B. [1,2]
C. ["foo:1","bar:2"]
D. ["foo","bar"]
Answer: D (LEAVE A REPLY)

NEW QUESTION: 26

□□□□□ □ □□□□ □□□□ □□□□□□□ □□□□□□□ □□□□□ □□□□□ □□ □□ □□□□ □□□□□ □. □□□□ □□□□ □□□□ □ □□ □□ nag□□ □□□□ □□□□□ 03□□ □□ □□□□□□.



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

getAvailableabilityMessage("newUserName")□ □□□□□ get Availability("newUserName")□ true □ □□□□ □ msg□ □□ □□□□□□?

A. □□□□□ □□

B. "□□□□□ □□□□□ □□□□□□"

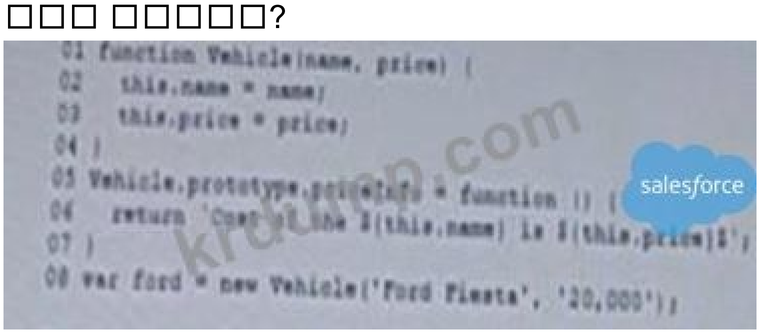
C. "□□ □□□□ □□□□ □□"

D. "newUserName"

Answer: C (LEAVE A REPLY)

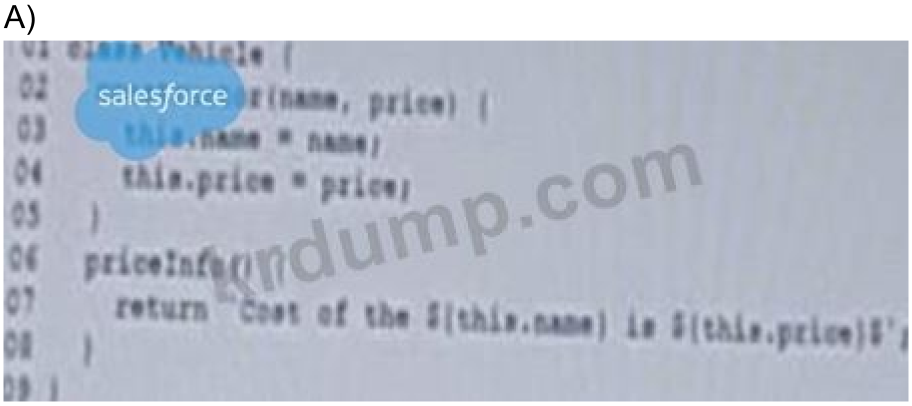
NEW QUESTION: 27

□□ □□□□ JavaScript □□□□ □□□□□ □□□□□□□□ □□ □□ □□□□ □□□□ □ □□□□ □□□□ □□□□ □□□□□□?



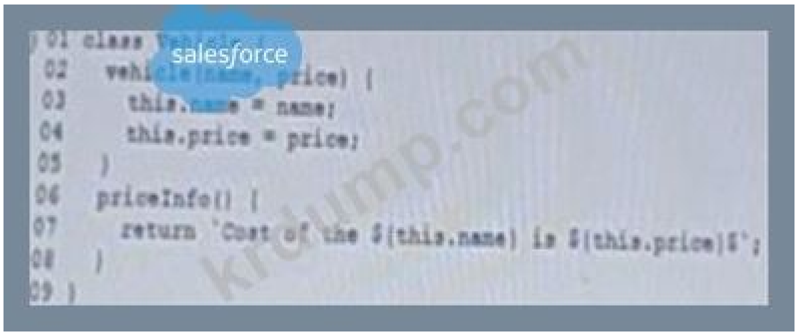
```
function Vehicle(name, price) {
  this.name = name;
  this.price = price;
}
Vehicle.prototype.priceInfo = function () {
  return `Cost of the ${this.name} is ${this.price}$`;
}
var ford = new Vehicle('Ford Fiesta', '20,000');
```

A)



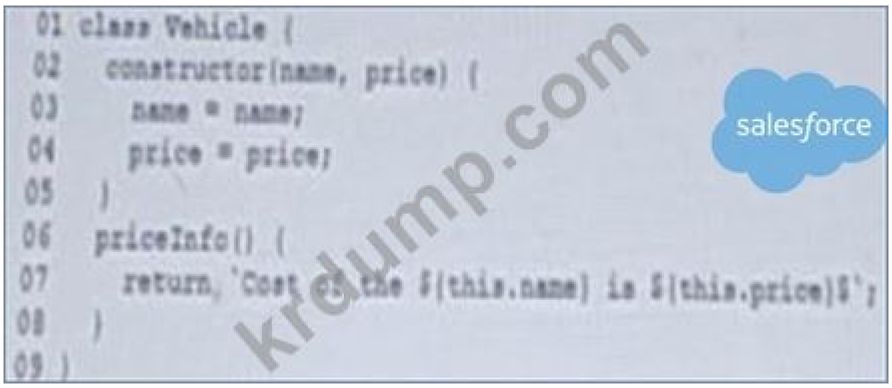
```
class Vehicle {
  constructor(name, price) {
    this.name = name;
    this.price = price;
  }
  priceInfo() {
    return `Cost of the ${this.name} is ${this.price}$`;
  }
}
```

B)



```
class Vehicle {
  vehicle(name, price) {
    this.name = name;
    this.price = price;
  }
  priceInfo() {
    return `Cost of the ${this.name} is ${this.price}$`;
  }
}
```

C)



```
class Vehicle {
  constructor(name, price) {
    name = name;
    price = price;
  }
  priceInfo() {
    return `Cost of the ${this.name} is ${this.price}$`;
  }
}
```

D)

```
class Vehicle {
  constructor() {
    this.name = name;
    this.price = price;
  }
  priceInfo() {
    return `Cost of the ${this.name} is ${this.price}$`;
  }
}
```

A. □□ B

B. □□ D

C. □□ C

D. □□ A

Answer: D (LEAVE A REPLY)

NEW QUESTION: 28

□□□□ □□ □□□ "Hello"□ □□□□ □□□ □□□□. □□ □□□□□□ □□ □□□□□ "World"□ □□□□ □□□ □□□□□□□. □□□ □□ □□□□□ " Hello World"□ □□□□ □□ □□.

□□□□ "Hello World"□ □□□□□ □□□ □□□□ □□ □□□ □ □ □□□□?

A. 2□□ console.log('Hello' , name());

B. 7□□) () ;

C. 9□□ sayHello(world) ()□ □□□□□.

D. 5□□ □□ □□( ) {□ □□□□□.

Answer: A (LEAVE A REPLY)

NEW QUESTION: 29

□□ □□□ □□ Node;js □□□□□?

A. □□□

B. □□

C. □□□

D. □□□

Answer: B,D (LEAVE A REPLY)

NEW QUESTION: 30

□□□ □□□□:

<□□ □□="□□" onchange="□□□□ □□()">

JavaScript □□□ □□□ □□□□.

01 □□ □□□□□□□(){

02 const □□□□ = document.querySelector('img');

03 const □□ = document.querySelector('□□[□□=□□]').□□[0];

04 //□□ 4 □□

05 reader.addEventListener("□□", () => {

06 □□□□.src = □□□.□□;

07 },□□);

08 // 8□ □□

09 }

04□□ 08□□□ □□□□ □□□□ □□□□ □□□ □ □□□ □□ □□

□□□ □ □□□□□ □□□□ □□□□□?

A. 04 const □□ = new File();
08 if(□□) URL.createObjectURL(□□);
B. 04 const □□ = new FileReader();
08 if(□□) URL.createObjectURL(□□);
C. 04 const □□ = new File();
08 if (□□) reader.readAsDataURL(□□);
D. 04 const □□ = new FileReader();
08 if (□□) reader.readAsDataURL(□□);
Answer: (SHOW ANSWER)

NEW QUESTION: 31
□□□□ □□□□□ □□
□□□ □□□ □□□ Node.js. □□□ □ □□□ □ □□□□.
□□ □□ HTML, CSS □
□□□□□□.
□□□□ □□□□ □□□□ □ □□□ □ □□ Node.js□ □ □□ □□□ □□□□□?
3□□ □□□ □□□□□.
A. □□ □□ □□□ □□ □□□ □□□ □□ .
B. □□□ □□ □□□ □□□ □□ □□ □ JavaScript □□□ □□□□□.
C. □ □□ □ □ □□ □□□□ □□□□ □□□□□.
D. □□ □□□□□□ □□ □ □□□□ □□ □□ □□□ □□□□ □□□□□.
E. □□ □□ □□□ □□ □□ □□□ □□□□ □□□ □□□ □□□□.
Answer: A,D,E (LEAVE A REPLY)

NEW QUESTION: 32
□□□□ □□□□ □□□ □□ □□□ □□□□□ □□□ □□□□□. □□ □□ □□□□ □□ □□ □□ □□□□ □□□□□.
□□ □□□(maxSpeed, □□){
this.maxspeed =masSpeed;
this.color = □□;
carSpeed = document.getElementById(' CarSpeed');
□□□;

fourWheels = new Car(carSpeed.value, 'red')□ □□□□□□.
□□ □□□ 06□□ □□□□□□ □□ □ □□□□□ □□□□□ □□□ □ □□ □ □□ □□□ □□□ □□□□□□?
2□□ □□□ □□□□□□.
A. Car Object□ □□ □□□ □□□□□ □□ □□□□□ □□□□□□.
B. carSpeed DOM □□□ □□□ □□□, □□□ □□□ □ □□ □□
C. carSpeed □ fourWheels □□□ □
D. window.localStorage □□□ □□□ □□
Answer: B,D (LEAVE A REPLY)

NEW QUESTION: 33
□□□□ Node.js□ □□□□□ □ □ □□□ □□□□. □□ □□□ □□□□□ □□ □□□□□ □□□ □□□□ □□ □□□□□□□ □□□□□.
□□ □□□□□□□ require□ □□□□ server□□ □□□ □□ □□□□□ □□□ □ □□□□. □□ □□ □□□□□ □□ □□□ □□□ □□ □□□ □□□□□□ □□□.
□□□ □□□□□ □□□ □□ □□□ □□□□□ □□ □□□ □□ □ □□□□□ □□ □□□ □□□ □□?
A)

```
01 server.catch((error) => {
02   console.log('ERROR', error);
03 });
```

B)

```
01 server.error((error) => {
02   console.log('ERROR', error);
03 });
```

C)

```
01 server.on('error', (error) => {
02   console.log('ERROR', error);
03 });
```

D)

```
01 try {
02   server.start();
03 } catch(error) {
04   console.log('ERROR', error);
05 }
```

A. □□ B
B. □□ A
C. □□ C
D. □□ D
Answer: C (LEAVE A REPLY)

NEW QUESTION: 34
□□ □□□ □□ □□ □□□□ □ □□ □□ □□□ □□□□□?
2□□ □□□ □□□□□□
A)

```
let countingDown = function(startNumber) {
    if (startNumber > 0) {
        console.log(startNumber);
        return countingDown(startNumber - 1);
    } 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);
};
```

A. □□ B

B. □□ C

C. □□ D

D. □□ A

Answer: B,C (LEAVE A REPLY)

NEW QUESTION: 35

□□□ □□□ □□ □□□ □□ □□□□ □□ □□?

A. □□□ □□□□ notStrict() □□□ □□□ □ □□□□.

B. □□ □□□□□ □□□ □□ □□□ □□ use non-strict □□□□ □□□□□.
□□□□ □□ □□.

C. □□□ □□□ □□ □□□ □□□□ □□ □□□ □□□□.

D. □□ □□ use strict =false; □□□□ □□□ □□ □□□ □□
□□□□ □□ □□.

Answer: A (LEAVE A REPLY)

NEW QUESTION: 36

□□□□ □□□□(UC)□ □□□ □□ □□□□ □□□□□ □□□□ □□□□□ □□□□ □□ □□□. □□□□ □ □□□ □□□□ □□ □□□ □□□□□□. □□ □□□□ □□ □□□□ □ □ □□ □□□□ □□□ □□□□□ □□ □□□ □□□ □□□□□ □□□□□□. □□□□□.

□□□ □□□□ □ □□ □□ □ □□□ □□□ □□□ □□ □□ □□ □□□ □□□ □ □□□ □?

A. □□, □□□□□

B. □□, □□( )

C. Console.timeLog( )

D. Console.getTime( )

Answer: (SHOW ANSWER)

NEW QUESTION: 37

□□ □□□□ □□□□□□□.

arr1 = [1, 2, 3, 4, 5]□□ □□□.

□□ □ □□ □□□ □ □□ □□□ □□□□□? arr2□ □□□□ arr2□ arr1□ □□ □□□ □□ □□.

A. arr2 = arr1.slice(0, 5);

B. arr2 = Array.from(arr1);

C. arr2 = arr1.sort();

D. arr2 = rr1□□□ □□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 38

□□□□ □□□□ □□□□ □□□□ □□□ □□□ □□□ □□□ □□□□□ □□ □□□ □□ □□□□□.

```
□□ □□ □□□ ( □□ ) {
□□ = 0□□ □□.
□□ += findSubTotal(□□);
□□ += addTax(□□);
□□ += addTip(□□);
□ □□;
}
```

□□□□ computeBill □□□ □ □□ □□□ □□□□ □□□ □ □□ □□□ □□□□□?

A. 03□□□ console.trace(total) □□□□ □□□□□.

B. console.log() □□□□□ findSubtotal □□.

C. 05□□□ □□□ □□□ □□□□□.

D. 03□□□ □□□ □□ □□

Answer: C (LEAVE A REPLY)

NEW QUESTION: 39

□□ □□□ □□□□□.

```
□□ str = '□□□□□□□';
Str[0] = 'J';
Str[4] = 'S';
```

□□□ □□□ □□ □□□ □ str□ □□ 'javascript'□□□. □ □□ □□□ □□□□□?

A. □□ □□ □□□ □ □□□□.

B. □□□ □□ □□ □□□ □ □□□□.

C. □□□ □□ □□ □□ □□□□□.

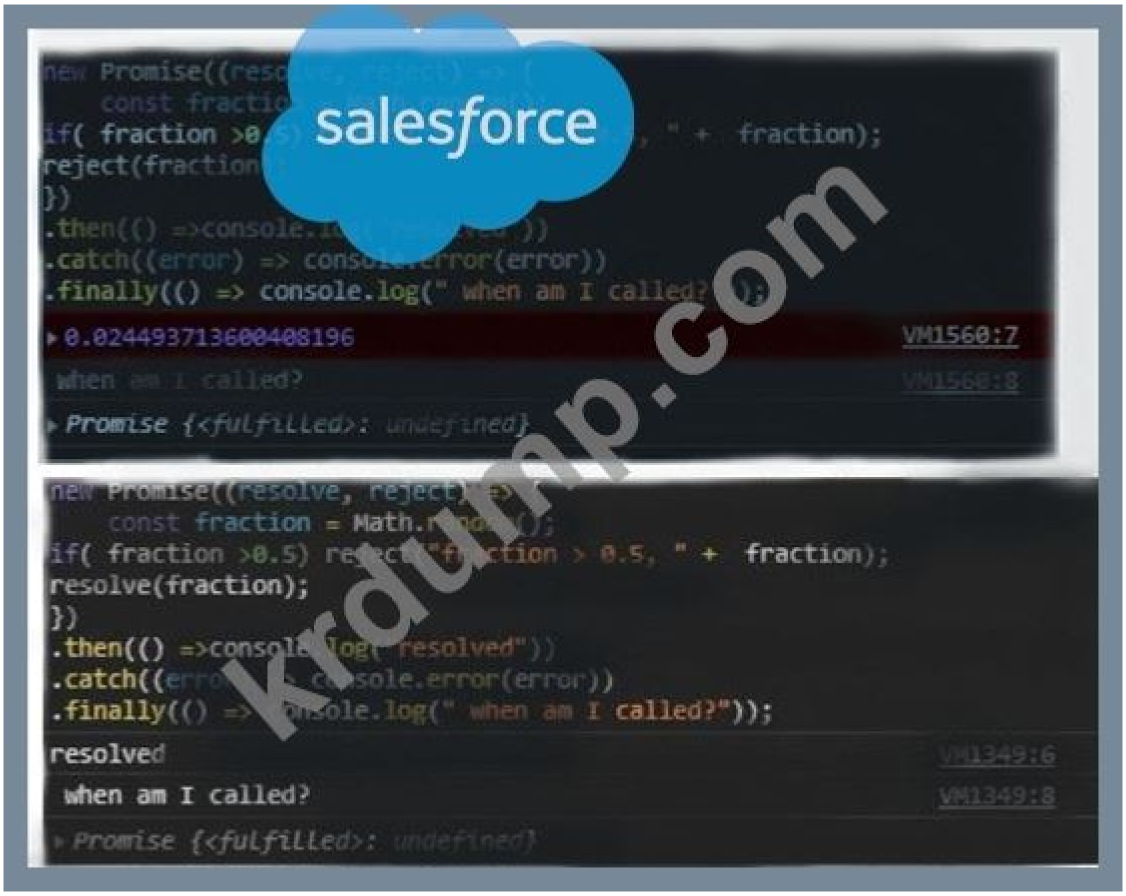
D. □□ □□ □□ □□□□□.

Answer: A (LEAVE A REPLY)

NEW QUESTION: 40

□□ □□□ □□□□□.

```
□□□ □□((□□, □□) => {
□□ □□ = Math.random();
if( fraction > 0.5) reject("fraction > 0.5, " + fraction);
□□(□□);
})
.then(() =>console.log("□□□"))
.catch((□□) => console.error(□□))
.finally(() => console.log(" □□ □□ □□□□□?"));
```



08□□ Promise.finally□ □□ □□□□□?

A. □□ □

B. □□ □□ □□□ □□

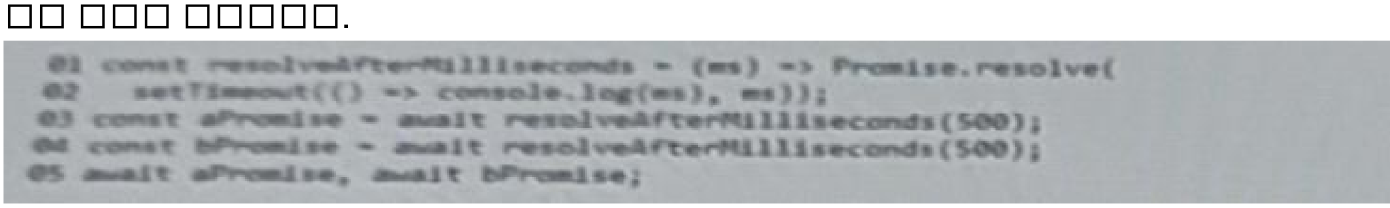
C. □□ □ □□□□□ □

D. □□□ □□

Answer: (SHOW ANSWER)

NEW QUESTION: 41

□□ □□□ □□□□□.



```
01 const resolveAfterMilliseconds = (ms) => Promise.resolve(
02   setTimeout(() => console.log(ms), ms));
03 const aPromise = await resolveAfterMilliseconds(500);
04 const bPromise = await resolveAfterMilliseconds(500);
05 await aPromise, await bPromise;
```

05□□ □□□ □□□ □□□□□?

A. aPromise □□ bPromise□ □□□□ □□□□.

B. □□□ □□□.

C. aPromise□ bPromise□ □□□□□ □□□□□.

D. Apromise□ bpromise□ □□□ □□□□□.

Answer: A (LEAVE A REPLY)

NEW QUESTION: 42

□□ □□□ □□□□□.

const □□□ = □□□ CustomEvent(

//□□ □□
);
obj.dispatchEvent(□□□);
□□□□ recordId□ □□ □□□ □□□ □□ update□□ □□□ □□ □□□□ □□□□ □□□.
□□□ □□□□ □□ □□□□ 02□□ □□ □□□□ □□□ □ □□ □ □□ □□□ □□□□
□?
2□□ □□□ □□□□□

A. { □□ : '□□□□', □□□ ID : '123abc' }

B. '□□□□', {
□□□□: {
□□□ ID : '123abc'
}
}

C. '□□□□' , '123abc'

D. '□□□□' , (
□□□ ID : '123abc'
(

Answer: B,D (LEAVE A REPLY)

NEW QUESTION: 43

□□□□ □□□ □□ □□□□ □□□ □□□ Node.JS□ □□□□ □□□ □□□ □□□□□ □ □□□□□ □□□. □□□ □□ □□ HTML, CSS □ JavaScript□ □□□□ □□□ □□□□□ API □□□ □□□□ □ □□□□□.
□□□□ □□□□ □□□□ □ □□□ □ □□ Node.js□ □ □□ □□□ □□□□□?
□□□□ □□□□ □□□□ □ □□□ □ □□ Node.js□ □ □□ □□□ □□□□□?

A. □□□ □□ □□□ □□□ □□ □□ □ JavaScript □□□ □□□□□.

B. □ □□ □ □ □□ □□□□ □□□□ □□□□□.

C. □□ □□ □□□ □□ □□□ □□□ □□□□□.

D. □□ □□□ □□□□ □□ □□□□ □□ □□□□□□ □□ □ □□□□□.

E. □□□ □□□ □□ □□ □□ □□ □□ □□□ □□□ □ □□□□.

Answer: E (LEAVE A REPLY)

NEW QUESTION: 44

□□□□ □□ □□□ JSON □□□□ □□□□ □□ □□□ □□ □□□ □□□ □□□□□.
01 const □□□ □□ ={
02 " □□□ " : "□□□-01″,
03 "□□□" : "user01@universalcontainers.demo",
04 "□□" : 25
□□ □ □□□ □□□ □□□ □□□□□ □ □□ □□□ □□□□□?
2□□ □□□ □□□□□

A. □□□ □□("□□□")
B. userInformation.get("□□□")
C. □□□ □□(□□□)
D. userInformation.email
Answer: (SHOW ANSWER)

NEW QUESTION: 45
□□ □□ □□□ □□□□□□□?
A. JSON.parse(" ' foo ' ");
B. JSON.parse(' " foo " ');
C. JSON.parse(" foo ");
D. JSON.parse(' foo ');
Answer: B (LEAVE A REPLY)

NEW QUESTION: 46
□□ □□□ □□□□□.
x = ('1' + 2) == (6 * 2);
false□ □□□□□□ □ □□□□□ □□□ □□□□□ □□□?
A. x = ('1' + 2) == (6 * 2);
B. x = (1 + 2) == ('6' / 2);
C. x = (1 + 2) == (6 / 2);
D. x = ('1' + ' 2') == (6 * 2);
Answer: A (LEAVE A REPLY)

NEW QUESTION: 47
□□□ □□□ HTML:
<div>
<div id ="row-uc"> □□ □□□□□</div>
<div id ="row-aa">□□ □□</div>
<div id ="row-bt"> □□□ □□ </div>
</div>
□□ □ □□ □□□□□ □□ priority = account CSS □□□□ □□□□ □□□□ □□□□□□?

A. □□ .queryElementById('row-uc').addclass('priority-account');
B. □□ .querySelector('#row-uc').classes.push('priority-account');
C. □□ .querySelector('#row-uc').classList.add('priority-account');
D. □□ .querySelectorALL('#row-uc').classList.add('priority-account');
Answer: A (LEAVE A REPLY)

NEW QUESTION: 48
□□□□□ □□□ □□ □□ □□□□□ □□□□□ □□ □□□ □ □□ □ □□ □□□□□□ API□ □ □□□□□?
3□□ □□□ □□□□□□
A. indexedDB
B. □□ □□□□□
C. □□ □□
D. IIFE
E. □□
Answer: A,B,D (LEAVE A REPLY)

NEW QUESTION: 49
□□□ □□ □□ □□□ □□□□ □□ □□□ □□□□□□□.
□□:
□□ □□(□□) {
□□(□□ +10) / 3;
}
x□ □□ □□□ 6□□ □□□□□□ □□□ □ 02□ □□□ □□□□ □□□?
x = □□ (8);
A. □□(□□(□□ +10) / 3;
B. □□ □□((num +10) /3);
C. □□ □□(num + 10) / 3;
D. □□ □□(num +10) /3;
Answer: A (LEAVE A REPLY)

NEW QUESTION: 50
Universal Containers□ □□ □□□□ □□□ □□□□□ □□ □□□ □□ □□□□ □□□□□ □.
□□. □□□□ □□ □□□□□□□ □□□□ □□ □□ □□□ □□□□□. □□ □□□□ □□□ □□□□ □□□ □□ DOM □□□ □□□□ □□□ 50□ □□□ □□□ HTML □□□ □ □□□□.
□□ □□ □ □□:

```
<!-- This is an ad -->
<div class="ad-library-item ad-hidden" onload="myFunction()">
  <img src="/ad-library/ad01.gif" />
</div>
```

□□ □□□ □□□ ad-library-item □□□□ □□□□ □□□□□ □□□ □□□ □□□□ □□□ □ □□□□ □□ □□□□ □□□□□.

A. DOM □□□□ □□□□ ad-library-item □□□□ □□□□ □□ □□□ □□□□□.

B. □□□□ □□□ □□□□ □□ □□□□ □□□□ □□□ □□ □□□□□ □□□□□.

C. □□□□□ □□□□ ad-library-item □□□□ □□□□ □□ □□□ □□□□ □□□□□ □□ □□□.

D. DOM □□□□ □□□□ □□ □□□□ □□□□ □□□ □□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 51

Universal Containers□ □□ □□□□ □□ □□□□ □□□□ □□ □□□ □□ □□□□ □□□ □□□. □□□□ □□ □□□□□□ □□□□ □□ □□ □□□ □□□□□. □□□□ □□□ □ □□□ □□ □□□ □□ DOM □□□ □□□□ □□□ 50□ □□□ □ HTML □□□ □□□□□.

```
<!-- This is an ad -->
<div class="ad-library-item ad-hidden" onload="myFunction()">
  <img src="/ad-library/ad01.gif" />
</div>
```

□□ □□□ □□□ ad-library-item □□□□ □□□□ □□□□□ □□□ □□□ □□□□ □□□ □ □□□□ □□ □□□□ □□□□□.

A. DOM □□□□ □□□□ ad-library-item □□□□ □□□□ □□ □□□ □□□□□.

B. □□□□ □□□ □□□□ □□ □□□□ □□□□ □□□ □□ □□□□□ □□□□□.

C. DOM □□□□ □□□□ □□ □□□□ □□□□ □□□ □□□.

D. □□□□□ □□□□ ad-library-item □□□□ □□□□ □□ □□□ □□□□ □□□□□ □□ □□□.

Answer: D (LEAVE A REPLY)

NEW QUESTION: 52

□□□□ □□ □□□ □□ □□□ □ □□□□ □□□□. □□□□ □□□□ □□□ □
□□ □□□ □□□□ □□□ □□□□ □□□ □□ □□
□□.
HTML □□ □□□ □□□ □□□□.
<□□ □□="□□□" □="□□□□□" □□="□□">
<□□ □□ ="□□" >□□□□□ </□□>
□□□□ □□ □□□□□□ □□□ □□□□□□.
Const □□ = document.querySelector('□□');
button.addEvenListener('□□', () => (
Const □□ = document.querySelector('□□');
console.log(input.getAttribute('□'));
□□□□ □□□ □□□□ □□□ □□ "Hello"□□□.
□ □□□ □□□□ □□□□□ □□□ □□ □□□?

A. 02□□ button.addEventListener("onclick", function() {

B. 02□□ button.addCallback("click", function() {

C. 04□□ console.log(input .value)□ □□□□.

D. □□ 03□ const input = document.getElementByName('input');

Answer: C (LEAVE A REPLY)

NEW QUESTION: 53

□□ HTML□ □□□□□□.

<div □□□="□□">

<□>

□□

□□

□□□

</div>

" Tony"□ "Mr. T."□ □□□□ JavaScript □□ □□□□□□?

A. document.querySelector('$main li:second-child').innerHTML = ' Mr. T. ';

B. document.querySelector('$main li:nth-child(2)'),innerHTML = ' Mr. T. ';

C. document.querySelectorAll('$main $TONY').innerHTML = ' Mr. T. ';

D. document.querySelector('$main li.Tony').innerHTML = ' Mr. T. ';

Answer: B (LEAVE A REPLY)

NEW QUESTION: 54

□□□□ □□□ □□□□ □□□ □□□ □□□□ □□ □□□□□□□ □ □□ □□□□ □□□ □□□□ □□ □□ □□□ □□□□□□. □□□□ □□□ □□□□ □□□□□□. □ □□ □□ □□□□□□□□. □□ □□□ □□□ □□□□.

□□□ □□ {

□□□(□□, □□) {

... // □□□ □□

}

}

□□□ SaleItem □□ □□ {

□□□(□□, □□, □□) {

...//□□□ □□

}

}

□□□□ □□□ □□□□ □□ □□□□ □□□□ □□□ □□□ □□ □□□ □□□□.

Item □ SaleItem□ □□ □□□ □□□□□.

Let regItem =new Item('□□□', 55);

saleItem = new SaleItem('□□' 80, -1);

Item.prototype.description = function() { return '□□□ ' + this.name□□□.

console.log(regItem.description());

console.log(saleItem.description());

SaleItem.prototype.description = function() { return '□□□ □□□ □□□□□.' +

□ □□; }
console.log(regItem.description());
console.log(saleItem.description());
□□ □□□ □□□ □ □□□ □□□□□?
A. □□□ □□□□□□□.
□□□ □□□□□
□□□ □□□□□□□.
□□□ □□□□□
B. □□□ □□□□□□□.
□□□ □□ TypeError:saleItem.description□ □□□ □□□□.
□□□ □□□□□
□□□□□ □□□□□
C. □□□ □□□□□□□.
□□□ □□ TypeError:saleItem.description□ □□□ □□□□.
□□□ □□□
□□□ □□□□□
D. □□□ □□□□□□□.
□□□ □□□□□
□□□ □□□□□□□.
□□□ □□□□□
Answer: A (LEAVE A REPLY)

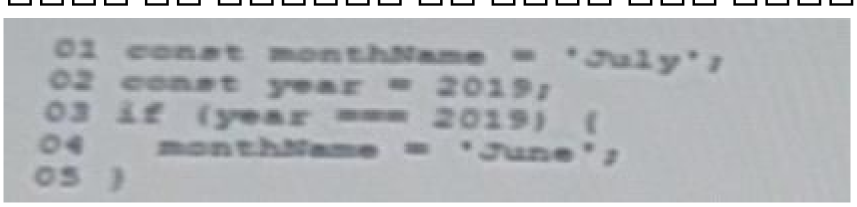
NEW QUESTION: 55
□□□□ □□ □□□ □□ □□□ □□ □□ □ □□□ □□ □□□□ sum3 □□□ □□□□□□
□□ □□ □□□ □□□□□□ □□□□□ □□□□□□□.
□□ □□□□ □□□ □□ □□ □ □□□ □□□□ □□ sum3□ □□□ □□□□□□□.

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

□□□□□ sum3 □□□□ □ □□□□ □□□ □ □□ □ □□ □□□ □□□□□?
2□□ □□□ □□□□□
A. 05□ □□□□ □□□□□.
B. 02□ □□□□ □□□□□.
C. 05□ □□□□ □□□□□.
D. □□ 02 □□□□ □□□□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 56
□□ □□□ □□□□□.

const □□□ = □□□ CustomEvent(
//□□ □□
);
obj.dispatchEvent(□□□);
□□□□ recordId□ □□ □□□ □□□ □□ update□□ □□□ □□ □□□□ □□□□ □□□.
□□□ □□□□ □□ □□□□ 02□□ □□ □□□□ □□□ □ □□ □ □□ □□□ □□□□
□?
2□□ □□□ □□□□□
A. '□□□□' , (
□□□ ID : '123abc'
(
B. '□□□□' , '123abc'
C. { □□ : '□□□□', □□□ ID : '123abc' }
D. '□□□□', {
□□□□: {
□□□ ID : '123abc'
Answer: A,D ([LEAVE A REPLY]())
}
}

NEW QUESTION: 57
□□□□ □□ □□□□□□ □□ □□□□ □□□ □□□□□ □□□ □□□□.

```
01 const monthName = 'July';
02 const year = 2019;
03 if (year === 2019) {
04   monthName = 'June';
05 }
```
□ □□□ □□□□□ □□ □□ □□□□ □□□?
A. 02 const year = 2020;
B. 02 let □□ = 2019 ;
C. 01 □□□ = '7□';
D. 03 if (□ == 2019) (
Answer: ([SHOW ANSWER]())

NEW QUESTION: 58
□□□□ firstName□ lastName□□□ □ □□ □□□ □□□□ □□□□ □□□□ formatName
□□□ □□□□. □□□ □□□ 5□ □□ □ □ □□□□□ □□□□□ □□□.
□ □□□ □□□□ □□□ □□□ □□□□□?
A. setTimeout('formatName', 5000, 'John", "Doe');
B. setTimeout(formatName(), 5000, "John", "BDoe");
C. setTimeout(formatName('John', "Doe'), 5000);
D. setTimout(() => { formatName("John', 'Doe') }, 5000);

Answer: A (LEAVE A REPLY)

NEW QUESTION: 59
□□ □□□□ □□□□ □□ □□ □□□ try... catch □□□ □□□□ □□□?
A. □□□ □□ □□□ □□ □□□ □□□□□.
B. □□□ □□□□□□□ □ □□ □□□ □□□□.
C. □□□ □□□ □□□ □□□□ □□□□.
D. □□□□ □□□□ □□ □□□ □□□ □□□ □ □□□□.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 60
□□ □□□□ □□□□□□□.
arr = [1,2, 3, 4, 5]□□ □□□.
□□ □ x□ [3, 4, 5]□ □□□□ □ □□ □□□ □□□□□?
3□□ □□□ □□□□□.
A. x= arr.splice(2,3);
B. x= arr.filter□□ □□((a) => (a<2));
C. x= arr.slice(2);
D. x= arr.filter((a) => (return a>2));
E. x = arr.slice(2,3);
Answer: A,C,D (LEAVE A REPLY)

NEW QUESTION: 61
□□□□ Node,js □□□ □□□□ □□ □□□ □□□ index,js□ □□□□□ □□□ □□□□. □ □□□□□ □□□ □ □□□ □□□□□. □□□□ □□□ □□□□ □□ □□□ □□□ □□□ □□□□□.
□□□□□□ □□□ □ □□ □□ □□□ □□□□□?
A. _□□ □□
B. this.path
C. □.□□
D. _dirname
Answer: A (LEAVE A REPLY)

NEW QUESTION: 62
□□□□ □□□ □□□□ □□□□□ □□□□ □□□ □□□□□ □□□.
□□□:
□□ Monster() { this.name = '□□□□□' };
Const z = □□□();
02□□ □ □□□□ □□□ □□□ □□□?
A. z □□□ □□□ □□□ □□□□□□□.
B. Window.name□ 'hello'□ □□□□ □□ z□ □□□□ □□ □□□ □□□□□.
C. z □□□ □□□ □□□ □□□□□□ this.name□ □□□□ □□ □□□ □□ □□□□.
D. Window.m□ □□□ □□□ □□□□□□□.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 63
□□ □□□□ □□□□ □□ NaN□□ □□□□ □ □□□ □ □□ □ □□ □□□ □□□□□?
3□□ □□□ □□□□□!
A. □ ! == □
B. Object.is(□, NaN)
C. □ === Number.NaN
D. Number.isNaN(□)
E. □ == NaN
Answer: (SHOW ANSWER)

NEW QUESTION: 64
□□ □ □□□□ □□□ □□□ □□ □□□□ □□□ □□□□ Node.js □ □□□ □□□□□ □
□□.
□□□□ □□ □□□□□ □□□ □□□□□ □□□ □□□□ □□ □□ □□□ □□□ □□□
□□□. □□ □□□□ □□, js □□□ □□□ □□□□□. □□□□ □□□□ □□□□ Node.js
□□□ □□□□□□ □□□.
□□□□ □□ □□□ □□□ CLI □□□□ □ □ □□□ □ □□ □□□ □□□□□?
A. □□ □□, js □□
B. □□ -i server.js
C. □□ □□ □□ □□, js
D. □□ □□ □□, js
Answer: D (LEAVE A REPLY)

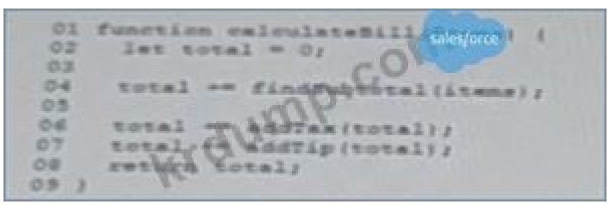
NEW QUESTION: 65
□□ □□□□ □□□□□□.

arr1 = [1, 2, 3, 4, 5]□□ □□□.
□□ □ arr2□ arr1□ □□ □□□ □□ □ □□ □□ arr2□ □□□□ □ □□ □□□ □□□□□?
2□□ □□□ □□□□□
A. arr2 = rr1□□□ □□□.
B. arr2 = arr1 □□□ □□□. □□□□(0, 5);
C. arr2 = □□□□□ □□□. (arr1)□□ ;
D. arr2 = arr1□□□ □□();
Answer: (SHOW ANSWER)

NEW QUESTION: 66
□□□□ □□ □□□ □□ □□□ □ □□□□ □□□□. □□□□ □□ □□□ □□□□ □□□
□ □□□ □□□□ □□□ □□ □□ □□□ □□□□□ □□□.
HTML □□ □□□ □□□ □□□□.
<□□ □□="□□□" □="□□□□□" □□="□□">
<□□ □□ ="□□" >□□□□□ </□□>
□□□□ □□ □□□□□□ □□□ □□□□□□.
Const □□ = document.querySelector('□□');
button.addEvenListener('□□', () => (
Const □□ = document.querySelector('□□');
console.log(input.getAttribute('□'));
□□□□ □□□ □□□□ □□□ □□ "Hello"□□□.
□ □□□ □□□□ □□□□□ □□□ □□ □□□?
A. 02□□ button.addCallback("click", function() {
B. 02□□ button.addEventListener("onclick", function() {
C. 04□□ console.log(input .value)□ □□□□.
D. 03□□ const input = document.getElementByName('input');
Answer: (SHOW ANSWER)

NEW QUESTION: 67
□□ □□□ □□□□□.
01 const exec = (□□, □□) =>{
02 □□□ □□(resolve => setTimeout(() => resolve(□□), □□)),
03 □□□ □□ runParallel() {
04 Const(□□1, □□2, □□3) = Promise.all□ □□□□□{
05 [exec('x', '100') , exec('y', 500), exec('z', '100')]
06);
07 return `□□ □□: $(result1) $(result2)$(result3)`;
08 }
}
}

runParallel() □□□ □□□□ □□□□ □ □□□□ □□□□□?
2□□ □□□ □□□□□
A. runParallel() .then(data);
B. runParallel() .then(function(data) □□ □□□
C. □□□ runParallel() .then(data);
D. □□ □□( ). □□(□□(□□□){
□□□ □□;
});
Answer: B,D (LEAVE A REPLY)

NEW QUESTION: 68
□□□□ □□ □□□□ □□□□ □□□ □□□ □□□ □□□ □□□□□ □□ □□□ □□□□
□.

```
01 function calculateBill( items ) {
02   let total = 0;
03
04   total += findSubtotal(items);
05
06   total += addTax(total);
07   total += addTip(total);
08   return total;
09 }
```
□□□□ computeBill □□□ □ □□ □□□ □□□□ □□□ □ □□ □□□ □□□□□?
A. □□□ □□□□□. 03□□ Trace(total) □□□.
B. 03□□□ □□□ □□□ □□□□□.
C. □□ .log □□□□□ findsubtotal □□.
D. 05□□□ □□□ □□□ □□□□□.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 69
UC(Universal Containers)□ □□ □□□ □□ JavaScript □□□□□□ □□□□□.
UC□ □□□□□ □ □□ □□ □□ □□□ □□ □□ □ □□ □□□ □□ □□□□ □□□□.
□□ □□ □□ □□□ □□□□□ □□ □□□ □□□□.
npm□ □□□□ UC□ □□□ □ □□ □□□ □□□□□?
A. □□ □□□□ □□□□ □□□ □□□ □□ □□ □□□ □□□□ □□□ □ □□□□.
□.
B. □□ □□□□□□ npm□□ □□□□ □□□ □□□□ git□ □□ □□□ □ □□□□.
C. □□ □□□□ □□□ □□ □ □□ □□□ □□ □□□□ □□□ □ □□□□.
□□□□□.
D. npm□ □□□ □□□□□□ □□□□ □□□ URL□ □□ □□□□ □□□ □ □□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 70
□□ □□□ □□□□□.

```
let timedFunction = () => {
    console.log('Timer called.');
};

let timerId = setTimeout(timedFunction, 1000);
```
□□ □ □□□□□ □□□ □□ □□□ □□□ □ □□□ □□□□ □□□□ □□□□□?

A. removeTimeout(timeFunction) ;

B. ClearTimeout(timeFunction);

C. ClearTimeout(timerId) ;

D. removeTimeout(timerId) ;

Answer: A (LEAVE A REPLY)

NEW QUESTION: 71

□□ □□□ □□□□□.

□□ □□(□□, □, □□□□) {

this.firstName = □ □□ □□;

this.lastName = □;

this.eyeColor = □ □□;

}

Person.job = '□□□';

const myFather = new Person('John', 'Doe');

console.log(myFather.job);

□□ □□ □ □□□ □□□□□?

A. □□□

B. □□□□ □□

C. ReferenceError: □□□□ □□ □□ "Person"□ □□

D. ReferenceError: eyeColor□ □□□□ □□□□□.

Answer: B (LEAVE A REPLY)

NEW QUESTION: 72

□□□□ □□□ □□□□ □□□ □□□ □□□□ □□ □□□□□□ □ □□ □□□□ □□□ □□□□ □□ □□ □□□ □□□□□. □□□□ □□□ □□□ □ □□ □□□ □□□□□. □□ □□□ □□□ □□□□.

□□□ □□ {

□□□(□□, □□) {

... // □□□ □□

}

}

□□□ SaleItem □□ □□ {

□□□(□□, □□, □□) {

...//□□□ □□

}
}
□□□□ Item □ SaleItem□ □□ □□□ □□□ □□□□ □□ □□□□ □□□□ □□ □□□
□□ □□□ □□□□.
Let regItem =new Item('□□□', 55);
saleItem = new SaleItem('□□' 80, -1);
Item.prototype.description = function() { return '□□□ ' + this.name□□□.
console.log(regItem.description()); console.log(saleItem.description());
SaleItem.prototype.description = function () { return '□□□ □□□ □□□□□' + this.name; }
console.log(regItem.description()); console.log(saleItem.description()); □□ □□□ □□□ □ □
□□ □□□□□?
A. □□□ □□□□□□.
□□□ □□ TypeError:saleItem.description□ □□□ □□□□.
□□□ □□□□□
□□□□□ □□□□□
B. □□□ □□□□□□.
□□□ □□□□□
□□□ □□□□□□.
□□□ □□□□□
C. □□□ □□□□□□.
□□□ □□ TypeError:saleItem.description□ □□□ □□□□.
□□□ □□□
□□□ □□□□□
D. □□□ □□□□□□.
□□□ □□□□□
□□□ □□□□□□.
□□□ □□□□□
Answer: D (LEAVE A REPLY)

NEW QUESTION: 73
□□□□ □□ □□□ □□ □□□ □ □□□□ □□□□. □□□□ □□ □□□ □□□□ □□□
□ □□□ □□□□ □□□ □□ □□ □□□ □□□□□ □□□.
HTML □□ □□□ □□□ □□□□.
<□□ □□="□□□" □="□□□□□" □□="□□">
<button type ="button" >□□□□□ </button> □□□□ □□ □□□□□□ □□□ □□□□□□.
Const □□ = document.querySelector('□□');
button.addEvenListener('□□', () => (
Const □□ = document.querySelector('□□');
console.log(input.getAttribute('□'));
□□□□ □□□ □□□□ □□□ □□ "Hello"□□□.
□ □□□ □□□□ □□□□□ □□□ □□□ □□ □□□?

A. 03□□ const input = document.getElementByName('input');
B. 02□□ button.addEventListener("onclick", function() {
C. 02□□ button.addCallback("click", function() {
D. 04□□ console.log(input .value)□ □□□□□.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 74
□□ □□□ □□□□□.
console.log("□□);
Promise.resolve('□□') .then(□□(□){
console.log('□□');
});
console.log('□□');
□□□ □□□□□ □□□ □ □□□ □□□□□?
A. □□
□□
□
B. □□
□
□□
C. □□
□□
□
D. □
□□
□□
Answer: B (LEAVE A REPLY)

NEW QUESTION: 75
□□ □□□□ □□□□□□.
arr1 = [1, 2, 3, 4, 5]□□ □□□.

□□ □ □ □□ □□□ □□□□ □ □□ □□□ arr2□ □□□□ □□□ □□□□□.
arr1□ □□ □□?

A. arr2 = Array.from(arr1);

B. arr2 = arr1.slice(0, 5)□□ □□□.

C. arr2 = rr1□□□ □□.

D. arr2 = arr1.sort();

Answer: A,B (LEAVE A REPLY)

NEW QUESTION: 76

□□□□ Node.js□ □□□□ □ □□□ □□□ □□□□. □ □□□ □□□□ □□□□ node server.js□□□. □ □□□ □□ □□□ □□□□ □□□□□□. □ □□□ □□ 1□□ □□ □□ □□ □□ □□□□ 5□□ □□ □□□ □ □ □□□□.
□□□ □□ □□ □□ □□□ □□□ □□ □□ □ □□□□ □□□ □ □□ □□□ □□□□□?

A. NODE_DEBUG=true □□ server.js

B. NODE_DEBUG=http,https □□ server.js

C. DEBUG=true □□ server.js

D. DEBUG=http, https □□ server.js

Answer: (SHOW ANSWER)

NEW QUESTION: 77

□□□□ □□□□ □□□□ HTML □□□ □□□ □□□□□ □□ □□□□□.
<button>□□□□</button>.
□□□□ □□□ □□□ □□□□ □□□□ □□□ class= "blue"□ □□□ □□□□. □□ □□□
□□ □ □□ □□□ □□□□ □□□□□.
□ □□□□ □□□□ □□□ □□ □□□ □□□□□?
A. □□□
B. □ □□
C. □□□
D. □□ □□
Answer: D (LEAVE A REPLY)

NEW QUESTION: 78
Universal Containers□ □□□□ HTML, CSS □
JavaScript□ □□□□ □□ □□□ □ □ □□□ PersonaliseContext□□ □□□□□ □□□□□.
□ □□□□ □□□ □□□ □(HTML □□□ □ □□ □□ □□) □□□□□ □□□.
□ □□ □□□ □□ □□□□ □□□□ □□.
□□ □□□ □□ PersonalizeWebsiteContent□ □□□□ □ □□□□ □□ □□ □□□□□?
□□□□ □□ □□?
A. window.addEventListener('onload', PersonalizeWebsiteContext);
B. document.addEventListener(''onDOMContextLoaded', PersonalizeWebsiteContext);
C. window.addEventListener('load',personalizeWebsiteContext);
D. Document.addEventListener('''DOMContextLoaded' , PersonalizeWebsiteContext);
Answer: C (LEAVE A REPLY)

NEW QUESTION: 79
□□□ □□□ □□ □□□ □□ □□□□ □□ □□?
A. □□□ □□ □□□ □□ □□□ □□□ □□□□□ □□ □□ □□□ □□□□ □□□□□.
B. □□□ □□□ □□ □□□ □□□□ □□ □□□ □□□□.
C. □□□ □□□□ notStrict() □□□ □□□ □ □□□□.
D. □□ □□ use strict =false; □□□ □□□ □□□□□□ □□□ □□ □ □□ □□□.
Answer: C (LEAVE A REPLY)

NEW QUESTION: 80
□□ □□ □□□□ □□□□□.
□□ = [1, 2, 3, 4, 4, 5, 4, 4]□□ □□□.
```
(let i =0; i < array.length; i++){
if (□□[i] === 4) {
array.splice(i, 1);
}
}
```
□□□ □□□ □ □□□ □□ □□□□□?

A. [1, 2, 3, 4, 5, 4, 4]
B. [1, 2, 3, 4, 5, 4]
C. [1, 2, 3, 4, 4, 5, 4]
D. [1, 2, 3, 5]
Answer: B (LEAVE A REPLY)

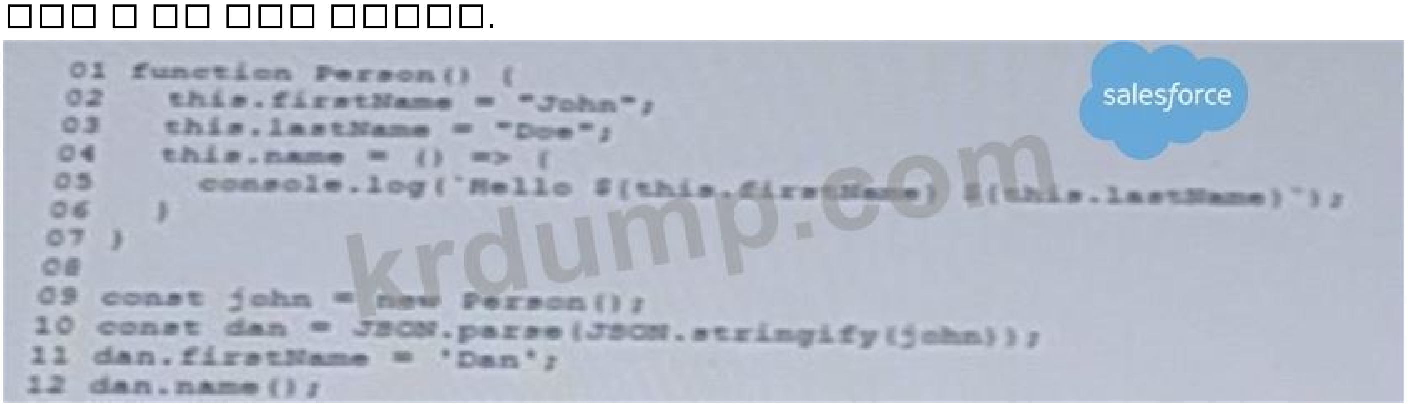
NEW QUESTION: 81
□□ □□ □□□ □□□□□□□.

```
let array = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11];
let output = 0;

for (let num of array) {
  if (output > 10) {
    break;
  }
  if (num % 2 == 0) {
    continue;
  }
  output += num;
}
```

□□ □□ □ □□ □□ □□□□□?
A. 36
B. 16
C. 11
D. 25
Answer: B (LEAVE A REPLY)

NEW QUESTION: 82
Universal Containers□□ □□ □□ JavaScript □□□ □□□□ □□□ □□□ □□□□. □□ □ □□□ □ □□ □□□ □□□□□.



```
function Person() {
  this.firstName = "John";
  this.lastName = "Doe";
  this.name = () => {
    console.log(`Hello ${this.firstName} ${this.lastName}`);
  }
}

const john = new Person();
const dan = JSON.parse(JSON.stringify(john));
dan.firstName = 'Dan';
dan.name();
```

□□ □□□ □□□ □□□□□?
A. □□ John Doe
B. TypeError: dan.name□ □□□ □□□□.
C. TypeError: □□ □□□ □□
D. □□ □□
Answer: (SHOW ANSWER)

NEW QUESTION: 83

□□ □□□ □□□□□.

let timeFunction =() => {
console.log('□□□□ □□□□□□□.");
};
let timerId = setTimeout(timedFunction, 1000);

□□ □ □□□□□ □□□ □□ □□□ □□□ □ □□□ □□□□ □□□□ □□□□□?

A. clearTimeout(timerId);
B. removeTimeout(timerId);
C. removeTimeout(timedFunction);
D. clearTimeout(timedFunction);

Answer: A (LEAVE A REPLY)

NEW QUESTION: 84

□□□□ □□ □□□□ □□□□ □□□□□ □□ □□□ □□□ □□□ □ □□□ □□□□ □ □□□□. □□ □□ □□□ 1.1.3□□□.

□□□□ □□ □□ □□□ □□ □ □□□ □□ □□□ □□□□□ □□□?

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

Answer: (SHOW ANSWER)

NEW QUESTION: 85

Universal Containers□ □□ □□ □□ □□□□ □□□□ □□□ □□ □□□□ □□□□□□. □□□□ □□ □□□□□□ □□□□ □□ □□ □□□ □□□□□. □□□□ □□□ □□□□ □□ □□□ □□ DOM □□□ □□□□ □□□ 50□ □□□ □ HTML □□□ □□□□□.

```
<!-- This is an ad -->
<div class="ad-library-item ad-hidden" onload="myFunction()">
  <img src="/ad-library/ad01.gif" />
</div>
```

□□ □□□□ □□□ □□ □□□□□ □□ □□□□ □□□□□□□. □□□□□ □□□ □□□ □□□□ □□□□ □□□□ □□ □□□□ □□□□□.

□□ □□□ □□□□□. □□□□ □□□□□ □□□□ □□□ □□□□ □□ □□□ □ □ □□ □□?

A. DOM □□□□ □□□□ ad-library -item □□□ □□□□ □□ □□□ □□□□□.
B. DOM □□□□ □□□□ □□ □□□□ □□□□ □□□ □□□.
C. □□□□ □□□ □□□□ ad-library-item □□□□ □□□□ □□ □□□ □□□□ □□□□□ □□□□□.
D. □□□□ □□□ □□□□ □□ □□□□ □□□□ □□□ □□ □□□□□ □□□□□.

Answer: C (LEAVE A REPLY)

NEW QUESTION: 86
□□□□ □□□ □□□□ □□ □□□□ □□□ □□□□□ □□ □□□ □□ □□□□ □□□□ □□.
'/path/PricePrettyPrint.js'□□ printPrice □□□□;
□□□ □□□□ □ □□□□□ □□□□□ PricePrettyPrint □□□ printPrice □□□ □□ □□□ □ □□ □□ □□□□□?
A. printPrice□ □□ exportc□□ □□□.
B. printPrice□ □□□ □□□□□□ □□□.
C. printPrice□ □□ □□□□□□ □□□.
D. printPrice□ □□ □□□□□□ □□□.
Answer: D (LEAVE A REPLY)

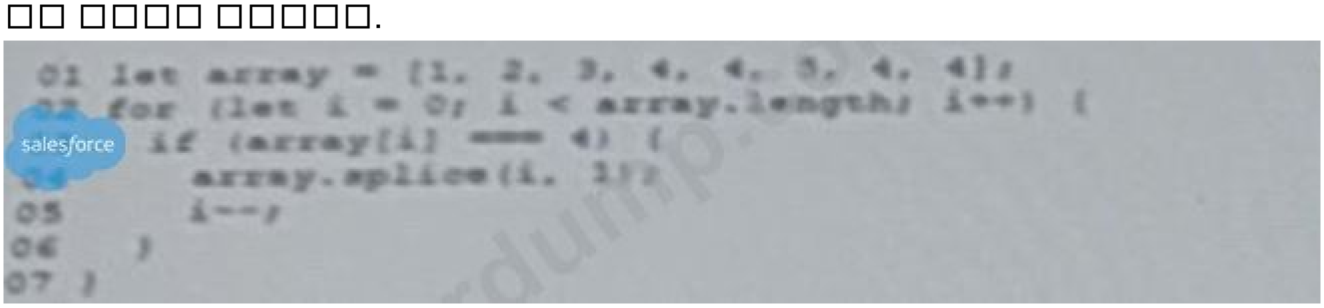
NEW QUESTION: 87
□□ □□□ □□□□□□□.
Const objBook = {
□□: '□□□□□□',
};
Object.preventExtensions(objBook);
Const newObjBook = objBook;
newObjectBook.author = '□□□';
objBook □ newObjBook□ □□ □□ □□□□□?
A. {□□: "□□□", □□: "□□□□□□}
□□□□ □□□ □□□□
B. {□□: "Robert", □□: "javaScript}
{□□: "□□□", □□: "□□□□□□}
C. {□□: "Robert"}
{□□: "□□□", □□: "□□□□□□}
D. [□□: "□□□□□□"] [□□: "□□□□□□"]
Answer: D (LEAVE A REPLY)

NEW QUESTION: 88
□□ □ □□□ NaN□ □□□□□□□?
2□□ □□□ □□□□□
A. Console.log(10/0);
B. Console.log(parseint ('two')) ;
C. Console.log(10 / 'five' ') ;
D. Console.log(10 / □□('5'));
Answer: (SHOW ANSWER)

NEW QUESTION: 89

□□ □□□ □□□□ □:
□□ myFunction(){
A = 5;
□□ b = 1;
}
myFunction();
console.log(a);
console.log(b);
□□ □□□ □□□□□?
A. 08□□ □□□ □□□□□ 09□□ □□□ □□□□□□.
B. 08□□ □□□ □□□□□ 09□□ □□□□ □□□□.
C. 08□□ 09□□ □□ □□□□ □□□ □□□□□.
D. 08□□ 09□□ □□ □□□□□ □□□□ □□ □□□□ □□□□.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 90
□□ □□□ □□□□□.
for(let number =2 ; number <= 5 ; number += 1) {
// □□□ □□ □□ □□□□□.
}
□□□□ □□□ □□□ □□ □□ □□□□ □□□. □□ □□□ □□□ □□ □□ □□□ □□□ □.
1. □□□ □□□□
2. □□ □□ □□□□ □□□ □ □□□ □□□□□.
3. □□□□□ Node.js □□□□ □□□□□.
□□ □□□ □□□□ □□ □□□□□?
A. console.debug(□□ % 2 === 0);
B. console.assert(□□ % 2 === 0);
C. □□(□□ % 2 === 0);
D. console.error(□□ % 2 === 0);
Answer: (SHOW ANSWER)

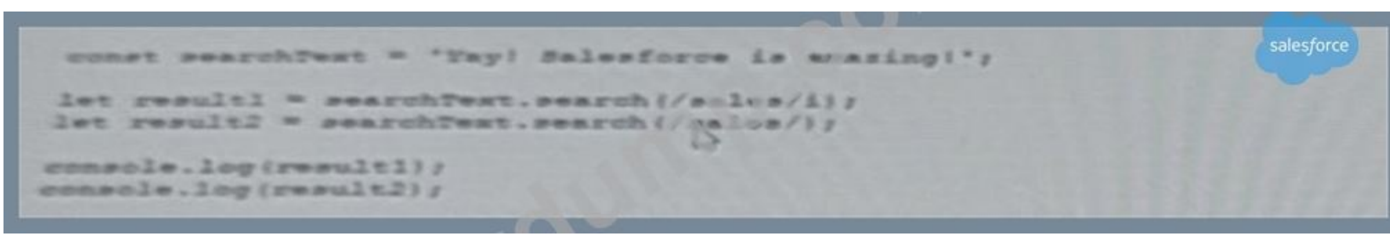
NEW QUESTION: 91
□□ □□□□ □□□□□.


```
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);
    i--;
  }
}
```

□□□ □□□ □ □□□ □□ □□□□□?

A. 1, 2, 3, 4, 5, 4
B. 1, 2, 3, 4, 5, 4, 4
C. 1, 2, 3, 5
D. 1, 2, 3, 4, 4, 5, 4
Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 92
□□ □□□ □□□□□□.



□ □□□ □□□ □ □□□ □□ □□□ □□□□□□?
A. > 5
> □□□□□ □□
B. > 5
>- 1
C. > 5
> 0
D. > □
> □□
Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 93
□□□ □□□□□ □□□□□□ JSON □□□□ □□□ □□□□□□□ □ □□ □□ □□□□□□□ □ □□□ □□□ □ □□□□?
A. JSON.stringify □ JSON.parse
B. JSON.parse □ JSON.deserialize
C. JSON.serialize □ JSON.deserialize
D. JSON.encode □ JSON.decode
Answer: ([SHOW ANSWER](#))

NEW QUESTION: 94

□□□□□□ □□□□ □□□ □ □□□□ □□□ □□ □□□ □□□□.

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

□□□□□□ □□□□ □□□ □ □□□□ □□□□ □□ □□ □ □ □□□□?

A. □ □□□ □□□□ □□□□□□.

B. □□□□ □□□□□□ □□

C. □□□□□□

D. □□□□ □□ □□

Answer: B ([LEAVE A REPLY]())

NEW QUESTION: 95

□□ □ □□□ □□□ □□□ □□□□ □□□ □□□ □□□□ □ □□ □□□ □□□□□?

3□□ □□□ □□□□□

A. x => (console.log(' □□□ ') ;)

B. [ ] => (console.log(' □□□ ') ;)

C. ( ) => (console.log(' □□□ ') ;)

D. (x,y,z) => (console.log(' □□□ ') ;)

E. X,y,z => (console.log(' □□□ ') ;)

Answer: A,D ([LEAVE A REPLY]())

NEW QUESTION: 96

□□ □□ □□□ □□□□ □□ □□□□ □□□ □□□ □ □□ □□□ □□□ □□□□□?

2□□ □□□ □□□□□

A. □□□ □□ □□□ □□□□ parentThis□□ □□ □□ □□□ □□ □□□ □□□□.

B. □□ □□□ □□ □□□□ □□ □□ □□□□ □□□□ □□□□□ □□□□□.

C. □□□ □□□□ □□□□ this□ □□□□□.

D. □□□ □□□□□ □□□□ □□□ □□□ □□□□ □□□□ □□□□ □ □□□□□.

Answer: B,D ([LEAVE A REPLY]())

NEW QUESTION: 97

□□ □□□ □□□□□.

```
const exec = (item, delay) =>
  new Promise(resolve => setTimeout(() => resolve(item), delay));

async function runParallel() {
  const [result1, result2, result3] = await Promise.all(
    [exec('x', '100'), exec('y', '500'), exec('z', '100')]
  );
  return `parallel is done: ${result1}${result2}${result3}`;
}
```

runparallel() □□□ □□□□ □□□□ □ □□□□ □□□□□?

2□□ □□□ □□□□□

A. runParallel () , (□□) (□□) { } 0;
B. □□□ runParalled(). □□ □□ (□□□) :
C. runParallel() , done (function (data)(return data; }};
D. runParralel() . □□ □□ (□□□);
Answer: A,B (LEAVE A REPLY)

NEW QUESTION: 98
□□ HTML□ □□□□□□.
<div id="□□">
<div id = " card-00">□ □□□ □ □□□□.</div>
<div id = "card-01">□ □□□ □□□ □□□
□□.</div>
</div>
ID card-01□ □□ □□□ □□ □□□ □□□□ □□□□ □□□□□?
A. document.getElementById(' card-01 ').getBoundingClientRest().width
B. document.getElementById(' card-01 ').innerHTML.lenght*e
C. document.getElementById(' card-01 ').style.width
D. document.getElementById(' card-01 ').width
Answer: A (LEAVE A REPLY)

NEW QUESTION: 99
□□ □□□ □□□□□.


□□□ □□□ □ □□ □□ □□□□□?
A. 5 - 5
B. 10 - 10
C. 5 - 10
D. 10 - 5
Answer: B (LEAVE A REPLY)

NEW QUESTION: 100
□□ □□□ □□□□ □:
01 □□ GameConsole(□□) {
02 this.name = □□;
03 }
04

05 GameConsole.prototype.load = □□(□□ □□) {
06 console.log(` $(this.name)□ □□□ □□ □□□□ : $(gamename) ...`);
07)
08 □□ □□ 16□□(□□) {
09 GameConsole.call(□, □□) ;
10 }
11 Console16bit.prototype = Object.create(GameConsole.prototype) ;
12 // □□□ □□ □□
13 console.log(` $(this.name)□ □□□□ □□□ □□ □□□□. $(gamename) ...`);
14 }
15 const console16bit = new Console16bit(' SNEGeneziz ');
16 console16bit.load(' □□ □□ 3x □□ ');

□□□□ □□□ □□□□ □□ □□□□□ □□□□ □□ 15□□ □□□ □□□□ □□□?
□□ ?
> SNEGeneziz□ □□□□ □□□ Super Monic 3x Force□ □□ □□□□. . .

A. Console16bit.prototype.load(□□□) {
B. Console16bit.prototype.load(□□ □□) = function() {
C. Console16bit = Object.create(GameConsole.prototype).load = □□
(□□□) {
D. Console16bit.prototype.load = function(gamename) {
Answer: D (LEAVE A REPLY)

NEW QUESTION: 101
UC(Universal Container)□ □□□□ □□ □□ □□□ □□□□ □□□ □□□ □□□ □□□ □ □□□ □□ □□ □□□□□ □□□□□. □ □□ □□□ □□□ □□□ □□ □□□□.
□ □□□ □□□□ □□ UC□ □□ □□□ □□ □□□□ □□ deboune □□□ □□□□□ □□ □□□. 3□□ □□□ □□□□□
A. □□ □□□ □□ □□□□ □□□□□ □□ □□ setTimeout□ timeId□ □□□□□.
B. □□□□ □□□ debounce □□□ true□ □□□□ □□□ □□□□□.
C. □□ settimeout□ □□ □□ □□□□ □□□□ □□ setTimeout□ □□□□□ □□□□ settimeout□ □□□□ □□ □□□□.
D. □□ setTimeout□ □□ □□ □□□□ □□□□ □□ □□ □□□ □□□□ □□ setTimeout□ □□□□ □ setTimeout□□ □□□□□.
E. □□ □□□□ □□□□ setTimeout □□□ □□□ □□□□ □□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 102
□□□□ □□□ □□ □□□□ □□□ □□□□□ □□□□□ □□ □□□ □□□□ □□□?
A. setInterim
B. setInterval
C. setTimeout

D. setPeriod
Answer: B (LEAVE A REPLY)

NEW QUESTION: 103
□□□□ □□ □□□ □□ □□□ □□ □ □□□ □□ □□□□ sum3 □□□ □□□□□ □□
□□ □□□ □□□□□□. □□□ □□:

```
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);
```

□□ □□□□ □□□ □□ □□ □□□ □□□ □□□□□ sum3□ □□□ □□□□□□□.
□□□□□ sum3 □□□□ □□□□ □□□ □ □□ □ □□ □□□ □□□□□□?
2□□ □□□ □□□□□
A. 02□ □□□□ □□□□□□.
B. 05□ □□□□ □□□□□□□.
C. 05□ □□□□ □□□□□□.
D. □□ 02 □□□□ □□□□□□.
Answer: A,B (LEAVE A REPLY)

NEW QUESTION: 104
□□□□ □□ □□□□ HTML □□□ □□□ □□□□□ □□ □□□□□□.
<□□>□□<□□>.
□□□□ □□□ □□□ □□□□ □□□□ □□□ □□□□ "□□□"□ □□□ □□□□. □□ □
□□ f□ □□□□ □□□□ □□□□□□.
□ □□□□ □□□□ □□□ □□ □□□ □□□□□□?
A. □□□
B. □□□
C. □□□
D. □ □□
Answer: D (LEAVE A REPLY)

NEW QUESTION: 105
□□□□ Node.js□ □□□□ □ □□□ □□□ □□□□. □□□□ □□□ □□□ □□□□□□□.
□ □□□ □□□ □□□ □□ □ □□ □□□□.
□□□□ □□ □□□□□ □□□□ □□□□□□. □, □□ □□□ index.js□ □□□□.
□□□ □□ □□. □□□□ chrome DevTools□ □□□□ □□□□□□ □□□□.
DevTools□ □□□□□ □□□ □□□□□ □□□□ □□ □□□ □ □□ □□□ □□□□□□?
A. □□ □□ index.js
B. □□ --inspect index.js
C. □□ --inspect-brk index.js
D. □□ -i index.js

Answer: B (LEAVE A REPLY)

NEW QUESTION: 106

□□ □□□ □□□□□.

```
01 new Promise((resolve, reject) => {
02   const fraction = Math.random();
03   if (fraction > 0.5) reject('fraction > 0.5, ' + fraction);
04   resolve(fraction);
05 })
06 .then(() => console.log('resolved'))
07 .catch((error) => console.error(error))
08 .finally(() => console.log('when am I called?'));
```

□□ □□□□□. □□□□□ □□ 08□□ □□□ □□□□?

A. □□ □ □□□□□ □

B. □□ □

C. □□ □□ □□□ □□

D. □□□ □□

Answer: (SHOW ANSWER)

NEW QUESTION: 107

□□ □□□ □□□□□.

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

□□□□ recordId□ □□ □□□ □□□ □□ update□□ □□□ □□ □□□□ □□□□ □□□. □□□ □□□□ □□ □□□□ 02□□ □□ □□□□ □□□ □ □□ □ □□ □□□ □□□□ □? 2□□ □□□ □□□□□

A. {□□ ; □□□□ ', □□□ ID : '123abc')

B. '□□□□', {

□□ □□ ; {

□□□ ID, '123abc

)

)

C. '□□□□' , '123abc'

D. '□□□□', (recordId ; 123abc'
)
Answer: B,D (LEAVE A REPLY)

NEW QUESTION: 108
□□ □□□□ □□□□□□□.
arr = [1, 2, 3, 4, 5]□□ □□□.
x□ (3, 4, 5)□ □□□□ □ □□ □□□ □□□□□?
3□□ □□□ □□□□□
A. x = arr.aplice(2, 3)□□ □□□.
B. x = arr.filter((a) => (a < 2)) □□ □□□.
C. x = arr□□□ □□. □□□□ (2, 3);
D. x = arr.filter((a) =>)return a > 2)) ;
E. x = arr.alice(2)□□ □□□.
Answer: A,D,E (LEAVE A REPLY)

NEW QUESTION: 109
□□□□ JavaScript □□□ □□□□□□.


```
function Person() {
  this.firstName = "John";
  this.lastName = "Doe";
  this.name = () => `${this.firstName},${this.lastName}`;
}

const john = new Person();
const dan = Object.assign(john);
dan.firstName = 'Dan';
```

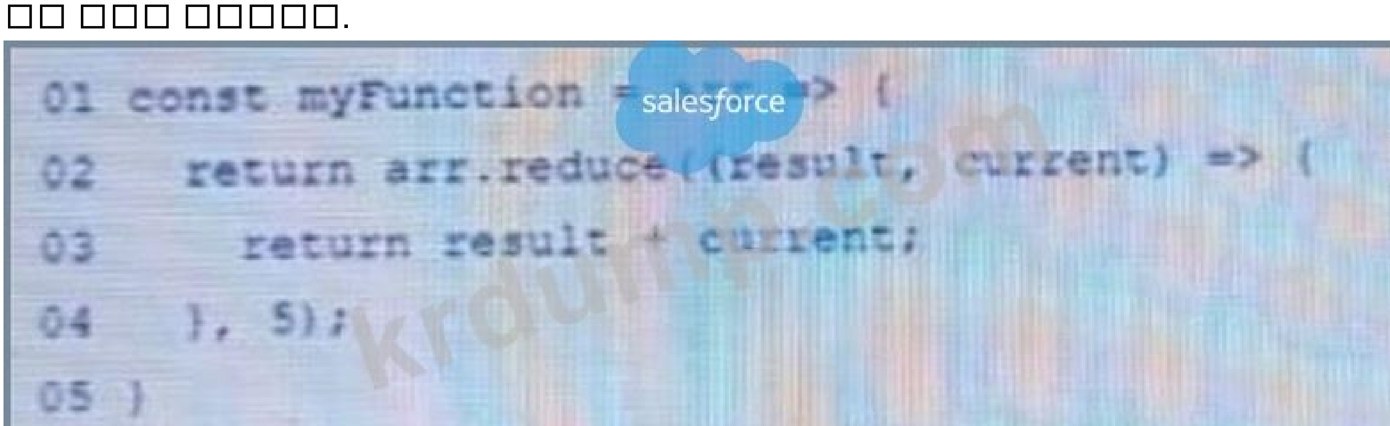
□□□□ dan□ forstName,lastName□ □□□ □□□□□□? 2□□ □□□ □□□□□
A. □, □□ = dan.lastName
B. □, □□( ) + □, □( )
C. □,□□
D. □,□□( )
Answer: (SHOW ANSWER)

NEW QUESTION: 110
□□ □□ □□□□ □□□□□.
□□ = [1, 2, 3, 4, 4, 5, 4, 4]□□ □□□.
For (let i =0; i < array.length; i++)
if (□□[i] === 4) {
array.splice(i, 1);

}
}
□□□ □□□ □ □□□ □□ □□□□□?
A. [1, 2, 3, 4, 4, 5, 4]
B. [1, 2, 3, 5]
C. [1, 2, 3, 4, 5, 4, 4]
D. [1, 2, 3, 4, 5, 4]
Answer: A (LEAVE A REPLY)

NEW QUESTION: 111
□□□□□□□ □ □□□ □□ □□□□□□□ □□□□□□□ □□□ □□□ □□□ □□□□ □ □□ □□□□□. □□□ □□ □□□□□ □□□□□ □ □□□ □□□□ □ □□□□.
□ □□□ □□□□ □□ □□ □ □□ □□□ □□□□□? 2□□ □□□ □□□□□
A. □□□ □□□ □□□□ □□□ □□ □□□□ □□□□□.
B. □□ □□□ □□□ □□□□□.
C. □ □□ □□ □□□ □□□□□.
D. □□ □□□ □ □ □□□ □□□□.
Answer: C,D (LEAVE A REPLY)

NEW QUESTION: 112
□□ □□□ □□□□□ □□ □□□□□?
A. JSON.parse('foo');
B. JSON.parse(" 'foo' ");
C. JSON.parse("foo ");
D. JSON.parse(" 'foo" ');
Answer: (SHOW ANSWER)

NEW QUESTION: 113
□□ □□□ □□□□□.

```
01 const myFunction = arr => {
02   return arr.reduce((result, current) => {
03     return result + current;
04   }, 5);
05 }
```
□ □□□ □□□ □ □ □□□ □□□ □□□□□?
A. NaN □□
B. 0 □□

C. □□ □□
D. □□ 5
Answer: D (LEAVE A REPLY)

NEW QUESTION: 114
□□□□ □□□ □ □□□ □□□□ □□□ □□□□□ □□□□□□ □ Node.js □□□ □□□ □ □□□□.
□□□:
* □ □□ □□□ □□□□ □□□ □□ □□□ □□□ □□□□□.
* require□ □□□□ ws□□ □□□ □□□ □ □□□□.
□□□□ □□ □□□ □□□ □□ □□ □□□ □□□□□ □□□□.
□ □□□ □□□□ □□ □□ □□□□□ □□ □□□ □□ □□□□ □ □□ □□□□ □□□□ □□ □□□□ □□□ □□□ □□□□□?
A)
```
ws.connect(() => {
  console.log('Connected to client');
}).catch((error) => {
  console.log('ERROR', error);
});
```
B)
```
ws.on('connect', () => {
  console.log('Connected to client');

  ws.on('error', (error) => {
    console.log('ERROR', error);
  });
});
```
C)
```
ws.on('connect', () => {
  console.log('Connected to client');
});

ws.on('error', (error) => {
  console.log('ERROR', error);
});
```
D)
```
try {
  ws.connect(() => {
    console.log('Connected to client');
  });
} catch(error) {
  console.log('ERROR', error);
}
```
A. □□ D
B. □□ C
C. □□ A
D. □□ B

Answer: (SHOW ANSWER)

NEW QUESTION: 115

□□□□ □□ □□□ □□ □□ □, name□ □□ N□□ □□□□ □□□ □□□□□ □□□.

const arrobj - [{"name "t "Zach"}, { "name t Kate ") , ("□□ □□□") < ('□□" t "□") , ("□□" t "□□□) , ("□□" t "Nathandle1' ')] □□ □□ □□□□ □□□□□.

```
01 arrObj.reduce((acc, curr) => {
02    // missing line 02
03    // missing line 03
04 }, 0);
```

□□ □□□ □ 02□ 03□ □□□ □□□ □□□□□?

A. conset sun + curr. □□□□ □□('N') 7 1:1;
□□ acc + □□ ;

B. □□ □□ = □□. □□ □□('N') 7 1 : 0
□□ acc + □□;

C. □□ □□ = □□. □□ startsWith ('N') 7 1 : 0;
□□ + □□□ □□□□□.

D. Const sum = curr. □□ startsWith ('N') 7 1 : 0;
□□ + □□□ □□□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 116

□□ □□□ □□□□□.

```
01 const server = require('server');
02
03 // Insert code here
```

□□□□ □ □□□ □□□□ □□□□□□ □□□□□. □□□ □□□□□□ □□□□ □□□ □ □□□ □□□ □□□□□.

□□□□ □□□□ □ □□□ □□□□□ 03□□ □□ □□□ □□□□ □□□?

A. □□ .start();

B. □□. on ('□□', (□□) => (
Console.log(□□□□ □□);
}};

C. □□((□□) =>(
Console.log ('□□ □', □□) ;
}};

D. □□( );

Answer: C (LEAVE A REPLY)

NEW QUESTION: 117

□□ □□□ □□□□□.

□□ =[[1, 2] , [3, 4, 5]];

□□ □ □□□ [1, 2, 3, 4, 5]□ □ □□□□ □□□□□?
2□□ □□□ □□□□□

A. [ ]. concat.apply(□□, [ ]);

B. [ ]. □□(... inArray);

C. [ ]. □□([....inArray]);

D. [ ]. Concat.apply([ ], inArray);

Answer: (SHOW ANSWER)

NEW QUESTION: 118

□□□ var1 □ var2□ □□□□ □ □□□□ □□□ □□□□ □□□□□ □□□□ □ □□ □□ □ □□□ □□□□□? 2□□ □□□ □□□□□

A. var1.toString( ) var2.toString( )

B. □□□(var1) .concat(var2)

C. string.concat(var1 + var2)

D. var1 + var2

Answer: A,C (LEAVE A REPLY)

NEW QUESTION: 119

□□ □□□ □□□□□.

```
□□□ □□((□□, □□) => {
□□ □□ = Math.random();
if( fraction > 0.5) reject("fraction > 0.5, " + fraction);
□□(□□);
})
.then(() =>console.log("□□□"))
.catch((□□) => console.error(□□))
.finally(() => console.log(" □□ □□ □□□□□?"));
```

```
ew Promise((resolve, reject) => {
   const fraction = Math.random();
f( fraction >0.5) reject("fraction > 0.5, " +  fraction);
eject(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
```

08□□ Promise.finally□ □□ □□□□□□?

A. □□ □ □□□□□□ □

B. □□ □□ □□□ □□

C. □□□ □□

D. □□ □

Answer: (SHOW ANSWER)

NEW QUESTION: 120

□□□□□ □□□ □□ □□ □□ □□ □□□ □□□ □□□□.

arr □□ - [7, 8, 5, 9]

□ □□□ □ □□ □□□ 15□ □□□ □□□ □□□□ □□□ □□□ □□□ □□ □□□?

□□□□□ □□□ □□□ □□□□□ □□□?

A. arr1 = arr.map((num) => num + 2 .filter((val) => val 15);

B. arr1 = arr.mapBy ((num) => return num + 2)) .filterBy ((val) => (return val > 15)) ;

C. arr1 = arr .map ((num => (num + 2)) .filterBy ((val => (val 15)) □□ □□□.

D. arr1 = arr .filter ((filter ((val) => (return val > 15)) .map ((num => (return num +2));

Answer: A (LEAVE A REPLY)

NEW QUESTION: 121

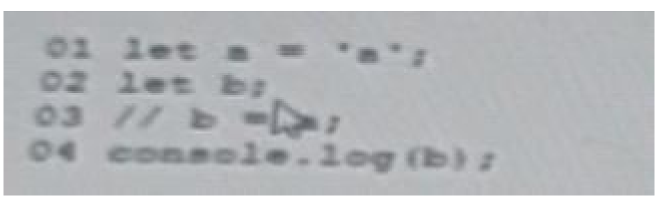
□□□□□ □□ □□□ □□□□ □□□□□□ □□□□ □□□ □□□□□ □□□.

□□ Monster() { this.name = '□□□□□' };

Const z = □□□();

02□□ □ □□□□ □□□ □□□ □□□?

A. z □□□ □□□ □□□ □□□□□□□ this.name□ □□□□ □□ □□□ □□ □□□□.
B. Window.m□ □□□ □□□ □□□□□□□□.
C. z □□□ □□□ □□□ □□□□□□□□.
D. Window.name□ 'hello'□ □□□□ □□ z□ □□□□ □□ □□□ □□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 122
□□ □□□ □□□□□.


□□□ □□□ □ □□□□ □□ □□□□□?
A. □□□□ □□
B. □
C. ReferenceError: b□ □□□□ □□□□□.
D. □
Answer: B (LEAVE A REPLY)

NEW QUESTION: 123
□□□□ □□ Promise□□ □□□ □□□ □ Promise.then □□ Promise.catch□ □□□□ □□ □□ □□□□.
□□ □ □□ □□□ □□□□□?
2□ □□ □□?
A. Promise.reject('□□□ □□ □□□ □□□□').catch(error => console.error(error));
B. Promise.reject('□□□ □□ □□□ □□□□.').then(error => console.error(error));
C. New Promise(() => (□□□ '□□ □□□ □□□□□'}).then(null, error => console.error(error)));
D. New Promise((resolve, reject) => (throw 'cool error here'}).catch(error => console.error(error)) ;
Answer: A,D (LEAVE A REPLY)

NEW QUESTION: 124
□□ □□□ □□□□□□.

```
□□ □□□(val) {
If (val === undefined) {
'□□□□ □□ □!' □□ ;
}
if (val === null) {
return 'Null □! ';
}
□□ □;
}
□□ x;
□□□(x);
```
13□□ □□ □□□ □□ □□□□ □□ □□□□□?
A. □□□□ □□
B. '□ □!'
C. 13□□□ □□□ □□□□□□.
D. '□□□□ □□ □□□□!'
Answer: (SHOW ANSWER)

NEW QUESTION: 125
□□ □□□ □□□□□.
```
01 const □□ = □□('□□');
02 /* □□□ □□ □□ */
```
□□□□ □ □□□ □□□□ □□□□□□ □□□□□. □□□ □□□□□□ □□□□ □□□□ □□□ □□□□ □□ □□
□□□□ □□□□ □ □□□ □□□□□ 03□□ □□ □□□ □□□□ □□□?
A. □□()
B. Server.start();
C. □□((□□) => (
D. server.on(' □□ ' , (□□) => {
console.log('□□ □', □□) ;})
E. console.log(' □□ □□ □ ', □□) ;
Answer: D (LEAVE A REPLY)

NEW QUESTION: 126
□□ □□:

```
function Animal(size, type){
this.size = size || "small";
this.type = type || "Animal";
this.canTalk =false;}
let Pet = function (size, type, name, owner){
    Animal.call(this, size, type);
    this.name = name;
    this.owner = owner;}
Pet.prototype = Object.create(Animal.prototype);
let  pet1 = new Pet();
console.log(pet1);
```
□□ □□□ □□□□ □□ □ □□ □□□ pet1□□ □□□□□?
3□□ □□□ □□□□□.
A. □□
B. □□□
C. □□
D. □□
E. □□
Answer: A,D,E (LEAVE A REPLY)

NEW QUESTION: 127
□□□□ □□□ □□ □□□ □□ □□□□□ □□□ □□□ □□□ □□□□ □□ □□□ □□ □□ □□□. □□□□ □□ □□□□ □□□ □□ □□□ □□□.
Const sumFunction = arr => {
□□ arr.reduce((□□, □□) => {
//
□□ += □□;
//
), 10);
);
□□□ □□□□ □□□□□ □□ □□□ □□□□□?
A. 03□□ if(arr.length == 0) (return 0;)□ □□
B. 04□□ result = result +current□ □□□□□.
C. 05□□ □□ □□□ □□□□□.
D. 02□□ return arr.map((result, current) => (
Answer: C (LEAVE A REPLY)

NEW QUESTION: 128

□□□□ File API□ □□□□ □□□□□□□ □□□ □□□ □□□□ □□□□□□ □□□. HTML□ □□□ □□□□.

```
<input type="file" onchange="previewFile()">
<img src="" height="200" alt="Image preview..."/>
```

JavaScript □□□ □□□ □□□□.

```
01 function previewFile() {
02   const preview = document.querySelector('img');
03   const file = document.querySelector('input[type=file]').files[0];
04   // line 4 code
05   reader.addEventListener("load", () => {
06     preview.src = reader.result;
07   }, false);
08   //line 8 code
09 }
```

04□□ 08□□□ □□□□ □□ □□□□□□ □□□□ □□□□ □□□□□□ □□□□ □□□ □ □□ □□□ □□□□□?

A. 04 const □□□ = □ □□( ) ;

08 if(□□) URL, createObjectURL(□□) ;

B. 04 const □□ = new FileReader( ) ;

08 if(□□) □□, createAsDataURL(□□) ;

C. 04 const □□□ = □ □□( ) ;

08 if(□□), □□, □□□ AsDataURL(□□) ;

D. 04 const □□ = new FileReader( ) ;

08 if(□□) URL, createObjectURL(□□) ;

Answer: (SHOW ANSWER)

NEW QUESTION: 129

□□□□ □□□□ □□□ □□ □□□ □□□□□ □□□ □□□□□. □□ □□ □□□□ □□ □ □□□□□.

□□□□ □□ □□□.

□□ □□□(maxSpeed, □□){

this.maxspeed =masSpeed;

this.color = □□;

carSpeed = document.getElementById(' CarSpeed');

□□□;

fourWheels = new Car(carSpeed.value, 'red')□ □□□□□.

□□ □□□ □□ 06□ □□□□□ □□ □ □ □□ □□□ □□□

□□□□ □□□□ □□□ □ □□□□?

2□□ □□□ □□□□□.

A. carSpeed □ fourWheels □□□ □

B. carSpeed DOM □□□ □□□ □□□, □□□ □□□ □ □□ □□

C. window.localStorage □□□ □□□ □□

D. Car Object□ □□ □□□ □□□□ □□ □□□□ □□□□□.

Answer: B,C (LEAVE A REPLY)