

# 300 JAVASCRIPT INTERVIEW MASTERY QUESTIONS

DIVE DEEP INTO JAVASCRIPT THEORY, SYNTAX, AND APIs  
AND INTERVIEW WITH CONFIDENCE



JONATHAN MIDDUAUGH

# 300 JAVASCRIPT INTERVIEW MASTERY QUESTIONS

DIVE DEEP INTO JAVASCRIPT THEORY, SYNTAX, AND APIs  
AND INTERVIEW WITH CONFIDENCE



JONATHAN MIDDUAUGH

# 300 JavaScript Interview Mastery Questions

Dive Deep into JavaScript Theory, Syntax, and APIs,  
and Interview with Confidence

Text copyright © 2020 Jonathan Middaugh all rights reserved.

## Introduction

This book contains questions covering JavaScript syntax, rules, functionality, and theory. It is designed to help you improve upon your JavaScript knowledge as well as expose you to some edge cases and unusual situations in JavaScript. I find that these are the types of examples that really solidify a developer's understanding of this fun language.

The book has a few questions specific to JavaScript running in the browser, but most questions are agnostic of environment. A few questions also include TypeScript, but they will be designated as such. The book is organized into 4 “tests” with each question linking to its answer in the back of the book.

As a bonus, the two links below provide another 50 free JavaScript interview mastery questions and 50 free TypeScript interview mastery question.

[50 JavaScript Interview Mastery Questions](#) (published on my Medium.com publication).

[50 TypeScript Interview Mastery Questions](#) (published on my Medium.com publication).

## About the Author

If you enjoy this book and want to continue learning JavaScript with me, take a look at the following:

- [My Medium.com Portfolio](#) – Filled with JavaScript articles and the occasional dev side hustle I’m working on
- [My YouTube Channel](#) – Same as my Medium.com portfolio, but I enjoy the interactive feel of making the videos. Plus I believe they can help different people learn in different ways
- [Twitter](#) – Follow me here for news about recent articles plus whatever is interesting in the dev community

## Table of Contents

[Test 1 Questions](#)

[Test 2 Questions](#)

[Test 3 Questions](#)

[Test 4 Questions](#)

[Test 1 Answers](#)

[Test 2 Answers](#)

[Test 3 Answers](#)

[Test 4 Answers](#)

# Test 1 Questions

Q1.

Which of the following prints first?

```
setTimeout(() => {  
    console.log('setTimeout wins');  
},0);  
  
queueMicrotask(() => {  
    console.log('queueMicrotask wins');  
});
```

- A) setTimeout wins
- B) queueMicrotask wins
- C) Compile time error

The answer is B. Tasks from queueMicrotask are called after the callstack is empty and before the event loop is called. Tasks from setTimeout are part of the eventQueue.

[Q1 Answered.](#)

Q2.

What will the code below output to the console and why?

```
let x = 10;  
const byValue = (y) => {  
    y = 20;  
}  
  
byValue(x);  
console.log(x);
```

- A) 20
- B) 10
- C) Compile time error

[Q2 Answered.](#)

Q3.

Consider the following code. What will it log to the console and why?

```
let x = 5;  
let x = 10;  
  
console.log(x);
```

- A) 5
- B) 10
- C) It will throw a compiler error.

[Q3 Answered.](#)

Q4.

Consider the following code. What does each line output?

```
console.log(new String("yes") === new String("yes")); // Line 1  
console.log("yes" === "yes"); // Line 2
```

- A) Line 1 prints true, Line 2 prints true
- B) Line 1 prints false, Line 2 prints true
- C) Line 1 prints true, Line 2 prints false
- D) Line 1 prints false, Line 2 prints false

[Q4 Answered.](#)

Q5.

What will the code below output to the console and why?

```
const x = new String("tricky");
const y = x;

console.log(x === y);
```

- A) True
- B) False

[Q5 Answered.](#)

Q6.

Are arrays objects or primitives in JavaScript?

- A) Objects
- B) Primitives

[Q6 Answered.](#)

Q7.

What is the return type of the async function below?

```
async function goodbye() {  
    return 'goodbye';  
}
```

- A) string
- B) Promise
- C) Async

[Q7 Answered.](#)

Q8.

True or False? The await keyword blocks all JavaScript code execution in an app until the awaited Promise is returned.

- A) True
- B) False

[Q8 Answered.](#)

Q9.

Recall that functions are objects in JavaScript. What does the following print?

```
function myFunc() {}  
console.log(typeof myFunc);
```

- A) object
- B) function
- C) true
- D) It throws an error

[Q9 Answered.](#)

Q10.

True or False? The following is valid syntax for printing ‘goodbye’.

```
let goodbye = async () => { return "Goodbye" };  
goodbye().then(console.log);
```

- A) True
- B) False

[Q10 Answered.](#)

Q11.

What of the following is not a difference between `typeof` and `instanceof`?

- A) `typeof` returns a type and `instanceof` returns a boolean
- B) `instanceof` requires TypeScript and `typeof` does not
- C) `typeof` takes the variable name on the right while `instanceof` takes a value on the left and right

[Q11 Answered.](#)

Q12.

Which of the following resolves when all promises are fulfilled, or rejects as soon as one promise is rejected?

- A) Promise.race()
- B) Promise.any()
- C) Promise.all()
- D) Promise.resolve()

[Q12 Answered.](#)

Q13.

What will the code below output to the console and why?

```
let x = 10;  
  
function myFunc(y){  
    y = 12;  
}  
myFunc(x);  
console.log(x);
```

- A) 10
- B) 12
- C) undefined

[Q13 Answered.](#)

Q14.

What does the below code output to the console and why?

```
console.log(11 & 3);
```

- A) 11
- B) 1
- C) 2
- D) 3

[Q14 Answered.](#)

Q15.

What is the value of Object.[[Prototype]]?

- A) Object
- B) null
- C) {}

[Q15 Answered.](#)

Q16.

What does the nullish coalescing operator do?

[Q16 Answered.](#)

Q17.

What operator will return the remainder when one operand is divided by another operand?

- A) \$
- B) #
- C) %
- D) None of the above

[Q17 Answered.](#)

Q18.

True or False? getElementsByTagName is a JavaScript function.

- A) True
- B) False

[Q18 Answered.](#)

Q19.

Give an example of when to use event delegation in JavaScript.

[Q19 Answered.](#)

Q20.

Does the code below evaluate to true or false?

`BigInt(1000000000000000000) === 1000000000000000000`

- A) true
- B) false

[Q20 Answered.](#)

Q21.

Which of the following is not a built-in JS error type?

- A) Error
- B) EvalError
- C) SyntaxError
- D) TypeError
- E) UndefinedError

[Q21 Answered.](#)

Q22.

How are numbers stored in JavaScript? Select all that apply.

- A) Double precision
- B) 64-bit
- C) Floating point numbers

[Q22 Answered.](#)

Q23.

If a function has no return value specified, what value does it return?

- A) null
- B) undefined
- C) nothing

[Q23 Answered.](#)

Q24.

What is the syntax for calling a method which may or may not exist on an object?

[Q24 Answered.](#)

Q25.

Which of the following is not a valid Promise method?

- A) Promise.some()
- B) Promise.allSettled()
- C) Promise.reject()
- D) Promise.race

[Q25 Answered.](#)

**Q26.**

How can you dynamically access an object property based on a variable value?

[Q26 Answered.](#)

Q27.

True or False? A string can be modified after it is created.

- A) True
- B) False

[Q27 Answered.](#)

Q28.

True or False? JavaScript is synchronous.

- A) True
- B) False

[Q28 Answered.](#)

Q29.

True or False? A nested catch in a promise chain can catch errors thrown higher up in the promise chain?

- A) True
- B) False

[Q29 Answered.](#)

Q30.

True or False? The Map object can have keys that are of types other than strings, such as number.

- A) True
- B) False

[Q30 Answered.](#)

Q31.

What will the code below output to the console and why?

```
const map = new Map();
map.set({},1);
console.log(map.get({}));
```

- A) undefined
- B) 1
- C) Compile time error

[Q31 Answered.](#)

Q32.

What will the code below output to the console and why?

```
let map1 = new Map([
  ['a', 1],
  ['b', 1],
  ['c', 1],
]);
```

```
let map2 = new Map([
  ['a', 2],
  ['b', 2]
]);
```

```
let spreadMap = new Map([...map1, ...map2]);
```

- A) Map(3) {"a" => 1, "b" => 1, "c" => 1}
- B) Map(3) {"a" => 2, "b" => 2, "c" => 1}
- C) Map(5) {"a" => 1, "b" => 1, "c" => 1, "a" => 2, "b" => 2}

[Q32 Answered.](#)

Q33.

True or False? Bracket notation can be chained just like dot notation?

- A) True
- B) False

[Q33 Answered.](#)

Q34.

What will the code below output to the console and why?

```
class Person {  
    name: string;  
    DOB: string;  
}  
console.log(typeof Person);
```

- A) “class”
- B) “function”
- C) “object”

[Q34 Answered.](#)

Q35.

Examine the code below. Diagnose an error that will occur. Assume that handleResponse and handleError are properly defined.

```
promise.then(  
    function(response) {  
        this.handleResponse();  
    },  
    function(error) {  
        this.handleError();  
    }  
)
```

[Q35 Answered.](#)

Q36.

What types of properties show up in for...in loops?

- A) Executable Properties
- B) Enumerable Properties
- C) Iterable Properties

[Q36 Answered.](#)

Q37.

True or False? .bind(this) immediately calls the bound function.

- A) True
- B) False

[Q37 Answered.](#)

Q38.

What does the code print?

```
function getName() {  
    var name = "Jon";  
    function printName() {  
        console.log(name);  
    }  
    printName();  
}  
getName();
```

- A) Jon
- B) undefined
- C) Compile time error

[Q38 Answered.](#)

Q39.

True or False? The await keyword is required with async functions?

- A) True
- B) False

[Q39 Answered.](#)

Q40.

What are the three ways for a function to refer to itself for recursion? Select three answers.

- A) Use self.run
- B) The function's name
- C) An in-scope variable that refers to the function
- D) Use argumentscallee

[Q40 Answered.](#)

Q41.

What will the code below output to the console and why?

```
const test = (incrementer) => {
  if (incrementer < 0){
    return;
  }
  console.log('first: ' + incrementer);
  test(incrementer - 1);
  console.log('second: ' + incrementer)
}
test(2);
```

[Q41 Answered.](#)

**Q42.**

True or False? The difference in Promise.any() and Promise.race() is that Promise.race() returns the first settled Promise (fulfilled *or* rejected), while Promise.any() continues to resolve until a promise is fulfilled, and only rejects if all promises are rejected.

- A) True
- B) False

[\*\*Q42 Answered.\*\*](#)

Q43.

True or False? JavaScript does not support overloading?

- A) True
- B) False

[Q43 Answered.](#)

**Q44.**

Without using class syntax, create a child object from a parent object.

[Q44 Answered.](#)

Q45.

What does a return statement do in a forEach loop on an Array?

- A) Nothing
- B) Returns the specified value
- C) Ends that loop

[Q45 Answered.](#)

Q46.

True or False? There are a maximum of three parameters in a for loop.

- A) True
- B) False

[Q46 Answered.](#)

Q47.

True or False? RegExp do not have any properties.

- A) True
- B) False

[Q47 Answered.](#)

Q48.

What does the code below output to the console and why?

```
function resolveQuickly(x) {  
    return new Promise(resolve => {  
        console.log(x)  
    });  
}
```

```
async function asyncTest() {  
    var x = resolveQuickly(10);  
    console.log(5);  
}
```

```
asyncTest();
```

- A) 10, 5
- B) 5, 10
- C) It does not print

[Q48 Answered.](#)

Q49.

What is blocking?

[Q49 Answered.](#)

Q50.

Which of the following is not a valid state for a Promise?

- A) Pending
- B) Fulfilled
- C) Rejected
- D) Completed

[Q50 Answered.](#)

Q51.

Which function executes a specified code block before the browser next repaints the display?

- A) requestAnimationFrame()
- B) setTimeout()
- C) requestRepaintDelay()

[Q51 Answered.](#)

Q52.

Assume you want to achieve a refresh rate of 60fps on an animation. Write a setInterval function that triggers the below pseudocoded animate function at 60fps.

```
function animate() {...}
```

[Q52 Answered.](#)

Q53.

What are the two types of exports in JavaScript?

- A) Abstract
- B) Whole
- C) Default
- D) Named

[Q53 Answered.](#)

Q54.

What statement would you use to iterate over asynchronous objects?

- A) if async then
- B) for await...of
- C) async iterable
- D) do while async

[Q54 Answered.](#)

Q55.

Why use an alias when importing a module?

[Q55 Answered.](#)

Q56.

True or False? The below code results in the same output.

```
var i, j;  
loop1:  
for (i = 0; i < 5; i++) {  
    loop2:  
    for (j = 0; j < 5; j++) {  
        if (i === 1 && j === 1) {  
            continue loop1;  
        }  
        console.log(`i: ${i}, j: ${j}`)  
    }  
}  
  
var i, j  
loop1:  
for (i = 0; i < 5; i++) {  
    loop2:  
    for (j = 0; j < 5; j++) {  
        if (i === 1 && j === 1) {  
            break loop2;  
        }  
        console.log(`i: ${i}, j: ${j}`)  
    }  
}
```

A) True

B) False

[Q56 Answered.](#)

Q57.

True or False? The following are equivalent.

```
var a = 1;  
var z = a;
```

and

```
var a, z = a = 1;
```

- A) True
- B) False

[Q57 Answered.](#)

Q58.

Use a reducer to find the min value of the below array.

```
const arr = [5, 10, 12, 4, 2];
```

[Q58 Answered.](#)

Q59.

What is a subroutine in JavaScript?

[Q59 Answered.](#)

Q60.

True or False? You might use eventHandlers *cut* and *copy* to keep a user from copying content from the browser to the clipboard?

- A) True
- B) False

[Q60 Answered.](#)

Q61.

What will the code below output to the console and why?

```
let i = 10
for(; i< 12;){
    console.log(++i);
}
```

- A) lint error
- B) runtime error
- C) infinite loop
- D) 11, 12

[Q61 Answered.](#)

Q62.

What are the three ways to create a new object?

- A) new Object()
- B) Object.create()
- C) factory pattern
- D) literal notation

[Q62 Answered.](#)

Q63.

What will the code below output to the console and why?

```
const a = {x: 1}
const b = {...a}
const c = {};
Object.assign(c, a);
a.x = 2;
console.log(a.x);
console.log(b.x);
console.log(c.x);
```

- A) 1, 1, 1
- B) 2, 2, 2
- C) 1, 2, 2
- D) 2, 1, 1
- E) 2, 1, 2
- F) 2, 2, 1

[Q63 Answered.](#)

Q64.

Implement the classic example of a closure that increments up each time it is called.

[Q64 Answered.](#)

Q65.

If myFunction is called as myFunction(5); what will be the return value?

```
function myFunction(x, y) {  
    const arr = Array.prototype.slice.call(arguments);  
    return arr.length;  
}
```

- A) 0
- B) 1
- C) 2

[Q65 Answered.](#)

Q66.

Which of the below does Object.seal() do? Select all that apply.

- A) prevents new properties from being added
- B) marks all existing properties as non-configurable
- C) prevents values of existing properties from being changed

[Q66 Answered.](#)

Q67.

Which of the below does Object.freeze() do? Select all that apply.

- A) prevents new properties from being added
- B) prevents existing properties from being removed
- C) prevents changing the enumerability of properties
- D) prevents changing the configurability of properties
- E) prevents changing the writability of properties
- F) prevents changing the values of properties
- G) prevents changing the prototype of the object

[Q67 Answered.](#)

Q68.

What is the difference between event.target and event.currentTarget?

- A) event.currentTarget returns the element the triggered listener was attached to
- B) event.target returns the most granular element the event was triggered on.
- C) event.target returns the element the triggered listener was attached to
- D) event.currentTarget returns the most granular element the event was triggered on.

[Q68 Answered.](#)

Q69.

If you need to call an anonymous function later, show what syntax you would use for creating the anonymous function.

[Q69 Answered.](#)

Q70.

What is the default sort of the Array sort() method?

- A) largest to smallest by numeric value
- B) smallest to largest by numeric value
- C) largest to smallest by character value
- D) smallest to largest by character value

[Q70 Answered.](#)

Q71.

What will the code below output to the console and why?

```
let scope = "global";
getScope = () => {
  var scope = "local";
  const inner = () => {return scope;}
  return inner;
}

getScope()();
```

- A) global
- B) local
- C) undefined

[Q71 Answered.](#)

Q72.

What will the code below output to the console and why?

```
console.log(3 + '2');
```

- A) 32
- B) 5
- C) runtime error

[Q72 Answered.](#)

Q73.

What is a race condition?

[Q73 Answered.](#)

Q74.

What is functional programming?

[Q74 Answered.](#)

Q75.

What is a higher order function?

- A) a function composed of multiple functions or interfaces
- B) a function that can take a function as an argument, or that returns another function
- C) a function that is a superclass for child objects

[Q75 Answered.](#)

## Test 2 Questions

Q1.

Consider the following code:

```
class Person {  
    firstName: string;  
    lastName: string;  
}  
  
const myPerson = new Person();  
console.log(myPerson.a);
```

Explain how the prototype chain is used to determine that myPerson.a is undefined?

[Q1 Answered.](#)

Q2.

What does the *class* keyword do in JavaScript?

- A) It conveys real meaning beyond what a typical object is
- B) It is syntactic sugar to make JavaScript more object oriented
- C) It creates an interface-like object that can be used for inheritance

[Q2 Answered.](#)

Q3.

What will the code below output to the console and why?

```
let x = 10;  
if(x){  
    let x = 12;  
    console.log(x);  
}
```

- A) 10
- B) 12
- C) It will throw an error

[Q3 Answered.](#)

Q4.

True or False? Tasks from the queueMicrotask queue are executed in a Last In, First Out basis.

- A) True
- B) False

[Q4 Answered.](#)

Q5.

What would be a name for this function that follows “clean code” principles?

```
const pntDT = () => {  
  const today = new Date();  
  const time = today.getHours() + ":" + today.getMinutes();  
  console.log(time);  
};  
  
pntDT();
```

[Q5 Answered.](#)

Q6.

Which of the following is not a step to create and register a new custom element?

- A) Create an HTML container in the DOM
- B) Create a javascript class that defines the behavior of the element
- C) Register the class and a DOMString representing the name of the element with the CustomElementRegistry object

[Q6 Answered.](#)

Q7.

Briefly explain the Shadow DOM API.

[Q7 Answered.](#)

Q8.

What method attaches a shadow DOM tree to the specified element and returns a reference to its ShadowRoot?

- A) Element.shadowAdd()
- B) Element.setDOMShadow()
- C) Element.attachShadow()

[Q8 Answered.](#)

Q9.

What value will arr[2][1] return?

```
const arr = [  
    ["a", "b", "c"],  
    ["d", "e", "f"],  
    ["g", "h", "i"]  
];
```

- A) d
- B) h
- C) b
- D) None of the above

[Q9 Answered.](#)

Q10.

What will the code below output to <app>?

```
document.getElementById("app").innerHTML = `  
<div>  
hewo  
    hewo2  
</div>`
```

- A) hewo hewo2
- B)
- hewo  
 hewo2
- C) hewo hewo2
- D) Does not compile

[Q10 Answered.](#)

Q11.

What pattern does jQuery use to make a single instance of it's globally available jQuery object?

- A) OOP
- B) Factory
- C) Singleton
- D) None

[Q11 Answered.](#)

Q12.

What is the difference between window.localStorage and window.sessionStorage? Select the best answer.

- A) localStorage stores key/value pairs in the web browser, and localStorage deletes after the browser closes
- B) sessionStorage stores key/value pairs in the web browser, and sessionStorage deletes after the browser closes
- C) Both store key/value pairs in the web browser, but localStorage deletes after the browser closes
- D) Both store key/value pairs in the web browser, but sessionStorage deletes after the browser closes

[Q12 Answered.](#)

Q13.

What will the code below output to the console and why?

```
let arr1 = [1, 2, 3];
arr1.push(4);
arr1.pop();
arr1.slice(1, 2);
arr1.splice(1, 2);
console.log(arr1);
```

- A) 1
- B) [1]
- C) [1,2]
- D) [2,3]
- E) Prints nothing.

[Q13 Answered.](#)

Q14.

True or False: The ! operator returns a boolean. Select one option.

- A) True
- B) False

[Q14 Answered.](#)

Q15.

Evaluate the code below. If both div and button have a click listener, how can you stop a click on the button from bubbling up to the div?

```
<div>
  <button>Click Me!</button>
</div>
```

- A) Event.stopPropagation in the button clickHandler
- B) Event.stopPropagation in the div clickHandler
- C) Event.stopBubble in the button clickHandler
- D) Event.stopBubble in the div clickHandler

[Q15 Answered.](#)

Q16.

What will the code below output to the console and why?

```
let xyz = "";  
xyz ||= "abc";  
console.log(xyz);
```

- A) abc
- B) “
- C) Compile time error

[Q16 Answered.](#)

Q17.

What is the difference in the output bewteen the two following functions?

```
void function voidExample() {  
    console.log("No Error!");  
}();
```

```
function noVoidExample() {  
    console.log("No Error!");  
}();
```

- A) The function noVoidExample executes and prints No Error, while voidExample throws an error.
- B) The function voidExample executes and prints No Error, while noVoidExample throws an error.
- C) Both throw an error

[Q17 Answered.](#)

Q18.

Which of the following is NOT a difference between NodeList and Array?

- A) NodeList is Web API, Array is JavaScript API
- B) NodeList cannot contain more than 10 items
- C) NodeList properties may be live properties that update if the DOM is updated (like `childNodes`)
- D) Many array functions are not available on NodeList

[Q18 Answered.](#)

Q19.

Which es6 array functions in JavaScript return a new array? Select all correct answers.

- A) map()
- B) forEach()
- C) filter()
- D) find()

[Q19 Answered.](#)

Q20.

What is the largest whole number JavaScript can reliably represent with the number primitive?

- A)  $2^{53} - 1$
- B)  $2^{32} - 1$
- C)  $2^{63} - 1$
- D) JavaScript can represent infinitely large numbers

[Q20 Answered.](#)

Q21.

Which function will modify a string to be URI compliant?

- A) JSON.stringify()
- B) encodeURI()
- C) encodeURIComponent()
- D) JSON.parse()

[Q21 Answered.](#)

Q22.

How are the 64 bits assigned to a number used to store numbers in JS?

[Q22 Answered.](#)

Q23.

In computer science, what is the difference between *null* and *undefined* ?

[Q23 Answered.](#)

Q24.

What is the technical name of the minus sign (-) in JavaScript?

- A) Unary Subtraction
- B) Binary Negation
- C) Unary Negation
- D) Binary Subtraction

[Q24 Answered.](#)

Q25.

True or False? The super keyword cannot be used outside of constructor functions.

- A) True
- B) False

[Q25 Answered.](#)

Q26.

What are two ways to access the color value on the cat object?

```
const cat = {  
    color: "orange"  
};
```

[Q26 Answered.](#)

Q27.

What is the Stack in JavaScript Runtime?

[Q27 Answered.](#)

Q28.

What is the syntax for hexidecimal escape sequence?

- A) '\xab1'
- B) '\10ab'
- C) '\x'

[Q28 Answered.](#)

Q29.

Write the code to create a custom made abstract class in vanilla JavaScript (no TypeScript)? The abstract keyword is not available as of ES2015.

[Q29 Answered.](#)

Q30.

What will the code below output to the console and why?

```
const badMap = new Map();
badMap.set('a', 1);
badMap.b = 2
console.log(badMap)
```

- A) Map(1) {"a" => 1}
- B) Map(1) {"b" => 2}
- C) Map(2) {"a" => 1, "b" => 2}
- D) Compile time error

[Q30 Answered.](#)

Q31.

What are the benefits of object-oriented programming (OOP)? Select three answers.

- A) Code is socketable or modular, so a class's source code can be maintained independently of the rest of an application's code.
- B) An object's code is kept private, meaning other objects only interact with an object's methods, so the internal code is kept hidden.
- C) Objects are easily reusable. They can be plugged into multiple parts of the code.
- D) Object-oriented code executes faster and computer resource usage is minimized.

[Q31 Answered.](#)

Q32.

Assume const map has the following values: Map(2) {"a" => 2, "b" => 2}  
In one line, transform map into a 2d array.

[Q32 Answered.](#)

Q33.

True or False? Only custom classes and objects can be extended in JavaScript?

- A) True
- B) False

[Q33 Answered.](#)

Q34.

True or false? An instance level variable can be accessed anywhere in a class as long as an instance has been created.

- A) True
- B) False

[Q34 Answered.](#)

Q35.

Why will this code not compile? Select one option.

```
let i: number;  
if(5%3>2){  
    i=0;  
}else if(5%3>1){  
    i=1;  
}  
console.log(i);
```

- A) It compiles fine
- B)  $5\%3 > 2$  and  $5\%3 > 1$  do not return booleans and thus cannot be used in an if statement
- C) i is a local variable and its value is never explicitly defined

[Q35 Answered.](#)

Q36.

True or False? length is a method on strings and a property on Arrays?

- A) True
- B) False

[Q36 Answered.](#)

Q37.

What is the first argument of function.call?

- A) A parameter for the function called
- B) It takes no options
- C) The instance object to invoke the function on

[Q37 Answered.](#)

Q38.

What is a technical name for the kind of function that printName is? Select all correct options.

```
function getName() {  
    var name = "Jon";  
    function printName() {  
        console.log(name);  
    }  
    printName();  
}  
getName();
```

- A) Inner function
- B) Contained function
- C) Closure

[Q38 Answered.](#)

Q39.

True or False? Arrow functions only require parenthesis if there are multiple arguments.

- A) True
- B) False

[Q39 Answered.](#)

Q40.

True or false? A class level variable can be accessed anywhere in a class.

- A) True
- B) False

[Q40 Answered.](#)

Q41.

What kind of error is thrown when you attempt to assign a new value to a const?

- A) ReferenceError
- B) Error
- C) RangeError
- D) TypeError

[Q41 Answered.](#)

**Q42.**

Which of the following are true of getters and setters? Select 2 options.

- A) Getters and setters increase code reusability.
- B) Getters and setters are an integral part of Object Oriented Programming
- C) Getters and setters control the flow of the program
- D) Getters and setters maintain encapsulation

[Q42 Answered.](#)

Q43.

Overriding and overloading are examples of which Object Oriented Programming principle?

- A) inheritance
- B) abstraction
- C) polymorphism
- D) encapsulation

[Q43 Answered.](#)

Q44.

True or False: static variables cannot be modified in non-static methods?

- A) True
- B) False

[Q44 Answered.](#)

Q45.

What happens if str === 'test1' in the below code?

```
switch (str) {  
    case "test1":  
    case "test2":  
        console.log("At test2");  
}
```

- A) “At test2” is printed
- B) Nothing
- C) Runtime Error

[Q45 Answered.](#)

**Q46.**

What does it mean when JavaScript is described as having “first-class functions”?

- A) functions can be recursive
- B) functions can be anonymous
- C) functions are treating like any other variable
- D) functions use the curly brace syntax

[\*\*Q46 Answered.\*\*](#)

Q47.

True or False? The following code compiles.

```
async function promiseMe() {  
    try {  
        let z = await Promise.reject(30);  
    }  
}
```

- A) True
- B) False

[Q47 Answered.](#)

Q48.

True or False? Date.getDate() returns the day of the month with values 1-31 while Date.getDay() returns the day of the week with values of 0-6.

- A) True
- B) False

[Q48 Answered.](#)

Q49.

Async operations are put into what kind of queue?

- A) blocking queue
- B) event queue
- C) async queue

[Q49 Answered.](#)

Q50.

What are the three functions JavaScript has long offered for executing code after time has elapsed? Select three answers.

- A) setTimeout()
- B) setInterval()
- C) callAsyncLoop()
- D) requestAnimationFrame

[Q50 Answered.](#)

Q51.

What Web API might you call if you wanted to measure the execution time of a code block?

- A) window.now()
- B) window.timer()
- C) performance.now()
- D) performance.timer()

[Q51 Answered.](#)

Q52.

What is a MIME type?

[Q52 Answered.](#)

Q53.

What is a simple syntax for exporting the below variables?

```
let myLet = 10;
```

[Q53 Answered.](#)

Q54.

What function property returns the number of args the function expects?

- A) Function.argsLength
- B) Function.length
- C) Function.parameters

[Q54 Answered.](#)

Q55.

What does it mean to "import a module for side effects only"?

[Q55 Answered.](#)

Q56.

True or False? The below code compiles and prints 3?

```
const test = () => {  
    return  
    1 + 2;  
}  
console.log(test());
```

- A) True
- B) False

[Q56 Answered.](#)

Q57.

True or False? The following are equivalent.

```
var a, z = 1;  
a = z;
```

and

```
var a = z = 1;
```

- A) True
- B) False

[Q57 Answered.](#)

Q58.

True or False? The code below returns 3.

```
const arr = [1, 2];
const reduced = arr.reduce((accumulator,element) => accumulator);
```

- A) True
- B) False

[Q58 Answered.](#)

Q59.

Where must the "use strict"; directive be placed in a file?

- A) Top
- B) Middle
- C) Bottom
- D) As an import

[Q59 Answered.](#)

Q60.

How can you use composition in vanilla JavaScript (no TypeScript)?

[Q60 Answered.](#)

Q61.

What will the code below output to the console and why?

```
for(let i = 1, j = 10; i< 5 < j; i += j/i){  
    console.log(Math.ceil(++i));  
}
```

- A) nothing
- B) divide by zero error
- C) 1, 12
- D) It will enter into an infinite loop

[Q61 Answered.](#)

Q62.

Show how object literals can be constructed without use of the colon :

[Q62 Answered.](#)

Q63.

What is the maximum size of an array in JavaScript?

- A)  $2^{16} - 1$
- B)  $2^{32} - 1$
- C)  $2^{16} - 1$
- D)  $2^8 - 1$

[Q63 Answered.](#)

Q64.

If a function is called with less parameters than the declared amount of parameters, what value is assigned to the missing parameters?

- A) null
- B) undefined
- C) 0

[Q64 Answered.](#)

Q65.

What is the Event Capturing phase?

[Q65 Answered.](#)

Q66.

True or False? Setters still work on a frozen object.

- A) True
- B) False

[Q66 Answered.](#)

Q67.

What does  $3 > 2 > 1$  return?

- A) true
- B) false
- C) 0

[Q67 Answered.](#)

Q68.

The below two code snippets return the same value. What is the difference in how that occurred?

```
'123'.length; //line 1  
new String(123).length; //line 2
```

- A) line 1 uses explicit coercion, line 2 uses implicit coercion
- B) line 1 uses implicit coercion, line 2 uses explicit coercion
- C) There is no difference

[Q68 Answered.](#)

Q69.

Which of the following is NOT a time that you would use an anonymous function?

- A) as a class method
- B) in a closure
- C) as a callback
- D) as an IIFE (immediately invoked function execution)

[Q69 Answered.](#)

Q70.

What is the difference in layout time and painting time in a web browser?

[Q70 Answered.](#)

Q71.

What will the code below output to the console and why?

```
let a = 8  
a >>>= 3;  
console.log(a);
```

- A) 1
- B) 10
- C) 4

[Q71 Answered.](#)

Q72.

What will the code below output to the console and why?

```
console.log([1, 2,,3].join(', '));
```

- A) lint error
- B) 1, 2, 3
- C) 1, 2, , 3

[Q72 Answered.](#)

Q73.

Give an example of how to avoid race conditions on network calls?

[Q73 Answered.](#)

Q74.

What is a pure function?

[Q74 Answered.](#)

Q75.

What is currying?

[Q75 Answered.](#)

## Test 3 Questions

Q1.

True or False? The compiler will throw an error for the below code.

```
const x = 10;  
x = 12;
```

- A) True
- B) False

[Q1 Answered.](#)

Q2.

What is the TypeScript type of the below code?

```
const x = "hello"
```

- A) "hello"
- B) const
- C) string
- D) No type is assigned

[Q2 Answered.](#)

Q3.

What will the code below output to the console and why?

```
let person = {  
    age: 25,  
    name: 'Justin'  
}  
  
const byRef = (myObj) => {  
    myObj.age = 30  
}  
  
byRef(person);  
console.log(person.age);
```

- A) 25
- B) 30
- C) Compile time error

[Q3 Answered.](#)

Q4.

What will the code below output to the console and why?

```
let x;  
if(x = true === true){  
    console.log(x);  
} else {  
    console.log('Not x');  
}
```

- A) true
- B) false
- C) Not x
- D) Compile time error

[Q4 Answered.](#)

Q5.

What will the code below output to the console and why?

```
let x = 10;  
  
function myFunc(y){  
    y = 12;  
}  
console.log(myFunc(x));
```

- A) 10
- B) 12
- C) undefined

[Q5 Answered.](#)

## Q6.

Assume the aggregator function below calculates how many days old you are, combines your first and last names, and splits apart your address. Select all the ways it violates clean code principles.

```
const aggregator = (DOB, firstName, lastName, address, skills) => {...}
```

- A) The function is poorly named
- B) The function should use the function keyword instead of being a lambda function
- C) The function is doing too many things and the function should be split up
- D) The function doesn't do anything with the skills parameter

[Q6 Answered.](#)

Q7.

What is the technical definition of the DOM from a JS perspective?

[Q7 Answered.](#)

Q8.

Given FancyButton class, how would you extend button to create a custom element named fancy-button?

- A) customElements.extend('fancy-button', FancyButton, { extends: 'button' });
- B) customElements.extend('fancy-button', FancyButton, { implements: 'button' });
- C) customElements.define('fancy-button', FancyButton, { extends: 'button' });
- D) customElements.define('fancy-button', FancyButton, { implements: 'button' });

[Q8 Answered.](#)

Q9.

Which of the following is correct string interpolation syntax?

const x = 'me';

- A) `Pick \${x}`
- B) `Pick &{x}`
- C) 'Pick \${x}'
- D) 'Pick &{x}'

[Q9 Answered.](#)

Q10.

What is the name of the pattern where the total number of instances of an object is limited to one?

- A) Singleton
- B) Factory
- C) Design
- D) OOP

[Q10 Answered.](#)

Q11.

Which of the following is not a method on the Set object?

- A) add()
- B) next()
- C) clear()
- D) delete()
- E) entries()

[Q11 Answered.](#)

Q12.

What window method allows you to store persistent key/value pairs in browser memory?

- A) window.browserStorage
- B) window.localStorage
- C) window.persistentStorage
- D) Nothing allows for this

[Q12 Answered.](#)

Q13.

All bitwise operations are performed on what bit binary numbers?

- A) 8
- B) 16
- C) 32
- D) 64

[Q13 Answered.](#)

Q14.

When does the delete operator not remove a property from an object?

[Q14 Answered.](#)

Q15.

What method should you call if you need to get the properties of an Object?

- A) Object.getKeyValues
- B) Object.getList
- C) Object.getNames
- D) Object.getOwnPropertyNames

[Q15 Answered.](#)

Q16.

What will the code below output to the console and why?

```
console.log(2 * 3 ** 3);
```

- A) 216
- B) 18
- C) 54
- D) Compile time error

[Q16 Answered.](#)

Q17.

Evaluate the code below. What is a one character change that can prevent this line from throwing an error?

```
const dog = {  
    color: "brown",  
    breed: {  
        pure: true  
    }  
};  
console.log(dog.hair.short)
```

[Q17 Answered.](#)

Q18.

Which array functions are not available on NodeList? Select all that apply.

- A) forEach()
- B) join()
- C) filter()
- D) map()

[Q18 Answered.](#)

Q19.

Which of the following is not a JSON method?

- A) reverse()
- B) parse()
- C) stringify()

[Q19 Answered.](#)

Q20.

True or False? There is a global Map Object in JavaScript.

- A) True
- B) False

[Q20 Answered.](#)

Q21.

What does it mean that a WeakSet object is weak?

[Q21 Answered.](#)

Q22.

What is one way to overcome the imprecision of floating point decimals?

- A) multiply and divide
- B) It works as-is
- C) run through an expander function

[Q22 Answered.](#)

Q23.

Evaluate the code below. How can it be refactored to use less code?

```
const temp = object.prop1;  
const test = ((temp === null || temp === undefined)) ? undefined :  
temp.prop2;
```

[Q23 Answered.](#)

Q24.

Evaluate the code below. What are the first and last outputs of the loop?

```
for(let i = 0; i <= 10; ++i){  
    console.log(i);  
}
```

- A) 0, 9
- B) 0, 10
- C) 1, 9
- D) 1, 10

[Q24 Answered.](#)

Q25.

Create a function on the below MyChild class that calls the parent printSomething method.

```
class MyParent {  
    constructor(){  
    }  
    printSomething(){  
        console.log(5);  
    }  
}  
class MyChild extends MyParent {  
    constructor(){  
        super()  
    }  
    printSomething(){  
        console.log(10);  
    }  
}
```

[Q25 Answered.](#)

Q26.

What does the code below return and why?

const value = 10 / -0;

- A) NaN
- B) -NaN
- C) Infinity
- D) -Infinity

[Q26 Answered.](#)

Q27.

What is Heap in JavaScript Runtime?

[Q27 Answered.](#)

Q28.

How does JavaScript handle it behind the scenes when a String Object method is called on a string literal value?

[Q28 Answered.](#)

Q29.

What will the code below output to the console and why?

```
const obj = {  
    a: 1,  
    b: 2  
};  
  
const arr = [];  
console.log(arr[0] = {...obj});
```

- A) 1, 2
- B) {...obj}
- C) {a: 1, b: 2}
- D) Does not compile

[Q29 Answered.](#)

Q30.

What does Map.delete(*key* ) return?

- A) It has no return
- B) *true* always
- C) *false* always
- D) *true* if the key was deleted, *false* if the item didn't exist and wasn't deleted

[Q30 Answered.](#)

Q31.

Which of the following is not an example of inheritance when using TypeScript? Select one option.

- A) An object has a field that its superclass has
- B) An object has a method that its superclass has
- C) An object has a method that its interface has
- D) An object has a static field that its superclass has

[Q31 Answered.](#)

Q32.

True or false? In JavaScript, an object can have only one subclass but unlimited parent classes.

- A) True
- B) False

[Q32 Answered.](#)

Q33.

True or false? An object can implement multiple interfaces. Assume TypeScript is enabled.

- A) True
- B) False

[Q33 Answered.](#)

Q34.

What does the code print? Select one option.

```
class MyClass {  
    static staticMethod() {  
        return this.name;  
    }  
    constructor() {  
        this.name = 'Jon';  
    }  
}  
  
const me = new MyClass();  
console.log(MyClass.staticMethod());
```

- A) Jon
- B) MyClass
- C) Compile time error
- D) Runtime error

[Q34 Answered.](#)

Q35.

True or False? A String is a primitive data type.

- A) True
- B) False

[Q35 Answered.](#)

Q36.

Show two ways to make an existing array empty.

[Q36 Answered.](#)

Q37.

What is the value of str after the code executes?

```
const str = " this".concat("that ").substring(3).trim().valueOf(2);
```

- A) 2
- B) h
- C) histhat
- D) Does not compile

[Q37 Answered.](#)

Q38.

What is a Generator function in JavaScript?

[Q38 Answered.](#)

Q39.

Which of the following is a valid modifier of class, assuming TypeScript is enabled? Select one option.

- A) private
- B) final
- C) abstract
- D) public
- E) default

[Q39 Answered.](#)

Q40.

Which of the following is not a reserved word in JavaScript? Select one option.

- A) abstract
- B) compile
- C) do
- D) enum

[Q40 Answered.](#)

Q41.

What is the output of this code?

```
class TestClass{  
    constructor() {  
        this.myInt = 5;  
    }  
  
    mainMethod(){  
        const test = new TestClass();  
        test.myInt = 6;  
        console.log(test.incrementIntAgain(test.incrementInt(test.myInt)));  
    }  
  
    incrementInt (theirInt) {  
        return theirInt++;  
    }  
  
    incrementIntAgain (theirInt) {  
        return ++theirInt;  
    }  
}
```

A) 5  
B) 6  
C) 7  
D) 8

[Q41 Answered.](#)

Q42.

What is the output of this code?

```
class TestClass{  
    constructor() {  
        this.myInt = 5;  
    }  
  
    mainMethod(){  
        const test = new TestClass();  
        test.myInt = 6;  
        console.log(test.incrementIntAgain(test.incrementInt(test.myInt)));  
    }  
  
    incrementInt (theirInt) {  
        return theirInt++;  
    }  
  
    incrementIntAgain (theirInt) {  
        return ++theirInt;  
    }  
}
```

A) 5  
B) 6  
C) 7  
D) 8

[Q42 Answered.](#)

Q43.

True or False: an instance level object can access a static variable?

- A) True
- B) False

[Q43 Answered.](#)

Q44.

What does the below code output to the console and why?

```
function Dog(name) {  
    this.name = name;  
}  
const buddy = new Dog("buddy");  
const rover = Object.create(buddy);  
  
console.log(Object.getPrototypeOf(rover));  
console.log(Object.getPrototypeOf(buddy));
```

[Q44 Answered.](#)

Q45.

What does the below code output to the console and why?

```
let x = (1, 2, 3);  
console.log(x);
```

- A) 1
- B) 2
- C) 3
- D) Compile time error

[Q45 Answered.](#)

Q46.

What are some differences in the RegExp literal notation and constructor?  
Select all that apply.

- A) The literal notation's parameters are enclosed between slashes and have no quotation marks
- B) The literal notation is evaluated at compile time
- C) The constructor function does not use slashes but has quotes
- D) The constructor is evaluated at runtime

[Q46 Answered.](#)

Q47.

What is the order of printing?

```
function resolveQuickly(x) {  
    return new Promise(resolve => {  
        setTimeout(() => {  
            console.log(x)  
        }, 1);  
    });  
}  
  
async function asyncTest() {  
    var x = resolveQuickly(10);  
    console.log(5);  
}  
  
asyncTest();
```

- A) 10, 5
- B) 5, 10
- C) Nothing prints.

[Q47 Answered.](#)

Q48.

Where did the name JavaScript originate?

[Q48 Answered.](#)

Q49.

Which of the following is NOT an advantage promises have over old-style callbacks?

- A) execution speed
- B) .then() chaining
- C) promise callbacks are always called in the order they are placed in the event queue
- D) better error handling
- E) promises avoid inversion of control

[Q49 Answered.](#)

Q50.

True or False? A do-while loop always executes at least once.

- A) True
- B) False

[Q50 Answered.](#)

Q51.

Show an example of setTimeout being called recursively to mimic setInterval.

[Q51 Answered.](#)

Q52.

What data types does JSON support? Select all appropriate answers.

- A) string
- B) number
- C) object
- D) null
- E) array
- F) boolean

[Q52 Answered.](#)

Q53.

What are two syntaxes for exporting theFunction as default?

[Q53 Answered.](#)

Q54.

What is the syntax difference in function declarations and function expressions?

[Q54 Answered.](#)

Q55.

True or False? *import* can be called as a function.

- A) True
- B) False

[Q55 Answered.](#)

Q56.

What does the following code print to the console and why?

```
console.log(x)
x = 5;
if(x){
    var x = 10;
    console.log(x);
}
```

- A) 5, 10
- B) 10, 10
- C) 5, 5

[Q56 Answered.](#)

Q57.

What is the purpose of the Array reduce()?

- A) reduce the array to callbacks
- B) reduce the array to a composite function
- C) reduce the array to a single value.

[Q57 Answered.](#)

Q58.

What is memoization?

- A) increasing a function's performance by caching its previously computed results
- B) something unavailable in JavaScript
- C) making a composition of object properties for future object use

[Q58 Answered.](#)

Q59.

Create an example of an abstract class in JavaScript.

[Q59 Answered.](#)

Q60.

Give an example of using String.prototype.match to run a string through a regex?

[Q60 Answered.](#)

Q61.

What will the code below output to the console and why?

```
for(let i = 6; i > 0; i-=2){  
    if(i%2){  
        continue;  
    }  
    console.log(i)  
}
```

- A) nothing
- B) 4, 2, 0
- C) 6, 4, 2
- D) 4, 2

[Q61 Answered.](#)

Q62.

True or False? Duplicated property names on an object are considered a syntax error.

- A) True
- B) False

[Q62 Answered.](#)

Q63.

True or False? The following is incorrect syntax.

```
function foo(arg) {  
    let arg = 1;  
}
```

- A) True
- B) False

[Q63 Answered.](#)

Q64.

What is the name of the array-like object that contains the values of all objects passed to the function?

- A) arguments
- B) args
- C) params
- D) parameters

[Q64 Answered.](#)

Q65.

What is the third parameter, *true* , used for in the below code?

```
element.addEventListener("click", e => console.log(element.eventPhase),  
true);
```

- A) enabling catching an event on capture phase
- B) enabling catching an event on bubbling phase
- C) it is not used in the addEventListener function

[Q65 Answered.](#)

Q66.

When considering coercion, what is a wrapper object?

[Q66 Answered.](#)

Q67.

What does  $3 > 2 == 1$  return?

- A) true
- B) false
- C) 0

[Q67 Answered.](#)

Q68.

What is a truthy value in javascript?

[Q68 Answered.](#)

Q69.

What is the name of the below syntax?

```
(function() {  
    console.log('IIFE');  
})();
```

[Q69 Answered.](#)

Q70.

What is the scope of arrow functions?

- A) global
- B) the scope in which they were defined
- C) block level

[Q70 Answered.](#)

Q71.

What will the code below output to the console and why?

```
let a = 8  
a >>>= 5;  
console.log(a);
```

- A) -1
- B) - 1167541818
- C) 0

[Q71 Answered.](#)

Q72.

True or False? The shift and pop methods on arrays return a new array without changing the value of the original array.

- A) True
- B) False

[Q72 Answered.](#)

Q73.

True or False? Objects assigned to const immutable.

- A) True
- B) False

[Q73 Answered.](#)

Q74.

What is the difference between imperative programming and declarative programming?

[Q74 Answered.](#)

Q75.

What is the difference between a statement and an expression?

- A) statements perform an action, expressions evaluate to a value
- B) statements evaluate to a value, expressions perform an action
- C) Neither of the above is correct

[Q75 Answered.](#)

## Test 4 Questions

Q1.

True or False? Evaluate the code below. bark() is now a function on the Dog class.

```
class Dog {  
    noise: string;  
    legs: number;  
}
```

```
Dog.bark = () => {  
    console.log('woof');  
}
```

```
let dog = new Dog();  
dog.bark();
```

- A) True
- B) False

[Q1 Answered.](#)

## Q2.

True or False? Evaluate the code below. bark() is now a function on the Puppy class.

```
class Dog {  
    noise: string;  
    legs: number;  
}  
  
Dog.prototype.bark = () => {  
    console.log('woof');  
}  
  
class Puppy extends Dog {  
    pottyTrained: boolean;  
}  
  
let dog = new Puppy();  
dog.bark();
```

- A) True
- B) False

[Q2 Answered.](#)

Q3.

Evaluate the following code. What is wrong with the getDOB function?  
TypeScript is enabled for this question.

```
const getDOB = (firstName: string, lastName: string) => {  
    return `${firstName} ${lastName}`;  
}
```

- A) getDOB has incorrect syntax in the string interpolation
- B) The typing of the parameters is wrong
- C) It returns undefined
- D) It is poorly named

[Q3 Answered.](#)

Q4.

Which of the following is the controller of custom elements?

- A) CustomHTML object
- B) HTMLElementRegistry object
- C) CustomController object
- D) CustomElementRegistry object

[Q4 Answered.](#)

Q5.

What class should a custom element extend?

- A) DOMElement
- B) HTMLElement
- C) CustomElement

[Q5 Answered.](#)

Q6.

What will the code below output to the console and why?

```
let x = 2.2 + 4.4 === 6.6;  
console.log(x);
```

- A) undefined
- B) True
- C) False
- D) Does not compile
- E) 6.6

[Q6 Answered.](#)

Q7.

What will the pre-es6 code below output to the console and why?  
TypeScript is NOT enabled.

```
function SomeClass() {  
    this.publicVar = 10;  
}  
SomeClass.staticVar = 10;  
  
const myClass = new SomeClass();  
myClass.publicVar = 12;  
myClass.staticVar = 15;  
  
const myClass2 = new SomeClass();  
console.log(myClass2.publicVar);  
console.log(myClass2.staticVar);  
console.log(SomeClass.staticVar);
```

- A) 10, undefined, 12
- B) 10, 12, 12
- C) 10, 15, 15
- D) 12, undefined, 10
- E) 10, undefined, 10

[Q7 Answered.](#)

Q8.

Write a function that prints the distinct vowels in the below string using one line of code.

```
const str = "abcdeeef";
```

[Q8 Answered.](#)

Q9.

What will the code below output to the console and why?

```
console.log(Math.floor(2.1) === new Float32Array(2.1));
```

- A) True
- B) False
- C) Compile Time Error

[Q9 Answered.](#)

Q10.

What will the code below output to the console and why?

```
console.log(Math.sin(Math.PI) === 0);
```

- A) True
- B) False

[Q10 Answered.](#)

Q11.

What will the code below output to the console and why?

```
z = 10;  
console.log(z);
```

- A) 10
- B) Compile time error
- C) z

[Q11 Answered.](#)

Q12.

What will the code below output to the console and why?

```
let max;  
const numberArray = [3, 5, 2];  
numberArray.forEach((element) => {  
    if (max < element) {  
        max = element;  
    }  
});  
console.log(max);
```

- A) 3
- B) 5
- C) 2
- D) undefined

[Q12 Answered.](#)

Q13.

Which bitwise operator inverts the bits of a 32 bit number?

- A) >>
- B) |
- C) &
- D) ~

[Q13 Answered.](#)

Q14.

What operator can remove properties from an object?

- A) remove
- B) delete
- C) return
- D) configure

[Q14 Answered.](#)

Q15.

What is the proper syntax for accessing the getter below?

```
const obj = {  
    count: [1, 2, 3],  
    get last() {  
        return this.count.pop();  
    }  
};
```

- A) obj.get(last)
- B) obj.getLast
- C) obj.last

[Q15 Answered.](#)

Q16.

When might you use new.target?

[Q16 Answered.](#)

Q17.

What method would you use to convert a nodelist to an array?

- A) Array.to()
- B) Array.from()
- C) NodeList.to()
- D) NodeList.from()

[Q17 Answered.](#)

Q18.

True or False? There are the two kinds of NodeList: static and dynamic.

- A) True
- B) False

[Q18 Answered.](#)

Q19.

Which object are Atomics not used with?

- A) Buffer
- B) SharedArrayBuffer
- C) ArrayBuffer

[Q19 Answered.](#)

Q20.

Which JavaScript Error type is commonly used as a base for user-defined error objects?

- A) ReferenceError
- B) SyntaxError
- C) TypeError
- D) Error

[Q20 Answered.](#)

Q21.

What property guarantees access to the global object regardless of environment?

- A) self
- B) globalThis
- C) this
- D) window

[Q21 Answered.](#)

Q22.

Which of the following is NOT true about undefined?

- A) It is a property of the global object
- B) It is a variable in global scope
- C) The initial value of undefined is the primitive value undefined
- D) It has a corresponding object named Undefined

[Q22 Answered.](#)

Q23.

True or false? Optional chaining can be stacked like below?

```
let stateStats = state.area?.land?.sqmiles;
```

- A) True
- B) False

[Q23 Answered.](#)

Q24.

What will the below code output to the console and why?

```
const a;  
a.test ??= 25;  
console.log(a.test);
```

- A) undefined
- B) 25
- C) Compile time error

[Q24 Answered.](#)

Q25.

What is the technical name of parenthesis () in JavaScript?

- A) Grouping Operator
- B) Parenthesis Operator
- C) Binding Operator

[Q25 Answered.](#)

Q26.

What does the below code output to the console and why?

```
let str = new String('aaa');
str.concat('bbb').concat('ccc');
console.log(str);
```

- A) aaa
- B) aaabbbccc
- C) ccc
- D) bbbccc

[Q26 Answered.](#)

Q27.

What is the Queue in JavaScript Runtime?

[Q27 Answered.](#)

Q28.

True or False? Vanilla JavaScript (no TypeScript) has abstract classes, as of ES2015.

- A) True
- B) False

[Q28 Answered.](#)

Q29.

What will the code below output to the console and why?

```
const b = {  
    a: 1,  
    b: 2,  
    3: 3  
}  
  
for(key in b){  
    console.log(b[key]);  
}
```

- A) 1 2 3
- B) 1 2
- C) Uncaught SyntaxError: Unexpected number

[Q29 Answered.](#)

Q30.

What is the syntax for initializing a Map object with multiple key/value pairs?

[Q30 Answered.](#)

Q31.

True or false? An interface can extend another interface. Assume TypeScript is enabled.

- A) True
- B) False

[Q31 Answered.](#)

Q32.

True or false? Assume TypeScript is enabled. An object inherits all the fields and methods of its parent class.

- A) True
- B) False

[Q32 Answered.](#)

Q33.

Which of the following are valid modifiers of TypeScript interface methods? Select all correct answers.

- A) public
- B) protected
- C) private
- D) abstract
- E) None of the above.

[Q33 Answered.](#)

Q34.

True or False? The static keyword is not a valid modifier in JavaScript. Rather, a typing superscript like TypeScript is required for *static* to work.

- A) True
- B) False

[Q34 Answered.](#)

Q35.

What one-line code block will print the following characters:



[Q35 Answered.](#)

Q36.

What, if anything, is in position 3 on the array?

```
const arr = [1, 2];  
arr.length = 3;
```

- A) It is empty
- B) It is not actually created
- C) null

[Q36 Answered.](#)

Q37.

Write a one line example of how to reverse the array.reduce function.

[Q37 Answered.](#)

Q38.

What are some limitations of arrow functions? Select all correct options.

- A) Arrow functions should not be used as methods
- B) Arrow functions do not have arguments, super, or new.target keywords
- C) Arrow functions cannot use yield keyword in it's body
- D) Arrow functions cannot be used as constructors

[Q38 Answered.](#)

Q39.

What is the difference in the default scope of traditional functions and arrow functions?

[Q39 Answered.](#)

Q40.

Which of the following are generally true regarding access modifiers?

- A) Fields should have the most narrow access modifier possible
- B) Methods should have the widest access modifier possible
- C) Avoid using public fields unless the field is a constant

[Q40 Answered.](#)

Q41.

What kind of error will be thrown if you forget a parenthesis in your for loop?

- A) SyntaxError
- B) TypeError
- C) ReferenceError

[Q41 Answered.](#)

Q42.

True or False? The below code is an example of overloading.

```
console.log({a: 1}.toString = 'Test');
```

- A) True
- B) False

[Q42 Answered.](#)

Q43.

True or False? JavaScript supports static variables on classes.

- A) True
- B) False

[Q43 Answered.](#)

**Q44.**

Give an example of using new ES6 syntax for creating a parent/child relationship.

[Q44 Answered.](#)

Q45.

True or False? A try statement must always be followed by at least one catch statement.

- A) True
- B) False

[Q45 Answered.](#)

Q46.

True or False? Use RegExp literal notation when the RegExp will be changing and use the RegExp constructor function when the RegExp will remain constant.

- A) True
- B) False

[Q46 Answered.](#)

Q47.

True or False? Promises are asynchronous in JavaScript.

- A) True
- B) False

[Q47 Answered.](#)

Q48.

What is the purpose of web workers?

- A) To execute expensive tasks on a separate thread than the main thread
- B) To offload events to different objects
- C) To speed up prototypical inheritance

[Q48 Answered.](#)

Q49.

What type of object does Promise.resolve() return?

[Q49 Answered.](#)

Q50.

What is the difference between setTimeout() and setInterval()? Select two options.

- A) setTimeout executes once after a specified time has elapsed
- B) setInterval executes repeatedly with a given delay between executions
- C) setTimeout executes repeatedly with a given delay between executions
- D) setInterval executes once after a specified time has elapsed

[Q50 Answered.](#)

Q51.

True or False? setInterval interval time includes the time it took to execute the code.

- A) True
- B) False

[Q51 Answered.](#)

Q52.

True or False? The below code compiles.

```
const array = [1, 2, 3];
```

```
for (let i = 0; i < array1.length; array1[i++] = String.fromCharCode(96 +  
i));
```

- A) True
- B) False

[Q52 Answered.](#)

Q53.

What is the difference in importing a default export vs a named, non-default export?

- A) curly braces
- B) the name is never included in the default
- C) nothing

[Q53 Answered.](#)

Q54.

True or False? Function declarations are hoisted, function expressions are not hoisted.

- A) True
- B) False

[Q54 Answered.](#)

Q55.

What statement will show the url from which a module was imported?

- A) import.path
- B) path.show
- C) path.directory
- D) import.meta

[Q55 Answered.](#)

Q56.

What is the difference in console.log, console.error, and console.warn?

[Q56 Answered.](#)

Q57.

What is the significance of seeing prototype in the following documentation: `Array.prototype.reduce()` as opposed to: `Array.from()`? Select all correct answers.

- A) prototype indicates reduce is an instance level method
- B) the lack of prototype indicates from is a class level method
- C) prototype indicates reduce is inherited from Object

[Q57 Answered.](#)

Q58.

How does the Maximum Call Stack Exceeded error get triggered?

[Q58 Answered.](#)

Q59.

Which of the following is not a method available on the Web API window object for resizing?

- A) resizeFrom()
- B) resizeBy()
- C) resizeTo()
- D) sizeToContent()

[Q59 Answered.](#)

Q60.

What two types can String.prototype.replace() take as the first parameter?  
Select all correct options.

- A) String
- B) regex
- C) Object

[Q60 Answered.](#)

Q61.

What will the code below output to the console and why?

```
for(let i = 6; i < 10; i*1.5){  
    if(i%2){  
        continue;  
    }  
    console.log(i)  
}
```

- A) 6
- B) 6, 9
- C) It enters into an infinite loop
- D) Lint error

[Q61 Answered.](#)

Q62.

Which of the following is a difference between Object.assign() and the spread syntax?

- A) spread syntax calls setters, Object.assign() does not
- B) Object.assign calls setters, spread syntax does not
- C) They actually behave identically

[Q62 Answered.](#)

Q63.

Give an example of a one line check to see if a property exists on object *dog* .

```
const dog = { sound: 'woof', legs: 4 }
```

[Q63 Answered.](#)

Q64.

Which of the following is not a way to convert the arguments function object to an array?

- A) arguments.assign(arguments)
- B) Array.prototype.slice.call(arguments);
- C) Array.from(arguments);
- D) [...arguments]

[Q64 Answered.](#)

Q65.

What is the target phase of event handling?

[Q65 Answered.](#)

Q66.

What does the double bang (!! ) do?

- A) nothing
- B) returns the truthy value of an object
- C) throws an error

[Q66 Answered.](#)

Q67.

what does  $1 < 2 < 3$  return?

- A) true
- B) false

[Q67 Answered.](#)

Q68.

How does JavaScript evaluate the truthy-ness or falsey-ness of a value or object in a boolean context?

- A) type coercion
- B) it's a property on each object
- C) some values never have truthy or falsey values

[Q68 Answered.](#)

Q69.

What is the purpose of using an IIFE? Select all possible answers.

- A) setting a variable value
- B) initializing values on an object
- C) use in a closure for initializing a value
- D) running a one-time function

[Q69 Answered.](#)

Q70.

What is an example of lexical scope? Select all possible answers.

- A) scope being defined dynamically based on variable input
- B) let self = this; //in a parent function, then self is available to an inner function
- C) the scope of an inner function contains the scope of the parent function

[Q70 Answered.](#)

Q71.

What will the code below output to the console and why?

```
let a = 3;  
a <= 1;  
console.log(a);
```

- A) 3
- B) 4
- C) 6
- D) 8

[Q71 Answered.](#)

Q72.

What will the code below output to the console and why?

```
badCode = () => {
let arr = [1,2,3,4];
const inner = () => {
return arr.pop(2)
}
    return inner;
}

console.log(badCode()());
```

- A) linting error
- B) 3, 4
- C) 34
- D) 4

[Q72 Answered.](#)

Q73.

What will the code below output to the console and why?

```
const obj = { a: 1 };
let obj2 = obj;
obj2.a = 2;
console.log(obj.a);
```

- A) 1
- B) 2
- C) undefined

[Q73 Answered.](#)

Q74.

What is a side effect in programming?

[Q74 Answered.](#)

Q75.

True or False? `Array.prototype.filter()` returns an array of all items that did NOT pass the test implemented by the provided function.

- A) True
- B) False

[Q75 Answered.](#)

# Test 1 Answers

**Q1 Answered.**

Which of the following prints first?

```
setTimeout(() => {  
    console.log('setTimeout wins');  
},0);  
  
queueMicrotask(() => {  
    console.log('queueMicrotask wins');  
});
```

- A) setTimeout wins
- B) queueMicrotask wins
- C) Compile time error

**Q1 Unanswered.**

**Q2 Answered.**

What will the code below output to the console and why?

```
let x = 10;  
  
const byValue = (y) => {  
    y = 20;  
}  
  
byValue(x);  
console.log(x);
```

- A) 20
- B) 10
- C) Compile time error

The answer is B. Unlike objects, when primitives are passed to a function only their value is passed, not the reference to the memory location. Therefore, changes to the parameter only affect the parameter in the function's scope.

**Q2 Unanswered.**

**Q3 Answered.**

Consider the following code. What will it log to the console and why?

```
let x = 5;  
let x = 10;  
  
console.log(x);
```

- A) 5
- B) 10
- C) It will throw a compiler error.

The answer is C. However, the below works because the scope of the declarations is different (It's not clean code, however).

```
let x = 5;  
  
if(x){  
let x = 10;  
console.log(x);  
}
```

[Q3 Unanswered.](#)

#### Q4 Answered.

Consider the following code. What does each line output?

```
console.log(new String("yes") === new String("yes")); // Line 1  
console.log("yes" === "yes"); // Line 2
```

- A) Line 1 prints true, Line 2 prints true
- B) Line 1 prints false, Line 2 prints true
- C) Line 1 prints true, Line 2 prints false
- D) Line 1 prints false, Line 2 prints false

The answer is B. Line 1 is comparing two objects, seeing that they are each unique (literally not the exact same object) and thus is false. Even a == loose check will result in false for Line 1. Line 2 passes because it is comparing two string primitives, not string objects.

#### Q4 Unanswered.

**Q5 Answered.**

What will the code below output to the console and why?

```
const x = new String("tricky");
const y = x;

console.log(x === y);
```

- A) True
- B) False

The answer is A. Unlike the previous question, where two unique objects were created, this time there is one unique object created, with two constants (x and y) pointing to this unique object in memory.

[Q5 Unanswered.](#)

**Q6 Answered.**

Are Arrays objects or primitives in JavaScript?

- A) Objects
- B) Primitives

The answer is A. Arrays are a special kind of object in JavaScript. They have properties that no other objects have, which are inherited from `Array.prototype`.

**Q6 Unanswered.**

**Q7 Answered.**

What is the return type of the async function below?

```
async function goodbye() {  
    return 'goodbye';  
}
```

- A) string
- B) Promise
- C) Async

The answer is B. async functions return Promises. To actually consume the value returned when the promise fulfills, you need a .then statement such as `goodbye.then((value) => console.log(value));`

[Q7 Unanswered.](#)

## Q8 Answered.

True or False? The await keyword blocks all JavaScript code execution in an app until the awaited Promise is returned.

- A) True
- B) False

The answer is B. The await keyword only blocks execution of code inside the specific async function containing the await keyword. The caller of the await's async function can continue with execution while awaiting the async functions resolution. This is called splitting execution flow.

## Q8 Unanswered.

## Q9 Answered.

Recall that functions are objects in JavaScript. What does the following print?

```
function myFunc() {}  
console.log(typeof myFunc);
```

- A) object
- B) function
- C) true
- D) It throws an error

The answer is B. Despite the fact that functions are indeed objects in JavaScript, `typeof myFunc` prints “function”.

## Q9 Unanswered.

**Q10 Answered.**

True or False? The following is valid syntax for printing ‘goodbye’.

```
let goodbye = async () => { return "Goodbye" };  
goodbye().then(console.log); // Line 2
```

- A) True
- B) False

The answer is A. Line 2 is a valid shorthand for passing the returned value of an async function to a callback.

[\*\*Q10 Unanswered.\*\*](#)

**Q11 Answered.**

What of the following is not a difference between `typeof` and `instanceof`?

- A) `typeof` returns a type and `instanceof` returns a boolean
- B) `instanceof` requires TypeScript and `typeof` does not
- C) `typeof` takes the variable name on the right while `instanceof` takes a value on the left and right

The answer is B. Neither `typeof` nor `instanceof` require TypeScript, both are native to JavaScript.

[\*\*Q11 Unanswered.\*\*](#)

## Q12 Answered.

Which of the following resolves when all promises are fulfilled, or rejects as soon as one promise is rejected?

- A) Promise.race()
- B) Promise.any()
- C) Promise.all()
- D) Promise.resolve()

The answer is C. Promise.all() is useful if you want the execution to wait until all Promises resolve (or any fail). Also, all of the above are valid Promise functions.

## Q12 Unanswered.

**Q13 Answered.**

What will the code below output to the console and why?

```
let x = 10;  
  
function myFunc(y){  
    y = 12;  
}  
myFunc(x);  
console.log(x);
```

- A) 10
- B) 12
- C) undefined

The answer is A. Primitives are passed by value, not by reference, so “changing them” in a function doesn’t affect the original variable.

**Q13 Unanswered.**

**Q14 Answered.**

What does the below code output to the console and why?

```
console.log(11 & 3);
```

- A) 11
- B) 1
- C) 2
- D) 3

The answer is D. & is a bitwise operator (very different from &&). 11 & 3 is the same as (in binary) 1011 & 0011. Only bits that are both 1 are kept at 1, so the output is 0011, which is the binary representation of 3.

[Q14 Unanswered.](#)

**Q15 Answered.**

What is the value of Object.[[Prototype]]?

- A) Object
- B) null
- C) {}

The answer is B. While Object is at the top of the prototype chain, when the browser is seeking the value of an accessed property, it will traverse the prototype chain until the value is found or until there are no more prototypes to traverse. While null is the value of Object.[[Prototype]], *undefined* is the returned value.

See here for more reading: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Inheritance\\_and\\_the\\_prototype\\_chain](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Inheritance_and_the_prototype_chain)

**Q15 Unanswered.**

**Q16 Answered.**

What does the nullish coalescing operator do?

It returns the right-hand side operand when the left hand operand is null or undefined. Otherwise, the left hand operand is returned. The syntax is ??.

**Q16 Unanswered.**

**Q17 Answered.**

What operator will return the remainder when one operand is divided by another operand?

- A) \$
- B) #
- C) %
- D) None of the above

The answer is C.

**Q17 Unanswered.**

**Q18 Answered.**

True or False? `getElementsByName` is a JavaScript function.

- A) True
- B) False

The answer is B. `getElementsByName` is actually Web API function. It is available like a normal JS function, but is accessible regardless of language being used.

**Q18 Unanswered.**

**Q19 Answered.**

Give an example of when to use event delegation in JavaScript.

One example: Use event delegation if an element that you want to listen for events on may not exist at page load. Instead, put event handler on a parent element and look at event.target.

Modern frameworks and libraries such as React make this less necessary.

**Q19 Unanswered.**

**Q20 Answered.**

Does the code below evaluate to true or false?

`BigInt(1000000000000000000) === 1000000000000000000`

- A) true
- B) false

The answer is B. However, `BigInt(1000000000000000000) == 1000000000000000000` is true

[Q20 Unanswered.](#)

**Q21 Answered.**

Which of the following is not a built-in JS error type?

- A) Error
- B) EvalError
- C) SyntaxError
- D) TypeError
- E) UndefinedError

The answer is E.

**Q21 Unanswered.**

**Q22 Answered.**

How are numbers stored in JavaScript? Select all that apply.

- A) Double precision
- B) 64-bit
- C) Floating point numbers

A, B, and C are all correct.

[\*\*Q22 Unanswered.\*\*](#)

**Q23 Answered.**

If a function has no return value specified, what value does it return?

- A) null
- B) undefined
- C) nothing

The answer is B.

[\*\*Q23 Unanswered.\*\*](#)

**Q24 Answered.**

What is the syntax for calling a method which may or may not exist on an object?

```
let value = myAPI.testMethod?.();
```

[\*\*Q24 Unanswered.\*\*](#)

**Q25 Answered.**

Which of the following is not a valid Promise method?

- A) Promise.some()
- B) Promise.allSettled()
- C) Promise.reject()
- D) Promise.race()

The answer is A.

[\*\*Q25 Unanswered.\*\*](#)

**Q26 Answered.**

How can you dynamically access an object property based on a variable value?

Answer with example variable:

```
let key = 'someValue';
myObject[key];
```

[\*\*Q26 Unanswered.\*\*](#)

**Q27 Answered.**

True or False? A string can be modified after it is created.

- A) True
- B) False

The answer is B. Strings are immutable in JavaScript. However, the variable pointing to a string can of course be reassigned to another string.

[\*\*Q27 Unanswered.\*\*](#)

**Q28 Answered.**

True or False? JavaScript is synchronous.

- A) True
- B) False

The answer is A. The Queue in JavaScript's Runtime is synchronous, thus the language is synchronous.

**Q28 Unanswered.**

## **Q29 Answered.**

True or False? A nested catch in a promise chain can catch errors thrown higher up in the promise chain?

- A) True
- B) False

The answer is B. From MDN docs: “Nesting is a control structure to limit the scope of catch statements. Specifically, a nested catch only catches failures in its scope and below, not errors higher up in the chain outside the nested scope”.

See here for more reading: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Using\\_promises](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Using_promises)

## **Q29 Unanswered.**

**Q30 Answered.**

True or False? The Map object can have keys that are of types other than strings, such as number.

- A) True
- B) False

The answer is A.

[\*\*Q30 Unanswered.\*\*](#)

Q31 Answered.

What will the code below output to the console and why?

```
const map = new Map();
map.set({},1);
console.log(map.get({}));
```

- A) undefined
- B) 1
- C) Compile time error

The answer is A. `map.get({})` is valid syntax, but the Object in the set and in the get are two different empty objects in memory. Therefore, the getter doesn't return a value. If the object used for `map.set` had first been assigned to a const, then that const was used in the get, the appropriate value would have been returned.

[Q31 Unanswered.](#)

**Q32 Answered.**

What will the code below output to the console and why?

```
let map1 = new Map([
  ['a', 1],
  ['b', 1],
  ['c', 1],
]);
```

```
let map2 = new Map([
  ['a', 2],
  ['b', 2]
]);
```

```
let spreadMap = new Map([...map1, ...map2]);
```

- A) Map(3) {"a" => 1, "b" => 1, "c" => 1}
- B) Map(3) {"a" => 2, "b" => 2, "c" => 1}
- C) Map(5) {"a" => 1, "b" => 1, "c" => 1, "a" => 2, "b" => 2}

The answer is B. Any identical keys in the second map will override keys in the first map.

[Q32 Unanswered.](#)

**Q33 Answered.**

True or False? Bracket notation can be chained just like dot notation?

- A) True
- B) False

The answer is A. `obj.prop1.prop2` and `obj['prop1']['prop2']` are equivalent.

[\*\*Q33 Unanswered.\*\*](#)

Q34 Answered.

What will the code below output to the console and why?

```
class Person {  
    name: string;  
    DOB: string;  
}  
console.log(typeof Person);
```

- A) “class”
- B) “function”
- C) “object”

The answer is B. This is a reminder that classes are simply syntactic sugar that make JavaScript feel more object oriented.

Q34 Unanswered.

### Q35 Answered.

Examine the code below. Diagnose an error that will occur. Assume that handleResponse and handleError are properly defined.

```
promise.then(  
  function(response) {  
    this.handleResponse();  
  },  
  function(error) {  
    this.handleError();  
  }  
)
```

The answer is *this* is not scoped properly. Instead, the code needs a way to have the proper scope on *this* , such as using *bind* or *var self = this;*

### Q35 Unanswered.

**Q36 Answered.**

What types of properties show up in for...in loops?

- A) Executable Properties
- B) Enumerable Properties
- C) Iterable Properties

The answer is B.

[Q36 Unanswered.](#)

**Q37 Answered.**

True or False? .bind(this) immediately calls the bound function.

- A) True
- B) False

The answer is B.

[\*\*Q37 Unanswered.\*\*](#)

**Q38 Answered.**

What does the code print?

```
function getName() {  
    var name = "Jon";  
    function printName() {  
        console.log(name);  
    }  
    printName();  
}  
getName();
```

- A) Jon
- B) undefined
- C) Compile time error

The answer is A. Inner functions have access to variables declared in the outer scope.

[\*\*Q38 Unanswered.\*\*](#)

**Q39 Answered.**

True or False? The await keyword is required with async functions?

- A) True
- B) False

The answer is B. An async function returns a promise and has many uses. The await keyword adds extra functionality and overhead that may not be needed. See [here](#) for an example.

**Q39 Unanswered.**

**Q40 Answered.**

What are the three ways for a function to refer to itself for recursion? Select three answers.

- A) Use self.run
- B) The function's name
- C) An in-scope variable that refers to the function
- D) Use argumentscallee

The answers are B, C, D.

[\*\*Q40 Unanswered.\*\*](#)

[Q41 Answered.](#)

What will the code below output to the console and why?

```
const test = (incrementer) => {
  if (incrementer < 0){
    return;
  }
  console.log('first: ' + incrementer);
  test(incrementer - 1);
  console.log('second: ' + incrementer)
}
test(2);
```

It outputs the following:

```
"first: 2"
"first: 1"
"first: 0"
"second: 0"
"second: 1"
"second: 2"
```

This function is recursive. It is interesting to see how it stacks function calls and the order the code executes in.

[Q41 Unanswered.](#)

## **Q42 Answered.**

True or False? The difference in Promise.any() and Promise.race() is that Promise.race() returns the first settled Promise (fulfilled *or* rejected), while Promise.any() continues to resolve until a promise is fulfilled, and only rejects if all promises are rejected.

- A) True
- B) False

The answer is A. Promise.any() and Promise.race() are very similar.

## **Q42 Unanswered.**

**Q43 Answered.**

True or False? JavaScript does not support overloading?

- A) True
- B) False

The answer is A. While JavaScript does not natively support overloading, TypeScript does. Also, overloading can be simulated in JavaScript by having a function return a different output when not all possible parameters are passed to it.

[\*\*Q43 Unanswered.\*\*](#)

**Q44 Answered.**

Without using class syntax, create a child object from a parent object.

One example solution:

```
function Dog(name) {  
    this.name = name;  
}  
const buddy = new Dog("buddy");  
const rover = Object.create(buddy);
```

If you examine `Object.getPrototypeOf(rover)`; you will see that `buddy` is `rover`'s parent prototype.

[\*\*Q44 Unanswered.\*\*](#)

**Q45 Answered.**

What does a return statement do in a forEach loop on an Array?

- A) Nothing
- B) Returns the specified value
- C) Ends that loop

The answer is A. If you need to return a value from a loop, don't use the forEach loop.

[\*\*Q45 Unanswered.\*\*](#)

## Q46 Answered.

True or False? There are a maximum of three parameters in a for loop.

- A) True
- B) False

The answer is B. An example:

```
for (var i = 0, j = 0; i <= 10; i++, j++){...}
```

This is valid for loop syntax.

## Q46 Unanswered.

**Q47 Answered.**

True or False? RegExp do not have any properties.

- A) True
- B) False

The answer is B. RegExps have many properties, such as .flags and .global.

**Q47 Unanswered.**

**Q48 Answered.**

```
function resolveQuickly(x) {  
    return new Promise(resolve => {  
        console.log(x)  
    });  
}
```

```
async function asyncTest() {  
    var x = resolveQuickly(10);  
    console.log(5);  
}
```

```
asyncTest();
```

- A) 10, 5
- B) 5, 10
- C) It does not print

The answer is A. This function has nothing asynchronous in the Promise, so the Promise resolves synchronously.

[Q48 Unanswered.](#)

**Q49 Answered.**

What is blocking?

Answer: When a slow or expensive process blocks the execution of remaining code until it is complete.

**Q49 Unanswered.**

**Q50 Answered.**

Which of the following is not a valid state for a Promise?

- A) Pending
- B) Fulfilled
- C) Rejected
- D) Completed

The answer is D.

**Q50 Unanswered.**

**Q51 Answered.**

Which function executes a specified code block before the browser next repaints the display?

- A) requestAnimationFrame()
- B) setTimeout()
- C) requestRepaintDelay()

The answer is A.

[\*\*Q51 Unanswered.\*\*](#)

## Q52 Answered.

Assume you want to achieve a refresh rate of 60fps on an animation. Write a setInterval function that triggers the below pseudocoded animate function at 60fps.

```
function animate() {...}
```

Answer:

```
setInterval(animate, Math.ceil(1000/60))
```

Read more about this example here: [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Timeouts\\_and\\_intervals](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Timeouts_and_intervals)

## Q52 Unanswered.

**Q53 Answered.**

What are the two types of exports in JavaScript?

- A) Abstract
- B) Whole
- C) Default
- D) Named

The answers are C, D.

**Q53 Unanswered.**

**Q54 Answered.**

What statement would you use to iterate over asynchronous objects?

- A) if async then
- B) for await...of
- C) async iterable
- D) do while async

The answer is B.

**Q54 Unanswered.**

[Q55 Answered.](#)

Why use an alias when importing a module?

Answer: Perhaps the module has a ver long name, or a name that doesn't express what the code does properly.

```
import {  
    longLongLongLongName as shortName  
}
```

[Q55 Unanswered.](#)

## Q56 Answered.

True or False? The below code results in the same output.

```
var i, j;  
  
loop1:  
for (i = 0; i < 5; i++) {  
    loop2:  
    for (j = 0; j < 5; j++) {  
        if (i === 1 && j === 1) {  
            continue loop1;  
        }  
        console.log(`i: ${i}, j: ${j}`)  
    }  
}  
  
var i, j  
  
loop1:  
for (i = 0; i < 5; i++) {  
    loop2:  
    for (j = 0; j < 5; j++) {  
        if (i === 1 && j === 1) {  
            break loop2;  
        }  
        console.log(`i: ${i}, j: ${j}`)  
    }  
}
```

A) True

B) False

The answer is A.

## Q56 Unanswered.

Q57 Answered.

True or False? The following are equivalent.

```
var a = 1;  
var z = a;
```

and

```
var a, z = a = 1;
```

- A) True
- B) False

The answer is A.

Q57 Unanswered.

**Q58 Answered.**

Use a reducer to find the min value of the below array.

```
const arr = [5, 10, 12, 4, 2];
```

A possible answer:

```
const reducedVal = arr.reduce((accumulator, element) => {  
    if (accumulator > element) {  
        return element;  
    }  
    return accumulator;  
});
```

[Q58 Unanswered.](#)

**Q59 Answered.**

What is a subroutine in JavaScript?

Answer: A subroutine is a function encountered in the main routine that is then saved to an object and stored for later use.

The execution scope (variables, params, etc) are stored with the subroutine.

**Q59 Unanswered.**

**Q60 Answered.**

True or False? You might use eventHandlers *cut* and *copy* to keep a user from copying content from the browser to the clipboard?

- A) True
- B) False

The answer is A. These event handlers are part of the Web API.

**[Q60 Unanswered.](#)**

Q61 Answered.

What will the code below output to the console and why?

```
let i = 10
for(; i< 12;){
    console.log(++i);
}
```

- A) lint error
- B) runtime error
- C) infinite loop
- D) 11, 12

The answer is D. A *for* loop doesn't require three statements, and the first can be empty.

Q61 Unanswered.

**Q62 Answered.**

What are the three ways to create a new object?

- A) new Object()
- B) Object.create()
- C) factory pattern
- D) literal notation

The answer is A, B, D. An example of literal notation: const obj = { a: 1}

**Q62 Unanswered.**

**Q63 Answered.**

What will the code below output to the console and why?

```
const a = {x: 1}
const b = {...a}
const c = {};
Object.assign(c, a);
a.x = 2;
console.log(a.x);
console.log(b.x);
console.log(c.x);
```

- A) 1, 1, 1
- B) 2, 2, 2
- C) 1, 2, 2
- D) 2, 1, 1
- E) 2, 1, 2
- F) 2, 2, 1

The answer is D.

[Q63 Unanswered.](#)

## **Q64 Answered.**

Implement the classic example of a closure that increments up each time it is called.

A potential answer:

```
var increment = (function () {  
    var incrementer = 0;  
    return function () {incrementer += 1; return incrementer}  
})();  
  
console.log(increment()); //outputs 1  
console.log(increment()); //outputs 2  
console.log(increment()); //outputs 3
```

## **Q64 Unanswered.**

**Q65 Answered.**

If myFunction is called as myFunction(5); what will be the return value?

```
function myFunction(x, y) {  
    const arr = Array.prototype.slice.call(arguments);  
    return arr.length;  
}
```

- A) 0
- B) 1
- C) 2

The answer is B.

[\*\*Q65 Unanswered.\*\*](#)

**Q66 Answered.**

Which of the below does Object.seal() do? Select all that apply.

- A) prevents new properties from being added
- B) marks all existing properties as non-configurable
- C) prevents values of existing properties from being changed

The answer is A, B. A similar but more rigorous function is Object.freeze()

[\*\*Q66 Unanswered.\*\*](#)

**Q67 Answered.**

Which of the below does Object.freeze() do? Select all that apply.

- A) prevents new properties from being added
- B) prevents existing properties from being removed
- C) prevents changing the enumerability of properties
- D) prevents changing the configurability of properties
- E) prevents changing the writability of properties
- F) prevents changing the values of properties
- G) prevents changing the prototype of the object

The answer is All of the Above.

**Q67 Unanswered.**

**Q68 Answered.**

What is the difference between event.target and event.currentTarget?

- A) event.currentTarget returns the element the triggered listener was attached to
- B) event.target returns the most granular element the event was triggered on
- C) event.target returns the element the triggered listener was attached to
- D) event.currentTarget returns the most granular element the event was triggered on

The answers are A, B. event.currentTarget changes as the event bubbles up, event.target stays the same.

**Q68 Unanswered.**

## **Q69 Answered.**

If you need to call an anonymous function later, show what syntax you would use for creating the anonymous function.

A possible answer:

```
let caller = function () {  
    console.log('call me maybe');  
};
```

The function is anonymous, but is stored to a variable for later reference.

## **Q69 Unanswered.**

**Q70 Answered.**

What is the default sort of the Array sort() method?

- A) largest to smallest by numeric value
- B) smallest to largest by numeric value
- C) largest to smallest by character value
- D) smallest to largest by character value

The answer is D. For example, 10 will be placed before 2 because the '1' will be evaluated against the '2'.

[\*\*Q70 Unanswered.\*\*](#)

**Q71 Answered.**

What will the code below output to the console and why?

```
let scope = "global";
getScope = () => {
  var scope = "local";
  const inner = () => {return scope;}
  return inner;
}
getScope()();
```

- A) global
- B) local
- C) undefined

The answer is B. The inner arrow function has the scope of the location it was defined.

[Q71 Unanswered.](#)

**Q72 Answered.**

What will the code below output to the console and why?

```
console.log(3 + '2');
```

- A) 32
- B) 5
- C) runtime error

The answer is A. 3 will be coerced into a string.

[\*\*Q72 Unanswered.\*\*](#)

**Q73 Answered.**

What is a race condition?

Answer: A race condition is when two threads or async processes must complete in the proper order and update some shared state, otherwise there will be a bug or unwanted outcome.

**Q73 Unanswered.**

**Q74 Answered.**

What is functional programming?

Answer: A coding paradigm utilizing declarative code and pure functions.

**Q74 Unanswered.**

**Q75 Answered.**

What is a higher order function?

- A) a function composed of multiple functions or interfaces
- B) a function that can take a function as an argument, or that returns another function
- C) a function that is a superclass for child objects

The answer is B.

**Q75 Unanswered.**

## Test 2 Answers

**Q1 Answered.**

Consider the following code:

```
class Person {  
    firstName: string;  
    lastName: string;  
}  
  
const myPerson = new Person();  
console.log(myPerson.a);
```

Explain how the prototype chain is used to determine that myPerson.a is undefined?

The browser checks myPerson.[[Prototype]] and does not see a property named *a* . Then the browser checks myPerson.[[Prototype]].[[Prototype]] (which is the Object prototype) and does not see a property named *a* . Then the browser checks myPerson.[[Prototype]].[[ Prototype]].[[ Prototype]], which is null, so the browser returns *undefined* .

[\*\*Q1 Unanswered.\*\*](#)

**Q2 Answered.**

What does the *class* keyword do in JavaScript?

- A) It conveys real meaning beyond what a typical object is
- B) It is syntactic sugar to make JavaScript more object oriented
- C) It creates an interface-like object that can be used for inheritance

The answer is B. Even when the class keyword is used, JavaScript still uses prototypical inheritance.

[Q2 Unanswered.](#)

**Q3 Answered.**

What will the code below output to the console and why?

```
let x = 10;  
if(x){  
    let x = 12;  
    console.log(x);  
}
```

- A) 10
- B) 12
- C) It will throw an error

The answer is B. The inner *x* has priority over the out *x* variable.

Keep in mind, this is not a great way to write code and is being used for an example.

[\*\*Q3 Unanswered.\*\*](#)

**Q4 Answered.**

True or False? Tasks from the queueMicrotask queue are executed in a Last In, First Out basis.

- A) True
- B) False

The answer is B. The tasks are in fact executed in a First In, First Out order.

**Q4 Unanswered.**

## **Q5 Answered.**

What would be a name for this function that follows “clean code” principles?

```
const pntDT = () => {  
  const today = new Date();  
  const time = today.getHours() + ":" + today.getMinutes();  
  console.log(time);  
};  
  
pntDT();
```

There are many possible function names that would comply better with clean code principles. For example, printDateAndTime() would be a better name because it leaves no doubt as to what the function does.

## **Q5 Unanswered.**

**Q6 Answered.**

Which of the following is not a step to create and register a new custom element?

- A) Create an HTML container in the DOM
- B) Create a javascript class that defines the behavior of the element
- C) Register the class and a DOMString representing the name of the element with the CustomElementRegistry object

The answer is A.

[\*\*Q6 Unanswered.\*\*](#)

**Q7 Answered.**

Briefly explain the Shadow DOM API.

The Shadow DOM API provides a way to attach a hidden separate DOM to an element not accessible through the usual JS DOM manipulation API. Shadow DOM provides for encapsulation of Web Components. Hidden DOM trees attach to elements in the regular DOM tree

[Q7 Unanswered.](#)

**Q8 Answered.**

What method attaches a shadow DOM tree to the specified element and returns a reference to its ShadowRoot?

- A) Element.shadowAdd()
- B) Element.setDOMShadow()
- C) Element.attachShadow()

The answer is C.

**Q8 Unanswered.**

**Q9 Answered.**

What value will arr[2][1] return?

```
const arr = [
    ["a", "b", "c"],
    ["d", "e", "f"],
    ["g", "h", "i"]
];
```

- A) d
- B) h
- C) b
- D) None of the above

The answer is B. First, arrays are zero-based in JavaScript. Therefore, arr[2][1] will access the 3<sup>rd</sup> element of the outer array, and the 2<sup>nd</sup> element of the inner array, returning a value of “h”.

[Q9 Unanswered.](#)

**Q10 Answered.**

What will the code below output to <app>?

```
document.getElementById("app").innerHTML = `  
<div>  
hewo  
    hewo2  
</div>`
```

- A) hewo hewo2
- B)
- hewo  
 hewo2
- C) hewo hewo2
- D) Does not compile

The answer is C.

**Q10 Unanswered.**

**Q11 Answered.**

What pattern does jQuery use to make a single instance of it's globally available jQuery object?

- A) OOP
- B) Factory
- C) Singleton
- D) None

The answer is C. One instance of the jQuery object is created globally.

[\*\*Q11 Unanswered.\*\*](#)

**Q12 Answered.**

What is the difference between window.localStorage and window.sessionStorage? Select the best answer.

- A) localStorage stores key/value pairs in the web browser, and localStorage deletes after the browser closes
- B) sessionStorage stores key/value pairs in the web browser, and sessionStorage deletes after the browser closes
- C) Both store key/value pairs in the web browser, but localStorage deletes after the browser closes
- D) Both store key/value pairs in the web browser, but sessionStorage deletes after the browser closes

The answer is D.

**Q12 Unanswered.**

**Q13 Answered.**

What will the code below output to the console and why?

```
let arr1 = [1, 2, 3];
arr1.push(4);
arr1.pop();
arr1.slice(1, 2);
arr1.splice(1, 2);
console.log(arr1);
```

- A) 1
- B) [1]
- C) [1,2]
- D) [2,3]
- E) Prints nothing.

The answer is B.

**Q13 Unanswered.**

**Q14 Answered.**

True or False: The ! operator returns a boolean. Select one option.

- A) True
- B) False

The answer is A. An example of a situation in which ! can be used is an if statement. if statements need a boolean returned in the evaluation, such as if( $x \neq y$ ).

**Q14 Unanswered.**

## **Q15 Answered.**

Evaluate the code below. If both div and button have a click listener, how can you stop a click on the button from bubbling up to the div?

```
<div>
  <button>Click Me!</button>
</div>
```

- A) Event.stopPropagation in the button clickHandler
- B) Event.stopPropagation in the div clickHandler
- C) Event.stopBubble in the button clickHandler
- D) Event.stopBubble in the div clickHandler

The answer is A.

## [Q15 Unanswered.](#)

**Q16 Answered.**

What will the code below output to the console and why?

```
let xyz = "";
xyz ||= "abc";
console.log(xyz);
```

- A) abc
- B) “
- C) Compile time error

If experimental syntax is enabled, the answer is A. Otherwise, there is a compile time error.

[Q16 Unanswered.](#)

## Q17 Answered.

What is the difference in the output between the two following functions?

```
void function voidExample() {  
    console.log("No Error!");  
}();
```

```
function noVoidExample() {  
    console.log("No Error!");  
}();
```

- A) The function noVoidExample executes and prints No Error, while voidExample throws an error.
- B) The function voidExample executes and prints No Error, while noVoidExample throws an error.
- C) Both throw an error

The answer is B. The void keyword allows for the execution of functions that may not otherwise execute.

## Q17 Unanswered.

**Q18 Answered.**

Which of the following is NOT a difference between NodeList and Array?

- A) NodeList is Web API, Array is JavaScript API
- B) NodeList cannot contain more than 10 items
- C) NodeList properties may be live properties that update if the DOM is updated (like `childNodes`)
- D) Many array functions are not available on NodeList

The answer is B.

[\*\*Q18 Unanswered.\*\*](#)

**Q19 Answered.**

Which es6 array functions in JavaScript return a new array? Select all correct answers.

- A) map()
- B) forEach()
- C) filter()
- D) find()

The answers are A, C.

[\*\*Q19 Unanswered.\*\*](#)

**Q20 Answered.**

What is the largest whole number JavaScript can reliably represent with the number primitive?

- A)  $2^{53} - 1$
- B)  $2^{32} - 1$
- C)  $2^{63} - 1$
- D) JavaScript can represent infinitely large numbers

The answer is A.

[\*\*Q20 Unanswered.\*\*](#)

Q21 Answered.

Which function will modify a string to be URI compliant?

- A) JSON.stringify()
- B) encodeURI()
- C) encodeString()
- D) JSON.parse()

The answer is B.

Q21 Unanswered.

[Q22 Answered.](#)

How are the 64 bits assigned to a number used to store numbers in JS?

Answer: 52 bits are dedicated to the value, eleven bits to the exponent, and one bit to the sign.

[Q22 Unanswered.](#)

**Q23 Answered.**

In computer science, what is the difference between *null* and *undefined* ?

*undefined* means a variable has no value because the developer has not set one yet. *null* means the variable was intentionally set to have no value.

**Q23 Unanswered.**

Q24 Answered.

What is the technical name of the minus sign (-) in JavaScript?

- A) Unary Subtraction
- B) Binary Negation
- C) Unary Negation
- D) Binary Subtraction

The answer is C.

Q24 Unanswered.

**Q25 Answered.**

True or False? The super keyword cannot be used outside of constructor functions.

- A) True
- B) False

The answer is B. For example, `super.functionOnParent([arguments]);` This is used to invoke a function on the parent of an object if the function has been overridden by the child.

[\*\*Q25 Unanswered.\*\*](#)

**Q26 Answered.**

What are two ways to access the color value on the cat object?

```
const cat = {  
    color: "orange"  
};
```

Answer: cat.color and cat['color']

[\*\*Q26 Unanswered.\*\*](#)

**Q27 Answered.**

**What is the Stack in JavaScript Runtime?**

The Stack is a line of Frames, which are function calls. The most recent function call is at the top of the stack, and must be executed and popped off the Stack before the next function call can be completed.

When one function calls another function, the newly called function is placed at the top of the Stack. The Stack is Last In, First Out, which means the most recently added function must be executed first.

[\*\*Q27 Unanswered.\*\*](#)

**Q28 Answered.**

What is the syntax for hexidecimal escape sequence?

- A) '\zfg'
- B) '\10ab'
- C) '\x'

The answer is C. For example, "\xB1" becomes "±"

**Q28 Unanswered.**

**Q29 Answered.**

Write the code to create a custom made abstract class in vanilla JavaScript (no TypeScript)? The abstract keyword is not available as of ES2015.

Example Answer:

```
class MyAbstract {  
    constructor() {  
        if(new.target === MyAbstract) {  
            throw new Error('Error Message');  
        }  
    }  
}
```

[\*\*Q29 Unanswered.\*\*](#)

Q30 Answered.

What will the code below output to the console and why?

```
const badMap = new Map();
badMap.set('a', 1);
badMap.b = 2
console.log(badMap)
```

- A) Map(1) {"a" => 1}
- B) Map(1) {"b" => 2}
- C) Map(2) {"a" => 1, "b" => 2}
- D) Compile time error

The answer is A. `badMap.b` will set the value `b` on the `badMap Object`, but `b` was not set in a way that properly interacted with the `Map` structure.

Q30 Unanswered.

### Q31 Answered.

What are the benefits of object-oriented programming (OOP)? Select three answers.

- A) Code is socketable or modular, so a class's source code can be maintained independently of the rest of an application's code.
- B) An object's code is kept private, meaning other objects only interact with an object's methods, so the internal code is kept hidden.
- C) Objects are easily reusable. They can be plugged into multiple parts of the code.
- D) Object-oriented code executes faster and computer resource usage is minimized.

The answers are A, B, C. A is describing the fact that an object can be modified without needing to recode anywhere else in an app. For example, if the product class of a Point of Sale system needs to be expanded to meet new business needs, there is likely nothing wrong with the rest of the code. OOP allows the product class to be modified while leaving the rest of the code as-is.

B is useful for proprietary code. For example, if a company develops and sells code that plugs into an existing customer application, the proprietary code needs to be kept invisible to the customer. If the customer only interacts with methods, the variables and logic of the classes in the code are kept proprietary.

C describes the ability of a class to be plugged into an app in multiple locations, or even into multiple apps. Older programming languages were linear in their execution. OOP lets an object be called repeatedly without having to rewrite the same code repeatedly.

### Q31 Unanswered.

[Q32 Answered.](#)

Assume const map has the following values: Map(2) {"a" => 2, "b" => 2}  
In one line, transform map into a 2d array.

Answer: Either [...map] or Array.from(map) will accomplish this.

[Q32 Unanswered.](#)

**Q33 Answered.**

True or False? Only custom classes and objects can be extended in JavaScript?

- A) True
- B) False

The answer is B. For example, custom classes and objects extend Object. Any built-in object can be extended.

**Q33 Unanswered.**

**Q34 Answered.**

True or false? An instance level variable can be accessed anywhere in a class as long as an instance has been created.

- A) True
- B) False

The answer is B. An instance level variable cannot be accessed in a static method.

**Q34 Unanswered.**

### Q35 Answered.

Assume TypeScript is enabled. Why will this code not compile? Select one option.

```
let i: number;  
if(5%3>2){  
    i=0;  
}else if(5%3>1){  
    i=1;  
}  
console.log(i);
```

- A) It compiles fine
- B)  $5\%3 > 2$  and  $5\%3 > 1$  do not return booleans and thus cannot be used in an if statement
- C) i is a local variable and its value is never explicitly defined

The answer is A. TypeScript may complain about i being used before it is assigned a value, but it will compile.

### Q35 Unanswered.

**Q36 Answered.**

True or False? length is a method on strings and a property on Arrays?

- A) True
- B) False

The answer is B. length is a property on strings and a method on Arrays.

[\*\*Q36 Unanswered.\*\*](#)

[Q37 Answered.](#)

What is the first argument of function.call?

- A) A parameter for the function called
- B) It takes no options
- C) The instance object to invoke the function on

Answer: the object to invoke the function on, i.e.

person.fullName.call(person2); where person2 has the properties necessary to invoke the person.fullName function. It can be thought of as a way of setting scope.

[Q37 Unanswered.](#)

### Q38 Answered.

What is a technical name for the kind of function that printName is? Select all correct options.

```
function getName() {  
    var name = "Jon";  
    function printName() {  
        console.log(name);  
    }  
    printName();  
}  
getName();
```

- A) Inner function
- B) Contained function
- C) Closure

The answers are A, C.

### Q38 Unanswered.

Q39 Answered.

True or False? Arrow functions only require parenthesis if there are multiple arguments.

- A) True
- B) False

False. Arrow functions also require parenthesis if there are zero args. If there is exactly one argument parenthesis are optional.

Q39 Unanswered.

**Q40 Answered.**

True or false? A class level variable can be accessed anywhere in a class.

- A) True
- B) False

The answer is A. A class level variable, also known as a static variable, can be accessed in any method, including static methods and constructors.

[Q40 Unanswered.](#)

**Q42 Answered.**

What kind of error is thrown when you attempt to assign a new value to a const?

- A) ReferenceError
- B) Error
- C) RangeError
- D) TypeError

The answer is D.

[\*\*Q42 Unanswered.\*\*](#)

**Q43 Answered.**

Overriding and overloading are examples of which Object Oriented Programming principle?

- A) inheritance
- B) abstraction
- C) polymorphism
- D) encapsulation

The answer is C.

[\*\*Q43 Unanswered.\*\*](#)

#### Q44 Answered.

True or False: static variables cannot be modified in non-static methods?

- A) True
- B) False

The answer is B. A non-static method can reference and modify a static variable with Class.Static syntax, like below.

```
class Test {  
    static MAX = 10;  
    getMax() {  
        Test.MAX = 12;  
        console.log(Test.MAX);  
    }  
}
```

#### Q44 Unanswered.

## Q45 Answered.

What happens if str === 'test1' in the below code?

```
switch (str) {  
    case "test1":  
    case "test2":  
        console.log("At test2");  
}
```

- A) “At test2” is printed
- B) Nothing
- C) Runtime Error

The answer is A. If *break* is not specified, a match will then fall through all remaining cases until it encounters the *break* keyword.

## Q45 Unanswered.

#### **Q46 Answered.**

What does it mean when JavaScript is described as having “first-class functions”?

- A) functions can be recursive
- B) functions can be anonymous
- C) functions are treating like any other variable
- D) functions use the curly brace syntax

The answer is C. This allows functions to be passed as parameters, returned from functions, and more.

#### **Q46 Unanswered.**

## Q47 Answered.

True or False? The following code compiles.

```
async function promiseMe() {  
    try {  
        let z = await Promise.reject(30);  
    }  
}
```

- A) True
- B) False

The answer is B. The rejection of a primitive is not a problem. However, the try without a corresponding catch or finally is a problem.

## Q47 Unanswered.

#### **Q48 Answered.**

True or False? Date.getDate() returns the day of the month with values 1-31 while Date.getDay() returns the day of the week with values of 0-6.

- A) True
- B) False

The answer is A. It can be difficult to remember which Date functions are zero index and which are one index.

#### **Q48 Unanswered.**

**Q49 Answered.**

Async operations are put into what kind of queue?

- A) blocking queue
- B) event queue
- C) async queue

The answer is B.

**Q49 Unanswered.**

**Q50 Answered.**

What are the three functions JavaScript has long offered for executing code after time has elapsed? Select three answers.

- A) setTimeout()
- B) setInterval()
- C) callAsyncLoop()
- D) requestAnimationFrame

The answers are A, B, D.

[\*\*Q50 Unanswered.\*\*](#)

## Q51 Answered.

What Web API might you call if you wanted to measure the execution time of a code block?

- A) window.now()
- B) window.timer()
- C) performance.now()
- D) performance.timer()

The answer is C. An example:

```
const t0 = performance.now();
runFunction();
const t1 = performance.now();
console.log(t0 - t1);
```

## Q51 Unanswered.

**Q52 Answered.**

What is a MIME type?

A MIME type is a string sent along with a file indicating the file type. For example, the MIME type sent with a JSON file is ‘application/json’.

**Q52 Unanswered.**

[Q53 Answered.](#)

What is a simple syntax for exporting the below variables?

```
let myLet = 10;
```

Answer:

```
export let myLet = 10;
```

The export keyword can be used at variable declaration. Many people will group exports together in a file but that is not necessary.

[Q53 Unanswered.](#)

**Q54 Answered.**

What function property returns the number of args the function expects?

- A) Function.argsLength
- B) Function.length
- C) Function.parameters

The answer is B. Functions are objects and thus have properties.

**Q54 Unanswered.**

**Q55 Answered.**

What does it mean to "import a module for side effects only"?

Answer: import the module without specifying anything to import, i.e.

`import '/someLib/my-class.js'; //No named or default export specified`

**Q55 Unanswered.**

## Q56 Answered.

True or False? The below code compiles and prints 3?

```
const test = () => {  
    return  
    1 + 2;  
}  
console.log(test());
```

- A) True
- B) False

The answer is B. *return* statements automatically have a semicolon placed after them by the compiler if the semicolon was left off. It is a good idea to use parenthesis if you ever need to wrap lines in a return statement.

## Q56 Unanswered.

Q57 Answered.

True or False? The following are equivalent.

```
var a, z = 1;  
a = z;
```

and

```
var a = z = 1;
```

- A) True
- B) False

The answer is B. z is declared globally. Otherwise, the values are the same.

Q57 Unanswered.

## Q58 Answered.

True or False? The code below returns 3.

```
const arr = [1, 2];
const reduced = arr.reduce((accumulator,element) => accumulator);
```

- A) True
- B) False

The answer is B. The accumulator does not use a `+=` return. Rather, the most recent return value is stored as the accumulator. The below statement would have the desired result:

```
const reduced = arr.reduce((accumulator, element) => accumulator + element);
```

## Q58 Unanswered.

**Q59 Answered.**

Where must the "use strict"; directive be placed in a file?

- A) Top
- B) Middle
- C) Bottom
- D) As an import

The answer is A.

**Q59 Unanswered.**

[Q60 Answered.](#)

How can you use composition in vanilla JavaScript (no TypeScript)?

One example: Create functions that are not methods on any object. Use Object.assign(obj) to assign the required functions to obj as methods on obj.

[Q60 Unanswered.](#)

Q61 Answered.

What will the code below output to the console and why?

```
for(let i = 1, j = 10; i< 5 < j; i += j/i){  
    console.log(Math.ceil(++i));  
}
```

- A) nothing
- B) divide by zero error
- C) 1, 12
- D) It will enter into an infinite loop

The answer is D.  $i < 5 < j$  will eval like this:  $(i < 5) < j$ . In the above code,  $i < 5$  is false in the first iteration. Then false  $< j$  evals to true.  $i$  keeps going up and thus is never less than 5, so the code loops continuously.

Q61 Unanswered.

**Q62 Answered.**

Show how object literals can be constructed without use of the colon :

```
let a = 1, b = 2, c = 3;  
let obj = { a, b, c }
```

**Q62 Unanswered.**

**Q63 Answered.**

What is the maximum size of an array in JavaScript?

- A)  $2^{16} - 1$
- B)  $2^{32} - 1$
- C)  $2^{16}$
- D)  $2^{32}$

The answer is B. If you attempt to set a value above  $2^{32} - 1$ , you will get a RangeError.

[\*\*Q63 Unanswered.\*\*](#)

**Q64 Answered.**

If a function is called with less parameters than the declared amount of parameters, what value is assigned to the missing parameters?

- A) null
- B) undefined
- C) 0

The answer is B.

**Q64 Unanswered.**

**Q65 Answered.**

What is the Event Capturing phase?

The phase in which the event trickles down the ancestors (window, div, etc) to the most granular element.

**Q65 Unanswered.**

**Q66 Answered.**

True or False? Setters still work on a frozen object.

- A) True
- B) False

The answer is B. The setters still exist and can be called. They give the appearance of working, but they do not actually change values.

[\*\*Q66 Unanswered.\*\*](#)

Q67 Answered.

What does  $3 > 2 > 1$  return?

- A) true
- B) false
- C) 0

The answer is B.  $3 > 2$  evaluates to true. Then *true*  $> 1$  evaluates to false.

Q67 Unanswered.

## **Q68 Answered.**

The below two code snippets return the same value. What is the difference in how that occurred?

```
'123'.length; //line 1  
new String(123).length; //line 2
```

- A) line 1 uses explicit coercion, line 2 uses implicit coercion
- B) line 1 uses implicit coercion, line 2 uses explicit coercion
- C) There is no difference

The answer is B.

## **Q68 Unanswered.**

[Q69 Answered.](#)

Which of the following is NOT a time that you would use an anonymous function?

- A) as a class method
- B) in a closure
- C) as a callback
- D) as an IIFE (immediately invoked function execution)

The answer is A.

[Q69 Unanswered.](#)

**Q70 Answered.**

What is the difference in layout time and painting time in a web browser?

The layout time is the time it takes to calculate the positioning of each object that will be painted. The layout is a 3d construct. Paint time is the time it actually takes to paint the points on the screen. Painting is a 2d construct.

[\*\*Q70 Unanswered.\*\*](#)

**Q71 Answered.**

What will the code below output to the console and why?

```
let a = 8  
a >>>= 3;  
console.log(a);
```

- A) 1
- B) 10
- C) 4

The answer is A. `>>>=` is known as the unsigned right shift assignment operator. It moves the bits to the right by the specified amount. In this case, 8 (1000 in binary) has its bits moved right by three (0001).

[Q71 Unanswered.](#)

**Q72 Answered.**

What will the code below output to the console and why?

```
console.log([1, 2,,3].join(', '));
```

- A) lint error
- B) 1, 2, 3
- C) 1, 2, , 3

The answer is C. Arrays can have empty spots. The value of the above array is [1, 2, *empty* , 3]

[Q72 Unanswered.](#)

**Q73 Answered.**

Give an example of how to avoid race conditions on network calls?

One possible answer: Create a cancel token or utilize a cancel API in your network calls. Cancel previous requests when you make a new one.

**Q73 Unanswered.**

**Q74 Answered.**

What is a pure function?

Answer:

A function which always returns the same output, given the same inputs and has no side effects.

[Q74 Unanswered.](#)

**Q75 Answered.**

What is currying?

Answer: Converting a single function with multiple arguments into a function called multiple times with a single argument.

**Q75 Unanswered.**

# Test 3 Answers

**Q1 Answered.**

True or False? The compiler will throw an error for the below code.

```
const x = 10;  
x = 12;
```

- A) True
- B) False

The answer is A. The *const* (short for *constant*) keyword denotes a value that cannot be changed.

[\*\*Q1 Unanswered.\*\*](#)

**Q2 Answered.**

What is the TypeScript type of the below code?

```
const x = "hello"
```

- A) "hello"
- B) const
- C) string
- D) No type is assigned

The answer is A. Since const values cannot be changed, the type can be locked into the actual string literal value.

[\*\*Q2 Unanswered.\*\*](#)

**Q3 Answered.**

What will the code below output to the console and why?

```
let person = {  
    age: 25,  
    name: 'Justin'  
}  
  
const byRef = (myObj) => {  
    myObj.age = 30  
}  
  
byRef(person);  
console.log(person.age);
```

- A) 25
- B) 30
- C) Compile time error

The answer is B. The reference to the person object in memory is actually passed to the byRef function. This means that any changes made to the object in the byRef function will truly affect the underlying object. This is true when objects are passed as parameters.

[Q3 Unanswered.](#)

#### **Q4 Answered.**

What will the code below output to the console and why?

```
let x;  
if(x = true === true){  
    console.log(x);  
} else {  
    console.log('Not x');  
}
```

- A) true
- B) false
- C) Not x
- D) Compile time error

The answer is A. Variables can be assigned in the evaluation of an if statement. When the variable is assigned, it is immediately used as the if evaluation.

#### **Q4 Unanswered.**

Q5 Answered.

What will the code below output to the console and why?

```
let x = 10;  
  
function myFunc(y){  
    y = 12;  
}  
console.log(myFunc(x));
```

- A) 10
- B) 12
- C) undefined

The answer is C. Evaluate the code carefully. You will notice that myFunc does not have a return value.

Q5 Unanswered.

## Q6 Answered.

Assume the aggregator function below calculates how many days old you are, combines your first and last names, and splits apart your address. Select all the ways it violates clean code principles.

```
const aggregator = (DOB, firstName, lastName, address, skills) => {...}
```

- A) The function is poorly named
- B) The function should use the function keyword instead of being a lambda function
- C) The function is doing too many things and the function should be split up
- D) The function doesn't do anything with the skills parameter

The answers are A, C, D. Clean code principles require good names and concise, specific code.

## Q6 Unanswered.

**Q7 Answered.**

What is the technical definition of the DOM from a JS perspective?

From the MDN Web Docs: The DOM is "A tree-like structure of connected nodes that represents the different elements and strings of text appearing in a markup document"

[https://developer.mozilla.org/en-US/docs/Web/Web\\_Components/Using\\_shadow\\_DOM](https://developer.mozilla.org/en-US/docs/Web/Web_Components/Using_shadow_DOM)

**Q7 Unanswered.**

**Q8 Answered.**

Given FancyButton class, how would you extend button to create a custom element named fancy-button?

- A) customElements.extend('fancy-button', FancyButton, { extends: 'button' });
- B) customElements.extend('fancy-button', FancyButton, { implements: 'button' });
- C) customElements.define('fancy-button', FancyButton, { extends: 'button' });
- D) customElements.define('fancy-button', FancyButton, { implements: 'button' });

The answer is C.

**Q8 Unanswered.**

**Q9 Answered.**

Which of the following is correct string interpolation syntax?

const x = 'me';

- A) `Pick \${x}`
- B) `Pick &{x}`
- C) 'Pick \${x}'
- D) 'Pick &{x}'

The answer is A.

[\*\*Q9 Unanswered.\*\*](#)

**Q10 Answered.**

What is the name of the pattern where the total number of instances of an object is limited to one?

- A) Singleton
- B) Factory
- C) Design
- D) OOP

The answer is A.

**Q10 Unanswered.**

**Q11 Answered.**

Which of the following is not a method on the Set object?

- A) add()
- B) next()
- C) clear()
- D) delete()
- E) entries()

The answer is B.

[\*\*Q11 Unanswered.\*\*](#)

**Q12 Answered.**

What window method allows you to store persistent key/value pairs in browser memory?

- A) window.browserStorage
- B) window.localStorage
- C) window.persistentStorage
- D) Nothing allows for this

The answer is B.

[\*\*Q12 Unanswered.\*\*](#)

**Q13 Answered.**

All bitwise operations are performed on what bit binary numbers?

- A) 8
- B) 16
- C) 32
- D) 64

The answer is C.

**Q13 Unanswered.**

**Q14 Answered.**

When does the delete operator not remove a property from an object?

When the object is created as non-configurable, like so:

```
var Car = {};  
Object.defineProperty(Car, 'name', {configurable: false})
```

**Q14 Unanswered.**

## Q15 Answered.

What method should you call if you need to get the properties of an Object?

- A) Object.getKeyValues
- B) Object.getList
- C) Object.getNames
- D) Object.getOwnPropertyNames

The answer is D. See the below code, for example:

```
const xy = {  
    a: 1,  
    b: 2  
};  
console.log(Object.getOwnPropertyNames(xy));
```

There are other methods that can also be used to retrieve the property names.

## Q15 Unanswered.

**Q16 Answered.**

What will the code below output to the console and why?

```
console.log(2 * 3 ** 3);
```

- A) 216
- B) 18
- C) 54
- D) Compile time error

The answer is C. `**` is the exponent operator. Furthermore, `**` has more precedence than `*`, so the code above is the same as `2 * 27`.

[Q16 Unanswered.](#)

## Q17 Answered.

Evaluate the code below. What is a one character change that can prevent this line from throwing an error?

```
const dog = {  
    color: "brown",  
    breed: {  
        pure: true  
    }  
};  
console.log(dog.hair.short);
```

The answer: change `console.log(dog.hair.short);` to `console.log(dog.hair?.short);`. This is called optional chaining.

## Q17 Unanswered.

**Q18 Answered.**

Which array functions are not available on NodeList? Select all that apply.

- A) forEach()
- B) join()
- C) filter()
- D) map()

The answer is A.

**Q18 Unanswered.**

**Q19 Answered.**

Which of the following is not a JSON method?

- A) reverse()
- B) parse()
- C) stringify()

The answer is A. The JSON object only has two methods (not including inherited methods), parse and stringify.

**Q19 Unanswered.**

**Q20 Answered.**

True or False? There is a global Map Object in JavaScript.

- A) True
- B) False

The answer is A. Not to be confused with Array.map(), the Map object holds key-value pairs and is iterable.

[\*\*Q20 Unanswered.\*\*](#)

**Q21 Answered.**

What does it mean that a WeakSet object is weak?

From the MDN docs: "If no other references to an object stored in the WeakSet exist, those objects can be garbage collected."

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/WeakSet](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/WeakSet)

[Q21 Unanswered.](#)

**Q22 Answered.**

What is one way to overcome the imprecision of floating point decimals?

- A) multiply and divide
- B) It works as-is
- C) run through an expander function

The answer is A. For example: const x = (0.1 \* 10 + 0.1 \* 10) / 10;

[Q22 Unanswered.](#)

**Q23 Answered.**

Evaluate the code below. How can it be refactored to use less code?

```
const temp = object.prop1;  
const test = ((temp === null || temp === undefined)) ? undefined :  
temp.prop2;
```

One possible answer: object. prop1?.prop2;

[Q23 Unanswered.](#)

**Q24 Answered.**

Evaluate the code below. What are the first and last outputs of the loop?

```
for(let i = 0; i <= 10; ++i){  
    console.log(i);  
}
```

- A) 0, 9
- B) 0, 10
- C) 1, 9
- D) 1, 10

The answer is B. Interestingly, i prints out with a value of zero in the initial iteration despite ++i.

[\*\*Q24 Unanswered.\*\*](#)

## Q25 Answered.

Create a function on the below MyChild class that calls the parent printSomething method.

```
class MyParent {  
    constructor(){  
    }  
    printSomething(){  
        console.log(5);  
    }  
}  
class MyChild extends MyParent {  
    constructor(){  
        super()  
    }  
    printSomething(){  
        console.log(10);  
    }  
}
```

One example answer:

```
printSomethingOld(){  
    super.printSomething()  
}
```

## Q25 Unanswered.

**Q26 Answered.**

What does the code below return and why?

const value = 10 / -0;

- A) NaN
- B) -NaN
- C) Infinity
- D) -Infinity

The answer is D. It returns -Infinity because 0 and -0 have two different values in memory due to the way numbers are stored.

[\*\*Q26 Unanswered.\*\*](#)

**Q27 Answered.**

What is Heap in JavaScript Runtime?

Heap is the region of memory where objects are stored.

[\*\*Q27 Unanswered.\*\*](#)

**Q28 Answered.**

How does JavaScript handle it behind the scenes when a String Object method is called on a string literal value?

Answer: From MDN docs - "JavaScript automatically converts the string literal to a temporary String object, calls the method, then discards the temporary String object."

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Text\\_formatting](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Text_formatting)

**Q28 Unanswered.**

Q29 Answered.

What will the code below output to the console and why?

```
const obj = {  
    a: 1,  
    b: 2  
};  
  
const arr = [];  
console.log(arr[0] = {...obj});
```

- A) 1, 2
- B) {...obj}
- C) {a: 1, b: 2}
- D) Does not compile

The answer is C. The ... syntax is known as the spread operator.

Q29 Unanswered.

[Q30 Answered.](#)

What does Map.delete(key ) return?

- A) It has no return
- B) *true* always
- C) *false* always
- D) *true* if the key was deleted, *false* if the item didn't exist and wasn't deleted

The answer is D.

[Q30 Unanswered.](#)

Q31 Answered.

Which of the following is not an example of inheritance when using TypeScript? Select one option.

- A) An object has a field that its superclass has
- B) An object has a method that its superclass has
- C) An object has a method that its interface has
- D) An object has a static field that its superclass has

The answer is C. Interfaces are a form of composition, not inheritance. Also, multiple interfaces can be implemented by a class, whereas a class can only extend one other class.

Q31 Unanswered.

### **Q32 Answered.**

True or false? In JavaScript, an object can have only one subclass but unlimited parent classes.

- A) True
- B) False

The answer is B. An object can have only one parent class but unlimited subclasses. For example, Vehicle class may have SUV, Coup and Truck as subclasses. It may have Transport as its parent class, but it cannot have any other parent class.

### **Q32 Unanswered.**

**Q33 Answered.**

True or false? An object can implement multiple interfaces. Assume TypeScript is enabled.

- A) True
- B) False

The answer is A. An object can be composed of many interfaces, but can inherit from only one class.

**Q33 Unanswered.**

Q34 Answered.

What does the code print? Select one option.

```
class MyClass {  
    static staticMethod() {  
        return this.name;  
    }  
    constructor() {  
        this.name = 'Jon';  
    }  
}  
  
const me = new MyClass();  
console.log(MyClass.staticMethod());
```

- A) Jon
- B) MyClass
- C) Compile time error
- D) Runtime error

The answer is B. The *this* keyword in staticMethod is not scoped to the instance level.

Q34 Unanswered.

**Q35 Answered.**

True or False? A String is a primitive data type.

- A) True
- B) False

The answer is A....and yet String is also an object. Usually the primitive type *string* is referred to in lower case. The String object is usually capitalized. But both exist in JavaScript.

**Q35 Unanswered.**

**Q36 Answered.**

Show two ways to make an existing array empty.

Two possible answers: `array.length(0)`, `array.splice(0, array.length)`

**Q36 Unanswered.**

**Q37 Answered.**

What is the value of str after the code executes?

```
const str = " this".concat("that ").substring(3).trim().valueOf();
```

- A) 2
- B) h
- C) histhat
- D) Does not compile

The answer is C. `valueOf` simply returns the value of whatever string it's called on.

[Q37 Unanswered.](#)

Q38 Answered.

What is a Generator function in JavaScript?

Answer: Generators are functions that can be exited and re-entered later. For example, a while loop in a generator function can be paused.

Read the MDN docs for more info: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/function\\*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/function*)

Q38 Unanswered.

**Q39 Answered.**

Which of the following is a valid modifier of class, assuming TypeScript is enabled? Select one option.

- A) private
- B) final
- C) abstract
- D) public
- E) default

The answer is C. The private and public modifiers are not applicable to classes.

[\*\*Q39 Unanswered.\*\*](#)

**Q40 Answered.**

Which of the following is not a reserved word in java? Select one option.

- A) abstract
- B) compile
- C) do
- D) enum

The answer is B; compile is not reserved. The *abstract* keyword was reservered but was removed from the reserved list in ECMA Script 5/6.

[\*\*Q40 Unanswered.\*\*](#)

Q41 Answered.

What is the output of this code?

```
class TestClass{  
    constructor() {  
        this.myInt = 5;  
    }  
  
    mainMethod(){  
        const test = new TestClass();  
        test.myInt = 6;  
        console.log(test.incrementIntAgain(test.incrementInt(test.myInt)));  
    }  
  
    incrementInt (theirInt) {  
        return theirInt++;  
    }  
  
    incrementIntAgain (theirInt) {  
        return ++theirInt;  
    }  
}
```

A) 5  
B) 6  
C) 7  
D) 8

The answer is C. This code was written in an intentionally difficult way for educational purposes, it is not a reflection of good code.

Q41 Unanswered.

**Q42 Answered.**

Package Exam;

```
public class Tester{
    private int myInt = 5;

    public static void main(String []args){
        Tester test = new Tester();
        test.myInt = 6;
        System.out.println(test.incrementIntAgain(test.incrementInt(test.myInt)));
    }

    public int incrementInt (int theirInt) {
        return theirInt++;
    }

    public int incrementIntAgain (int theirInt) {
        return ++theirInt;
    }
}
```

What is the output of this code?

- A) 5
- B) 6
- C) 7
- D) 8

The answer is C. Passing the returned value of one method as the parameter of another method is fine.

**Q42 Unanswered.**

**Q43 Answered.**

True or False: an instance level object can access a static variable?

- A) True
- B) False

The answer is B. Static variables can only be accessed directly from the class, i.e. Class.StaticVar.

**Q43 Unanswered.**

**Q44 Answered.**

What does the below code output to the console and why?

```
function Dog(name) {  
    this.name = name;  
}  
const buddy = new Dog("buddy");  
const rover = Object.create(buddy);  
  
console.log(Object.getPrototypeOf(rover));  
console.log(Object.getPrototypeOf(buddy));
```

Answer: Dog {name: "buddy"} and a reference to the Dog class.

Object.getPrototypeOf(object) returns a reference to the prototype of an object. The prototype of rover is the buddy object, and the prototype of the buddy object is the Dog class.

[Q44 Unanswered.](#)

**Q45 Answered.**

What does the below code output to the console and why?

```
let x = (1, 2, 3);  
console.log(x);
```

- A) 1
- B) 2
- C) 3
- D) Compile time error

The answer is C. The comma separator returns the final value.

**Q45 Unanswered.**

**Q46 Answered.**

What are some differences in the RegExp literal notation and constructor?  
Select all that apply.

- A) The literal notation's parameters are enclosed between slashes and have no quotation marks
- B) The literal notation is evaluated at compile time
- C) The constructor function does not use slashes but has quotes
- D) The constructor is evaluated at runtime

All of the above are correct.

**Q46 Unanswered.**

[Q47 Answered.](#)

What is the order of printing?

```
function resolveQuickly(x) {  
    return new Promise(resolve => {  
        setTimeout(() => {  
            console.log(x)  
        }, 1);  
    });  
}  
  
async function asyncTest() {  
    var x = resolveQuickly(10);  
    console.log(5);  
}  
  
asyncTest();
```

- A) 10, 5
- B) 5, 10
- C) Nothing prints.

The answer is B. There is no await on the call to resolveQuickly. Since the Promise execution contains a setTimeout, it will be asynchronous.

[Q47 Unanswered.](#)

**Q48 Answered.**

Where did the name JavaScript originate?

JavaScript was originally developed at Netscape in 1995 and was supposed to be called LiveScript. However, the name was changed to JavaScript because Java was a popular language.

Source: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/A\\_introduction\\_to\\_JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_introduction_to_JavaScript)

**Q48 Unanswered.**

**Q49 Answered.**

Which of the following is NOT an advantage promises have over old-style callbacks?

- A) execution speed
- B) .then() chaining
- C) promise callbacks are always called in the order they are placed in the event queue
- D) better error handling
- E) promises avoid inversion of control

The answer is A.

Read more here: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Introducing>

**Q49 Unanswered.**

**Q50 Answered.**

True or False? A do-while loop always executes at least once.

- A) True
- B) False

The answer is A. The ‘do’ portion of the do-while is entered into before any truthy evaluation is performed.

[\*\*Q50 Unanswered.\*\*](#)

**Q51 Answered.**

Show an example of setTimeout being called recursively to mimic setInterval.

```
setTimeout(function recur() {  
  console.log('called me');  
  setTimeout(recur, 1000);  
}, 1000);
```

**Q51 Unanswered.**

**Q52 Answered.**

What data types does JSON support? Select all appropriate answers.

- A) string
- B) number
- C) object
- D) null
- E) array
- F) boolean

All of the above are correct.

**Q52 Unanswered.**

**Q53 Answered.**

What are two syntaxes for exporting theFunction as default?

The answer:

```
export { theFunction as default };  
export default function theFunction () { ... }
```

[Q53 Unanswered.](#)

**Q54 Answered.**

What is the syntax difference in function declarations and function expressions?

Answer:

```
function myFunc() {console.log('hoisted');}  
var myFunc2 = function () {console.log('not hoisted')}
```

[\*\*Q54 Unanswered.\*\*](#)

**Q55 Answered.**

True or False? *import* can be called as a function.

- A) True
- B) False

The answer is A. *import* can be called dynamically and it returns a promise. Do this if a module is not needed at load time.

[Q55 Unanswered.](#)

## Q56 Answered.

What does the following code print to the console and why?

```
console.log(x)
x = 5;
if(x){
    var x = 10;
    console.log(x);
}
```

- A) 5, 10
- B) 10, 10
- C) 5, 5

The answer is B. If you run this code in a browser dev tools console, it prints 10, 10.

It will throw lint errors in an IDE environment.

## Q56 Unanswered.

**Q57 Answered.**

What is the purpose of the Array reduce()?

- A) reduce the array to callbacks
- B) reduce the array to a composite function
- C) reduce the array to a single value.

The answer is C. How? By calling a function on every element of the array and "accumulating" the return values in an accumulator variable

**Q57 Unanswered.**

**Q58 Answered.**

What is memoization?

- A) increasing a function's performance by caching its previously computed results
- B) something unavailable in JavaScript
- C) making a composition of object properties for future object use

The answer is A.

**Q58 Unanswered.**

**Q59 Answered.**

Create an example of an abstract class in JavaScript.

Answer: you can only create an abstract class if TypeScript or another typing superset is enabled. Example:

```
abstract class Printer {  
    constructor() {}  
  
    print1(): void {  
        console.log("printing");  
    }  
  
    abstract print2(): void;  
}
```

**Q59 Unanswered.**

**Q60 Answered.**

Give an example of using String.prototype.match to run a string through a regex?

One possible answer:

```
const sentence = 'Heyo!';
const regex = /[A-Z]/g;
console.log(sentence.match(regex));
```

[\*\*Q60 Unanswered.\*\*](#)

Q61 Answered.

What will the code below output to the console and why?

```
for(let i = 6; i > 0; i-=2){  
    if(i%2){  
        continue;  
    }  
    console.log(i)  
}
```

- A) nothing
- B) 4, 2, 0
- C) 6, 4, 2
- D) 4, 2

The answer is C. *i* is always even, so the continue is never hit. The -= is valid syntax.

[Q61 Unanswered.](#)

## Q62 Answered.

True or False? Duplicated property names on an object are considered a syntax error.

- A) True
- B) False

The answer is B. In ECMAScript 5 strict mode, duplicate property names threw a SyntaxError. However, since ECMAScript 2015 it is no longer an error. Instead, the most recent assignment overrides the old assignment. This decision was made because property names can now be computed.

## Q62 Unanswered.

**Q63 Answered.**

True or False? The following is incorrect syntax.

```
function foo(arg) {  
    let arg = 1;  
}
```

- A) True
- B) False

The answer is A. This is known as a redeclaration of a formal parameter.

[\*\*Q63 Unanswered.\*\*](#)

**Q64 Answered.**

What is the name of the array-like object that contains the values of all objects passed to the function?

- A) arguments
- B) args
- C) params
- D) parameters

The answer is A. The arguments object is accessible inside all non-arrow functions.

**Q64 Unanswered.**

## Q65 Answered.

What is the third parameter, *true* , used for in the below code?

```
element.addEventListener("click", e => console.log(element.eventPhase),  
true);
```

- A) enabling catching an event on capture phase
- B) enabling catching an event on bubbling phase
- C) it is not used in the addEventListener function

The answer is A. Many developers are familiar with event bubbling; fewer are familiar with the less-discussed event capture phase. The default for the third parameter is *false* , but enabling it means the even will fire on event capture.

## Q65 Unanswered.

**Q66 Answered.**

When considering coercion, what is a wrapper object?

An object wrapped around a primitive in order to “give” the primitive properties. For example:

```
const str = '123';
console.log(str.length); //returns 3
```

str doesn't actually have properties. However, it gets wrapped in a String object when you try to access String object properties.

[\*\*Q66 Unanswered.\*\*](#)

Q67 Answered.

What does  $3 > 2 == 1$  return?

- A) true
- B) false
- C) 0

The answer is A.  $3 > 2$  returns true,  $true == 1$  returns true.

Q67 Unanswered.

**Q68 Answered.**

What is a truthy value in javascript?

From the MDN docs: A truthy value is a value that is considered *true* in a boolean context.

Source: <https://developer.mozilla.org/en-US/docs/Glossary/Truthy#:~:text=In%20JavaScript%2C%20a%20truthy%20value,type%20coercion%20in%20Boolean%20contexts.>

**Q68 Unanswered.**

[Q69 Answered.](#)

What is the name of the below syntax?

```
(function() {  
    console.log('IIFE');  
})();
```

Answer:

Immediately Invoked Function Execution. The function executes immediately due to the () after the function.

[Q69 Unanswered.](#)

**Q70 Answered.**

What is the scope of arrow functions?

- A) global
- B) the scope in which they were defined
- C) block level

The answer is B.

[\*\*Q70 Unanswered.\*\*](#)

Q71 Answered.

What will the code below output to the console and why?

```
let a = 8  
a >>>= 5;  
console.log(a);
```

- A) -1
- B) - 1167541818
- C) 0

The answer is C. The unsigned right shift assignment operator won't flip a positive number to negative.

Q71 Unanswered.

**Q72 Answered.**

True or False? The shift and pop methods on arrays return a new array without changing the value of the original array.

- A) True
- B) False

The answer is B. The shift method removes the first element (index 0) and returns it, and it *does* modify the original array. The pop method does the same but with the last element in the array.

[\*\*Q72 Unanswered.\*\*](#)

**Q73 Answered.**

True or False? Objects assigned to const immutable.

- A) True
- B) False

The answer is B. The name binding cannot be changed, but the underlying object can be.

[Q73 Unanswered.](#)

Q74 Answered.

What is the difference between imperative programming and declarative programming?

Answer: Imperative programming focuses on how to do things, declarative programming focuses on what to do. Declarative programming may take advantage of built-in APIs to accomplish things, while imperative programming explicitly controls the path of the code execution.

Q74 Unanswered.

**Q75 Answered.**

What is the difference between a statement and an expression?

- A) statements perform an action, expressions evaluate to a value
- B) statements evaluate to a value, expressions perform an action
- C) Neither of the above is correct

The answer is A. Statements perform an action (i.e. a for loop), expressions evaluate to a value (i.e.  $5*5$ ).

**Q75 Unanswered.**

## Test 4 Answers

## **Q1 Answered.**

True or False? Evaluate the code below. bark() is now a function on the Dog class.

```
class Dog {  
    noise: string;  
    legs: number;  
}  
  
Dog.bark = () => {  
    console.log('woof');  
}  
  
let dog = new Dog();  
dog.bark();
```

- A) True
- B) False

The answer is B. Dog.prototype.bark is the correct syntax for adding bark to the Dog object.

[Q1 Unanswered .](#)

## Q2 Answered.

True or False? Evaluate the code below. bark() is now a function on the Puppy class.

```
class Dog {  
    noise: string;  
    legs: number;  
}  
  
Dog.prototype.bark = () => {  
    console.log('woof');  
}  
  
class Puppy extends Dog {  
    pottyTrained: boolean;  
}  
  
let dog = new Puppy();  
dog.bark();
```

- A) True
- B) False

The answer is A. When the Dog prototype had bark() added to it, anything extending Dog will also have bark. This is because the browser will look for bark on the Puppy prototype, and not finding it there, will look for bark on the Dog prototype.

## Q2 Unanswered.

### **Q3 Answered.**

Evaluate the following code. What is wrong with the getDOB function?  
TypeScript is enabled for this question.

```
const getDOB = (firstName: string, lastName: string) => {  
    return `${firstName} ${lastName}`;  
}
```

- A) getDOB has incorrect syntax in the string interpolation
- B) The typing of the parameters is wrong
- C) It returns undefined
- D) It is poorly named

The answer is D. Syntactically, there is nothing wrong with this function. However, it does something completely different than what the function name implies.

### [Q3 Unanswered.](#)

**Q4 Answered.**

Which of the following is the controller of custom elements?

- A) CustomHTML object
- B) HTMLElementRegistry object
- C) CustomController object
- D) CustomElementRegistry object

The answer is D.

[Q4 Unanswered.](#)

**Q5 Answered.**

What class should a custom element extend?

- A) DOMElement
- B) HTMLElement
- C) CustomElement

The answer is A.

[Q5 Unanswered.](#)

**Q6 Answered.**

What will the code below output to the console and why?

```
let x = 2.2 + 4.4 === 6.6;  
console.log(x);
```

- A) undefined
- B) True
- C) False
- D) Does not compile
- E) 6.6

The answer is C.  $2.2 + 4.4$  is not 6.6, but rather something like 6.60000000001 due to the fixed precision of numbers in JavaScript.

[Q6 Unanswered.](#)

## Q7 Answered.

What will the pre-es6 code below output to the console and why?  
TypeScript is NOT enabled.

```
function SomeClass() {  
    this.publicVar = 10;  
}  
SomeClass.staticVar = 10;  
  
const myClass = new SomeClass();  
myClass.publicVar = 12;  
myClass.staticVar = 15;  
  
const myClass2 = new SomeClass();  
console.log(myClass2.publicVar);  
console.log(myClass2.staticVar);  
console.log(SomeClass.staticVar);
```

- A) 10, undefined, 12
- B) 10, 12, 12
- C) 10, 15, 15
- D) 12, undefined, 10
- E) 10, undefined, 10

The answer is E. myClass only updates the publicVar for its particular instance of SomeClass. staticVar is not accessible from an instance. myClass.staticVar actually added an instance var named staticVar and set the value to 15.

## Q7 Unanswered.

## **Q8 Answered.**

Write a function that prints the distinct vowels in the below string using one line of code.

```
const str = "abcdeeef";
```

There may be multiple answers to this, but one possible answer is below:  
console.log(...new Set(str));

## **Q8 Unanswered.**

**Q9 Answered.**

What will the code below output to the console and why?

```
console.log(Math.floor(2.1) === new Float32Array(2.1));
```

- A) True
- B) False
- C) Compile Time Error

The answer is B.

[Q9 Unanswered.](#)

**Q10 Answered.**

What will the code below output to the console and why?

```
console.log(Math.sin(Math.PI) === 0);
```

- A) True
- B) False

The answer is B. This *should* be true, however Math.PI cannot be exactly equal to PI, because PI is infinite. Instead, Math.PI is an approximation. So Math.sin(Math.PI) is very close to zero, but not exactly zero.

[Q10 Unanswered.](#)

**Q11 Answered.**

What will the code below output to the console and why?

```
z = 10;  
console.log(z);
```

- A) 10
- B) Compile time error
- C) z

The answer actually depends on if strict mode is enabled or not. If strict mode is not enabled, z is a global variable initialized as 10, so the answer is A. If strict mode is on, it will not allow z to be initialized without the var, let, or const keyword, so the answer is B.

[Q11 Unanswered.](#)

**Q12 Answered.**

What will the code below output to the console and why?

```
let max;  
const numberArray = [3, 5, 2];  
numberArray.forEach((element) => {  
    if (max < element) {  
        max = element;  
    }  
});  
console.log(max);
```

- A) 3
- B) 5
- C) 2
- D) undefined

The answer is D. The variable max was not initialized but was used in an evaluation. The evaluation will not behave as expected.

[Q12 Unanswered.](#)

**Q13 Answered.**

Which bitwise operator inverts the bits of a 32 bit number?

- A) >>
- B) |
- C) &
- D) ~

The answer is D.

**Q13 Unanswered.**

**Q14 Answered.**

What operator can remove properties from an object?

- A) remove
- B) delete
- C) return
- D) configure

The answer is B.

**Q14 Unanswered.**

## Q15 Answered.

What is the proper syntax for accessing the getter below?

```
const obj = {  
    count: [1, 2, 3],  
    get last() {  
        return this.count.pop();  
    }  
};
```

- A) obj.get(last)
- B) obj.getLast
- C) obj.last

The answer is C.

## Q15 Unanswered.

**Q16 Answered.**

When might you use new.target?

When you want to throw a specific error when a class or certain functions are not called with the new keyword preceding the call.

**Q16 Unanswered.**

**Q17 Answered.**

What method would you use to convert a nodelist to an array?

- A) Array.to()
- B) Array.from()
- C) NodeList.to()
- D) NodeList.from()

The answer is B.

**Q17 Unanswered.**

**Q18 Answered.**

True or False? There are two kinds of NodeList: static and dynamic.

- A) True
- B) False

The answer is A. Dynamic nodeLists update as the DOM changes. Static is a static list.

[Q18 Unanswered.](#)

**Q19 Answered.**

Which object are Atomics not used with?

- A) Buffer
- B) SharedArrayBuffer
- C) ArrayBuffer

The answer is A.

[\*\*Q19 Unanswered.\*\*](#)

**Q20 Answered.**

Which JavaScript Error type is commonly used as a base for user-defined error objects?

- A) ReferenceError
- B) SyntaxError
- C) TypeError
- D) Error

The answer is D.

[\*\*Q20 Unanswered.\*\*](#)

**Q21 Answered.**

What property guarantees access to the global object regardless of environment?

- A) self
- B) globalThis
- C) this
- D) window

The answer is B.

[\*\*Q21 Unanswered.\*\*](#)

**Q22 Answered.**

Which of the following is NOT true about undefined?

- A) It is a property of the global object
- B) It is a variable in global scope
- C) The initial value of undefined is the primitive value undefined
- D) It has a corresponding object named Undefined

The answer is D.

**Q22 Unanswered.**

**Q23 Answered.**

True or false? Optional chaining can be stacked like below?

```
let stateStats = state.area?.land?.sqmiles;
```

- A) True
- B) False

The answer is A.

**Q23 Unanswered.**

## Q24 Answered.

What will the below code output to the console and why?

```
const a;  
a.test ??= 25;  
console.log(a.test);
```

- A) undefined
- B) 25
- C) Compile time error

The answer is C. The constant a was not initialized, therefore a.test will be flagged by the compiler.

## Q24 Unanswered.

**Q25 Answered.**

What is the technical name of parenthesis () in JavaScript?

- A) Grouping Operator
- B) Parenthesis Operator
- C) Binding Operator

The answer is A.

[\*\*Q25 Unanswered.\*\*](#)

**Q26 Answered.**

What does the below code output to the console and why?

```
let str = new String('aaa');
str.concat('bbb').concat('ccc');
console.log(str);
```

- A) aaa
- B) aaabbbccc
- C) ccc
- D) bbbccc

The answer is A. concat returns a new string, the original string is not affected.

[Q26 Unanswered.](#)

**Q27 Answered.**

What is the Queue in JavaScript Runtime?

The Queue is list of messages to be processed, and each message has an associated function to call.

Different than the stack because stack is a list of dependent functions, where the more recent must be executed in order to execute te older properly. The Queue is First In, First Out, meaning oldest messages are processed first.

[\*\*Q27 Unanswered.\*\*](#)

## **Q28 Answered.**

True or False? Vanilla JavaScript (no TypeScript) has abstract classes, as of ES2015.

- A) True
- B) False

The answer is B. The abstract modifier cannot be used for creating a class which cannot be instantiated.

## **Q28 Unanswered.**

## Q29 Answered.

What will the code below output to the console and why?

```
const b = {  
    a: 1,  
    b: 2,  
    3: 3  
}  
  
for(key in b){  
    console.log(b[key]);  
}
```

- A) 1 2 3
- B) 1 2
- C) Uncaught SyntaxError: Unexpected number

The answer is C. In JavaScript, keys must be strings. If you want to use a type other than string, use the Map object.

## Q29 Unanswered.

**Q30 Answered.**

What is the syntax for initializing a Map object with multiple key/value pairs?

Answer: Create a new Map object initialized with a two dimensional array of key/value pairs. For example: const map = new Map([['a',1], ['b',2], ['c',3]]);

[Q30 Unanswered.](#)

### **Q31 Answered.**

True or false? An interface can extend another interface. Assume TypeScript is enabled.

- A) True
- B) False

The answer is A. This is different than behavior in some programming languages (Java, for example). When an interface extends another interface, it has the properties of the parent interface, and the child interface simply adds more properties (or overwrites properties).

### **Q31 Unanswered.**

### Q32 Answered.

True or false? Assume TypeScript is enabled. An object inherits all the fields and methods of its parent class.

- A) True
- B) False

The answer is B. An object does not inherit fields and methods that have the “private” modifier.

### Q32 Unanswered.

**Q33 Answered.**

Which of the following are valid modifiers of TypeScript interface methods? Select all correct answers.

- A) public
- B) protected
- C) private
- D) abstract
- E) None of the above.

The answer is E. All interface methods are public and abstract in TypeScript interfaces, but actually putting the public or abstract modifier in the interface is incorrect syntax.

[\*\*Q33 Unanswered.\*\*](#)

**Q34 Answered.**

True or False? The static keyword is not a valid modifier in JavaScript. Rather, a typing superscript like TypeScript is required for *static* to work.

- A) True
- B) False

The answer is B.

[\*\*Q34 Unanswered.\*\*](#)

**Q35 Answered.**

What one-line code block will print the following characters:



One possible answer: `console.log(String.fromCodePoint(9731, 9000, 8000, 0x2FF));`

This creates a string from Unicode code points, which are an extensive collection of characters and symbols.

[Q35 Unanswered.](#)

**Q36 Answered.**

What, if anything, is in position 3 on the array?

```
const arr = [1, 2];  
arr.length = 3;
```

- A) It is empty
- B) It is not actually created
- C) null

The answer is A. It is an empty slot. `console.log(arr);` will print `[1,2,undefined];`

[\*\*Q36 Unanswered.\*\*](#)

**Q37 Answered.**

Write a one line example of how to reverse the array.reduce function.

Answer: There's actually a built in Array function called Array.reduceRight.

[Q37 Unanswered.](#)

**Q38 Answered.**

What are some limitations of arrow functions? Select all correct options.

- A) Arrow functions should not be used as methods
- B) Arrow functions do not have arguments, super, or new.target keywords
- C) Arrow functions cannot use yield keyword in it's body
- D) Arrow functions cannot be used as constructors

The answer is all of the above.

**Q38 Unanswered.**

**Q39 Answered.**

What is the difference in the default scope of traditional functions and arrow functions?

Answer: Traditional functions default this to the window scope. Arrow functions execute in the scope in which they are created.

**Q39 Unanswered.**

**Q40 Answered.**

Which of the following are generally true regarding access modifiers?

- A) Fields should have the most narrow access modifier possible
- B) Methods should have the widest access modifier possible
- C) Avoid using public fields unless the field is a constant

The answers are A, C. All members (both fields and methods) should have the most narrow access modifier possible. This keeps methods from being used in unintended ways and keeps fields from having their values changed unintentionally.

[Q40 Unanswered.](#)

**Q41 Answered.**

What kind of error will be thrown if you forget a parenthesis in your for loop?

- A) SyntaxError
- B) TypeError
- C) ReferenceError

The answer is A.

[\*\*Q41 Unanswered.\*\*](#)

**Q42 Answered.**

True or False? The below code is an example of overloading.

```
console.log({a: 1}.toString = 'Test');
```

- A) True
- B) False

The answer is B. The default `toString` method is *overridden* here, not overloaded.

[Q42 Unanswered.](#)

**Q43 Answered.**

True or False? JavaScript supports static variables on classes.

- A) True
- B) False

The answer is A. This is a recent addition to JavaScript and not all browsers support it as of 2020.

**Q43 Unanswered.**

**Q44 Answered.**

Give an example of using new ES6 syntax for creating a parent/child relationship.

```
let dog = { name: "buddy", legs: 4 };
let animal = { type: "mammal" };
Object.setPrototypeOf(dog, animal);
console.log(Object.getPrototypeOf(dog));
```

**Q44 Unanswered.**

#### **Q45 Answered.**

True or False? A try statement must always be followed by at least one catch statement.

- A) True
- B) False

The answer is B. A try statement may be followed by a finally statement instead of a catch statement. A finally statement always executes.

#### **Q45 Unanswered.**

#### **Q46 Answered.**

True or False? Use RegExp literal notation when the RegExp will be changing and use the RegExp constructor function when the RegExp will remain constant.

- A) True
- B) False

The answer is B. This is the opposite of how to use them: Use RegExp constructor function notation when the RegExp will be changing and use the RegExp literal notation when the RegExp will remain constant.

#### **Q46 Unanswered.**

## Q47 Answered.

True or False? Promises are asynchronous in JavaScript.

- A) True
- B) False

The answer is B. Promises are synchronous, however, they often do asynchronous things like making calls to an API. See this example below that shows they are synchronous. This prints 10, then 5.

```
function resolveQuickly(x) {
  return new Promise(resolve => {
    console.log(x)
  });
}

async function asyncTest() {
  var x = resolveQuickly(10);
  console.log(5);
}

asyncTest();
```

## Q47 Unanswered.

**Q48 Answered.**

What is the purpose of web workers?

- A) To execute expensive tasks on a separate thread than the main thread
- B) To offload events to different objects
- C) To speed up prototypical inheritance

The answer is A.

**Q48 Unanswered.**

**Q49 Answered.**

What type of object does Promise.resolve() return?

Answer: A promise in a resolved state with a given resolved value.

**Q49 Unanswered.**

**Q50 Answered.**

What is the difference between setTimeout() and setInterval()? Select two options.

- A) setTimeout executes once after a specified time has elapsed
- B) setInterval executes repeatedly with a given delay between executions
- C) setTimeout executes repeatedly with a given delay between executions
- D) setInterval executes once after a specified time has elapsed

The answers are A, B.

[\*\*Q50 Unanswered.\*\*](#)

**Q51 Answered.**

True or False? setInterval interval time includes the time it took to execute the code.

- A) True
- B) False

The answer is A. If the interval is set to 100ms and the code block takes 50ms to execute, there will only be a 50ms gap before the block kicks off again.

[\*\*Q51 Unanswered.\*\*](#)

**Q52 Answered.**

True or False? The below code compiles.

```
const array = [1, 2, 3];
```

```
for (let i = 0; i < array1.length; array1[i++] = String.fromCharCode(96 +  
i));
```

- A) True
- B) False

The answer is A. A for loop can have an empty statement after it.

**Q52 Unanswered.**

**Q53 Answered.**

What is the difference in importing a default export vs a named, non-default export?

- A) curly braces
- B) the name is never included in the default
- C) nothing

The answer is A. Default exports can still be named.

**Q53 Unanswered.**

**Q54 Answered.**

True or False? Function declarations are hoisted, function expressions are not hoisted.

- A) True
- B) False

The answer is A.

[\*\*Q54 Unanswered.\*\*](#)

**Q55 Answered.**

What statement will show the url from which a module was imported?

- A) import.path
- B) path.show
- C) path.directory
- D) import.meta

The answer is D. This will work if experimental support for import.meta is on.

[\*\*Q55 Unanswered.\*\*](#)

**Q56 Answered.**

What is the difference in console.log, console.error, and console.warn?

These three functions log differently in different browsers, but usually error has an 'error' icon or red color, warn has 'warn' icon or yellow color.

**Q56 Unanswered.**

[Q57 Answered.](#)

What is the significance of seeing prototype in the following documentation: `Array.prototype.reduce()` as opposed to: `Array.from()`? Select all correct answers.

- A) prototype indicates reduce is an instance level method
- B) the lack of prototype indicates from is a class level method
- C) prototype indicates reduce is inherited from Object

The answers are A, B.

[Q57 Unanswered.](#)

**Q58 Answered.**

How does the Maximum Call Stack Exceeded error get triggered?

Answer: Usually a function is recursively calling itself, so the "Stack" of function calls waiting to execute grows until a maximum is reached.

**Q58 Unanswered.**

**Q59 Answered.**

Which of the following is not a method available on the Web API window object for resizing?

- A) resizeFrom()
- B) resizeBy()
- C) resizeTo()
- D) sizeToContent()

The answer is A.

[\*\*Q59 Unanswered.\*\*](#)

**Q60 Answered.**

What two types can String.prototype.replace() take as the first parameter?  
Select all correct options.

- A) String
- B) regex
- C) Object

The answer is A, B. However, if another primitive is passed, it will be converted to a string.

[\*\*Q60 Unanswered.\*\*](#)

**Q61 Answered.**

What will the code below output to the console and why?

```
for(let i = 6; i < 10; i*1.5){  
    if(i%2){  
        continue;  
    }  
    console.log(i)  
}
```

- A) 6
- B) 6, 9
- C) It enters into an infinite loop
- D) Lint error

The answer is C.  $i * 1.5$  doesn't change the value of  $i$ . Rather,  $i *= 1.5$  would properly update the value.

**Q61 Unanswered.**

**Q62 Answered.**

Which of the following is a difference between Object.assign() and the spread syntax?

- A) Spread syntax calls setters, Object.assign() does not
- B) Object.assign calls setters, spread syntax does not
- C) They actually behave identically

The answer is B.

**Q62 Unanswered.**

**Q63 Answered.**

Give an example of a one line check to see if a property exists on object *dog* .

```
const dog = { sound: 'woof', legs: 4 }
```

One possible answer:

```
let propExists = 'sound' in woof;
```

**Q63 Unanswered.**

**Q64 Answered.**

Which of the following is not a way to convert the arguments function object to an array?

- A) arguments.assign(arguments)
- B) Array.prototype.slice.call(arguments);
- C) Array.from(arguments);
- D) [...arguments]

The answer is A.

[\*\*Q64 Unanswered.\*\*](#)

**Q65 Answered.**

What is the target phase of event handling?

The target phase is the second of the three phases of event handling. The first is capture, the third is bubbling. The target phase is simply the event reaching the target.

**Q65 Unanswered.**

**Q66 Answered.**

What does the double bang (!! ) do?

- A) nothing
- B) returns the truthy value of an object
- C) throws an error

The answer is B. For example, !!0 returns false.

[\*\*Q66 Unanswered.\*\*](#)

**Q67 Answered.**

- A) true
- B) false

The answer is A.  $1 < 2$  returns true,  $true < 3$  returns true.

**Q67 Unanswered.**

**Q68 Answered.**

How does JavaScript evaluate the truthy-ness or falsey-ness of a value or object in a boolean context?

- A) type coercion
- B) it's a property on each object
- C) some values never have truthy or falsey values

The answer is A.

**Q68 Unanswered.**

**Q69 Answered.**

What is the purpose of using an IIFE? Select all possible answers.

- A) setting a variable value
- B) initializing values on an object
- C) use in a closure for initializing a value
- D) running a one-time function

The answers are A, C, D. However, one could probably even use an IIFE to accomplish B if so desired.

[Q69 Unanswered.](#)

**Q70 Answered.**

What is an example of lexical scope? Select all possible answers.

- A) scope being defined dynamically based on variable input
- B) let self = this; //in a parent function, then self is available to an inner function
- C) the scope of an inner function contains the scope of the parent function

The answers are B, C.

**Q70 Unanswered.**

**Q71 Answered.**

What will the code below output to the console and why?

```
let a = 3;  
a <=> 1;  
console.log(a);
```

- A) 3
- B) 4
- C) 6
- D) 8

The answer is C. The left shift operator moves the bits to the left by the specified amount and assigns the value. In this case, 3 (11 in binary) is moved to 110 in binary, which is a value of 6.

[Q71 Unanswered.](#)

## Q72 Answered.

What will the code below output to the console and why?

```
badCode = () => {
let arr = [1,2,3,4];
const inner = () => {
return arr.pop(2)
}
    return inner;
}

console.log(badCode()());
```

- A) linting error
- B) 3, 4
- C) 34
- D) 4

The answer is D. *pop* doesn't accept any parameters, but in vanilla JavaScript it simply discards any passed parameters.

## Q72 Unanswered.

**Q73 Answered.**

What will the code below output to the console and why?

```
const obj = { a: 1 };
let obj2 = obj;
obj2.a = 2;
console.log(obj.a);
```

- A) 1
- B) 2
- C) undefined

The answer is B.

[Q73 Unanswered.](#)

**Q74 Answered.**

What is a side effect in programming?

Answer: A side effect is when a function modifies a state or observable value that affects other parts of the application. This is as opposed to the main effect of the function, which is the value the function returns.

**Q74 Unanswered.**

**Q75 Answered.**

True or False? `Array.prototype.filter()` returns an array of all items that did NOT pass the test implemented by the provided function.

- A) True
- B) False

The answer is B. The opposite is correct, filter returns the items that did pass the test implemented by the provided function.

**Q75 Unanswered.**