10 Most Asked ES6 Interview Questions & Answers For Developers

Difficulty Level : Easy • Last Updated : 27 May, 2021

When JavaScript came into the picture of the programming world, the name was chosen for marketing reasons. At that time Java was getting popular all around the world (but you better know both are different). Later it was submitted to ECMA (European Computer Manufacturers Association) to standardize the language and its specification.

Later it was named ECMAScript of ES. The first edition was released in June 1997. ES6 is the sixth edition of the language, and it was later renamed ECMAScript 2015. The edition introduced many new features such as class, modules, iterators, for/of loop, arrow functions, typed arrays, promises, reflection, etc.

that you should prepare for the interviews. These questions will be helpful in clearing the interviews specially for the frontend development role.

1. What are the Common Features of ES6?

Below are the common features of ES6…

- Support for constants/immutable variables.
- Block scope support for both variables, constants, functions.
- Arrow functions
- Extended parameter Handling
- Default parameters
- Template literals and extended literals.
- Destructuring Assignment
- Arrow functions
- Promises
- Classes
- Modules
- Support for Map/Set & WeakMap/WeakSet
- Promises, localization, meta-programming, internationalization

Let and const both are blocks scoped. If you declare a variable using these keywords then it only exists within the innermost block that surrounds them. Suppose you declare a variable using let inside a block(if a condition or for-loop) then it will be accessed only within a block.

Below is an example of the let keyword…

Javascript

```javascript
if(true) {
    let a=0;
    console.log(a); //prints 0;
}

console.log(a); throws ReferenceError: a is not defined.
```

Const is immutable in JavaScript. You can not change or reassign its value once it is declared.

Javascript

```javascript
const a=0;
a=1; // TypeError: Assignment to constatnt variable.
const b= [1, 2];
b.push(3); //[1, 2, 3]
b[3]= 4; // [1, 2, 3, 4]
```

Make a practice to use let and const instead of var keyword.

3. What is the Arrow function? What is the difference between a normal function and an arrow function?

In ES6 arrow function is defined to define the function and use it. Basically, it's a shorthand

Javascript

```
function add(a, b) {
    return a+b;
};

//Implementation with arrow function

const add = (a, b) =>a+b;

// With single argument, no parenthesis required

const add5 = a => 5+a;
```

Let’s discuss some differences between the arrow functions and the normal function.

- They don’t have their own versions, and they close over this.
- They can have a concise body instead of a verbose one. (They can also have a verbose body)
- You can not use them as constructors. You’re not allowed to use new with an arrow function. This simply means an arrow function can not have a prototype property on them.
- There isn’t generator syntax for the arrow function. For example: There is no arrow equivalent to function *foo() {…}.

4. What is set?

Set is the collection of new values. In Set, there shouldn’t be any duplicate value. All the values should be unique. These values can be primitive types or object references.

Javascript

```
var mySet = new Set();

mySet.add(1); // Set [1]
```

Read Discuss Courses Practice Video

This is a new feature in ES6. Generator function allows you to generate many values over time, returning an object. We can iterate this object, and we can pull values from the function one value at a time. When you call a generator function it returns an iterable object. We use * sign for a generator function with a new ‘yield’ keyword in ES6.

Javascript

```
function *infiniteNumbers() {
    let n=1;
    while(true) {
        yield n++;
    }
}

const numbers = infiniteNumbers(); // returns an iterable object

numbers.next(); // { value: 1, done: false}
numbers.next(); // { value: 2, done: false}
numbers.next(); // { value: 3, done: false}
```

6. What is the spread operator in ES6?

Spread operator is used to obtain the list of parameters. It is represented by three dots (…). Basically spread operator takes an iterable (such as array or string) and expands it into individual elements. In JavaScript, it is mainly used to make shallow copies of JS. It makes your code concise and increases the readability of your code.

You can use the spread operator to combine or to perform the concatenation between arrays. An example is given below…

Javascript

```
let num1 = [40,50,60];
```

7. What is Destructuring in ES6?

In ES6 destructuring was introduced to extract data from arrays and objects into an individual variable. It allows you to extract the smaller fragment from objects and arrays. An example is given below…

Javascript

```javascript
let fullname =['Alan','Rickman'];
let [fname,lname] = fullname;
console.log (fname,lname);
```

8. Define Map in ES6.

Before the ES6 was introduced we were using an object to map the keys and values. Map becomes a new way in ES6 to represent the data in key-value pairs. Map is ordered, and it remembers the insertion order of the keys. You can traverse the elements in their insertion order.

Below is the representation of Map

```
var map = new Map([iterable]);
```

9. Explain Promises in ES6.

In JavaScript, there is a concept of asynchronous programming. In asynchronous programming, you run the processes individually from the main thread. In ES6 promises are the easiest way to deal with asynchronous programming. A promise can either be rejected or resolved depending on the operation outcome. Before promises were introduced in ES6, callbacks were used to deal with asynchronous programming.

But it created the problem of callback hell and to overcome this problem promises were

JavaScript callback is helpful in working with events. We pass a function into another function as an argument to another function.

When we use callback in our web application, a lot of times happens that callback becomes nested. Excessive use of callbacks makes your web application messy and results in callback hell.

Conclusion

We have included 10 important questions of ES6 which are really important for interview purposes. Apart from these questions, there are some other concepts important in ES6. For example…

- Modules in JavaScript
- Hoisting in JavaScript
- Concept of Babel
- Concept of Webpack
- Concept of Weakset
- Concept of Weakmap
- Template literals

To Learn JavaScript, please refer to **JavaScript Tutorial**

♡ Like 4

Next ›

Read Discuss Courses Practice Video

Related Articles

1. 7 Most Asked ReactJS Interview Questions & Answers
2. Top 10 Most Commonly Asked Web3 Interview Questions and Answers
3. 15 Commonly Asked Blockchain Interview Questions & Answers
4. Django Interview Questions & Answers With Practical Tips For Junior Developers
5. Top 50 Most Asked Google Tricky Interview Questions
6. Commonly asked JavaScript Interview Questions | Set 1
7. SQL Interview Questions asked in Top Tech Companies
8. JavaScript Interview Questions and Answers
9. JavaScript Interview Questions and Answers | Set 3
10. JavaScript Interview Questions and Answers | Set-2

Article Contributed By :

anuupadhyay
@anuupadhyay

Read

Discuss

Courses

Practice

Video

Article Tags : GBlog, JavaScript, Web Technologies

Improve Article

Report Issue

GeeksforGeeks

A-143, 9th Floor, Sovereign Corporate Tower,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Company

About Us

Careers

In Media

Contact Us

Privacy Policy

Copyright Policy

Advertise with us

Learn

DSA

Algorithms

Data Structures

SDE Cheat Sheet

Machine learning

CS Subjects

Video Tutorials

Courses

News

Top News

Technology

Work & Career

Business

Finance

Languages

Python

Java

CPP

Golang

Web Tutorials	Write an Article
Django Tutorial	Improve an Article
HTML	Pick Topics to Write
JavaScript	Write Interview Experience
Bootstrap	Internships
ReactJS	Video Internship
NodeJS	