MindMajix

Home > Blog > General >

ES6 Interview Questions

Do you want to become a web developer or a **JavaScript developer?** Are you in process of interview preparation? Do you want guidance on the type of **JavaScript ES6 Interview Questions** that could be asked? Then spare a moment and check out the interview questions we have compiled for you. You will find here the ES6 Interview Questions that have a high chance of being asked.

Rating: 4.5 ★★★★½ 237

GET TRAINED AND CERTIFIED

JavaScript ES6 is nothing but the 6th version of JavaScript introduced in 2015. It is also called ECMAScript 6 or ECMAScript 2015. The standard used by JavaScript is ECMAScript only. ECMAScript specifies the way of working of the general-purpose programming language, JavaScript. It is the first major update after the ES5 standardized in 2009. ES6 overcomes several limitations of the core language with its new syntax and features including template strings, arrow fiction, and many more. Therefore, it makes coding easier for the developers because more work can be done with less code.

Let us have some insights on ES6 before moving further-

1. ES6 is used by around 3295 companies in their tech stacks. These include StackShare, Slack, Discord, Glovo, KAVAK, Tech Stack, HENNGE K.K, and Hepsiburada.
2. A JavaScript developer with knowledge of ES6 can earn an average salary of $104,270 per annum in the USA. And with experience of just 1-2 years, one can earn around $111,018 per annum.

Let us move towards **ES6 Interview Questions and Answers – 2022 (updated)** one by one for each of the following-

Top 10 ES6 Interview Questions and Answers

ES6 Interview Questions and Answers for Freshers:

1. Differentiate between ES5 and ES6.

ES5	ES6
This is the 5th edition standardized in 2009. All the web browsers have successfully implemented ES5.	This is the 6th edition standardized in 2015. Most web browsers have partially implemented ES6.
It is mandatory to use the function keyword for defining functions.	We don’t need the function keyword for defining the function.
It is also mandatory to use the return keyword for defining functions.	Again, we don’t need the return keyword for defining functions.
It shows relatively low performance.	It shows relatively higher performance.

2. What are the reasons for using arrow functions in ES6?

1. Compactness
2. Safety of scope
3. Clarity

3. What are the advantages of spread syntax in ES6?

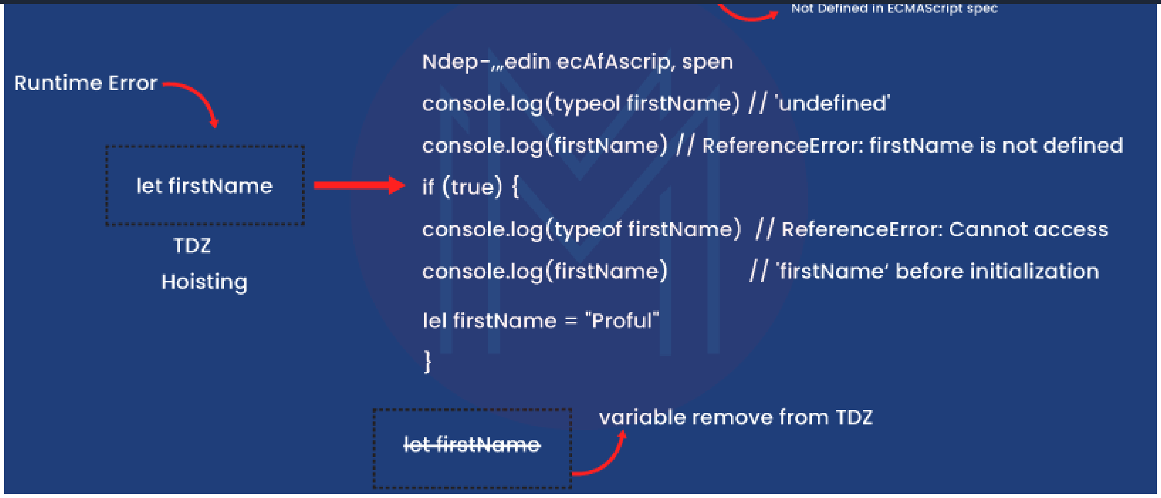
The spread syntax is highly beneficial in ES6 for coding in a functional paradigm. We can use the spread syntax for creating copies of objects and arrays easily. We don't require slice, Object.create, or any library function while using spread syntax. It has wide use in rx.js and Redux projects.

4. How are .apply and .call different from each other?

.call	.apply
We use this for invoking the functions. Its value within the function is derived from the first parameter.	We use this for invoking the functions as well but after taking an array of arguments. This array performs as the next argument and the function performs this way.
This calls the method taking the object as its argument.	This is used for writing several methods able to perform on several objects.
The parameters accepted by it are arguments and objectInstance.	The parameters accepted by it are arrayofArguments and objectInstance.
Several arguments are taken by it separated by commas.	Several arguments are taken by it using an array of arguments.

5. What is meant by Temporal Dead Zone in ES6?

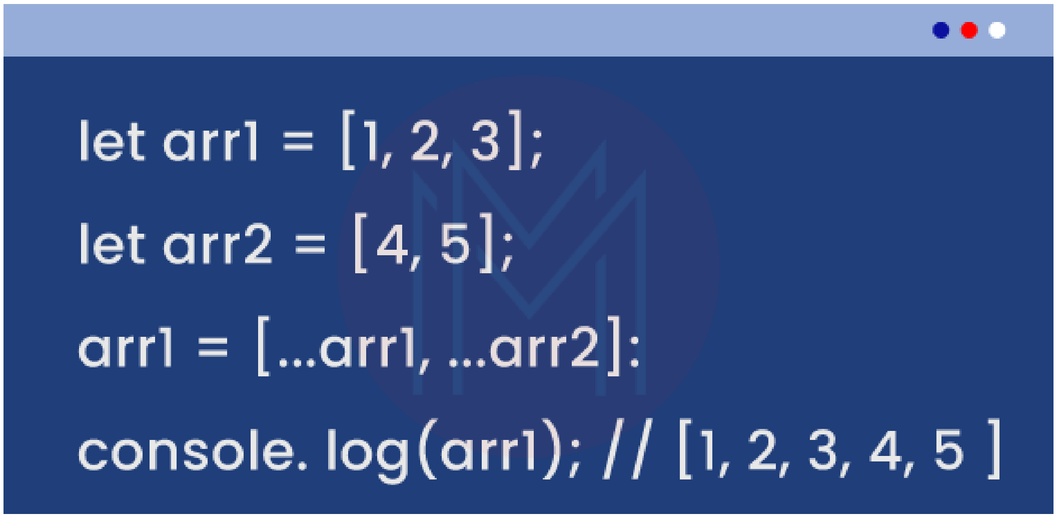
The temporal Dead Zone is the period existing between the variable binding and the declaration in the program. The purpose of its existence is to identify program errors as a variable that we can access before the declaration. Also, it can be used to make const work properly. We can also use it for proof guarding JavaScript.



```
Ndep-„,edin ecAfAscrip, spen
console.log(typeol firstName) // 'undefined'
console.log(firstName) // ReferenceError: firstName is not defined
if (true) {
console.log(typeof firstName)  // ReferenceError: Cannot access
console.log(firstName)          // 'firstName' before initialization
lel firstName = "Proful"
}
```

6. What is a spread operator?

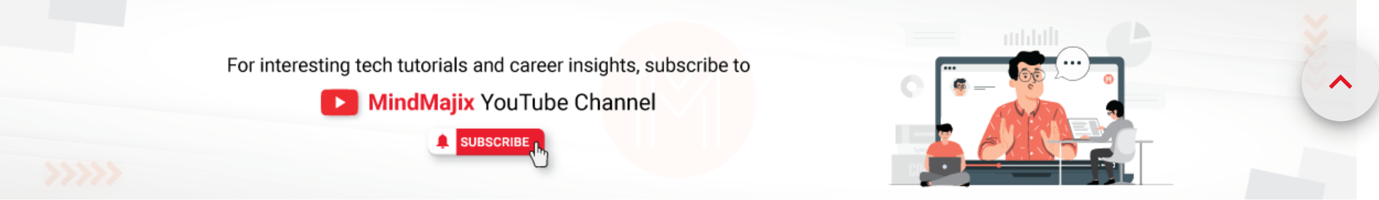
The spread operator is denoted by '…' and the variable follows it. For instance, its syntax looks like- '…Y'. We can manipulate arrays and objects using the spread operator which is the main reason for using it in ES6. We can use it to copy one object's property to another.



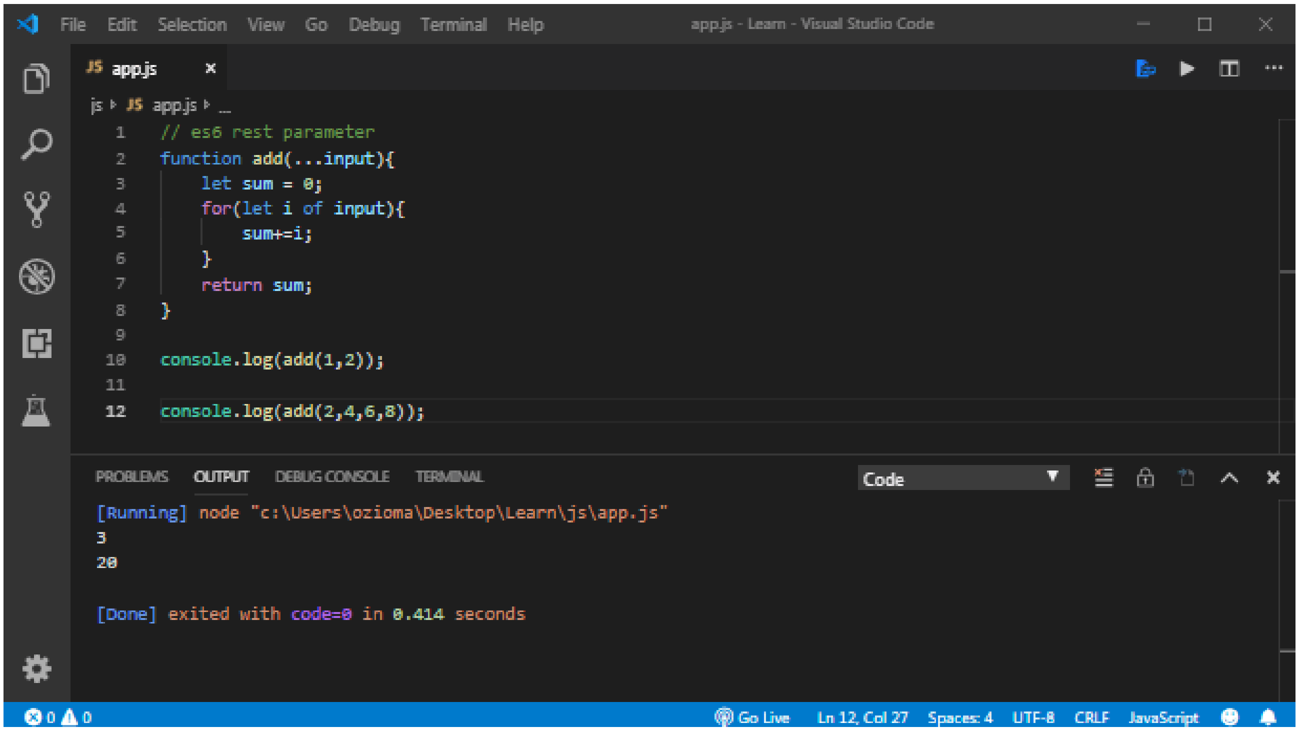
```
let arr1 = [1, 2, 3];
let arr2 = [4, 5];
arr1 = [...arr1, ...arr2]:
console. log(arr1); // [1, 2, 3, 4, 5 ]
```

7. What is meant by default operator?

The default operator is used for initializing a function by default values. The parameter can take any value- a function or a number or even null.



All the arguments needed for invoking a function can be recovered using this operator. Hence we can separate the items category-wise with the help of the rest parameter. We can combine the parameters in a common array parameter.


File Edit Selection View Go Debug Terminal Help
app.js - Learn - Visual Studio Code
app.js
js ▸ app.js ▸ ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
⊗ 0 ⚠ 0
Go Live Ln 12, Col 27 Spaces: 4 UTF-8 CRLF JavaScript

```
// es6 rest parameter
function add(...input){
    let sum = 0;
    for(let i of input){
        sum+=i;
    }
    return sum;
}

console.log(add(1,2));

console.log(add(2,4,6,8));
```

```
[Running] node "c:\Users\ozioma\Desktop\Learn\js\app.js"
3
20

[Done] exited with code=0 in 0.414 seconds
```

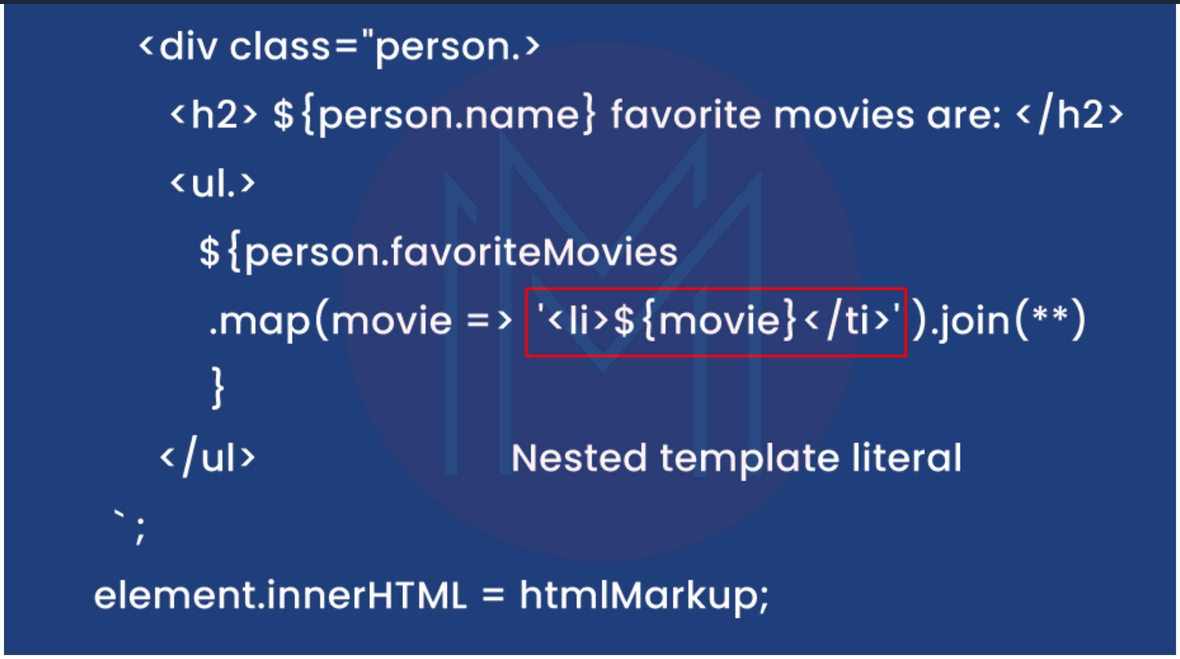
9. What are proxies and classes?

Proxies- Objects can be created using proxies and they can be hosted with a wide range of behavior. Profiling, as well as logging, is also helped by proxies.

Classes- patterns can be easily created using class declaration based on OOP (Object-Oriented Programming). It supports base class access and works with constructors, static methods, and inheritance.

10. What is the purpose of template literals in ES6?

Template literals are the strings with code embedded and having variables inside. We can interpolate and concatenate using template literals more comprehensively than the previous versions of ES.



```
    <div class="person.>
      <h2> ${person.name} favorite movies are: </h2>
      <ul.>
        ${person.favoriteMovies
         .map(movie => '<li>${movie}</ti>').join(**)
         }
      </ul>
  `;
element.innerHTML = htmlMarkup;
```

Nested template literal

11. What do you mean by Babel?

The most popular transpiler of JavaScript and the standard in the industry is Babel only. We can write our code in ES6 and then it can be converted back into the browser supported pre-ES6 JavaScript.

12. How do you install Babel?

We need node.js and NPM for installing Babel. Therefore we need to ensure that node.js is already installed on the server.

We can check the availability of nodes on our server by running the given commands-

```
node –v
npm -v
```

Next, babel can be installed locally by the given commands-

```
npm install –save-dev babel-cli
```

13. Differentiate between const and object.freeze.

- Object.freeze- It performs its functions on values (object values particularly). But the object becomes immutable and its properties can't be changed.

14. What do you mean by IIFEs?

IIFE (Immediately Invoked Function Expression) gets executed immediately after its creation. We use this pattern often for avoiding the pollution of the global namespace. That is because all the variables we use inside IIFE are invisible outside its scope.

```
(function IIFE(){
         console.log( “Hello!” );
})();
// “Hello!”
```

15. What new string methods were introduced in ES6?

The following string methods are introduced in ES6-

- String.startsWith()
- String.endsWith()
- String.includes()
- String.repeat()

ES6 Interview Questions and Answers for Experienced

1. List the advantages of Arrow Function.

The advantages of the arrow function are given below-

a. The code size is reduced.
b. There is the optionality of the return statement for the one-line function.
c. The context is bound lexically.
d. There is the optionality of functional braces for a one-line statement.

2. Explain the de-structuring assignment in ES6.

Example-

```
let fullname = [‘Ron’,’Weasely’];
let[fname,lname] = fullname;
console.log(fname,lname);
```

Output-

```
Ron Weasely
```

3. What is meant by Map in ES6?

We used to map values and keys with an object before the introduction of ES6. Now ES6 allows the representation of data in key-value pairs by using Map. The insertion order is remembered by Map. And the elements can be traversed in the order of insertion.

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

4. Explain for...of the loop with an example.

This kind of loop is used to iterate the iterables (string, array, etc.).

Example-

```
var fruits = [‘Grapes’, ‘Mango’, Melon’];
for(letvalue of fruits)
{
  console.log(value);
}
```

Output-

```
Grapes
Mango
```

5. What is meant by Weakset?

The Weakset allows us to store the weakly held objects in a group or collection. Duplicate values can't be stored in Weakset just like a set. We can’t iterate the Weakset.

Only the has(value), add(value), and delete(value) methods of object set are included in the Weakset.

6. What is meant by Weakmap?

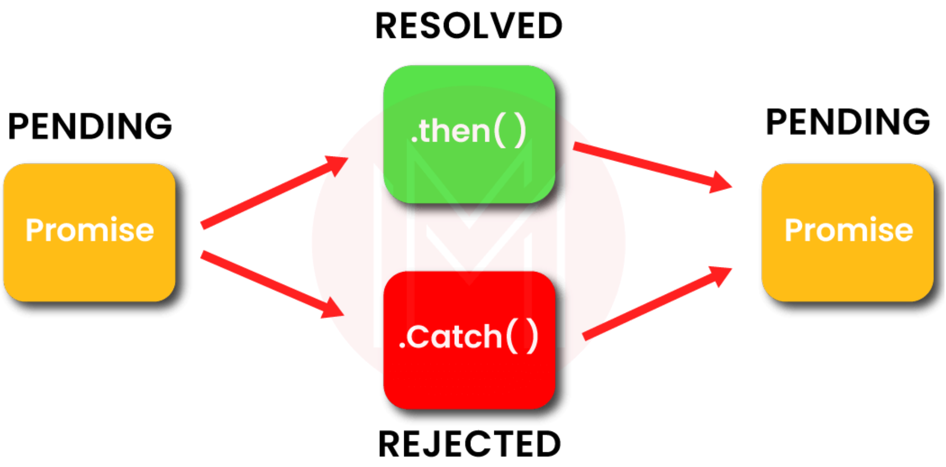
Weak maps are mostly like maps, but the weak map keys should be the objects. Each element is stored as a key-value pair where the weakly references are keys. Here, the objects include keys and arbitrary include values.

The elements are iterated by a weak map in their insertion order. Only the has(key), get(key), set(key), and delete(key) methods are included.

7. What is meant by Promises in ES6?

The easiest way of working in JavaScript with Asynchronous programming is using ES6 promises. The individual running of processes from the main thread is included in Asynchronous programming and the main thread is notified when it gets completed.

Earlier, the Callbacks were used to perform the task. Now, the promises overcome the Callback hell problem.



8. Mention the 3 states of promises in ES6.

- Pending: This is the beginning stage of every promise. It signifies the pending result.
- Fulfilled: It signifies that the operation has been completed.
- Rejected: it signifies if any failure has occurred during the computation.

9. What is meant by Callback and Callback hell?

Callback- The execution of a function is handled by it after another function has been executed. It helps while working with events. We can pass a function to another as an argument in the callback. It is a great way to deal with basic cases like minimal asynchronous operations.

Callback hell- working with callback becomes messy when a web application including a lot of code is developed. This massive callback nesting is called callback hell.

10. List some new array methods that were introduced in ES6.

The main array methods introduced in ES6 are-

- Array.from()
- Array.of()
- Array.prototype.find()
- Array.prototype.copyWithin()
- Array.prototype.findIndex()
- Array.prototype.keys()
- Array.prototype.entries()
- Array.prototype.values()
- Array.prototype.fill()

11. Give an example spread operator in ES6.

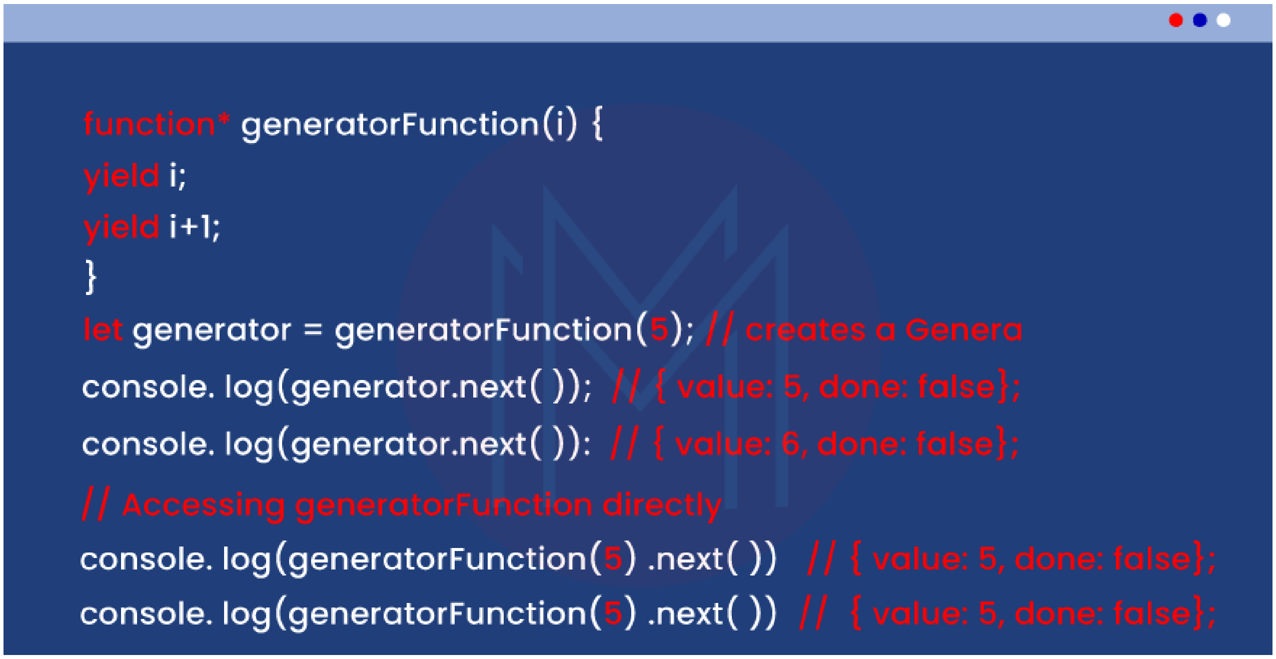
```
let num1 = [40,50,60];
let num2 = [10,20,30,…num1,70,80,90,100];
console.log(num2);
```

Output

```
[
     10, 20, 30, 40, 50,
     60, 70, 80, 90, 100
]
```

A new way of working with functions and iterators is provided by a generator. It is a different type of function that can pause once or multiple times but will resume later. The declaration function* defines the generator function.

The code doesn't run when we call the generator. A special object- Generator object is returned for managing the execution.



```
function* generatorFunction(i) {
yield i;
yield i+1;
}
let generator = generatorFunction(5); // creates a Genera
console. log(generator.next( )); // { value: 5, done: false};
console. log(generator.next( )): // { value: 6, done: false};
// Accessing generatorFunction directly
console. log(generatorFunction(5) .next( ))  // { value: 5, done: false};
console. log(generatorFunction(5) .next( )) //  { value: 5, done: false};
```

13. What are modules in JavaScript?

Module refers to the JavaScript code piece that is written in a file. It becomes easy to maintain code, debug code, and reuse code with the help of modules. Each module contains a piece of code that is executed when we load it.

Preparing for JavaScript Interview? Here are Top JavaScript Interview Questions and Answers

14. When shouldn't you use arrow functions?

We shouldn't use the arrow functions in the given cases-

a. Object methods
b. Function hoisting, named functions
c. This/argument
d. Callback functions with dynamic context

The export statement arises by the import statement when exporting objects, functions, and variables to the JavaScript modules is needed.

Named export- It is used for exporting multiple values. The imported module and exported module should match each other.

Default export- It is used for exporting a single module such as a class, function, object, etc. here, the names of imports are completely anonymous.

Frequently Asked Questions and Answers on ES6 Interview:

1. Name some popular tools that have integration with ES6.

The following are some of the popular tools that can integrate with ES6-

a. Apache OpenWhis
b. Gatsby
c. Backpack
d. Draggable JS
e. Luxon
f. Trails
g. Esbuild
h. ImageBoss

2. List down the new features of ES6.

The new features of ES6 are listed below-

- Arrow functions
- Support for immutable variables or constants
- Extended handling of parameter
- Both constant and variable functions are supported
- De-structuring assignment
- Classes, modules, iterators, generators
- The regular expression is enhanced
- Extended literals and template literals
- Object properties are enhanced
- Meta-programming, promises, localization, and internationalization

3. How can the code be migrated from ES5 to ES6?

ES5 is a subset of ES6. Therefore nothing specific has to be done. The ES5 code is present automatically in ES6. The increment adoption of the newer version becomes extremely convenient for this reason.

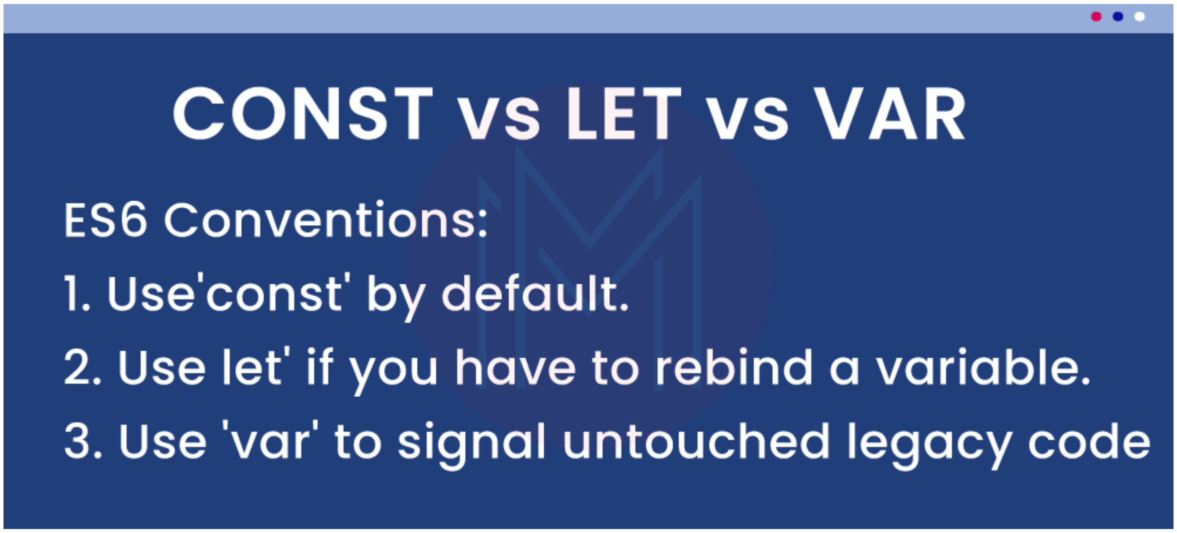
4. Mention some alternatives to ES6.

a. CoffeeScript
b. jQuesry
c. Python
d. HTML
e. PHP
f. TypeScript
g. JAVA

5. What are let and const in ES6? And how do they differ from var?

We used the var keyword to declare a variable in JavaScript. Var keyword falls under the function scoped category. The variables can also be accessed. Thus the code was wrapped in a function whenever a new scope had to be created.

Both let and const fall under the blocks scoped category. A variable exists within only the innermost block surrounding the keywords if a variable is declared. For instance, a variable declared inside a block using let inside can be accessed within the block only.



CONST vs LET vs VAR

ES6 Conventions:

1. Use'const' by default.

2. Use let' if you have to rebind a variable.

3. Use 'var' to signal untouched legacy code

6. Give an example of let keyword.

```
        let  a=0;
        console.log(a); //prints  0;
}
console.log(a); throws that ReferenceError: a is not defined.
```

7. What is meant by set?

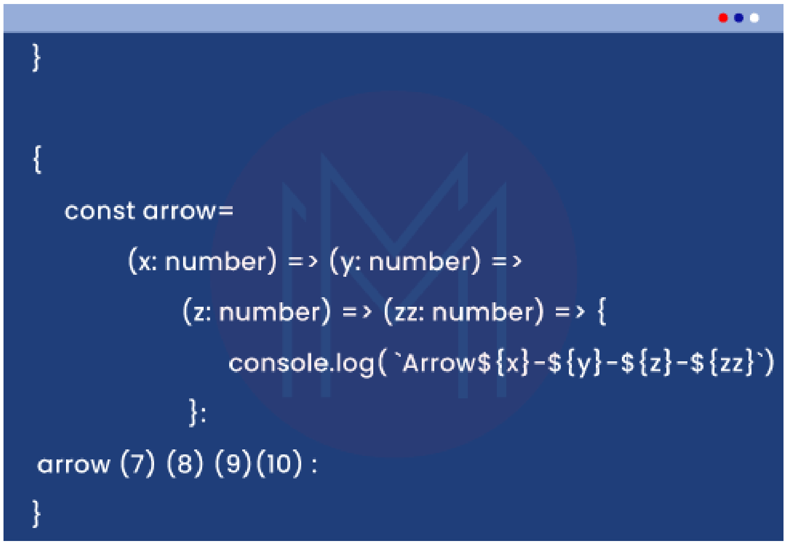
The set collects all the new values. Any duplicate value shouldn’t be there in Set. They all should be unique. But the values can be object references as well as primitive types.

```
Var mySet = new Set () :

mySet.add(1); // Set [1]
mySet.add(5); // Set [1,5]
mySet.add(5); // Set [1,5] --ignored
```

8. What is meant by the Arrow function?

ES6 arrow function is used for defining and using a function. It is Arrow functions’ shorthand notation. The parameter list (....) can be passed to the arrow function. And a => marker, as well as function body, follows it. Parentheses are not needed when the arrow function is declared with a single argument.



```
}


{
    const arrow=
          (x: number) => (y: number) =>
                (z: number) => (zz: number) => {
                      console.log( `Arrow${x}-${y}-${z}-${zz}`)
                };
 arrow (7) (8) (9)(10) :
}
```

9. How do arrow functions differ from normal functions?

- They can't be used as constructors. Using new with the arrow function is not allowed. This implies that there is no prototype property of the arrow function.
- The arrow function doesn't have generator syntax.

10. List the object-oriented features available in WS6.

The following object-oriented features are available in ES6-

a. Classes
b. Methods
c. Inheritance

Conclusion

The main purpose of ES6 is to reduce the complex processes of software development. The resulting code is modern as well as easily readable. It provides incredible performance with limited consumption of memory. This makes ES6 highly favorable among the developers in every industry. Hence the opportunities for these developers have increased by leaps and bounds. And the ES6 Interview Questions provided here intend to take you closer to these opportunities only.

Course Schedule

Name	Dates	
JavaScript Training	Dec 17 to Jan 01	VIEW DETAILS
JavaScript Training	Dec 20 to Jan 04	VIEW DETAILS
JavaScript Training	Dec 24 to Jan 08	VIEW DETAILS
JavaScript Training	Dec 27 to Jan 11	VIEW DETAILS

Last updated: 14 December 2022

About Author

SaiKumar Kalla

Kalla Saikumar is a technology expert and is currently working as a content associate at MindMajix. Write articles on multiple platforms such as ServiceNow, Business Analysis,...

Read More

MindMajix

Kotlin Training

30 hours 163

★★★★★ 4.8

12 /15

EXPLORE COURSES

Start a conversation ...

Privacy - Terms

Terms of use - Privacy - Report a bug

powered by GraphComment