[1. Explain the difference between ES5 and ES6?](https://www.bestinterviewquestion.com/question/explain-the-difference-between-es5-and-es6-fpee34865pb)

**Answer**

| **ES5** | **ES6** |
| --- | --- |
| ES5 is the 5th edition of ECMAScript which was standardized in 2009. This version has been successfully implemented in all web browsers. | ES6 is the 6th edition of ECMAScript which was standardized in 2015. This version has been partially implemented in most web browsers. |
| The function keyword is mandatory to use to define the functions. | In this, to define the function, the function keyword is not needed. |
| The return keyword is also mandatory to use to define the functions | The return keyword is not needed to define the functions. |
| ES5 has lower performance due to the absence of a few features. | ES6 has a comparatively higher performance. |

Is this helpful?  Yes  No

 9  0

[2. What do you mean by IIFEs?](https://www.bestinterviewquestion.com/question/what-do-you-mean-by-iifes-immediately-invoked-function-expressions-wh89c4866la)

**Answer**

IIFEs or Immediately Invoked Function Expressions are the expressions which are immediately executed as soon as it is created. IIFEs are used to avoid the excessive use of global namespace because all the variables which are used inside the IIFE cannot be viewed outside its scope. In other words, IIFEs are design patterns also known as Self Executing Anonymous Functions because it executes on its own.

Is this helpful?  Yes  No

 8  0

[3. Why are arrow functions used in ES6?](https://www.bestinterviewquestion.com/question/why-are-arrow-functions-used-in-es6-ka0ii4867lf)

**Answer**

Arrow functions are one of the major reasons for the popularity of ES6 due to the reasons mentioned below:

* Safety of scope: Consistent use of arrow functions guarantee to use the same thisObject as the root. Using the arrow functions provides scope safety because even if a single standard function callback is mixed with some arrow functions, the probability of scope getting disturb increases.
* Compactness: It is easier to read and write when it comes to arrow functions as compared to other functions. This provides a less complex approach of using ES6 as well as the arrow functions.
* Provides clarity: A developer can easily identify what “ thisObject” is and find out the next-higher function statement after the arrow function statement if he/she uses arrow function in most of the program. It will provide huge clarity and easy detection of errors in any program.

Is this helpful?  Yes  No

 7  0

[4. Explain the motivation behind Symbols to ES6.](https://www.bestinterviewquestion.com/question/explain-the-motivation-behind-symbols-to-es6-nugho4868c6)

**Answer**

Symbols are a particular kind of object, newly added in the sixth version of ECMAScript which can be used as a unique property name in the objects. Using Symbol in the place of String (as it was done in the previous versions) allows the programmer to use different modules that can create properties that will not be problematic to each other. Another huge benefit of Symbols is that they can be made private and the properties defined cannot be accessed by any user or developer who does not have direct access to the Symbol. They comprise of a function that can be used to create different symbols that can work differently. But they do not have any literal syntax, unlike the other primitives of JavaScript.

Is this helpful?  Yes  No

 4  4

[5. Explain the benefits of using spread syntax in ES6. How is it different from the rest of the syntax?](https://www.bestinterviewquestion.com/question/explain-the-benefits-of-using-spread-syntax-in-es6-how-is-it-different-from-the-rest-of-the-syntax-msd574869dq)

**Answer**

As far as coding in a functional paradigm is concerned, using the spread syntax in ES6 can prove to be heavily beneficial. The spread syntax can be easily used to create copies of arrays or objects. Using spread syntax in a program does not need to resort to Object.create, slice or any library function. The spread syntax is widely used in Redux and rx.js projects.

While on the other hand, the rest syntax feature of ES6 is used to pass an arbitrary number of arguments to a function. It can also be considered as a shorthand in JavaScript programming. Rest syntax can be considered as the opposite of spread syntax. It takes data and stuffs it into an array instead of unpacking the data. In addition to that, it works suitably in function arguments, arrays as well as to object destructuring assignments.

*Note: This question is asked in almost all ES6 Interview Questions.*

Is this helpful?  Yes  No

 5  1

[6. Differentiate between ES6 class and ES5 function constructors.](https://www.bestinterviewquestion.com/question/differentiate-between-es6-class-and-es5-function-constructors-sdkha4870nc)

**Answer**

| **ES6 class** | **ES5 Function Constructors** |
| --- | --- |
| ES6 class basically does the work of defining a new object and appending functions to its prototype. | ES5 Function constructors work and look the same but the main difference is observed when the developer uses the Inheritance property. |
| They can be considered as the syntax base for constructor functions. | These can only be executed with the help of a new operator. |
| ES6 class allows the developers to instantiate objects using the new operator. | ES5 function constructors focus on how the objects are instantiated. |
| They also ensure the developer that this keyword which is basically used inside the class only refers to the object that is being created by the developer. | ES5 function constructor focus on implementing the reusable object creation code. Any function can be used as a constructor. |

Is this helpful?  Yes  No

 7  1

[7. Differentiate between .call and .apply.](https://www.bestinterviewquestion.com/question/differentiate-between-call-and-apply-xeajj4871pd)

**Answer**

| **.call** | **.apply** |
| --- | --- |
| .call is used to invoke the functions, and the first parameter will become the value of this within the function created by the developer. | .apply is also used to invoke the functions but it takes an array of arguments. This array works as the next argument and the function works this way. |
| .call is used to call the method which takes the owner object as its argument. | .apply method is used to write methods. These methods can be used on several objects. |
| It accepts two parameters namely objectInstance and arguments | It accepts two parameters namely objectInstance and arrayofArguments |
| .call takes several arguments separated by commas. | .apply takes several arguments with the help of an array of arguments. |

Is this helpful?  Yes  No

 7  1

[8. Explain the reason for using ES6 classes.](https://www.bestinterviewquestion.com/question/explain-the-reason-for-using-es6-classes-nr4sk4872hb)

**Answer**

ES6 classes have been found to be very useful to the developers. Some of the major uses of ES6 classes are as follows:

* ES6 classes have a simpler and less error-prone syntax.
* As far as setting up inheritance hierarchies are concerned, ES6 is considered as the best option as it uses new syntax with the old syntax which minimizes the errors and eases the process.
* ES6 classes help in defending the developers from failing to use new ones properly with the constructor function. This is one of the most common errors that occur with the developers during the use of a new operator. Classes remove this error by having the constructor throw an exception if this proves to be an invalid object for the constructor.
* Classes also help to call the method which is of the prototype’s version. This version is much simpler with the new ES6 syntax than the old versions.

Is this helpful?  Yes  No

 5  0

[9. What is a generator in JavaScript?](https://www.bestinterviewquestion.com/question/what-is-a-generator-in-javascript-6yn3x4873qm)

**Answer**

With the launch of ES6, a new way of working with functions and iterators came into work. These are called generators. A generator is nothing but a function that can stop at the point of time when the user commands it to do so and then continue working just from the point where it stopped. In other words, a generator has properties of a function but also works as an iterator. It does not work like other lines of code which need to be completed for execution.

Generators work on a run-to-completion model which makes it easier for the developers to compile a program. Generators can also be defined as a special class of functions that work to simplify the task of coding iterators in a program. It produces a sequence of results rather than producing a single value. It enables the developer to generate a series of values at a time.

*This is very important javascript ES6 Interview Questions.*

Is this helpful?  Yes  No

 6  0

[10. Define Temporal Dead Zone in ES6.](https://www.bestinterviewquestion.com/question/define-temporal-dead-zone-in-es6-hncuf4874ay)

**Answer**

The temporal dead zone can be defined as the time span which exists between the time taken for binding of a variable and its declaration in a program. The temporal dead zone exists so that program errors can be easily identified as a variable that can be made accessible before its declaration. Also for making const work properly, the temporal dead zone can be used. It is also used to proof guard JavaScript which is basically a mechanism used for runtime.

Is this helpful?  Yes  No

 2  0

[11. What is spread, default and rest parameter?](https://www.bestinterviewquestion.com/question/what-is-spread-default-and-rest-parameter-kkbb64875on)

**Answer**

Spread Operator- Donated by ‘...’ and is followed by the variable. The syntax of the spread operator would look like this for example:- ‘...X’. Spread operators manipulated objects and arrays which is the prime reason it is used in ES6. It is used for copying the property of one object to another.

Default Operator- In order to initialize a function using default values, the default operator is deployed. The parameter’s value can be anything- a number or a function or null.

Rest Parameter- This operator is used for recovering all the arguments that are required to invoke a function. This allows us to put items belonging to different categories separate. The rest parameter allows combining the parameters in a common array parameter.

Is this helpful?  Yes  No

 3  0

[12. What is Webpack ?](https://www.bestinterviewquestion.com/question/what-is-webpack-rafk74876bi)

**Answer**

Webpack allows you to create a pile of the javascript files that are usable in the browser. With Webpack, we can build the dependency graph, do application processing. With the dependency graph, one can map every module of the project. This helps in matching up with project requirements.

Is this helpful?  Yes  No

 3  0

[13. What is the advantage of using Webpack?](https://www.bestinterviewquestion.com/question/what-is-the-advantage-of-using-webpack-q8p4i4877u4)

**Answer**

The advantage of a web pack is explained through its description above. It helps in making a bundle of the project modules into a single javascript file. Webpack combined the Dev server which helps in updating the code and asset management.

Is this helpful?  Yes  No

 3  0

[14. What are the classes and proxies?](https://www.bestinterviewquestion.com/question/what-are-the-classes-and-proxies-jtwoc4878lf)

**Answer**

Class- You can use the patterns easily with OOP based class declaration. It works readily with constructors, supports base class access, inheritance and static methods. OOP refers to Object-Oriented Programming.

Proxies- With proxies, you get to create objects and you can host the objects with a huge behavior diversity. Proxies can help in profiling and logging as well.

Is this helpful?  Yes  No

 2  1

[15. What are constants?](https://www.bestinterviewquestion.com/question/what-are-constants-4cfyi4879iy)

**Answer**

You can call the immutable variables as constants. The value of a constant once defined remains the same and we cannot change it. This is the value that you introduce during the declaration. For example, const Y= 6.0. This value of Y would continue to be 6 throughout and it won’t be changed.