

Searching by Amin Orai

Advanced JavaScript Interview Questions And Answers

Here are 13 advanced [JavaScript interview questions](#), including why interviewers may ask them and ways to answer:

1. How would you explain functional programming?

The interviewer may want to determine whether you are sufficiently passionate about programming and have studied all of its facets. A correct response typically includes background information on functional programming, an explanation of how it relates to JavaScript and examples of functional programming languages and JavaScript features that support them.

Example: *"Functional programming is a fundamental Java Script concept that refers to the process of developing programmes through the application and composition of mathematical functions. Since the 1950s, it has been used to create programming languages, including Lisp, Haskell, Clojure and Machine Learning. Additionally, there are many JavaScript features that support functional programming, including higher-order functions, arguments or values as functions and first-class functions."*

Related: [How To Learn JavaScript \(With Advantages And Functions\)](#)

2. Explain the JavaScript event delegation model

This question aims to gauge your understanding of the event delegation model and how it works.

Example: *"There is some cool stuff in JavaScript that makes it the best of all. The delegation model is one of them. When capturing and bubbling enable functions to implement a single handler for multiple elements at the same time, we refer to this as event delegation. Instead of specifying nodes, event delegation enables you to add event listeners to a single parent. This listener analyses bubbled events to locate a match on the child elements. Many people believe it is complex, but it is actually quite simple once one understands it."*

3. Explain arrow functions

Searching by Amin Orai

This function generally relates to writing function expressions. Explain its use case in your answer with an example.

Example: *"An arrow function is a concise way to write function expressions. Arrow functions are not constructors and do not support the keywords this, arguments, super or new.target. It is best suited for functions that are not methods. An arrow function looks something like this:"*

```
<em>const  
hello=<span><span>()</span>=></span>{<span>console</span>.log(<span>'good  
morning'</span>);}</em>  
hello();</em>
```

4. Explain NaN and its role

When an interviewer asks to define a term, the term is usually something commonly used. In this case, the term is NaN, which stands for "not a number," an important concept that JS developers may require understanding when working with numerical values.

Example: *"When a value in an operation is not a number, it returns NaN. It may occur in a few instances. For example, if an operation can return a suitable output because a portion of the function was non-numeric or the result has a value that is not numeric."*

5. What are the primitive data types in JavaScript?

Such questions usually assess your theoretical knowledge of JavaScript. While answering this question, consider listing all the primitive data types available in JavaScript concisely.

Example: *"A primitive data type is a fundamental data type that is not constructed from other data types. It is limited to representing a single value. By definition, all primitives are built-in data types and the compiler must know them. But not all built-in data types are primitives. There are five primitive data types available in JavaScript:"*

Searching by Amin Orai

- `Undefined`
- `Null`
- `Boolean`
- `String`
- `Number`

Everything else in Javascript is an object."

Related: [JavaScript Vs. jQuery: What Is The Difference? \(With FAQs\)](#)

6. How do you calculate Fibonacci numbers in JavaScript?

Creating Fibonacci numbers is a very popular question for programmers. Try to answer it correctly with an efficient code example.

Example: *"Fibonacci number sequence is a numbers sequence where each value is the sum of the previous two, starting with 0 and 1. The first seven values are 0, 1, 1, 2, 3, 5, 8. This JavaScript function will return Fibonacci numbers sequence up to a given numbers of terms."*

```
// ask for input from the user
const number = 
parseInt(prompt('Enter the no. of terms: '));
let n1 = 0, n2 = 1, nextNumber;

console.log('Fibonacci Sequence:');

for (let i = 1; i <=
number; i++) {
  console.log(n1);
  nextTerm = n1 + n2;
  n1 = n2;
  n2 = nextNumber;
}
```

Searching by Amin Orai

7. How to add and remove object properties dynamically in Javascript?

This question may test both your theoretical and practical JavaScript knowledge. You can use an example code to explain.

Example: *"You can add a property to an object using `object.property_name=value` and delete a property using `object.property_name`. Here is an example showing the same:"*

```
<span><em>let</em></span><em> person = <span>new</span>
<span>Object</span>();
<span>// adding a property</span>
person.name=<span>'Rohan'</span>;
user.age=<span>35</span>;
<span>console</span>.log(person);
<span>delete</span> person.age;
<span>console</span>.log(person);
</em>
```

Related: [Top 50 JavaScript Interview Questions and Answers](#)

8. Explain the difference between `Object.freeze()` vs `const`

This explains your flexibility detailed understanding when it comes to handling variables and functions.

Example: *"const and Object.freeze are two different functions in JavaScript. const applies to bindings variables. It creates an immutable binding, which means it is not possible to assign a new value to the binding."*

```
<em>const <span>human</span> = {
name: <span>"Rakesh"</span>
};
<span>let</span> <span>animal</span> = {
species: <span>"Monkey"</span>
};
<span>human</span> = animal; // ERROR <span>"person"</span> is read-
only</em>
```

Searching by Amin Orai

Object.freeze works on object values. It makes an object immutable, which means changing its properties is not possible.

```
<span><em>let</em></span><em> human = {  
name: <span>"Rakesh"</span>  
};  
<span>let</span> animal = {  
species: <span>"Monkey"</span>  
};  
Object.freeze(person);  
human.name = <span>"Rohan"</span>; //TypeError: Cannot assign  
<span>to</span> read <span>only</span> property <span>'name'</span>  
<span>of</span> object  
console.log(person);  
</em>
```

9. Explain JavaScript cookies

This question can help the interviewer gauge your basic knowledge of working in web development. Include its role with an example use of cookies in your answer.

Example: *"Cookies are the tiny text files that are properly stored in a computer and they develop when the user goes to the websites to store some information that they require. Examples are username details and information about the shopping cart content from earlier visits."*

Related: [15 Java 8 Interview Questions \(With Example Answers\)](#)

10. Explain the role of deferred scripts in JavaScript

Performance is key to any project. Such questions can help the interviewer find your understanding of the features of well-coded programs.

Example: *"The parsing for HTML code while the page is loading waits by default until the scripts have paused executing. If your server is a little slow or the script is specifically heavy, then your webpage can take more time to load. When deferred is used, scripts delay the execution till the time the HTML parser is running. This lessens the apparent loading time of web pages by allowing the starting part of the page to load fast."*

Searching by Amin Orai

Related: [8 Next.js Interview Questions \(With Sample Answers And Tips\)](#)

11. What is the use of break and continue statements in JavaScript?

These two terminologies relate to the looping aspect of programming. Consider explaining how the break and continue statement affect the loop differently.

Example: *"The break statement jumps out from a running loop, whereas the continue statement allows the loop to keep running after breaking just one iteration. Both statements allow writing a complex loop function with various potential results and required actions based on a given function outcome."*

12. Why wrap the content of a JS source file in a function block?

This process is becoming popular in several JS libraries you might use as a developer. Answering this question can show your knowledge of ancillary processes that might be library-specific.

Example: *"This technique encloses the contents of a file creating a private namespace. This implementation of a namespace avoids disharmony between JavaScript objects and libraries they are housed in."*

Related: [15 Software Developer Skills And How To Develop Them](#)

13. What are the advantages and disadvantages of monolithic and microservice architectures?

The interviewer is probably looking to see if you understand the distinctions between the two and are knowledgeable enough about them to determine the most practical approach in any given situation.

Example: *"Monolithic architectures are typically implemented as a single application, which simplifies the addition of many cross-cutting concerns, such as rate-limiting, logging and various security features. While they may appear to be more practical at first, as an application develops, it becomes more difficult to maintain the code or scale it."*

Searching by Amin Orai

Whereas microservices are likely to have a wide range of cross-cutting concerns that were not anticipated during the design phase. In the long run, they are typically preferred due to their autonomous organisation, which makes them easy to restructure as you scale up or wish to change their purpose."