**JavaScript Interview Questions**

1. Please tell me what equality means in JavaScript.

2. Can you explain what IIFEs refer to in JavaScript?

3. Please explain what the bind() function does in JavaScript.

4. Could you tell us the difference between const and Object.freeze() in JavaScript?

5. Please explain what generators are in JavaScript.

6. Can you explain what hoisting is in JavaScript?

7. Please explain what the prototype design pattern does in JavaScript.

8. Could you tell me what the temporal dead zone is in ES6?

9. What is the primary difference between map() and forEach()?

10. Please explain the difference between null and undefined variables.

11. Can you tell me the difference between WeakMap and ES6 maps?

12. Please explain whether JavaScript is a pass-by-value or pass-by-reference language.

13. Please tell me your best advanced JavaScript soft skill.

14. Please tell us your best advanced technical JavaScript skill.

15. Could you tell us what closures do in JavaScript?

16. Can you tell us what the close() function does in JavaScript?

17. Please tell us what imports are in JavaScript.

18. Please tell us what exports are in JavaScript.

19. Can you explain the use of strict JavaScript directives?

20. Please explain what typecasting means in JavaScript.

21. Please tell us the main primitive data types in JavaScript.

22. Could you tell us what you know about the different pop-up boxes in JavaScript?

23. Can you explain what event bubbling means in JavaScript?

24. Please explain the key differences between the ES5 function and ES6 class constructors.

**1. Please tell me what equality means in JavaScript.**

In javascript, equality can be tested in two main ways

**Loose equality “==”**

Using double equals “==” operator, which checks for value equality with type coercion.

The “==” operator check for equality after converting both values to a common type. This means it perform type coercion which can lead to unexpected results.

**Strict equality “===”**

Using triple equals “===” operator which checks for strict equality without type coercion.

The “===” operator checks for equality without type coercion. This means the values must be the same type and have the same value.

2. Can you explain what IIFEs refer to in JavaScript?

3. Please explain what the bind() function does in JavaScript.

4. Could you tell us the difference between const and Object.freeze() in JavaScript?

|  |  |  |
| --- | --- | --- |
| **Feature** | “const” | “Object.freeze()” |
| **Declaration** | Keyword used to declare variables | Method used to freeze objects |
| **Reassignment** | Not allowed; once assigned, cannot change | Not applicable; no effect on variable assignment |
| **Object Mutability** | Object properties can be changed | Object properties cannot be changed |
| **Array Mutability** | Array elements can be changed | Array elements cannot be changed |
| **Scope** | Block scoped | Not applicable; affects object itself |
| **Nested object/Array mutability** | Nested objects/array can be changed. | Nested object/array can still be changed/shallow freeze |
| **Use case** | Ensuring a variable cannot be reassigned | Ensuring object properties cannot be changed |
| **Effect on primitive values** | None beyond assignment immutability | Not applicable; affects objects and array only |
| **Syntax example** | `const x=10;` | `Object.freeze(obj);` |
| **Error on modification** | Throw error on reassignment | Fails silently / throws in strict mode |
| **Common use** | constants that should not be reassigned | Immutable configuration/ settings |

5. Please explain what generators are in JavaScript.

Generators in javascript are a special type of function that can pause execution and resume at a later point. They are useful for working with sequences of data, such as iterating over collections, handling asynchronous programming, or managing large data sets that need to be processed incrementally.

**Key concepts of generators**

* **Generator function syntax**

Defined using the “function\*” syntax

Use the “yield” keyword to pause execution and produce a value.

* Generator object

When a generator function is called, it returns a generator object

The generator object conforms to both the iterable and iterator protocols

* Iteration and control

The generator object has a “next()” method, which resumes the generator function execution and returns object with “value” and “done” properties.

“value” is the value yielded by the generator.

“done” is a Boolean indicating if the generator has completed execution.

Uses of generator

* Lazy evaluation

Generators can produce values on demand, making them useful for dealing with large / potentially infinite sequences.

* Asynchronous programming

Generators can be combined with “promises” to handle asynchronous operations more elegantly.

* Complex iteration logic

They allow encapsulating complex iteration logic inside a generator function, making the code cleaner and more maintainable.

6. Can you explain what hoisting is in JavaScript?

7. Please explain what the prototype design pattern does in JavaScript.

8. Could you tell me what the temporal dead zone is in ES6?

9. What is the primary difference between map() and forEach()?

10. Please explain the difference between null and undefined variables.

11. Can you tell me the difference between WeakMap and ES6 maps?

12. Please explain whether JavaScript is a pass-by-value or pass-by-reference language.

13. Please tell me your best advanced JavaScript soft skill.

14. Please tell us your best advanced technical JavaScript skill.

15. Could you tell us what closures do in JavaScript?

16. Can you tell us what the close() function does in JavaScript?

17. Please tell us what imports are in JavaScript.

18. Please tell us what exports are in JavaScript.

19. Can you explain the use of strict JavaScript directives?

20. Please explain what typecasting means in JavaScript.

21. Please tell us the main primitive data types in JavaScript.

22. Could you tell us what you know about the different pop-up boxes in JavaScript?

23. Can you explain what event bubbling means in JavaScript?

24. Please explain the key differences between the ES5 function and ES6 class constructors.