

JavaScript Interview Questions & Answers for Playwright Automation Testing

1. What is JavaScript?

JavaScript is a high-level, interpreted programming language used primarily for web development to add interactivity to web pages.

Example:

```
console.log("Hello, JavaScript!");
```

2. Difference between var, let, and const?

`var` is function-scoped, `let` and `const` are block-scoped. `const` variables cannot be reassigned.

Example:

```
var a = 10;  
let b = 20;  
const c = 30;
```

3. What is hoisting?

Hoisting is JavaScript's default behavior of moving declarations to the top of the current scope before code execution.

Example:

```
console.log(a); // undefined  
var a = 5;
```

4. What are data types in JavaScript?

JavaScript supports Primitive (String, Number, Boolean, null, undefined, Symbol, BigInt) and Reference (Object, Array, Function) types.

Example:

```
let str = "Hello";  
let num = 10;  
let obj = { key: "value" };
```

5. How do you declare a constant in JS?

Constants are block-scoped variables that cannot be reassigned after declaration.

Example:

```
const PI = 3.14;
```

6. How to check data type of a variable?

Use `typeof` operator to determine the type of a variable.

Example:

```
typeof "hello"; // "string"
```

7. What is the difference between null and undefined?

`undefined` means a variable has been declared but not assigned, while `null` is an assignment value that represents no value.

Example:

```
let a;  
let b = null;
```

8. What is type coercion?

Type coercion is JavaScript's automatic or implicit conversion of values from one data type to another.

Example:

```
'5' + 1; // "51"  
'5' - 1; // 4
```

9. How to convert a string to a number?

Use `Number()` or `parseInt()` to convert a string to a number.

Example:

```
let n = Number("123"); // 123
```

10. How to convert a number to a string?

Use `String()` or `.toString()` to convert a number to a string.

Example:

```
let str = String(123); // "123"
```

11. What is the difference between == and ===?

`==` checks for equality with type coercion, while `===` checks for strict equality (without coercion).

Example:

```
5 == "5"; // true
5 === "5"; // false
```

12. Logical operators in JS?

Logical operators include `&&` (AND), `||` (OR), and `!` (NOT).

Example:

```
true && false; // false
true || false; // true
!true; // false
```

13. What is the ternary operator?

A shorthand for `if...else`. It has the syntax `condition ? expr1 : expr2`.

Example:

```
let result = (age >= 18) ? "Adult" : "Minor";
```

14. What are falsy values in JS?

Falsy values are `false`, `0`, ``, `null`, `undefined`, and `NaN` — values that evaluate to false in a Boolean context.

Example:

```
if (!false) console.log("Falsy");
```

15. Write an if/else condition in JS

Conditional statements are used to perform actions based on different conditions.

Example:

```
if (score > 50) {  
  console.log("Pass");  
} else {  
  console.log("Fail");  
}
```

16. Switch-case example

Used to execute one block of code among many based on matching a value.

Example:

```
switch(day) {  
  case "Mon": console.log("Work"); break;  
  default: console.log("Rest");  
}
```

17. What is the typeof NaN?

`NaN` stands for 'Not-a-Number' but its type is actually 'number'.

Example:

```
typeof NaN; // "number"
```

18. What is short-circuit evaluation?

Logical expressions are evaluated from left to right and may stop evaluating once the result is determined.

Example:

```
false && doSomething(); // doSomething() not called
```

19. Difference between prefix and postfix increment?

Prefix increments before the value is used, postfix increments after.

Example:

```
let x = 1;
console.log(++x); // 2
console.log(x++); // 2 (then x = 3)
```

20. How to check if a variable is an array?

Use `Array.isArray()` method to check if a value is an array.

Example:

```
Array.isArray([1, 2]); // true
```

21. Function declaration vs expression?

A function declaration defines a named function and is hoisted. A function expression can be anonymous and is not hoisted.

Example:

```
function greet() {}
const greet2 = function() {};
```

22. Arrow function example

Arrow functions provide a concise syntax and do not bind their own `this` context.

Example:

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

23. What is a callback function?

A callback is a function passed into another function to be executed later.

Example:

```
setTimeout(() => console.log("Hello"), 1000);
```

24. What is closure in JavaScript?

A closure gives access to an outer function's scope from an inner function.

Example:

```
function outer() {  
  let count = 0;  
  return function inner() {  
    return ++count;  
  }  
}
```

25. What is lexical scope?

Lexical scope means the scope of a variable is defined by its position in the source code.

Example:

```
function outer() {  
  let a = 10;  
  function inner() {  
    console.log(a);  
  }  
  inner();  
}
```

26. What is IIFE?

Immediately Invoked Function Expression executes right after it's defined.

Example:

```
(function() {  
  console.log("IIFE");  
})();
```

27. Difference between arguments and parameters?

Parameters are variables in a function definition; arguments are actual values passed.

Example:

```
function add(a, b) {}  
add(2, 3);
```

28. What is the 'this' keyword?

`this` refers to the object from which the function was called.

Example:

```
const obj = {  
  name: "JS",  
  greet() { console.log(this.name); }  
};
```

29. What is the use of `bind`, `call`, and `apply`?

They are used to set the `this` context of a function.

Example:

```
function greet() { console.log(this.name); }  
const user = { name: "Alice" };  
greet.call(user);
```

30. Difference between function and arrow function?

Arrow functions don't have their own `this`, `arguments`, or `prototype`.

Example:

```
const obj = {  
  value: 10,  
  regular: function() { return this.value; },  
  arrow: () => this.value  
};
```

31. How do you create an array in JavaScript?

An array is a special type of object used to store ordered collections.

Example:

```
let arr = [1, 2, 3];
```

32. What is the use of push and pop in arrays?

`push` adds an element to the end; `pop` removes the last element.

Example:

```
arr.push(4);  
arr.pop();
```

33. What is the use of shift and unshift?

`shift` removes the first element; `unshift` adds an element to the beginning.

Example:

```
arr.unshift(0);  
arr.shift();
```

34. How to loop through an array using forEach?

The `forEach()` method executes a function for each array element.

Example:

```
arr.forEach(el => console.log(el));
```

35. What does map() do?

`map()` creates a new array with the results of calling a function on every element.

Example:

```
arr.map(x => x * 2);
```

36. What does filter() do?

`filter()` returns a new array with all elements that pass a test.

Example:

```
arr.filter(x => x > 1);
```

37. What does reduce() do?

`reduce()` executes a reducer function on each element and returns a single value.

Example:

```
arr.reduce((a, b) => a + b);
```

38. How to find an element in an array?

`find()` returns the value of the first element that satisfies the condition.

Example:

```
arr.find(x => x === 2);
```

39. How to check if a value exists in an array?

`includes()` checks if an array contains a specified value.

Example:

`arr.includes(2);`

40. How to sort an array in JavaScript?

`sort()` sorts the elements of an array in place.

Example:

`arr.sort((a, b) => a - b);`