



Top 10



What are the different data types present in JavaScript?

JavaScript has seven primitive data types: String, Number, Boolean, Null, Undefined, Symbol, and BigInt. It also includes one complex data type: Object.



Explain the difference between null and undefined.

undefined means a variable has been declared but not yet assigned a value. null is an assignment value that represents no value or no object.



What is the purpose of the use strict directive in JavaScript?

The use strict directive defines that JavaScript code should be executed in "strict mode." This mode helps catch common coding errors and "unsafe" actions, such as assigning values to undeclared variables.



What are Immediately Invoked Function Expressions (IIFE)?

An IIFE is a JavaScript function that runs as soon as it is defined. It is used to avoid polluting the global namespace by creating a scope for variables.

```
(function() {  
    // Code here  
})();
```



Explain the concept of hoisting in JavaScript.

Hoisting is JavaScript's default behavior of moving declarations to the top of the current scope. This means that variables and function declarations are moved to the top of their scope before code execution.



What is event bubbling in JavaScript?

Event bubbling is a type of event propagation where the event starts from the target element and bubbles up to the outer elements. It means that if an event occurs on an element, it will trigger the event handlers on that element and then on its parent elements.



What is the difference between function declarations and function expressions?

Function Declaration: A function, declared as a separate statement, is hoisted.

```
function foo() { }
```

Function Expression: A function assigned to a variable, is not hoisted

```
const foo = function() { }
```



What are template literals in JavaScript?

Template literals are string literals allowing embedded expressions. They are enclosed by backticks (```) instead of single or double quotes and can contain placeholders denoted by `${expression}`.

```
const name = 'John';  
const greeting = `Hello, ${name}!`;
```



What is the difference between call, apply, and bind?

call: Invokes a function with a given this value and arguments provided individually.

```
func.call(thisArg, arg1, arg2, ...);
```

apply: Invokes a function with a given this value and arguments provided as an array.

```
func.apply(thisArg, [argsArray]);
```

bind: Returns a new function with a given this value and initial arguments.

```
const boundFunc = func.bind(thisArg, arg1, arg2, ...);
```



What is the difference between let, const, and var?

- **var:** Function-scoped, can be redeclared and updated, hoisted to the top of its scope.
- **let:** Block-scoped, can be updated but not redeclared within the same scope, not hoisted.
- **const:** Block-scoped, cannot be updated or redeclared, requires an initializer at declaration, not hoisted.



