

Interview Question

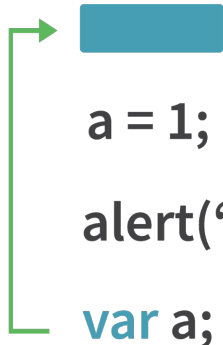
Q1. Explain Hoisting in javascript? (Uber)

Ans.

Hoisting is the default behaviour of javascript, where all the variable and function declarations are moved on top.

Both functions and variables are hoisted in javascript.

Declaration
moves to top



```
a = 1;  
alert(' a = ' + a);  
var a;
```

```
hoistedVariable = 3;  
console.log(hoistedVariable); // outputs 3 even when the variable is declared after it is initialized  
var hoistedVariable;
```

Q2. What is the role of a strict mode in JavaScript ? (Twitter)

Ans.

The JavaScript strict mode is used to generate silent errors. It provides a "use strict"; expression to enable the strict mode. This expression can only be placed as the first statement in a script or a function.

```
"use strict";  
x=10;  
console.log(x);
```

Q3. What is the requirement of debugging in JavaScript? (eBay)

Ans.

JavaScript didn't show any error message in a browser. However, these mistakes can affect the output. The best practice to find out the error is to debug the code. The code can be debugged easily by using web browsers like Google Chrome, Mozilla Firefox.

To perform debugging, we can use any of the following approaches:

- Using console.log() method
- Using debugger keyword

Q4. What is the use of the debugger keyword in JavaScript? (Airbnb)

Ans.

JavaScript debugger keyword sets the breakpoint through the code itself. The debugger stops the execution of the program at the position it is applied. Now, we can start the flow of execution manually. If an exception occurs, the execution will stop again on that particular line.

```
function display( )  
{  
  x = 10;  
  y = 15;  
  z = x + y;  
  debugger;  
  console.log(z) ;  
  console.log(a) ;  
}  
display( ) ;
```

- ❖ In the above example, the **variable “a”** is not declared anywhere, so it throws an error, but using the debugger keyword flow of code stops there, and now you can manually check line by line where the code breaks and throws an error.

Q5. How to handle exceptions in JavaScript? (Groupon)

Ans.

By the help of try/catch block, we can handle exceptions in JavaScript. JavaScript supports **try, catch, finally and throw keywords** for exception handling.

Q6. Explain Closures in JavaScript.

Ans.

Closures is an ability of a function to remember the variables and functions that are declared in its outer scope.

In JavaScript, we need closures when a variable that is defined outside the scope in reference is accessed from some inner scope.

```
var num = 10;  
function sum( ){  
    console.log(num+num);  
}  
sum( );
```