

```
1 console.clear();
2
3 /*
4
5 Hoisting is a behavior where variable and function declarations are moved to the top of their scope (either global or function)
  before code execution.
6
7 Variables declared with var are hoisted to the top and initialized with undefined. You can use them before declaration (not
  recommended).
8
9 Variables declared with let or const are hoisted but not initialized. They are in a Temporal Dead Zone (TDZ) from the start of
  the block until the declaration is encountered.
10
11 */
12
13 console.log(a); // undefined
14
15 var a = 10;
16
17 console.log(a); // 10
18
19 /*
20
21 Behind the scenes:
22
23 var a;          // hoisted and initialized to undefined
24
25 console.log(a); // undefined
26
27 a = 10;
28
29 console.log(a); // 10
30
31 */
32
33 console.log(b); // ✖ ReferenceError: Cannot access 'b' before initialization
34
35 let b = 20;
```

```
36
37 console.log(c); // ❌ ReferenceError: Cannot access 'c' before initialization
38
39 const c = 30;
40
41 // Calling the function before its declaration works perfectly
42
43 greet(); // ✅ Output: "Hello from a hoisted function!"
44
45 function greet() {
46
47     console.log("Hello from a hoisted function!");
48 }
```