JS Hoisting

Hoisting in JavaScript is a behavior of the JS engine where it moves all declarations to the top of their block's scope before execution. This results in being able to access classes, functions, and variables before their declaration in the code.

Different declarations behave differently when hoisted. Hoisted functions - When a function is hoisted, the function can be called before its declaration and execute properly.

Hoisted variables(var) - When a var variable is hoisted, its declaration is moved to the top and it is also initialized with the value of undefined. This means that before the declaration, accessing a var variable will return undefined instead of its user initialized value.

Hoisted variables(let and const) - When a let or const variable is hoisted, its declaration is moved to the top, but unlike a var variable, they aren't initialized. This means that a let or const variable called before its declaration will not be initialized and return an error.

Hoisted classes - When a class is hoisted, its situation is similar to that of a hoisted let or const variable. A class declaration will be moved to the top, but without initialization. So if its getting called, it'll return an error.