

# JSSCOPE & HOISTING

## WHAT

Scope and Hoisting relate to the **visibility / readability** of a variable (or a function).

## SCOPE

WHAT IS IT ?



APPLIES TO



RESULT

Scope is **THE VISIBILITY** of a variables.

Variables.

Every variable declared inside the function, lives only inside that function, and **cannot be accessed from outside**.

### ● GLOBAL SCOPE

Variables declared **outside** the function can be accessed from anywhere in the Javascript.

### ● LOCAL SCOPE

Variables declared **inside** the function can only be accessed from inside the function where they are declared.

### ● BLOCK SCOPE

Variables **LET** and **CONST** have Block Scope, when declared inside a block { } cannot be accessed from outside.

Variable **VAR** don't have **Block Scope**, when declared inside a block { } can be accessed from outside.

ES6+

WHAT IS IT ?



APPLIES TO



RESULT

## HOISTING

Hoisting is **THE DEFAULT BEHAVIOUR** of moving declarations to the top of the current scope.

Variable declarations AND function declarations.

Javascript functions can be called **before** they are declared.

Arrow functions are not hoisted. They must be defined before they are used.

ES6+

Variables **LET** and **CONST** are hoisted to the top of the block, but **NOT** initialized. Using them before being declared, will return **ReferenceError**.