***Difference between Var and Let in JavaScript:***

***var:***

Variables declared with var are function-scoped or globally scoped, but not block-scoped. This means that a variable declared with var is accessible throughout the entire function in which it is declared, or if declared outside a function, it becomes a global variable accessible in the entire JavaScript environment.

Variables declared with var are hoisted, which means that the variable declarations are moved to the top of their scope during the compilation phase. However, the variable assignments remain in their original positions.

var allows variable redeclaration within the same scope, which can lead to unexpected behavior.

***let:***

Variables declared with let are block-scoped. They are only accessible within the block in which they are declared (e.g., within a loop, conditional statement, or a block enclosed by curly braces {}).

Unlike var, variables declared with let are not hoisted. They exist in the temporal dead zone (TDZ) until they are declared.

let does not allow redeclaration of variables within the same scope. If you try to redeclare a variable with let, it will result in an error.

In general, it is recommended to use let and const instead of var in modern JavaScript code. let provides better block scoping and helps avoid some of the common issues associated with var, such as hoisting and unintended global scope. const should be used for variables that you don't intend to reassign.ax