Javascript

Difference between let, const, and var

@coding__comics

var:

- Variables declared with **var** are function-scoped, meaning they are only available within the function in which they are declared. If declared outside any function, they become global.

- **var** variables are hoisted, which means the declaration is moved to the top of its scope during the compilation phase, but the assignment (if any) remains in place.

```
1  console.log(x); // undefined (hoisted)
2  var x = 5;
```

let:

- Variables declared with **let** are block-scoped, meaning they are only available within the block (a pair of curly braces) in which they are defined.

- **let** variables are not hoisted in the same way as **var**, and attempting to access them before declaration results in a **ReferenceError**.

```
1  console.log(y); // ReferenceError: y is not defined
2  let y = 10;
```

const:

- Similar to **let**, **const** is block-scoped.
- The key difference is that variables declared with **const** must be assigned a value at the time of declaration, and their values cannot be reassigned.

```
const z = 15;
z = 20; // Error: Assignment to constant variable
```

- It's important to note that while const prevents the reassignment of the variable itself, it does not make objects or arrays immutable. The properties or elements of a const object or array can still be modified.