

Day 133 -

⇒ what's difference b/w. let, const and var?

var → function scope, redeclare ok, hoisting undefined.

let → block scope, no redeclare, TDZ.

const → block scope, no reassign.

⇒ Can you reassign a const variable?  
why or why not?

• NO

• ● It can change the content of object/array.

⇒ what does it mean that let and const are block-scoped?

- `let/const` only work within block `{ }`.

- if outside use then error.

⇒ What is hoisting? How does it differ for `var` vs `let/const`?

- Declaration moves up

- `var` → hoisted = undefined

- `let/const` → hoisted but TDZ (error before use)

⇒ What is destructuring in JS? Give example.

- <sup>values</sup> ~~the~~ books directly into variables.

- `const [a, b] = [1, 2];`

- `const {x, y} = obj;`



⇒ How do you assign default values in destructuring?

```
const {x = 10, y = 20} = {};
```

```
// x = 10, y = 20.
```

⇒ How can destructuring be used in function parameters?

```
function use({name, age}) {  
  console.log(name, age);  
}
```

⇒ How are template literals different from normal strings?

- Backticks <sup>(\)</sup> used.
- `{ }` & variables.
- Multiline support.

⇒ what are tagged template literals?

- ~~for~~ Template string doing function process.
- example:

tag 'Hello \${name}';

⇒ can you write multi-line string using template literals?

const str =

line 1

line 2

';