

Day #15

⇒ use-case of spread in object copy or update an object without mutating original:

```
const user = { name: "Ali", age: 22 };  
const updated = { ...user, age: 23 };
```

⇒ Merge arrays/objects with spread

```
const arr1 = [1, 2], arr2 = [3, 4];  
const mergedArr = [...arr1, ...arr2];  
const obj1 = { a: 1 }, obj2 = { b: 2 };  
const mergedObj = { ...obj1, ...obj2 };
```

⇒ syntax of arrow function

```
const add = (a, b) => a + b;
```

⇒ this in arrow functions

- Arrow function don't have their own this.

- They use this from the surrounding scope (lexical binding)

⇒ why not as object methods?

Because this won't refer to the object itself, but to outer scope.

⇒ Ternary operator

condition ? value if True : value if False;

Example

let age = 18;

let canVote = age >= 18 ? "Yes" : "No";

⇒ Rewrite if/else using ternary
if (x > 0) { result = "positive"; }
else { result = "Negative"; }

let
result = x > 0 ? "positive"
: "Negative";

⇒ Nesting ternaries

yes, you can, but it hurts
readability. Better to use
if/else

let grade = score > 80 ? "A"
: score > 60 ? "B" : "C";