const

VS

freeze

in JavaScript

[Don’t Miss It]

www.scribbler.live

const

- Declares a variable that cannot be reassigned.
- Primarily used to define variables whose value should not change.
- While the reference to the variable cannot change, the contents of objects or arrays defined with const can still be modified.

JS

```
const user = { name: "Alice", age: 30 };
user.age = 31; // This is allowed
// user = { name: "Bob", age: 25 };
// This will throw an error
```

Object.freeze()

- Makes an object immutable, preventing any changes to its properties.

- Used to freeze an object so that its properties cannot be added, removed, or modified.

- It makes the entire object immutable, but does not affect nested objects unless they are also frozen.

```JS
const user = { name: "Alice", age: 30 };
Object.freeze(user);
user.age = 31; // This will not change the age property
// user.name = "Bob";
// This will also not change the name property
```

Reassignment vs. Modification:

- Object.freeze() prevents any modification to the object's properties.

- const prevents reassignment of the variable itself, but allows modification of the contents if it's an object or array.

Practical Usage

- const is great for defining constants or variables that should not be reassigned.
- Object.freeze() is ideal for creating truly immutable objects where no changes should be allowed.

Deep Freeze

- For true immutability, nested objects need to be frozen as well.

JS

```
function deepFreeze(obj) {
  Object.keys(obj).forEach(name => {
    const prop = obj[name];
    if (typeof prop === 'object' && prop !== null)
        deepFreeze(prop);
  });
  return Object.freeze(obj);
}

const user = { name: "Alice", details: { age: 30 } };
deepFreeze(user);
user.details.age = 31;
// This will not change the age property
```

Keep Exploring Javascript with us!

Share this with a friend who needs it and make sure to practice these in scribbler.

Scribbler.live

Free and Open Interface to experiment JavaScript