

object.freeze in javascript

`Object.freeze()` in JavaScript is a static method that freezes an object, making it immutable. This means that once an object is frozen:

Attempts to add new properties will fail silently in non-strict mode and throw a `TypeError` in strict mode.

Deleting properties will also fail silently or throw an error in strict mode.

The values of existing properties cannot be changed, and their attributes (enumerability, configurability, writability) cannot be altered.

How to use `Object.freeze()`:

You simply pass the object you want to freeze as an argument to the method. It returns the same object, not a frozen copy.

JavaScript

```
const myObject = {
  name: 'Alice',
  age: 30,
  address: {
    city: 'New York',
    zip: '10001'
  }
};

Object.freeze(myObject);

// Attempts to modify the frozen object:
myObject.age = 31; // Fails silently (or throws TypeError in strict mode)
myObject.newProperty = 'value'; // Fails silently (or throws TypeError in strict mode)
delete myObject.name; // Fails silently (or throws TypeError in strict mode)

console.log(myObject); // { name: 'Alice', age: 30, address: { city: 'New York', zip: '10001' } }
```

Important Considerations:

`Object.freeze()` performs a shallow freeze. This means it only freezes the immediate properties of the object. If an object contains other objects or arrays as properties, those nested objects/arrays are not frozen and can still be modified. To achieve a "deep freeze," you would need to recursively freeze all nested objects.

`Object.freeze()` is a powerful tool for enforcing immutability, which can be beneficial for data integrity, preventing unintended side effects, and creating constant-like objects.

In strict mode, attempts to modify a frozen object will throw a `TypeError`, which can help in identifying and debugging code that inadvertently tries to alter frozen objects.

Revision #2

Created 29 October 2025 02:43:38 by AI API

Updated 5 December 2025 09:24:55 by AI Channel