writeable vs enumerable vs configurable

In JavaScript, every object property comes with a set of attributes (called "property descriptors"): **writable**, **enumerable**, and **configurable**. These control how the property behaves beyond just its value.

Writable

- **What it means:** Determines if the property's value can be changed with assignment (`object.prop = newValue`).
- **Default:** `true` for regular object properties.
- **Example:**

```
const obj = {};
Object.defineProperty(obj, 'name', {
  value: 'Akshat',
  writable: false
});
obj.name = 'Jane';
console.log(obj.name); // Output: 'Akshat'
```

- **Summary:**
  - If `writable: false`, any direct attempt to change the property's value fails (silent in non-strict mode, error in strict mode).[1] [2]

Enumerable

- **What it means:** Controls whether the property shows up in `for...in` loops or `Object.keys()`.
- **Default:** `true` for properties you declare directly on an object.
- **Example:**

```
const obj = { name: 'Akshat' };
Object.defineProperty(obj, 'name', {
  enumerable: false
});
console.log(Object.keys(obj)); // Output: []
for (let key in obj) { console.log(key); } // Nothing logs
```

- **Summary:**

- If `enumerable: false`, the property is hidden from loops and most methods that list keys. [2] [3] [1]

Configurable

- **What it means:** Determines if the property can be deleted or its descriptor attributes (`writable`, `enumerable`, etc.) can be changed.
- **Default:** `true` for normal properties.
- **Example:**

```
const obj = {};
Object.defineProperty(obj, 'name', {
  value: 'Akshat',
  configurable: false
});
delete obj.name;          // Fails: name remains
Object.defineProperty(obj, 'name', { enumerable: true }); // Fails: cannot reconfigu
```

- **Key Points:**
  - If `configurable: false`:
    - The property cannot be deleted with `delete`.
    - Its attributes (`enumerable`, `configurable`, `get`, `set`) cannot be changed.
    - `writable` can be changed from `true` to `false` (but not vice versa); once set to `false`, it's irreversible.
    - The property cannot be turned back to configurable once set to `false`. [4] [5] [6]
- **Summary:**
  - `configurable: false` locks down the property and its settings except for making it non-writable.

Summary Table

Attribute	Controls...	Default	Can...
writable	If value can be modified	true	Be set to false to make value read-only
enumerable	If property is visible in loops/key listing	true	Be set to false to hide from enumeration
configurable	If descriptor/prop can be changed or deleted	true	Be set to false to prevent deletion/redefinition

In practice:

- Most normal object properties are writable, enumerable, and configurable.
- Use these flags for advanced patterns, such as locking down config, hiding sensitive/internal data, or creating “read-only” properties. [6] [1] [2]

⁂

1. https://javascript.info/property-descriptors
2. https://akshatv.hashnode.dev/javascript-object-properties-configuration
3. https://www.w3schools.com/jsref/jsref_object_defineproperty.asp
4. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/defineProperty
5. https://stackoverflow.com/questions/23590502/difference-between-configurable-and-writable-attributes-of-an-object
6. https://www.tektutorialshub.com/javascript/javascript-property-descriptors-enumerable-writable-configurable/