# what if we change a nornal key value in shadowcopy

If you change a **normal (non-nested/primitive) key value** in a shallow copy, the original object is **not affected**. Only the top-level properties are duplicated and become independent in the copy.

For example:

```
const original = { a: 1, b: { c: 2 } };
const shallowCopy = { ...original };

shallowCopy.a = 99;      // Only changes shallowCopy.a
console.log(original.a); // Output: 1  (original remains unchanged)
```

- **Top-level primitive changes** (like `a` above) in the shallow copy do **not change the original**.
- **Nested object changes** (like `b.c`) in either the copy or the original *do* change both, because their references are shared.[1] [2] [3]

**Summary:**

- Modifying a direct key/value (not an object or array) in a shallow copy will not affect the original object — they are independent at the top level.[2] [3] [1]

⁂

1. https://developer.mozilla.org/en-US/docs/Glossary/Shallow_copy
2. https://mayallo.com/deep-copy-vs-shallow-copy-in-javascript/
3. https://stackoverflow.com/questions/29050004/modifying-a-copy-of-a-javascript-object-is-causing-the-original-object-to-change