Deep Copy vs Shallow Copy

by Shivam Pandey

Shallow Copy

A shallow copy duplicates only the top-level properties. If those properties are references (like objects or arrays), the copy will reference the same objects.

```
let original = { a: 1, b: { c: 2 } };
let copy = { ...original };
copy.b.c = 3; // Changes 'original.b.c'
```

by Shivam Pandey

Deep Copy

A deep copy creates a complete clone of the original object, duplicating all nested objects and arrays.

```
let original = { a: 1, b: { c: 2 } };
let copy = JSON.parse(JSON.
stringify(original));
copy.b.c = 3; // 'original.b.c'
// remains 2!
```

by Shivam Pandey

When to use Shallow Copy

- Small objects with primitive data types.
- Situations where performance is critical.
- Cases where changes to nested objects should reflect in all copies.

by Shivam Pandey

When to use Deep Copy

- Complex objects with nested structures.
- Scenarios where complete independence from the original object is needed.
- Preventing unintended side-effects from shared references.

by Shivam Pandey

FOLLOW ME

for more such content

Did you find this helpful?

Comment down below