Here's a solid set of exercises (progressive and practical) to help you master the `spread` operator and `map()` in JavaScript. These cover arrays, objects, and more advanced patterns.

---

## ◈ Spread Operator Exercises

### ✅ Basic Level

1. Clone an Array

```
const nums = [1, 2, 3];
const copy = [...nums];
// Practice: Log both and verify independence
```

2. Merge Two Arrays

```
const arr1 = [1, 2];
const arr2 = [3, 4];
const merged = [...arr1, ...arr2];
```

3. Clone an Object

```
const user = { name: "Alice", age: 25 };
const clone = { ...user };
```

4. Merge Two Objects

```
const a = { x: 1 };
const b = { y: 2 };
const merged = { ...a, ...b };  // What happens if both have key 'x'?
```

5. Add Properties Dynamically

```
const person = { name: "John" };
const updated = { ...person, age: 30 };
```

---

## ◈ map() Method Exercises

### ✅ Basic Level

6. Double Each Number

```
const nums = [1, 2, 3];
const doubled = nums.map(n => n * 2);
```

7. Convert to Uppercase

```
const names = ["alice", "bob"];
const upper = names.map(name => name.toUpperCase());
```

8. Extract Property from Array of Objects

```
const users = [
  { id: 1, name: "A" },
  { id: 2, name: "B" },
];
const names = users.map(u => u.name);
```

9. Format Strings

```
const products = ["apple", "banana"];
const formatted = products.map(p => `Product: ${p}`);
```

10. Return Boolean for Each Item

```
const nums = [3, 6, 9];
const isEven = nums.map(n => n % 2 === 0);
```

---

## 💠 Combo: Spread + map()

✅ Intermediate Level

11. Clone Array of Objects with a New Property

```
const users = [{ name: "Anna" }, { name: "Tom" }];
const withAge = users.map(user => ({ ...user, age: 25 }));
```

12. Flatten Nested Arrays

```
const nested = [[1, 2], [3, 4]];
const flat = nested.flat();  // or use spread inside map in custom ways
```

13. Immutably Update an Item in an Array

```
const list = [{ id: 1 }, { id: 2 }];
const updatedList = list.map(item =>
  item.id === 2 ? { ...item, done: true } : item
);
```

14. Remove a Property from Each Object

```
const users = [{ name: "A", pwd: "123" }, { name: "B", pwd: "456" }];
const safeUsers = users.map(({ pwd, ...rest }) => rest);
```

15. Transform Data + Merge with Defaults

```
const entries = [{ name: "Sam" }, { name: "Lee" }];
const defaultData = { active: true };
const final = entries.map(e => ({ ...defaultData, ...e }));
```