📌 How many times can you call `resolve()` or `reject()` in a Promise?

👉 Only once!

Once a Promise is **settled** (either **resolved** or **rejected**), it becomes **immutable**—its state is **locked**, and any further calls to `resolve()` or `reject()` are ignored silently.

🧠 Let's break it down:

- ✅ You can call either `resolve(value)` or `reject(error)` **once**.
- 🚫 Calling `resolve()` or `reject()` **after it has already been settled has no effect**.
- ⚠️ If both are present, **only the first one called will take effect**.

🔍 Example:

```
const promise = new Promise((resolve, reject) => {
  resolve("✅ First resolve");

  // This will be ignored
  reject("❌ This reject is ignored");

  // This will also be ignored
  resolve("✅ Second resolve is ignored");
});

promise.then(result => console.log(result)).catch(err => console.log(err));
```

Output:

```
✅ First resolve
```

🧾 Summary:

Action	Effect
First `resolve()` or `reject()`	✅ Takes effect
Subsequent `resolve()` or `reject()`	❌ Ignored silently
