

JavaScript Nullish Checking & Defaults Cheat Sheet

Safe Property Access

1. Optional Chaining (Modern - ES2020)

javascript

```
const role = obj?.property?.nestedProperty;
```

Returns: `property` value if `obj` exists, `undefined` if any chain link is nullish

2. Traditional Safe Access

javascript

```
const role = obj && obj.property && obj.nestedProperty;
```

Returns: Last truthy value or first falsy value in chain

Providing Default Values

1. Nullish Coalescing (Modern - ES2020)

javascript

```
const role = possiblyNull ?? 'default';
```

Returns: `possiblyNull` if it's **NOT** `null` or `undefined`, otherwise `'default'`

2. Logical OR (Classic)

javascript

```
const role = possiblyFalsy || 'default';
```

Returns: `possiblyFalsy` if **truthy**, otherwise `'default'`

 **Careful!** Falsy values include: `false`, `0`, `""`, `null`, `undefined`, `NaN`

Combination Patterns

Safe Access + Default Value

javascript

```
// Modern
const role = obj?.property ?? 'default';
```

```
// Traditional
const role = (obj && obj.property) || 'default';
```

Quick Reference Table

Scenario	Modern Syntax	Traditional Syntax	Returns
Safe property access	<code>obj?.prop</code>	<code>obj && obj.prop</code>	prop value or undefined
Deep safe access	<code>obj?.a?.b?.c</code>	<code>obj && obj.a && obj.b && obj.c</code>	Nested value or undefined
Default for null/undefined	<code>value ?? 'default'</code>	<code>value !== null ? value : 'default'</code>	Value or default
Default for any falsy	<code>`value`</code>	<code>'default'`</code>	<code>`value`</code> 'default' Value or default
Safe access + default	<code>obj?.prop ?? 'default'</code>	<code>`(obj && obj.prop)`</code>	<code>'default'`</code> Property value or default

Real-world Examples

javascript

```
const user = { profile: { name: 'John' } };

// Safe property access
const userName = user?.profile?.name;           // 'John'
const userAge = user?.profile?.age;              // undefined

// Default values
const displayName = userName ?? 'Anonymous';     // 'John'
const displayAge = userAge ?? 'Age not set';      // 'Age not set'
const role = user?.role ?? 'guest';              // 'guest'

// Combined pattern
const greeting = `Hello ${user?.profile?.name ?? 'Guest'}`; // 'Hello John'
```

Common Pitfalls

javascript

```
// ❌ WRONG - returns object, not property
const bad = user && user.profile; // Returns profile object, not name!

// ✅ CORRECT - gets the actual property
const good = user && user.profile && user.profile.name;

// ❌ WRONG - || catches empty strings, 0, false
const name = user.name || 'Anonymous'; // 'Anonymous' if name is ""

// ✅ CORRECT - ?? only catches null/undefined
const name = user.name ?? 'Anonymous'; // "" stays as ""
```

Pro Tip

Use **optional chaining + nullish coalescing** for clean, safe code:

javascript

```
// Instead of:
const value = obj && obj.a && obj.a.b ? obj.a.b : 'default';

// Write:
const value = obj?.a?.b ?? 'default';
```

Remember: `?.` for safe access, `??` for smart defaults!