

JavaScript String Methods Cheat Sheet

Core methods, comparison behavior, and defect patterns for QA, BA, and IT professionals.

1. Core String Methods

Method	Does	Watch For
<code>trim()</code>	Removes leading/trailing whitespace	Must reassign the result — strings are immutable
<code>slice(start,end)</code>	Extracts substring by position	End index is exclusive — off-by-one risk
<code>split(sep)</code>	Breaks string into an array	Breaks silently if delimiter count changes upstream
<code>replace()</code> / <code>replaceAll()</code>	Substitutes matched text	<code>replace()</code> without <code>/g</code> only replaces the first match
<code>includes()</code> / <code>indexOf()</code>	Finds a substring	<code>indexOf()</code> returns -1, not false, when not found
<code>toUpperCase()</code> / <code>toLowerCase()</code>	Converts case	Locale rules can affect certain characters
<code>padStart()</code> / <code>padEnd()</code>	Pads to a fixed length	Common for formatting IDs/codes to a required width

2. String Comparison Behavior

Comparison	Behavior	Risk
<code>===</code>	Exact match, case-sensitive	Fails on invisible whitespace/Unicode differences
<code>==</code>	Allows type coercion	Can mask genuine type mismatches
<code>localeCompare()</code>	Locale-aware sort comparison	Missing use causes accented-character sort defects

3. Six Defect Patterns to Recognize on Sight

Missing reassignment after a method call

`trim()`, `replace()`, `toUpperCase()` all return new strings. Calling them without capturing the result changes nothing.

Off-by-one in `slice()`/`substring()`

The end index is exclusive. `"Hello".slice(0,3)` returns `"Hel"`, not `"Hell"`.

indexOf() treated as a boolean

-1 is truthy in JavaScript. Always compare explicitly: `indexOf(x) !== -1`.

Invisible Unicode characters

Smart quotes, non-breaking spaces, and zero-width characters from copy-pasted or PDF-extracted text look identical but fail strict equality.

Partial replace() due to missing global flag

Without `/g` or `replaceAll()`, only the first match is replaced — easy to miss on small test samples.

Silent type coercion with +

"5" + 3 produces "53", not 8. Unvalidated form input concatenated this way is a common defect source.

Edge case: a non-breaking space (U+00A0) looks identical to a regular space in almost every log viewer, but fails a strict equality check against a string built with regular spaces. A frequent, hard-to-spot cause of 'the values look the same but the test fails.'