sonar RULES

Products ⌄

- Ø Secrets
- SAP ABAP
- APEX Apex
- C C
- C++ C++
- CloudFormation
- COBOL COBOL
- C# C#
- CSS CSS
- Flex Flex
- GO Go
- HTML HTML
- Java Java
- JS **JavaScript**
- Kotlin Kotlin
- Objective C
- php PHP
- PL/I PL/I
- PL/SQL PL/SQL
- Python Python
- RPG RPG
- Ruby Ruby
- Scala Scala
- Swift Swift
- Terraform Terraform
- Text Text
- TS TypeScript
- T-SQL T-SQL
- VB VB.NET
- VB6 VB6
- XML XML

JS

JavaScript static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your JAVASCRIPT code

| All rules 285 | 🔒 Vulnerability 29 | 🐛 Bug 62 | Security Hotspot 43 | Code Smell 151 | Quick Fix 41 |

Tags ⌄   Search by name... 🔍

Functions should always return the same type
⊗ Code Smell

Arguments to built-in functions should match documented types
⊗ Code Smell

Literals should not be thrown
⊗ Code Smell

Functions should not be called both with and without "new"
⊗ Code Smell

Array indexes should be numeric
⊗ Code Smell

Assertion arguments should be passed in the correct order
⊗ Code Smell

Ternary operators should not be nested
⊗ Code Smell

"delete" should not be used on arrays
⊗ Code Smell

Variables and functions should not be redeclared
⊗ Code Smell

"indexOf" checks should not be for positive numbers
⊗ Code Smell

"arguments.caller" and "arguments.callee" should not be used
⊗ Code Smell

Multiline blocks should be enclosed in curly braces
⊗ Code Smell

Unicode Grapheme Clusters should be avoided inside regex character classes

Analyze your code

🐛 Bug   ⊘ Major ?   🏷 regex

When placing Unicode Grapheme Clusters (characters which require to be encoded in multiple Code Points) inside a character class of a regular expression, this will likely lead to unintended behavior.

For instance, the grapheme cluster ë requires two code points: one for `c`, followed by one for the *umlaut* modifier `'\u{0308}'`. If placed within a character class, such as `[ë]`, the regex will consider the character class being the enumeration `[c\u{0308}]` instead. It will, therefore, match every `c` and every *umlaut* that isn't expressed as a single codepoint, which is extremely unlikely to be the intended behavior.

This rule raises an issue every time Unicode Grapheme Clusters are used within a character class of a regular expression.

Noncompliant Code Example

```
"cëdd".replace(/[ëd]/g, "X"); // result is "XXXXXX" and not
```

Compliant Solution

```
"cëdd".replace(/ë|d̈/g, "X"); // result is "cXXd"
```

Available In:
sonarlint | sonarcloud | sonarqube

Code Smell

Boolean expressions should not be gratuitous

⊗ Code Smell

Variables should be used in the blocks where they are declared

⊗ Code Smell

Parameters should be passed in the correct order

⊗ Code Smell

Two branches in a conditional structure should not have exactly the same implementation