# sonar RULES

**Products** ⌄

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

**TS**

# TypeScript static code analysis

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

| All rules 279 | 🔒 Vulnerability 27 | 🐞 Bug 51 | 🛡 Security Hotspot 43 | ⊙ Code Smell 158 | 🐞 Quick Fix 50 |

Tags ⌄          Search by name...

### 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

### 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
⊙ Code Smell

### Unused assignments should be removed
⊙ Code Smell

### Function parameters with default

## Alternatives in regular expressions should be grouped when used with anchors

**Analyze your code**

🐞 Bug      ⊘ Major ⑦      🏷 regex

In regular expressions, anchors `^` and `$` have higher precedence than the `|` operator. So in a regular expression like `^alt1|alt2|alt3$`, `alt1` would be anchored to the beginning, `alt3` to the end and `alt2` wouldn't be anchored at all. Usually the intended behavior is that all alternatives are anchored at both ends. To achieve this, a non-capturing group should be used around the alternatives.

In cases where it is intended that the anchors only apply to one alternative each, adding (non-capturing) groups around the anchors and the parts that they apply to will make it explicit which parts are anchored and avoid readers misunderstanding the precedence or changing it because they mistakenly assume the precedence was not intended.

**Noncompliant Code Example**

```
^a|b|c$
```

**Compliant Solution**

```
^(?:a|b|c)$
```

or

```
^a$|^b$|^c$
```

or, if you do want the anchors to only apply to `a` and `c` respectively:

```
(?:^a)|b|(?:c$)
```

Available In:

**sonar**lint 😊  |  **sonar**cloud ☁  |  **sonar**qube ⌇

values should be last

⊗ Code Smell

---

**Functions should not be defined inside loops**

⊗ Code Smell

---

**"switch" statements should not have too many "case" clauses**

⊗ Code Smell

---

**Only "while", "do", "for" and "switch" statements should be labelled**

⊗ Code Smell

---

Sections of code should not be