




 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



TypeScript static code analysis

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

All rules279

Vulnerability27

Bug51

Security Hotspot43

Code Smell158

Quick Fix50

Tags ▾

Search by name... 🔍

Code Smell

Primitive types should be omitted from initialized or defaulted declarations

Code Smell

Non-null assertions should not be used

Code Smell

"undefined" should not be assigned

Code Smell

Trailing commas should not be used

Code Smell

Array constructors should not be used

Code Smell

Quotes for string literals should be used consistently

Code Smell

Statements should end with semicolons

Code Smell

Comments should not be located at the end of lines of code

Code Smell

Loops should not contain more than a single "break" or "continue" statement

Code Smell

Variable, property and parameter names should comply with a naming convention

Code Smell

Lines should not end with trailing whitespaces

Code Smell

An open curly brace should be located at the end of a line

Analyze your code

Code Smell

Minor ?

convention

Shared naming conventions allow teams to collaborate effectively. This rule raises an issue when the brace-style is not respecting the convention setup in parameter:

- 1tbs (default)
- allman
- stroustrup

Noncompliant Code Example

Using the parameter default (1tbs):

```
if (condition)
{
    doSomething();
}
else {
    doSomethingElse();
}
```

Compliant Solution

```
if (condition) {
    doSomething();
} else {
    doSomethingElse();
}
```

Exceptions

- Object literals appearing as arguments can start on their own line.

```
functionWithObject(
{
    g: "someValue"
});
```

- When blocks are inlined (left and right curly braces on the same line), no issue is triggered.

```
if(condition) {doSomething();}
```

Available In:

sonarlint | sonarcloud | sonarqube

https://rules.sonarsource.com/typescript/RSPEC-1105

1/2

Files should contain an empty newline at the end

 Code Smell

An open curly brace should be located at the end of a line

 Code Smell

Tabulation characters should not be used

 Code Smell

Function and method names should comply with a naming convention

 Code Smell

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.
[Privacy Policy](#)