

































-  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 rules 279
-  Vulnerability 27
-  Bug 51
-  Security Hotspot 43
-  Code Smell 158
-  Quick Fix 50

-  Code Smell



Primitive types should be omitted from initialized or defaulted declarations

 Code Smell
-  Code Smell



Non-null assertions should not be used

 Code Smell
-  Code Smell



"undefined" should not be assigned

 Code Smell
-  Code Smell

Trailing commas should not be used

 Code Smell
-  Code Smell

Array constructors should not be used

 Code Smell
-  Code Smell

Quotes for string literals should be used consistently

 Code Smell
- Code Smell

Statements should end with semicolons

 Code Smell
- Code Smell

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

 Code Smell
- Code Smell

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

 Code Smell
- Code Smell

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

 Code Smell
- Code Smell

Lines should not end with trailing whitespaces

 Code Smell

Tags ▾

Search by name... 🔍

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

Analyze your code

-  Code Smell
-  Minor ?
-  brain-overload

Restricting the number of `break` and `continue` statements in a loop is done in the interest of good structured programming.

Only one `break` or one `continue` statement is acceptable in a loop, since it facilitates optimal coding. If there is more than one, the code should be refactored to increase readability.

Noncompliant Code Example





```
for (var i = 1; i <= 10; i++) { // Noncompliant - 2 continu
  if (i % 2 == 0) {
    continue;
  }

  if (i % 3 == 0) {
    continue;
  }

  alert("i = " + i);
}
```

Available In:

sonarlint  sonarcloud  sonarqube 

| |
|---|
| <div>Files should contain an empty newline at the end</div> <div> Code Smell</div> |
| <div>An open curly brace should be located at the end of a line</div> <div> Code Smell</div> |
| <div>Tabulation characters should not be used</div> <div> Code Smell</div> |
| <div>Function and method names should comply with a naming convention</div> <div> Code Smell</div> |