




 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

Tags ▾

Search by name... 🔍

Code Smell
Wrapper objects should not be used for primitive types
Code Smell
Multiline string literals should not be used
Code Smell
Local variables should not be declared and then immediately returned or thrown
Code Smell
Function call arguments should not start on new lines
Code Smell
"switch" statements should have at least 3 "case" clauses
Code Smell
A "while" loop should be used instead of a "for" loop
Code Smell
Unnecessary imports should be removed
Code Smell
Boolean literals should not be used in comparisons
Code Smell
Extra semicolons should be removed
Code Smell
Class names should comply with a naming convention
Code Smell
Track uses of "TODO" tags
Code Smell

Comma operator should not be used

Analyze your code

Code Smell

Major ?

The comma operator takes two expressions, executes them from left to right and returns the result of the second one. Use of this operator is generally detrimental to the readability and reliability of code, and the same effect can be achieved by other means.

Noncompliant Code Example

```
i = a += 2, a + b; // What's the value of i ?
```

Compliant Solution

```
a += 2;  
i = a + b;
```

Exceptions

Use of comma operator is tolerated:





- in initialization and increment expressions of for loops.

```
for(i = 0, j = 5; i < 6; i++, j++) { ... }
```

- If the expression sequence is explicitly wrapped in parentheses.

Available In:

sonarlint | sonarcloud | sonarqube

Web SQL databases should not be used  Vulnerability
Variables declared with "var" should be declared before they are used  Code Smell
Track lack of copyright and license headers  Code Smell
Reading the Standard Input is security-sensitive  Security Hotspot