

































-  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


Object literal shorthand syntax should be used





 Strings and non-strings should not be added





 Primitive types should be omitted from initialized or defaulted declarations





 Non-null assertions should not be used




 "undefined" should not be assigned



 Trailing commas should not be used



 Array constructors should not be used



 Quotes for string literals should be used consistently



 Statements should end with semicolons



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




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





 Variable, property and parameter

### Variables should not be shadowed

Analyze your code




 Code Smell

 Major ?

 suspicious pitfall

Overriding or shadowing a variable declared in an outer scope can strongly impact the readability, and therefore the maintainability, of a piece of code. Further, it could lead maintainers to introduce bugs because they think they're using one variable but are really using another.





Available In:

 sonarlint  sonarcloud  sonarqube

© 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](#)

https://rules.sonarsource.com/typescript/RSPEC-1117

1/2

|   |
|---|
| <div>names should comply with a naming convention</div> <div> Code Smell</div>               |
| <div>Lines should not end with trailing whitespaces</div> <div> Code Smell</div>             |
| <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> |
|   |