




 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...

directly




Code Smell

Comparison operators should not be used with strings




Code Smell

Private properties that are only assigned in the constructor or at declaration should be "readonly"




Code Smell

Property getters and setters should come in pairs




Code Smell

JavaScript parser failure




Code Smell

The ternary operator should not be used




Code Smell

"==" and "!=" should be used instead of "===" and "==="




Code Smell

Functions should not have too many lines of code




Code Smell

Track comments matching a regular expression




Code Smell

Statements should be on separate lines




Code Smell

Magic numbers should not be used



Code Smell


Collapsible "if" statements should be merged





Code Smell

"indexOf" checks should not be for positive numbers

Analyze your code

 Code Smell

 Major

 suspicious

Most checks against an `indexOf` call against an array compare it with `-1` because `0` is a valid index. Any checks which look for values `>0` ignore the first element, which is likely a bug. If you're merely checking the presence of the element, consider using `includes` instead. Before using `includes` method make sure that your browser version is supporting it.

Noncompliant Code Example

```
var color = "blue";
var name = "ishmael";

var arr = [color, name];

if (arr.indexOf("blue") > 0) { // Noncompliant
  // ...
}
```

Compliant Solution

```
var color = "blue";
var name = "ishmael";

var arr = [color, name];

if (arr.indexOf("blue") >= 0) {
  // ...
}
if (arr.includes("blue")) {
  // ...
}
```

See

[Array.prototype.includes\(\)](#) documentation at MDN

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-2692

1/2

<div>Standard outputs should not be used directly to log anything</div> <div> Code Smell</div>
<div>Files should not have too many lines of code</div> <div> Code Smell</div>
<div>Lines should not be too long</div> <div> Code Smell</div>
<div>Debugger statements should not be used</div> <div> Vulnerability</div>