




 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



JavaScript static code analysis

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

- All rules 285
- Vulnerability 29
- Bug 62
- Security Hotspot 43
- Code Smell 151
- Quick Fix 41

Object literal shorthand syntax should be used	Code Smell
Strings and non-strings should not be added	Code Smell
Object literal syntax should 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

Tags ▾

Search by name... 🔍

Unused function parameters should be removed

Analyze your code

- Code Smell
- Major
- Quick Fix
- unused

Unused parameters are misleading. Whatever the values passed to such parameters, the behavior will be the same.

Noncompliant Code Example

```
function doSomething(a, b) { // "a" is unused
  return compute(b);
}
```

Compliant Solution

```
function doSomething(b) {
  return compute(b);
}
```

or

```
function doSomething(_a, b) {
  return compute(b);
}
```

Exceptions

When arguments is used in the function body, no parameter is reported as unused.

```
function doSomething(a, b, c) {
  compute(arguments);
}
```





Also, the rule ignores all parameters whose name starts with an underscore (_). This is a common practice to acknowledge the fact that some parameter is unused (e.g. in TypeScript compiler).

```
function doSomething(_a, b) {
  return compute(b);
}
```

Available In:

sonarlint | sonarcloud | sonarqube

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](#)


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