sonar RULES

Products ⌄

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

JS

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

Tags ⌄ | Search by name...

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

Standard outputs should not be used directly to log anything

Code Smell

Files should not have too many lines of code

Code Smell

Lines should not be too long

Code Smell

Debugger statements should not be used

Vulnerability

"alert(...)" should not be used

Vulnerability

Regular expressions using Unicode character classes or property escapes

Variables and functions should not be redeclared

Analyze your code

Code Smell   Major   confusing

This rule checks that a declaration doesn't use a name that is already in use. Indeed, it is possible to use the same symbol multiple times as either a variable or a function, but doing so is likely to confuse maintainers. Further it's possible that such reassignments are made in error, with the developer not realizing that the value of the variable is overwritten by the new assignment.

This rule also applies to function parameters.

Noncompliant Code Example

```
var a = 'foo';
function a() {}    // Noncompliant
console.log(a);    // prints "foo"

function myFunc(arg) {
  var arg = "event"; // Noncompliant, argument value is lost
}

fun(); // prints "bar"

function fun() {
  console.log("foo");
}

fun(); // prints "bar"

function fun() {  // Noncompliant
  console.log("bar");
}

fun(); // prints "bar"
```

Compliant Solution

```
var a = 'foo';
function otherName() {}
console.log(a);

function myFunc(arg) {
  var newName = "event";
}

fun(); // prints "foo"

function fun() {
  print("foo");
}

fun(); // prints "foo"
```

should enable the unicode flag

Bug

The base should be provided to "parseInt"

Bug

Function declarations should not be made within blocks

Bug

Writing cookies is security-sensitive

Security Hotspot

"continue" should not be used

```
function printBar() {
  print("bar");
}

printBar(); // prints "bar"
```

Available In:

sonarlint | sonarcloud | sonarqube