



JavaScript

Objective C

Kotlin

PHP

PL/I

PL/SQL

Python

RPG

Ruby

Scala

Swift

Text

T-SQL

VB.NET

VB₆

XML

Terraform

TypeScript

OiD.

A

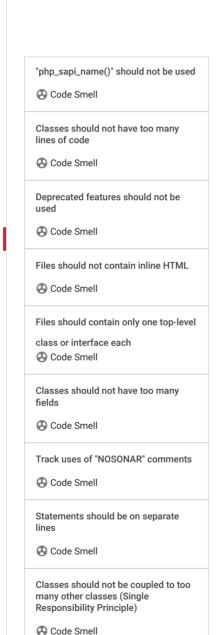


PHP static code analysis

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



Tags



"switch case" clauses should not have

Assignments should not be made from within sub-expressions

Files should not have too many lines

too many lines of code

Code Smell

A Code Smell

Code Smell

of code

```
Multiline blocks should be
                                               Analyze your code
enclosed in curly braces
Curly braces can be omitted from a one-line block, such as with an if statement or
for loop, but doing so can be misleading and induce bugs.
This rule raises an issue when the whitespacing of the lines after a one line block
indicates an intent to include those lines in the block, but the omission of curly
braces means the lines will be unconditionally executed once.
Note that this rule considers tab characters to be equivalent to 1 space. If you mix
spaces and tabs you will sometimes see issues in code which look fine in your
editor but are confusing when you change the size of tabs.
Noncompliant Code Example
  if ($condition)
    firstActionInBlock();
    secondAction(); // Noncompliant; executed unconditionally
  thirdAction();
  if($condition) firstActionInBlock(); secondAction(); // Non
  if($condition) firstActionInBlock(); // Noncompliant
    secondAction(); // Executed unconditionally
  $str = null:
```

Search by name...

Compliant Solution

\$str = \$arrav(\$i1:

for (\$i = 0; \$i < count(\$array); \$i++)

```
if ($condition) {
  firstActionInBlock();
  secondAction();
thirdAction();
if($condition) { firstActionInBlock(); secondAction(); }
if($condition) {
  firstActionInBlock();
  secondAction();
$str = null;
for ($i = 0; $i < count($array); $i++) {
 $str = $array[$i];
  doTheThing($str);
```

doTheThing(\$str); // Noncompliant; executed only on last

Lines should not be too long
Code Smell

HTTP response headers should not be vulnerable to injection attacks
Vulnerability

"sleep" should not be called
Vulnerability

Static members should be referenced with "static::"

See

• MITRE, CWE-483 - Incorrect Block Delimitation

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