

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- 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



PHP static code analysis

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

All rules 268

Vulnerability 40

Bug 51

Security Hotspot 33

Code Smell 144

Tags ▾

Search by name... 🔍

Security Hotspot
"goto" statement should not be used
Code Smell
Character classes in regular expressions should not contain only one character
Code Smell
Superfluous curly brace quantifiers should be avoided
Code Smell
Non-capturing groups without quantifier should not be used
Code Smell
WordPress option names should not be misspelled
Code Smell
WordPress options should not be defined at the end of "wp-config.php"
Code Smell
Constants should not be redefined
Code Smell
Regular expressions should not contain empty groups
Code Smell
Regular expressions should not contain multiple spaces
Code Smell
Single-character alternations in regular expressions should be replaced with character classes
Code Smell
Reluctant quantifiers in regular expressions should be followed by an expression that can't match the empty

Control structures should use curly braces

Analyze your code

Code Smell

Critical ?

pitfall

While not technically incorrect, the omission of curly braces can be misleading, and may lead to the introduction of errors during maintenance.

Noncompliant Code Example

```
if (condition) // Noncompliant
    executeSomething();
```

Compliant Solution


```
if (condition) {
    executeSomething();
}
```

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](#)


string

 Code Smell

Character classes in regular expressions should not contain the same character twice

 Code Smell

Regular expressions should not be too complicated

 Code Smell

PHPUnit assertTrue/assertFalse should be simplified to the corresponding dedicated assertion

 Code Smell

Methods should not have identical
implementation