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

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... 🔍

Code Smell

Unicode-aware versions of character classes should be preferred

☢ Code Smell

Alias functions should not be used

☢ Code Smell

Perl-style comments should not be used

☢ Code Smell

Superglobals should not be accessed directly

☢ Code Smell

Colors should be defined in upper case

☢ Code Smell

String literals should not be concatenated

☢ Code Smell

"final" should not be used redundantly

☢ Code Smell

Source code should comply with formatting standards

☢ Code Smell

"elseif" keyword should be used in place of "else if" keywords

☢ Code Smell

PHP keywords and constants "true", "false", "null" should be lower case

☢ Code Smell

Closing tag "?>" should be omitted on files containing only PHP

☢ Code Smell

Classes should not have too many methods

Analyze your code

☢ Code Smell   ◈ Major ?   🏷 brain-overload

A class that grows too much tends to aggregate too many responsibilities and inevitably becomes harder to understand and therefore to maintain. Above a specific threshold, it is strongly advised to refactor the class into smaller ones which focus on well defined topics.

Available In:

sonarlint 😖 | sonarcloud ⚙ | sonarqube 📶

Only LF character (Unix-like) should be used to end lines

⚠ Code Smell

More than one property should not be declared per statement

⚠ Code Smell

The "var" keyword should not be used

⚠ Code Smell

"<?php" and "<?=" tags should be used

⚠ Code Smell

File names should comply with a naming convention

⚠ Code Smell