

- 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... 🔍

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 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

Constant names should comply with a naming convention

Analyze your code

Code Smell Critical ? convention

Shared coding conventions allow teams to collaborate efficiently. This rule checks that all constant names match a provided regular expression.

### Noncompliant Code Example

With the default regular expression `^[A-Z][A-Z0-9]*(_[A-Z0-9]+)*$`:

```
define("const1", true);

class Foo {
    const const2 = "bar";
}
```

### Compliant Solution

```
define("CONST1", true);

class Foo {
    const CONST2 = "bar";
}
```

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](#)

Methods should not have identical implementations

 Code Smell

Functions should use "return" consistently

 Code Smell

Assertion arguments should be passed in the correct order

 Code Smell

Ternary operators should not be nested

 Code Smell