

- 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...

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 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

String literals should not be duplicated

Analyze your code

Code Smell

Critical

design

Duplicated string literals make the process of refactoring error-prone, since you must be sure to update all occurrences.

On the other hand, constants can be referenced from many places, but only need to be updated in a single place.

Noncompliant Code Example

With the default threshold of 3:

```
function run() {
    prepare('this is a duplicate'); // Non-Compliant - 'this
    execute('this is a duplicate');
    release('this is a duplicate');
}
```

Compliant Solution

```
MESSAGE = 'this is a duplicate';

function run() {
    prepare(MESSAGE);
    execute(MESSAGE);
    release(MESSAGE);
}
```

Exceptions

- No issue will be raised on:
- strings with less than 5 characters
 - strings with only letters, numbers, underscores and hyphens

```
$severity = $request->getParam('severity-score');{code}
```


Available In:

sonarlint | sonarcloud | sonarqube

For unit assertions, assertions should be simplified to the corresponding dedicated assertion

 Code Smell

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