
































-  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... 🔍

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
Comments should not be located at the end of lines of code
 Code Smell
Local variable and function parameter names should comply with a naming convention
 Code Smell
Field names should comply with a naming convention
 Code Smell
Lines should not end with trailing whitespaces
 Code Smell
Files should contain an empty newline at the end
 Code Smell
Modifiers should be declared in the correct order
 Code Smell
An open curly brace should be located at the beginning of a line

Method and function names should comply with a naming convention Analyze your code

 Code Smell

 Minor ?

 convention

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

Noncompliant Code Example

With default provided regular expression: `^[a-z][a-zA-Z0-9]*$`:

```
function DoSomething(){...}
```

Compliant Solution

```
function doSomething(){...}
```

Exceptions

Methods with an `@inheritdoc` annotation, as well as magic methods (`__construct()`, `__destruct()`, `__call()`, `__callStatic()`, `__get()`, `__set()`, `__isset()`, `__unset()`, `__sleep()`, `__wakeup()`, `__toString()`, `__invoke()`, `__set_state()`, `__clone()`, `__debugInfo()`) are ignored.

```
function __construct(){...}
function __destruct(){...}

/**
 * @inheritdoc
 */
function myFunc(){...}
```

Available In:

 sonarlint

 sonarcloud

 sonarqube

 Code Smell

An open curly brace should be located at the end of a line

 Code Smell

Tabulation characters should not be used

 Code Smell

Method and function names should comply with a naming convention

 Code Smell

Creating cookies with broadly defined "domain" flags is security-sensitive

 Security Hotspot