
































-  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


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

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

Track uses of "FIXME" tags

Analyze your code

 Code Smell  Major  cwe

FIXME tags are commonly used to mark places where a bug is suspected, but which the developer wants to deal with later.

Sometimes the developer will not have the time or will simply forget to get back to that tag.

This rule is meant to track those tags and to ensure that they do not go unnoticed.

Noncompliant Code Example

```
function divide($numerator, $denominator) {  
    return $numerator / $denominator;           // FIXME  
}
```


See

- [MITRE, CWE-546](#) - Suspicious Comment


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](#)

 Code Smell


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

 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