

- 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 | Code Smell |

Empty statements should be removed

Analyze your code

Code Smell

Minor

unused

Empty statements, i.e. `;`, are usually introduced by mistake, for example because:

- It was meant to be replaced by an actual statement, but this was forgotten.
- There was a typo which lead the semicolon to be doubled, i.e. `;;`.

Noncompliant Code Example

```
function doSomething() {  
    ; // Noncompliant  
}  
  
function doSomethingElse($p) {  
    echo $p;; // Noncompliant  
}  
  
for ($i = 1; $i <= 10; doSomething($i), $i++); // Noncompliant
```

Compliant Solution

```
function doSomething() {}  
  
function doSomethingElse($p) {  
    echo $p;  
  
    for ($i = 1; $i <= 10; $i++) {  
        doSomething($i);  
    }  
}
```

Available In:

sonarlint

sonarcloud

sonarqube


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