

- 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

Classes should not be coupled to too many other classes (Single Responsibility Principle)

Analyze your code

Code Smell

Major

brain-overload

According to the Single Responsibility Principle, introduced by Robert C. Martin in his book "Principles of Object Oriented Design", a class should have only one responsibility:

If a class has more than one responsibility, then the responsibilities become coupled.
Changes to one responsibility may impair or inhibit the class' ability to meet the others.
This kind of coupling leads to fragile designs that break in unexpected ways when changed.

Classes which rely on many other classes tend to aggregate too many responsibilities and should be split into several smaller ones.

Nested classes dependencies are not counted as dependencies of the outer class.

Noncompliant Code Example

```
class Foo {                                // Noncompliant - Foo depends on too many other classes
    /**
     * @var T1
     */
    public $a1;                             // Foo is coupled to T1
    /**
     * @var T2
     */
    protected $a2;                         // Foo is coupled to T2
    /**
     * @var T3
     */
    private $a3;                            // Foo is coupled to T3

    /**
     * @param T5
     * @param T6
     *
     * @return T4
     */
    public function compute(T5 $a, $b) { // Foo is coupled to T5 and T6
        $result = new T7();             // Foo is coupled to T7
        return $result;
    }
}
```

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