

- 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



Python static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your PYTHON code

All rules 216 Vulnerability 29 Bug 55 Security Hotspot 31 Code Smell 101

Tags

Search by name...

Code Smell
Function parameters' default values should not be modified or assigned
Code Smell
Some special methods should return "NotImplemented" instead of raising "NotImplementedError"
Code Smell
Custom Exception classes should inherit from "Exception" or one of its subclasses
Code Smell
Bare "raise" statements should not be used in "finally" blocks
Code Smell
Arguments given to functions should be of an expected type
Code Smell
`str.replace` should be preferred to `re.sub`
Code Smell
Unread "private" attributes should be removed
Code Smell
Cognitive Complexity of functions should not be too high
Code Smell
The first argument to class methods should follow the naming convention
Code Smell
Method overrides should not change contracts
Code Smell
Wildcard imports should not be used

Backticks should not be used Analyze your code

Code Smell Blocker python3

Backticks are a deprecated alias for repr (). Don't use them any more, the syntax was removed in Python 3.0.

Noncompliant Code Example


```
return `num` # Noncompliant
```

Compliant Solution


```
return repr(num)
```

Available In: sonarlint sonarcloud sonarqube


wildcard imports should not be used

 Code Smell


String literals should not be duplicated

 Code Smell

Functions and methods should not be empty

 Code Smell

Server-side requests should not be vulnerable to forging attacks

 Vulnerability

Non-empty statements should change control flow or have at least one side-effect

 Bug