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...

sensitive

Security Hotspot

Constants should not be used as conditions

Code Smell

"SystemExit" should be re-raised

Code Smell

Bare "raise" statements should only be used in "except" blocks

Code Smell

Comparison to None should not be constant

Code Smell

"self" should be the first argument to instance methods

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

Functions returns should not be invariant

Analyze your code

Code Smell | Blocker

When a function is designed to return an invariant value, it may be poor design, but it shouldn’t adversely affect the outcome of your program. However, when it happens on all paths through the logic, it is surely a bug.

This rule raises an issue when a function contains several return statements that all return the same value.

Noncompliant Code Example

```
def foo(a):  # NonCompliant
    b = 12
    if a == 1:
        return b
    return b
```

Available In:

sonarlint | sonarcloud | sonarqube

`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