sonar RULES

Products

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

Code Smell

Methods and field names should not differ only by capitalization

Analyze your code

Code Smell | Blocker | confusing

Looking at the set of methods and fields in a `class` and finding two that differ only by capitalization is confusing to users of the class.

This situation may simply indicate poor naming. Method names should be action-oriented, and thus contain a verb, which is unlikely in the case where both a method and a field have the same name (with or without capitalization differences). However, renaming a public method could be disruptive to callers. Therefore renaming the member is the recommended action.

Noncompliant Code Example

```
class SomeClass:
    lookUp = false
    def lookup():       # Non-compliant; method name differs
        pass
```

Compliant Solution

```
class SomeClass:
    lookUp = false
    def getLookUp():
        pass
```

Available In:

sonarlint | sonarcloud | sonarqube

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