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...

Functions should not have too many lines of code
Code Smell

Track uses of "NOSONAR" comments
Code Smell

Track comments matching a regular expression
Code Smell

Statements should be on separate lines
Code Smell

Functions should not contain too many return statements
Code Smell

Files should not have too many lines of code
Code Smell

Lines should not be too long
Code Smell

Methods and properties that don't access instance data should be static
Code Smell

New-style classes should be used
Code Smell

Parentheses should not be used after certain keywords
Code Smell

Track "TODO" and "FIXME" comments that do not contain a reference to a person
Code Smell

Module names should comply with a naming convention
Code Smell

Type checks shouldn't be confusing

Analyze your code

Code Smell | Major

Checking that variable X has type T with type annotations implies that X's value is of type T or a subtype of T. After such a check, it is a good practice to limit actions on X to those allowed by type T, even if a subclass of T allows different actions. Doing otherwise will confuse your fellow developers.

Just to be clear, it is common in python to perform an action without checking first if it is possible (see "Easier to ask for forgiveness than permission."). However when type checks are performed, they should not contradict the following actions.

This rule raises an issue when an action performed on a variable might be possible, but it contradicts a previous type check. The list of checked actions corresponds to rules {rule:python:S2159}, {rule:python:S3403}, {rule:python:S5607}, {rule:python:S5756}, {rule:python:S5644}, {rule:python:S3862}, {rule:python:S5797}, {rule:python:S5795} and {rule:python:S5632}. These other rules only detect cases where the type of a variable is certain, i.e. it cannot be a subclass.

Noncompliant Code Example

```
def add_the_answer(param: str):
    return param + 42  # Noncompliant. Fix this "+" operatio
    # Note: In practice it is possible to create a class inh
```

Compliant Solution

```
def add_the_answer(param: str):
    return param + "42"
```

See

- Python documentation - `isinstance` function
- Python glossary - "Easier to ask for forgiveness than permission."

Available In:

sonarlint | sonarcloud | sonarqube

Comments should not be located at the end of lines of code

☢ Code Smell

Lines should not end with trailing whitespaces

☢ Code Smell

Files should contain an empty newline at the end

☢ Code Smell

Long suffix "L" should be upper case

☢ Code Smell