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

Function names should comply with a naming convention

Analyze your code

Code Smell Major convention

Shared coding conventions allow teams to collaborate efficiently. This rule checks that all function names match a provided regular expression.

Noncompliant Code Example

With the default provided regular expression: `^[a-z_][a-z0-9_]*$`

```
def MyFunction(a,b):
    ...
```

Compliant Solution

```
def my_function(a,b):
    ...
```

Available In:

sonarlint | sonarcloud | sonarqube

Code Smell

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