

- 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

Class names should comply with a naming convention

Analyze your code

Code Smell Minor convention

Shared coding conventions allow teams to collaborate effectively. This rule allows to check that all class names match a provided regular expression.

The default regular expression is based on PEP-8 standard. It allows "CapWords" convention and "snake_case" in lowercase. The "snake_case" convention is accepted by PEP-8 when the class is primarily used as a callable (ex: decorator, context manager, etc...). However the "CapWords" convention is recommended in every case.

Noncompliant Code Example

With default provided regular expression `^_?([A-Z_][a-zA-Z0-9]*|[a-z_][a-z0-9_]*)*$`:

```
class myClass: # Noncompliant
...

class my_CONTEXT_manager: # Noncompliant
    def __enter__(self):
        pass
    def __exit__(self, type, value, traceback):
        pass
```

Compliant Solution

```
class MyClass:
...

class my_context_manager:
    def __enter__(self):
        pass
    def __exit__(self, type, value, traceback):
        pass
```

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