

- 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

A subclass should not be in the same "except" statement as a parent class

Analyze your code

Code Smell Minor ? unused

Repeating an exception class in a single except statement will not fail but it is not what the developer intended. Either the class is not the one which should be caught, or this is dead code.

Having a subclass and a parent class in the same except statement is also useless. It is enough to keep only the parent class.

This rule raises an issue when an exception class is duplicated in an except statement, or when an exception class has a parent class in the same except statement.

Noncompliant Code Example

```
try:
    raise NotImplementedError()
except (NotImplementedError, RuntimeError): # Noncompliant.
    print("Foo")

try:
    raise NotImplementedError()
except (RuntimeError, RuntimeError): # Noncompliant.
    print("Foo")
```

Compliant Solution

```
try:
    raise NotImplementedError()
except RuntimeError:
    print("Foo")
```


See

- Python Documentation - [Handling Exceptions](#)

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