Code Smell (101)

Search by name...



RPG RPG

Ruby

Scala

Swift

Terraform

Text

Ts TypeScript

T-SQL

VB VB.NET

VB6 VB6

XML XML



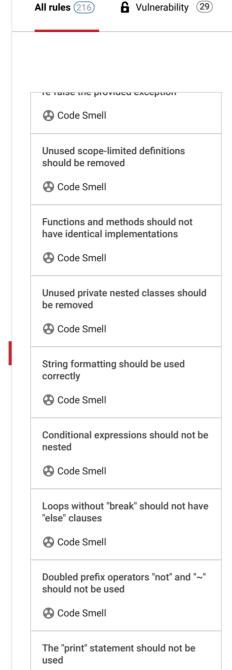
## Python static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your PYTHON code

**∰** Bug (55)

Tags

Security Hotspot 31



Code Smell

inequality

Code Smell

"<>" should not be used to test

Two branches in a conditional structure should not have exactly the

Unused assignments should be

same implementation

Code Smell

Comparison to None should Analyze your code not be constant Code Smell Critical suspicious Checking if a variable or parameter is None should only be done when you expect that it can be None. Doing so when the variable is always None or never None is confusing at best. At worse, there is a bug and the variable is not updated properly. This rule raises an issue when expressions X is None, X is not None, X == None or X != None are constant, i.e. X is always None or never None. **Noncompliant Code Example** mynone = None result = mynone is None: # Noncompliant. Always True. if mynone == None: # Noncompliant. Always True. if mynone is not None: # Noncompliant. Always False. pass if mynone == None: # Noncompliant. Always False. pass myint = 42 result = myint is None: # Noncompliant. Always False. if myint == None: # Noncompliant. Always False. pass if myint is not None: # Noncompliant. Always True. if myint == None: # Noncompliant. Always True. pass See • Python documentation - Identity comparisons • Python documentation - eq operator Available In: sonarlint ⊕ | sonarcloud ♦ | sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the

removed

Code Smell

A field should not duplicate the name of its containing class

Code Smell

Function names should comply with a naming convention

Code Smell

Functions and lambdas should not reference variables defined in enclosing loops

Code Smell

Sections of code should not be

commented out