sonar RULES

Products

- 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

Character classes should be preferred over reluctant quantifiers in regular expressions

Analyze your code

Code Smell | Minor | regex

Using reluctant quantifiers (also known as lazy or non-greedy quantifiers) in patterns can often lead to needless backtracking, making the regex needlessly inefficient and potentially vulnerable to catastrophic backtracking. Particularly when using `.*?` or `.+?` to match anything up to some terminating character, it is usually a better idea to instead use a greedily or quantified negated character class containing the terminating character. For example `<.+?>` should be replaced with `<[^>]*>` or `<[^>]+>`.

Noncompliant Code Example

```
r'<.+?>'
r'".*?"'
```

Compliant Solution

```
r'<[^>]+>'
r'"[^"]*"'
```

Exceptions

This rule only applies in cases where the reluctant quantifier can easily be replaced with a negated character class. That means the repetition has to be terminated by a single character or character class. Patterns such as the following, where the alternatives without reluctant quantifiers are more complicated, are therefore not subject to this rule:

```
/<!--.*?-->/
-/\*.*?\*/-
```

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