- 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... 🔍

Character classes should be preferred over reluctant quantifiers in regular expressions

☢ Code Smell

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

☢ Code Smell

Walrus operator should not make code confusing

☢ Code Smell

Jump statements should not be redundant

☢ Code Smell

"pass" should not be used needlessly

☢ Code Smell

"except" clauses should do more than raise the same issue

☢ Code Smell

Boolean checks should not be inverted

☢ Code Smell

Unused local variables should be removed

☢ Code Smell

Local variable and function parameter names should comply with a naming convention

☢ Code Smell

Field names should comply with a naming convention

☢ Code Smell

Class names should comply with a naming convention

☢ Code Smell

Regex alternatives should not be redundant

Analyze your code

🐛 Bug   ⬣ Major ⓘ   🏷 regex

If an alternative in a regular expression only matches things that are already matched by another alternative, that alternative is redundant and serves no purpose.

In the best case this means that the offending subpattern is merely redundant and should be removed. In the worst case it's a sign that this regex does not match what it was intended to match and should be reworked.

Noncompliant Code Example

```
r"[ab]|a"   # The "|a" is redundant because "[ab]" alre
r".*|a"     # .* matches everything, so any other alter
```

Compliant Solution

```
r"[ab]"
r".*"
```

Available In:

sonarlint ∞ | sonarcloud ⊕ | sonarqube ⌇

Code Smell

Method names should comply with a naming convention

⊗ Code Smell

Track uses of "TODO" tags

⊗ Code Smell

HTML autoescape mechanism should not be globally disabled

🔓 Vulnerability

Variables, classes and functions should be either defined or imported

🐞 Bug

"__exit__" should accept type, value,