
































-  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	
Track uses of "NOSONAR" comments	
Track comments matching a regular expression	
Statements should be on separate lines	
Functions should not contain too many return statements	
Files should not have too many lines of code	
Lines should not be too long	
Methods and properties that don't access instance data should be static	
New-style classes should be used	
Parentheses should not be used after certain keywords	
Track "TODO" and "FIXME" comments that do not contain a reference to a person	
Module names should comply with a naming convention	

## Conditional expressions should not be nested

Analyze your code

 Code Smell

 Major 

 confusing

Just because you *can* do something, doesn't mean you should, and that's the case with nested conditional expressions. Nesting conditional expressions results in the kind of code that may seem clear as day when you write it, but six months later will leave maintainers (or worse - future you) scratching their heads and cursing.

Instead, err on the side of clarity, and use another line to express the nested operation as a separate statement.

### Noncompliant Code Example

```
class Job:
    @property
    def readable_status(self):
        return "Running" if job.is_running else "Failed" if
```

### Compliant Solution

```
class Job:
    @property
    def readable_status(self):
        if job.is_running:
            return "Running"
        return "Failed" if job.errors else "Succeeded"
```

### Exceptions

No issue is raised on conditional expressions in comprehensions.

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