




 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



TypeScript static code analysis

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

- All rules 279
- Vulnerability 27
- Bug 51
- Security Hotspot 43
- Code Smell 158
- Quick Fix 50

Code Smell
Primitive types should be omitted from initialized or defaulted declarations
Code Smell
Non-null assertions should not be used
Code Smell
"undefined" should not be assigned
Code Smell
Trailing commas should not be used
Code Smell
Array constructors should not be used
Code Smell
Quotes for string literals should be used consistently
Code Smell
Statements should end with semicolons
Code Smell
Comments should not be located at the end of lines of code
Code Smell
Loops should not contain more than a single "break" or "continue" statement
Code Smell
Variable, property and parameter names should comply with a naming convention
Code Smell
Lines should not end with trailing whitespaces
Code Smell

"if ... else if" constructs should end with "else" clauses

Analyze your code

Code Smell Critical

This rule applies whenever an `if` statement is followed by one or more `else if` statements; the final `else if` should be followed by an `else` statement.

The requirement for a final `else` statement is defensive programming.

The `else` statement should either take appropriate action or contain a suitable comment as to why no action is taken. This is consistent with the requirement to have a final default clause in a `switch` statement.

Noncompliant Code Example

```
if (x == 0) {
    doSomething();
} else if (x == 1) {
    doSomethingElse();
}
```

Compliant Solution

```
if (x == 0) {
    doSomething();
} else if (x == 1) {
    doSomethingElse();
} else {
    throw "Unexpected value for x";
}
```

Available In:

sonarlint | sonarcloud | sonarqube

<div>Files should contain an empty newline at the end</div> <div> Code Smell</div>
<div>An open curly brace should be located at the end of a line</div> <div> Code Smell</div>
<div>Tabulation characters should not be used</div> <div> Code Smell</div>
<div>Function and method names should comply with a naming convention</div> <div> Code Smell</div>