




 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





## JavaScript static code analysis


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


All rules285

 Vulnerability29

 Bug62

 Security Hotspot43


 Code Smell151

 Quick Fix41


Tags ▾

Search by name... 🔍


Object literal shorthand syntax should be used

 Code Smell


Strings and non-strings should not be added

 Code Smell


Object literal syntax should 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

Function constructors should not be used

Analyze your code

 Code Smell

 Critical



In addition to being obtuse from a syntax perspective, function constructors are also dangerous: their execution evaluates the constructor's string arguments similar to the way eval works, which could expose your program to random, unintended code which can be both slow and a security risk.

In general it is better to avoid it altogether, particularly when used to parse JSON data. You should use ECMAScript 5's built-in JSON functions or a dedicated library.

Noncompliant Code Example

```
var obj = new Function("return " + data()); // Noncompliant
```

Compliant Solution

```
var obj = JSON.parse(data);
```

Exceptions

Function calls where the argument is a string literal (e.g. (Function('return this'))()) are ignored.




See

- OWASP Top 10 2017 Category A1 - Injection

Deprecated

This rule is deprecated; use {rule:javascript:S1523} instead.

Available In:

 |  | 

© 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 property of their respective owners. All rights are expressly reserved.  
[Privacy Policy](#)

https://rules.sonarsource.com/javascript/RSPEC-3523

1/2

<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>