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

# JavaScript static code analysis

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

All rules 285 | Vulnerability 29 | Bug 62 | Security Hotspot 43 | Code Smell 151 | Quick Fix 41

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

## Array constructors should not be used

Analyze your code

Code Smell Minor suspicious

Array literals should always be preferred to Array constructors.

Array constructors are error-prone due to the way their arguments are interpreted. If more than one argument is used, the array length will be equal to the number of arguments. However, using a single argument will have one of three consequences:

- If the argument is a number and it is a natural number the length will be equal to the value of the argument.

```
let arr = new Array(3); // [empty × 3]
```

- If the argument is a number, but not a natural number an exception will be thrown.

```
let arr = new Array(3.14);  // RangeError: Invalid array len
```

- Otherwise the array will have one element with the argument as its value.

```
let arr = new Array("3");  // ["3"]
```

Note that even if you set the length of an array, it will be empty. That is, it will have the number of elements you declared, but they won't contain anything, so no callbacks will be applied to the array elements.

For these reasons, if someone changes the code to pass 1 argument instead of 2 arguments, the array might not have the expected length. To avoid these kinds of weird cases, always use the more readable array literal initialization format.

### Noncompliant Code Example

```
let myArray = new Array(x1, x2, x3);   // Noncompliant. Resu
let emptyArray = new Array();          // Noncompliant. Resu

let unstableArray = new Array(n);      // Noncompliant. Vari

let arr = new Array(3); // Noncompliant; empty array of leng
arr.foreach((x) => alert("Hello " + x)); // callback is not
let anotherArr = arr.map(() => 42); // anotherArr is also em
```

### Compliant Solution

```
let myArray = [x1, x2, x3];
let emptyArray = [];

// if "n" is the only array element
let unstableArray = [n];
// or,  if "n" is the array length (since ES 2015)
```

Files should contain an empty newline at the end

Code Smell

An open curly brace should be located at the end of a line

Code Smell

Tabulation characters should not be used

Code Smell

Function and method names should comply with a naming convention

Code Smell

```
let unstableArray = Array.from({length: n});

let arr = ["Elena", "Mike", "Sarah"];
arr.foreach((x) => alert("Hello " + x));
let anotherArr = arr.map(() => 42);  // anotherArr now holds
```

Available In:

sonarlint | sonarcloud | sonarqube

© 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