[JavaScript] Bad Parts

□□□□□□□□□□□□

with Statement

```
// @Deprecated □□□□□□□
with ([1, 2, 3]) {
console.log(toString()); // 1,2,3
}
```

eval

□□□□□

continue Statement

> The continue statement jumps to the top of the loop. I have never seen a piece of code that was not improved by refactoring it to remove the continue statement.

switch Fall Through

□□□□default

Block-less Statements

□□□□□□□□□□□□

Style □□

> In my own practice, I observed that when I used ++ and--,my code tended to be too tight, too tricky, too cryptic.

Bitwise Operators

□□JS□□□□□□□□□□□□

> In most languages, these operators are very close to the hardware and very fast. In JavaScript, they are very far from the hardware and very slow.

The function Statement Versus the function Expression

□□□hoisted□□□□□□ if □□□□□□□

```
<span style="font-family: '; font-size: 13.028571128845215px">function f() {}; // func

<span style="font-family: '; font-size: 13.028571128845215px">const f = function () {
```

Typed Wrappers

□□□□□□

new

□□□new Function □□□□□□□□ new □□□□□□□blabla□□□
□□□new □□□□□□□ bug □□□□□□□□□ new □□ Java □□□□□□ Java □□ new Class()□

void

> In many languages, void is a type that has no values. In JavaScript, void is an operator that takes an operand and returns undefined. This is not useful, and it is very confusing. Avoid void.

Date: 2026-03-04
Words: 423
Time to read: 2 mins

Older

2025-07-22

HTTP series