

# Theory: JavaScript Code Style

🕒 12 minutes

0 / 3 problems solved

Skip this topic

Start practicing

531 users solved this topic. Latest completion was 2 days ago.

A professional programmer writes code that not only works but also looks clear. It makes it easier for other developers to work with it and saves everybody time. Let's see why this is so important and how to write clear code.

## §1. What is a Style Guide?

Imagine you work in a big IT company where everyone programs in their own style. Let's say you prefer to use two spaces as an indent, name variables with camelCase, and add an empty string between statements. Another developer likes to use four spaces as an indent, prefers underscore notation, and removes all the blank lines.

When you look at code written by another developer, you have to spend a lot of time understanding it because you are used to different code style principles. It will be hard for you to find errors in another developer's code and maintain it. To minimize difficulties, it is better to have some general rules for writing code. It also helps you write code faster because you no longer need to think about minor style principles.

The list of best practices for writing code is called the **style guide**. Following it makes your code understandable and easy to read by other developers.

In most cases, companies do not need to create their own style recommendation because they can follow the style guides created by other companies. The most popular are:

- [Airbnb JavaScript Style Guide](#)
- [Google JavaScript Style Guide](#)

## §2. Airbnb JavaScript Style Guide

Let's take a look at the most common style guide, the **Airbnb JavaScript Style Guide**. This manual describes recommendations for spaces, empty strings, semicolons, line length, and so on. Let's consider a few of them!

1. Sometimes developers write long lines of code; to read them fully, you need to use horizontal scrolling. The best alternative is to set the maximum line length and move the other characters below. As stated in the Airbnb JavaScript Style Guide, 100 characters for a string is sufficient.

```
1 // good
2 const thePaymentIsActive = isPaymentFromToday(payment.date)
3   && isPaymentFromAvaliableAdresses(payment.id)
4   && isPaymentToShow(payment.id)
5
6 // bad
7
const thePaymentIsActive = isPaymentFromToday(payment.date) && isPaymentFromAvalia
bleAdresses(payment.id) && isPaymentToShow(payment.id)
```

2. There is an eternal argument between programmers whether to use spaces or tabs as an indent. The Airbnb JavaScript Style Guide provides an answer to this question and recommends using two spaces. If you like to use tabulation, you can set it to two spaces.

Current topic:

JavaScript Code Style ...

Topic depends on:

✗ Template literals ...

✗ Conditional operators ...

Table of contents:

[1 JavaScript Code Style](#)

[§1. What is a Style Guide?](#)

[§2. Airbnb JavaScript Style Guide](#)

[§3. Linters](#)

[§4. Conclusion](#)

[Feedback & Comments](#)

```
1 // good
2 function myFunction() {
3     const name = "Alex";
4 }
5
6 // only one space is bad
7 function myFunction() {
8     const name = "Alex";
9 }
10
11
12 // four spaces are bad
13
14 function myFunction() {
15
16     const name = "Alex";
17
18 }
19
```

3. In functions, you should place one space before the first bracket.

```
1 // good
2 function show() {
3     console.log("My name is Helen");
4 }
5
6
7 // bad
8 function show(){
9     console.log("My name is Helen");
10 }
11
```

4. In the control statements, it is better to put only one space before the opening bracket.

```
1 // good
2 if (a === b) {
3     console.log(a)
4 }
5
6 // bad
7 if(a === b) {
8     console.log(a)
9 }
10
```

5. In the case of method chaining, it is a good idea to break it into multiple lines, each calling a new method.

```
1 // good
2 document
3     .getElementById("reviewsContainer")
4     .classList
5     .remove("active");
6
7 // bad
8 document.getElementById("reviewsContainer").classList.remove("active");
9
```

6. You should only use one empty line to separate your code.

```

1 // good
2 if (a > b) {
3   console.log ("a is bigger than b");
4 }
5
6 // bad
7 if (a > b) {
8
9
10   console.log ("a is bigger than b");
11
12
13 }

```

7. It's good practice to terminate the file with a single line character and not to use extra spaces at the end of lines of code.

```

1 // good
2 function myFunction() {
3   // ...
4 }
5
6
export default myFunction;␣ // the ␣ symbol is used instead of an extra space for clarity
7
8 // bad
9 function myFunction() {
10   // ...
11 }
12
13
14 export default myFunction;
15
16
17 // bad
18 function myFunction() {
19   // ...
20 }
21
22
23 export default myFunction;␣ // the ␣ symbol is used instead of an extra space for clarity
24
25 ␣ // the ␣ symbol is used instead of an extra space for clarity

```

8. Don't forget to write the semicolon at the end of your statements. Otherwise, JavaScript may split your code incorrectly.

```

1 // good
2 const a = 5;
3 const b = 7;
4 const c = 10;
5
6 // bad
7 const a = 5
8 const b = 7
9 const c = 10

```

9. It is better to use **camelCase** for naming functions, objects, and methods. **camelCase** means that for names consisting of multiple words, the first word should start with a lowercase letter, and the rest of the words start with a capital letter.

```
1 // good
2 function myFunctionToCheckValues() {
3   //...
4 }
5
6 // bad
7 function MyFunctionToCheckValues() {
8   //...
9 }
10
11
12 // bad
13
14 function myfunctiontocheckvalues() {
15
16   //...
17
18 }
```

Above, we have reviewed ten essential rules for writing high-quality code. The official documentation has over a hundred recommendations on how to do this: check them out in the full [Airbnb JavaScript Style Guide](#).

### §3. Linters

You have learned a few rules for writing clear code. Now the question is: how do you remember them all when you write code? We are ready to offer you a tool that will help you check all patterns automatically: **linters**. Now you don't need to remember all the patterns.

With linters, you need to specify the necessary settings, and the tool will notify you if you wrote something wrong. The most popular linters are:

- [JSLint](#) — a linter without an extensive list of configuration settings.
- [JSHint](#) — this style has more config parameters than JSLint.
- [ESLint](#) — one of the most common linters. It can not only show you errors but also fix your scripts automatically.

### §4. Conclusion

It's a nightmare to maintain code written in different styles. Each company should create a unified style pattern or use one that already exist, such as Airbnb JavaScript Style Guide. Also, linters can help you monitor compliance with the rules when writing code. Please don't forget to use style guides and linters in your code: this will save you and other developers a lot of time and energy.

 Report a typo

67 users liked this theory. 2 didn't like it. What about you?



Start practicing

Comments (6)

Hints (0)

Useful links (0)

Show discussion