

JavaScript Best Practices & Coding Standards

Reference: Airbnb JavaScript Style Guide

Official URL: <https://github.com/airbnb/javascript>

This is the gold standard for JavaScript style and best practices. Interns should bookmark and refer to it often.

General Guidelines





Rule	Standard
Max lines per function	◆ 30-50 lines max per function. If it grows longer, break it into smaller helper functions.
Max function parameters	◆ Prefer no more than 3 parameters . Use an object if more are needed.
Line length	◆ Limit lines to 100–120 characters (Airbnb uses 100).
Indentation	◆ Use 2 spaces (no tabs).
Semicolons	◆ Always use semicolons .
Quotes	◆ Use single quotes ' for strings unless you need interpolation (<code>` ` `</code>).
Trailing commas	◆ Use trailing commas in multi-line objects/arrays.

Naming Conventions

Type	Convention	Example
Variable & function names	camelCase	getUserData ()

Constants	UPPER_CASE_SN AKE	MAX_RETRIES
Classes & Constructors	PascalCase	UserModel
Private variables (JS # syntax)	Prefix with #	#internalId




Functions & Methods

-  Prefer **arrow functions** for callbacks and anonymous functions.
-  Avoid using `function` keyword unless for named functions or constructors.
-  Use **default parameters** when possible.
-  Keep functions **pure** – avoid side effects.



```
// Good
const getFullName = (first, last) => `${first} ${last}`;
```


```
// Bad
function getFullName(first, last) {
  return first + " " + last;
}
```

Objects & Arrays

-  Prefer **object destructuring** and **array spread/rest** syntax.
-  Clone objects using `{ ...obj }` or `structuredClone()` (if supported).
-  Avoid `for...in` with arrays. Use `for...of` or `array.forEach()`.

Conditions & Booleans

-  Use **truthy/falsy** checks clearly.
-  Avoid complex nested ternaries.

-  Use `??` and `?.` (nullish coalescing & optional chaining).

// Good

```
const value = user?.preferences?.theme ?? 'light';
```

Code Quality

- Use **ESLint** with Airbnb config:

```
npm install eslint-config-airbnb-base eslint-plugin-import --save-dev
```

In `.eslintrc`:

```
{  
  "extends": "airbnb-base"  
}
```

- Use **Prettier** to auto-format your code:



```
npm install --save-dev prettier
```

Testing & Debugging

- Always **write unit tests** for reusable functions.
- Use `console.log()` during development, but remove before production.
- Use proper **error handling** with `try/catch` and log meaningful errors.

Common Mistakes to Avoid

-  Avoid global variables.

-  Avoid callback hell – use `async/await`.
-  Don't mix `var`, always use `const` or `let`.

Must-Follow Airbnb Highlights

Rule	Description
<code>const</code>	Use <code>const</code> by default, <code>let</code> only when reassignment is needed
Template Strings	Always use backticks <code>`</code> for strings with variables
Shorthand Properties	Use <code>{ name }</code> instead of <code>{ name: name }</code>
Method Shorthand	Use <code>sayHi() {}</code> instead of <code>sayHi: function() {}</code>

Tools to Enforce These Standards

- **ESLint** (Airbnb base config)
- **Prettier**
- **Husky + lint-staged** for pre-commit checks
- **Jest or Vitest** for unit testing