Kayla Young

Professor David Conner

ICS 314

26 September 2024

Coding Standards Are the Foundation of Quality Code

Coding standards are often considered as the fine details of software development that minor visual aspects such as indentation or brace placements. While these may seem trivial, coding standards extend far beyond visual preferences. They can serve as the backbone of maintainable, scalable, and robust software. Not only do they enforce consistency and readability, but they also serve as an effective learning tool for developers, especially when learning a new language or framework.

On the surface, coding standards might seem like a set of rules that developers follow to maintain uniformity. But, they contribute to much more than that. For example, ESLint, is a widely used tool for enforcing coding standards in JavaScript and TypeScript. When I first started using ESLint with VSCode, I thought it would simply flag errors and help me correct basic syntax issues. But in practice, ESLint does more than that, it also teaches new things and encourages the user to problem solve. When ESLint points out an error, it can guide you toward better practices. It forces you to adhere to language-specific patterns that you might otherwise overlook. For instance, ESLint's suggestions on avoiding unused variables or preferring const over let where possible can nudge developers toward writing cleaner, more efficient code. In this way, coding standards facilitate an immersive learning process, where you gradually internalize language rules through consistent application.

When I first started using ESLint, I found resolving all the errors to be a bit tedious, especially when the code seemed functional despite the warnings. It felt like chasing after minute details that did not impact the core logic. However, the process slowly became more useful than painful. As an example, I learned the importance of being consistent with a code style and not just for myself, but for others who might work on the same codebase in the future. This collaborative aspect of coding standards is often overlooked but essential. When multiple developers follow the same conventions, the project becomes easier to maintain, debug, and scale.

What initially felt like over-polishing code turned into a valuable lesson: coding standards are there not to enforce perfection but to prevent small issues from growing into larger, harder-to-track problems. Linting tools like ESLint become particularly useful when working on large projects as maintaining code quality across many files and contributors is crucial.

Coding standards are not just about personal habits, but they are important when collaborating as a team. In large codebases or projects, consistency is key. Imagine reading code written by five different developers, each with their own style. Even if the logic is perfect, the experience is disorienting and unnecessarily complicated. Coding standards act as a common framework that binds individual contributions into a cohesive whole. By standardizing how code is written, reviewed, and shared, teams can work more efficiently and spend less time deciphering each other’s code.

Moreover, coding standards can prevent common mistakes. For example, ESLint can warn you about accidental global variable declarations or unused functions, issues that are easy to overlook but can cause significant bugs in production. Without such standards, these errors might slip through, leading to preventable downtime or costly debugging.

Initially, I viewed coding standards through a narrow lens as something that was cosmetic or a burden in some ways. But my perspective has since shifted. There's a lot more to coding standards than enforcing stylistic choices. They foster a deeper understanding of the language, encourage better coding practices, and significantly improve code quality. While adhering to them can feel tedious at times, the long-term benefits outweigh the short-term pain. The more you work with them, the more you realize their value in making code understandable, reusable, and resilient.

All in all, coding standards are not merely about correcting the syntax of your code. They represent a commitment to clarity, consistency, and collaboration in software development. As my experience with ESLint has shown, these standards help us learn the nuances of a language and reinforce good practices that lead to higher-quality code. Far from being a trivial aspect of software engineering, coding standards are an essential tool for any developer striving for excellence.