Clean code is all about writing code that is easy to understand, maintain, and modify. It's code that prioritizes clarity and readability, making it a joy to work with for both the original developer and anyone who comes after. Here are some key principles of clean code:

Readability:

- **Meaningful Names:** Use descriptive variable and function names that clearly convey their purpose. Avoid cryptic abbreviations or single-letter names.
- **Comments Sparingly:** Comments should be clear and concise, explaining non-obvious logic or complex sections. However, well-written code often speaks for itself, so avoid excessive commenting.
- **Consistent Formatting:** Adhere to established coding conventions (e.g., indentation, spacing) to improve readability and maintainability.

Simplicity:

- **KISS (Keep It Simple Stupid):** Favor simpler solutions over complex ones. Avoid unnecessary abstractions or over-engineering.
- **Short Functions:** Break down large functions into smaller, more focused functions that perform a single task. This improves code modularity and reusability.
- **DRY (Don't Repeat Yourself):** Eliminate code duplication. If you find yourself writing similar logic in multiple places, refactor it into a reusable function or class.

Maintainability:

- **Boy Scout Rule:** Leave the codebase cleaner than you found it. This means refactoring messy code, fixing bugs, and documenting your work.
- **Error Handling:** Implement proper error handling to gracefully manage unexpected situations and provide informative error messages.
- **Testing:** Write unit tests to verify the correctness of your code. This helps catch regressions and increases confidence in code modifications.

Additional Considerations:

- **Follow Established Standards:** Use common coding conventions and style guides specific to your language or project to ensure consistency.
- **Think About the Future:** Write code with the future in mind. Consider how the code might need to be changed or extended in the future.

By following these principles, you can write clean code that is not only functional but also a pleasure to work with. There's a great book by Robert C. Martin called "Clean Code: A Handbook of Agile Software Craftsmanship" that dives deeper into these principles and offers practical advice for writing maintainable code.