

Clean Code Development Cheat Sheet

Descriptive Naming:

Aim for names that express the goal or objective of the code by giving variables, functions, classes, and other entities names that are both clear and descriptive.

Keep It Short and Sweet:

Focusing on a specific task and keeping procedures and functions brief. In general, shorter functions are simpler to comprehend and update.

Comments Where Necessary:

Adding remarks in moderation and emphasizing the why over the what when doing so. Code ought to be self-explanatory but, when necessary, comments can offer context.

Consistent Formatting:

Keeping your formatting and coding standards constant across your codebase. This covers space, indentation, and other artistic components.

Avoid Magic Numbers and Strings:

To make code easier to read and maintain, magic numbers and strings should be swapped out for named constants or variables.

Error Handling:

Providing clear error notifications and gracefully handling problems. preventing quiet failures and reporting or logging faults as appropriate.

Refactor Regularly:

Refactor the code frequently to enhance its organization and design. Refactoring enhances maintainability and helps get rid of code smells.

Unit Testing:

To ensure that the code is correct, write unit tests. This guarantees that the code functions as intended and makes future modifications and updates easier.

DRY Principle (Don't Repeat Yourself):

Preventing redundant code. Incorporating similar functionality into classes, modules, or methods to encourage reuse and cut down on duplication.

Version Control Best Practices:

Making good use of version control systems. Writing insightful commit statements, branch according to the demands of your project, and commit often.