Writing better code and best coding practice for programmers:

1.Keep It Simple:

The first and the most basic tip, is to keep your code as simple and readable as possible. Don't get fancy if you don't have to, and don't over-complicate problems (a common issue among software developers). By keeping it simple you can produce higher quality code, solve problems faster, work better in developer groups and have a more flexible code base, among other things.

2. Make Sure Your Code Is Readable:

Yes, we all want to be as efficient and optimised as possible. And sometimes coders may want to show their skill and talent by writing an entire method or function in one line. After all, many of us became coders because we love to solve complex coding challenges. But it doesn't matter what challenge you solved or how optimised your code is if no one can read it.

Understand Your Code:

As a beginner, even if you are writing simple code having an 'if else' statement, start by realising the code on a piece of paper. The algorithm and the whole compiler process will look more meaningful once you understand the idea behind the code. Even for experts, the best way to solve a complex problem or formulate an algorithm to solve any complex problem is by breaking it into sub-parts and then try to formulate a solution for each. Once you start internalising the code and solving problems you will quickly build up your confidence.

3.Comment And Document:

When it comes to coding best practices, one habit that could help you is to start each function or method you create with a comment. In this comment, you can outline exactly what the function or method does, what its parameters are, and what it returns. You'll also want to describe any possible errors or exceptions. You can also summarise the major steps if your code is particularly complex, as well as outline the role of each file and class and any contents in each class field.

4.Plan Your Approach:

Before you dive straight into the problem and get your hands dirty with some code, it is essential for the programming enthusiast to plan their approach accordingly. It is not always the best idea for you to jump into coding for a complex problem. It is usually a better proposition to effectively plan your course of action and dwell on the specifics.

5. Don't Repeat Yourself (DRY):

The DRY principle, formulated by Any Hunt and Dave Thomas in *The Pragmatic Programmer*, is the use of functions, classes and instances to allow you to avoid retyping code that has already been written once. This fundamental principle allows developers to avoid duplication to produce much cleaner code compared to the programmer who uses unnecessary repetition. Optimising the code is what often separates a great coder from an average one.

6. Indent Your Code:

Imagine you go to a supermarket and there is no consistency over how the items are placed in the area. Some dairy products are at the clothing sections, others are at the cosmetics area, and bread products are placed with the vegetables. Indentation in the code is much like the arrangement that you need in a supermarket or any other place in the real world. When your code is indented, it becomes more readable and easier to find what you're looking for.

7. Naming Convention:

This is one tip that keeps popping up in every single article about the correct way of working on any programming language, and still people tend to forget or neglect it. Having a proper naming convention is extremely important in a code as the doors for future edits and updating is always wide open. Having irrelevant or contradicting names to your pages, variables, functions or arrays will only create troubles for you in the future. Therefore, name elements on the basis of what they are and make it a habit to maintain a convention throughout your code.

8.Develop Projects:

The best part about coding with any programming language is the large heaps of fabulous projects that you can develop with them. You can create a ton of unique and amazing projects to add to your portfolio or resume, or you can construct these projects just for fun to learn something new and gain further knowledge.

10. Effective Debugging:

One of the essential aspects of programming is encountering and tackling errors. Effective debugging is a crucial aspect that every coder must get used to because no matter how good you are at solving programming blocks or questions, you will always enter a situation where you are stuck at the problem, and you don't have an idea on how to proceed.
5. Develop Projects:

The best part about coding with any programming language is the large heaps of fabulous projects that you can develop with them. You can create a ton of unique and amazing projects to add to your portfolio or resume, or you can construct these projects just for fun to learn something new and gain further knowledge.

11.Avoid Hard-Coding

As much as possible, avoid hard-coding anything, other than constants. System-level settings, usernames, passwords, and other configuration information should never be hard-coded. Some developers may try to take shortcuts when developing prototypes, and then these prototypes can find their way into production environments. This is the opposite of best coding practices. Hard coding in these cases is not technical debt, but could be seen as irresponsible with major consequences. Should that code ever become accessible, it represents a huge security risk when endpoints and access credentials are exposed.

12.Test Runs

Having a rectangular box on display because the browser was not capable to output the content is a major turn off for users. To avoid these situations as a developer, your final work must be to check how your website displays on different devices and across different and make the required changes accordingly. The saying, "Don't judge a book by its cover," does not hold true for programming languages because the better your display looks, the more appreciation your work gets.