

## EMERGING METHODS FOR EARLY DETECTION FOREST FIRES DEEP LEARNING

TEAM ID:PNT2022TMID30386

### The Importance of Code Readability

What is code readability?

Readable code is simply code that clearly communicates its intent to the reader. Most likely, the code we write will be read by other developers, who will either want to understand or modify the way our code works. They may need to test the code, fix a problem, or add a new feature.

These other developers reading our code could be our teammates, a consultant, a new junior developer, or even developers from another company who are programming a specific portion of the application. And more importantly, that developer could be ourselves, when we revisit our own code. With most languages, the ratio of time spent reading vs. writing is well over 10:1.

What is good code?

Good code must satisfy 2 qualities:

It must be correct, it must produce a result that is expected when executed.

The code must be easily readable and understandable by ourselves and other developers.

Why should we produce readable code?

Since our software code can be shared with other developers, we'll want to make it easier to work with this shared code. To do this, we'll have to make sure that everyone involved, including ourselves, understands the same thing easily and quickly. Code that is readable for one person is not necessarily readable for another.

Code that is not readable takes more time to understand and can make us lose a lot of time on what should be a simple task. The worst-case scenario is that it could even take several iterations to fix some problems. In some cases, we might spend so much time trying to understand code that we might want to rewrite it completely.

Poorly written code would even make refactoring difficult. When rewriting poorly written code, by removing the original code and writing a new implementation, we might not cover all the use cases of that code, especially if the documentation of the requirements is limited or absent. As

a result, the time spent rewriting will be greater than the time it took to write the original code.

Furthermore, by not understanding the code well, we may misinterpret its use and by modifying the code, we may unintentionally create new problems. Code that is not easily readable can therefore increase the risk of defects.

### Benefits of Creating Readable Code

On the other hand, readable and well-tested code is what makes it easier to refactor, extend and modify parts of the system because it is easier and less time-consuming to understand. Readable and well-tested code is the foundation of a solid base, where developers are confident and quick to make changes.

Good readable code obviously takes more effort to produce than quickly written and poorly planned code. At Direct Impact, we believe that the benefits of readable code win over quickly produced code that may contain more errors.

In order to produce good readable code, we will have to follow good coding practices.

How do we make our code more readable?

It would be easy to say that by simply adding comments to our code, it will automatically become more readable. But readable code involves much more. Adding relevant comments everywhere in our code would be a very laborious task and could become overly commented code. Overly commented code would have the opposite effect, making the code harder to read and understand.

Here are videos we've produced on improving code readability without using comments, and differentiating between meaningful and not meaningful comments.

This picture shows a cursor in a block of code adding spacing between different elements of code

Developers can enhance code readability by adding spacing between blocks of code

Another important point is that in a multilingual company, like Direct Impact Solutions, translation of these comments could, at some point, be necessary. So we have to find a balance between several techniques. So what can we do to make our code easier to understand without having to comment on everything?

These four code readability best practices are a good place to start:

Have a naming convention

Take care of your code

Have a well-organized relationship diagram

Use reusable code

These four categories of best practices can have positive (or negative if not applied properly) impacts on the performance of our applications. We will continue to illuminate these concepts in our code readability blog series. We hope you find this information helpful and easy to reproduce

How To Write Readable Code With Code Examples

How To Write Readable Code With Code Examples

In this tutorial, we will try to find the solution to How To Write Readable Code through programming. The following code illustrates this.

1. Always name your properties or variables in such a way that leaves absolutely no room for confusion/misunderstanding/ambiguity
2. Always capture a complicated piece of logic in a very descriptive, self-reading variable.
3. Avoid double negative (``const isNotBlocked = false` --> `const isBlocked = true``)

We have shown how to address the How To Write Readable Code problem by looking at a number of different cases.