Guide to reviewing my code:

Hello, my name is Joon. This document is a user guide to suggest how to review and revise my code, so that I could improve my code by absorbing your insights the best way I can. This document includes my views on a successful code review, my communication methods, my unintended habits that may annoy you, and in general about me that could let you know more about what parts to focus on during the code review. Thank you for agreeing to participate in letting me improve - Here is a guide to working with me.

How I view success

In programming, I am able to write readable and functional code with the least number of complications. Most of the time, the code is short, simple and works without much maintenance most of the time. During a code review where I get to hear the insights and knowhow of other developers, I want to improve on the efficiency of the code and the architecture to make code more efficient.

In terms of a successful code review, I think that having a smooth and well structured review session for both the reviewer and the reviewee is important. The reviewers would need to receive the required code before any deadlines, and the developers would need to streamline the process of understanding and accepting the comments that have been issued on the code. Also, for the developer, the code would have to be documented in an agreed upon fashion or at least in a clean, standardized way for the reviewer to comprehend quickly. This can be done using documentations, doc strings, testing, formatting and refactoring, variable naming conventions et cetera.

I do not mind big overhauls of architecture of the code if it is deemed necessary between the collaborators, or if the review session reveals possible efficiency upgrades and optimizations. I believe that success comes in creating a higher quality code with optimized versions of what I previously had - more work is not a problem.

How I communicate

Feedback is very important to me. I do realize that in a team, there are things that I miss and misunderstand. I don’t take comments and issues personally in a working environment unless they attack me directly on skills, or other personal issues. The interaction would proceed with me asking questions and taking notes on where I could improve on and making different methods on solving these issues. Location and means of communications rarely matter, but I do prefer face to face. I don’t have much sensitivity to the feedback themselves, so feel free to give comments without spinning your words. Efficiency is key to a quick and effective code review. I struggle to express my concerns in my code due to not being able to phrase them correctly, but most of the time these issues I try to mention are resolved by the reviewer. I am contactable by messenger, email or phone, and I do not have set work hours.

Things I do that may annoy you

I am currently working on detail. When I code, I look at the big picture and worry about the functionalities working, and not too much on how small things are looking. Before working on git hooks and actions that do the small detailing and checks before me, I had issues in committing code too many times because I found small errors that killed the program one by one. The implementation of pylint to most of my projects have helped address this issue vastly, and the code I present to you will not contain these issues - however, while we work together, it would be great to discuss how we could implement changes as it would be great to focus on detail along with the big picture.

My code may have multiple flow issues, because of the problems I mentioned above. While I want the code to be efficient and pretty, my priorities are to get the code working first, then fix the errors. I am trying to work on formatting and creating a flowchart of how the program would work to make it more efficient and readable. I am currently using python's black and other refactoring tools, but please comment on how I could approach this better during our review.

I also am considered to be a very skeptical person. It's a personality thing that I am in the progress of improving, and I have come a long way. On the occasions where I am pessimistic about something that annoys you, please point it out.

My strengths

All in all, I love coding. I like it when I can see myself improving in the quality of code that I write, as well as just seeing things work after a long period of struggling to execute some tasks. I am very open to comments and pointers that may improve my programming in a more efficient and standardized way. I do not fear big changes in the way I code, which is why I will respect and appreciate your collaboration style in working together, or being in a code review session. I learn and adapt quickly, so the pointers you provide will not go to waste.

My growth areas

As I have mentioned in the section "Things I do that may annoy you", I am working on coding with minor details, testing after checking bugs that are blatant, and working on a more efficient flow. These are all areas that I had been commented on before in prior code reviews. Please point out any other aspects where I could work on to improve how I code as a programmer.

Thanks!