Rules for Good commit messages

Good commit messages are crucial in managing and maintaining any collaborative project, or even for your future self to understand changes made. Here are some best practices for writing commit messages:

1. ***Use clear and concise language**: The person reading your commit message should immediately understand what the commit does. If the message is unclear, it can cause confusion and may lead to unnecessary mistakes.*
2. ***Use the imperative mood**: This is more of a convention rather than a rule, but it's widely followed. Messages should preferably be in the format "Fix bug" or "Add feature" instead of "Fixed bug" or "Added feature."*
3. ***Start with a short summary**: The first line of the commit message should be a brief summary of what changes the commit contains. This summary should not exceed 50 characters as it helps in quickly scanning through the commit history. It's often referred to as the subject of the commit.*
4. ***Detail in the body**: If the commit is complex and involves a lot of changes, include a detailed explanation in the body of the commit message, after the summary. It's a good place to explain the 'what' and 'why' of the changes, not the 'how.' Each line should be wrapped around 72 characters.*
5. ***Reference relevant issues**: If the commit is related to an issue in the project's issue tracker, it's a good practice to reference those issues in the commit message. This makes it easier for other contributors or even your future self to find more context about the change.*
6. ***Avoid generic messages**: Messages like "bug fixes" or "updates" are not helpful to anyone. Every commit message should clearly explain the purpose of the commit.*
7. ***Use bullet points for multiple changes**: If your commit includes several changes (although it's better to make one commit per significant change), list them as bullet points in the body of the commit message.*

Remember, the key to a good commit message is clarity. A good commit message can save a lot of time for anyone looking at your commit and trying to understand what it does, including your future self.