1. structure and aggregate

- others and your future self
    - to understand the purpose and functionality
    - of different sections of the code.

- balance
    - Too many comments can make the code cluttered and harder to read,
    - Too few can leave readers confused.

- **Describe the purpose**:
    - Explain what the code is intended to do, rather than how it does it. The code itself shows the "how", so focus your comments on the "why".

- **Keep it concise**:
    - Long comments can be just as confusing as no comments at all. Try to be clear and concise in your explanations.

- **Update your comments**:
    - If you update your code, don't forget to update your comments as well. Outdated comments can be misleading.

- **Don't state the obvious**:
    - Comments should provide additional insight, not repeat what's already clear from the code.

Source(s)

1. en.wikipedia.org

2. why

- Organization:
  - Comments can help divide the code into logical sections,
    - making it easier to locate specific parts of the code[12].
  - This is especially helpful in large codebases.

- Context:
  - Comments provide context about what the code is supposed to do[123].
    - which can be particularly useful when the purpose of the code
    - is not immediately apparent from reading it.

- Onboarding:
  - For new team members or contributors, comments can serve as a guide to understanding the codebase[12].
  - They can quickly grasp the purpose and functionality of different parts of the code.

- Maintenance:
  - When updating or debugging the code, comments can help developers understand the original intent of the code[124].
  - making it easier to make changes without introducing bugs.

- Collaboration:
  - In a team setting, comments can facilitate better collaboration by providing a clear understanding of the code's functionality to all team members[12].

- Used judiciously:
  - Over-commenting, or stating the obvious, can make the code more cluttered and harder to read[12].
    - cluttered
    - harder to read.
  - The best comments provide insight and context that the code alone cannot[12].
    - insight
    - context

Source(s)

1. [Best practices for writing code comments - Stack Overflow](#)

2. [Putting comments in code: the good, the bad, and the ugly.](#)

3. [Comments: Definition, Types, Uses, Syntax for Different Languages - Toppr](#)

4. [Programming - Commenting - University of Utah](#)