VI. Most computer languages including Python enable the inclusion of explanatory comments. Some developers do this as a matter of course, others feel with appropriate structure, function and data names, etc. their code is self-documenting and does not require comments. Please share your thoughts about this.

Ans.

The practice of including explanatory comments in code is a topic of debate among developers, and there are different schools of thought regarding its necessity.

Advantages of Including Explanatory Comments:

I. Code Understanding: Comments can provide additional context and explanations that help other developers (including yourself) understand the code's purpose, logic, and any complex algorithms or business rules involved.

II. Documentation: Comments can serve as a form of documentation, describing the code's behavior, inputs, outputs, and any assumptions or constraints.

III. Collaboration: When working on a team, comments can facilitate collaboration by enabling clear communication and making it easier for team members to understand and maintain the codebase.

Advantages of Self-Documenting Code:

I. Readability: Well-structured code with meaningful names for functions, variables, and classes can often convey its purpose and behavior without needing excessive comments.

II. Maintenance: By writing clean and self-explanatory code, you reduce the reliance on comments for understanding the code, making it easier to maintain and modify in the future.

III. Simplicity: Reducing the number of comments can lead to more concise and focused code, reducing clutter and potential confusion.

IV. Finding a Balance: The best approach lies in finding a balance between comments and self-documenting code. While self-explanatory code is valuable, there are scenarios where comments can greatly enhance understanding, especially for complex logic or critical sections.

Here are some guidelines to consider:

I. Use Comments Sparingly: Focus on using comments where they add significant value or when the code's purpose may not be immediately obvious.
II. Be Clear and Concise: Write comments that are clear, concise, and to the point. Avoid unnecessary or redundant comments.

III. Explain Why, Not What: Instead of describing what the code does (which should be evident from the code itself), focus on explaining the rationale behind the code design. choices or any important considerations.

IV. Update Comments: Remember to keep comments up to date as the code evolves. Outdated or misleading comments can be more harmful than no comments at all.

Ultimately, the goal should be to write code that is both readable and maintainable without relying solely on comments. Strive for self-documenting code by using meaningful names, following best practices, and organizing code in a logical manner. When necessary, judiciously include comments to provide clarity and context where code alone may not suffice.