



# Review GitHub Copilot's "explain" and "document" features

5 minutes

GitHub Copilot makes it easier for developers to work with complex or unfamiliar code. In Visual Studio Code, developers have direct access to GitHub Copilot features that analyze, explain, or document code.

## Analyze and explain code using GitHub Copilot

The following GitHub Copilot features can be used to analyze and explain code:

- Ask mode: Use the Chat view in ask mode to analyze your entire workspace and ask questions about coding topics, technology concepts, and general programming practices.
- Inline Chat: Start an inline chat session directly from the editor to ask about specific code sections.
- Smart Actions: Run the Explain smart action to explain selected code without even having to write a prompt.
- Quick Chat: Ask a quick question and get back into what you're doing.

## Generate inline code documentation using GitHub Copilot

The following GitHub Copilot features can be used to generate inline code documentation:

- Ask mode: Use the ask mode to suggest inline code documentation for a code file, class, or method. Review the suggested inline documentation in the Chat view. Request updates as needed in the chat conversation. Once you're satisfied with the proposed documentation, you can insert the suggested documentation into the corresponding code file.
- Edit mode: Use the edit mode to suggest inline code documentation for a code file, class, or method. Review the suggested documentation in the editor, navigating between

suggested updates. Accept or reject individual suggestions, or modify suggestions as needed.

- **Agent mode:** Use the agent mode to autonomously generate inline code documentation for specified code projects, files, classes, or methods. Have the agent perform other tasks during or after the documentation process. Accept or reject the updates when the agent has completed the assigned tasks.
- **Inline chat** - Use an inline chat session to work directly in the code editor. The inline chat is useful when you want to document specific code (classes, methods, or code blocks) without leaving the editor. You can provide specific instructions in the prompt or use the `/doc` slash command without specific information if you want. Updates are displayed in the code editor and can be accepted or rejected. You can also choose the option to rerun the prompt with or without updates.
- **Generate Docs smart action:** Use the Generate Docs smart action to generate inline code documentation for a selected class, method, or code block. The Generate Docs smart action is useful when you don't have specific requirements that would require a prompt. Updates are displayed in the code editor and can be accepted or rejected. You can also rerun the smart action.

## Generate project documentation using GitHub Copilot

The following GitHub Copilot features can be used to generate project documentation:

- **Chat view in ask mode:** Use the ask mode to analyze the codebase and gain an understanding of the code projects, files, and classes. Use ask mode to preview suggested project documentation. Review suggestions and request updates or other details as needed. Insert the suggested documentation into a project documentation file (README.md).
- **Chat view in Edit mode:** Generate a project documentation file (README.md).
- **Chat view in agent mode:** Generate a collection of project documentation files: create/update the README.md, create/update the UsageExamples.md, and create/update the ChangeLog.md. Include links between the documentation files, cross-reference classes and methods, and ensure consistency across docs.

### Important

When you use the Chat view in agent mode, GitHub Copilot may make multiple premium requests to complete a single task. Premium requests can be used by user-initiated prompts and follow-up actions Copilot takes on your behalf. The total premium requests used will depend on the complexity of the task, the number of steps involved, and the model selected.

## Summary

GitHub Copilot helps you analyze and document your code quickly and accurately. The chat view, inline chat, quick chat, and smart action features provide various levels of interaction directly within the Visual Studio Code environment.

### Next unit: Examine GitHub Copilot prompts and keywords

[< Previous](#)[Next >](#)