GitHub Copilot: How to Use AI to Improve Your Coding Workflow



Artificial intelligence is gradually transforming the work of software developers. With GitHub Copilot, developers can receive code suggestions, understand existing programs, and accelerate repetitive tasks directly within their development environment.

However, GitHub Copilot does not replace technical skills or critical thinking. It should be treated as an assistant that proposes solutions for developers to review, test, and improve.

What Is GitHub Copilot?

GitHub Copilot is an AI-powered coding assistant. It integrates with Visual Studio Code and other supported development environments.

Depending on the project context and the instructions provided, Copilot can:

- complete a line or block of code;
- generate a function from a comment;
- explain how existing code works;
- suggest a correction for an error;
- generate unit tests;
- improve or simplify existing code;
- produce technical documentation.

The developer remains responsible for reviewing the results before adding them to a project.

Save Time Without Losing Control

GitHub Copilot can accelerate code creation, especially for common structures and repetitive tasks.

For example, a developer can describe the purpose of a function in a comment and then review the generated suggestion. The suggestion can be accepted, modified, or rejected.

This approach allows developers to dedicate more time to business logic, architecture, code quality, and complex problem-solving.

Use GitHub Copilot Chat

Copilot Chat allows developers to interact with the assistant using natural language. It can help them:

- understand a file or function;
- investigate an error;
- request an improvement;
- generate comments;
- prepare a README file;
- create tests;
- compare technical solutions.

The quality of the response depends heavily on the context and clarity of the request.

How Can You Write Better Prompts?

An effective prompt clearly describes the expected result. It can specify:

- the programming language;
- the project context;
- the function’s input data;
- the expected output;
- technical constraints;
- the desired style or structure;
- special cases that must be handled.

Instead of simply asking Copilot to “create a function,” it is better to describe its purpose, parameters, output format, and required error handling.

Best Practices for Responsible Use

AI-generated code can contain errors, security vulnerabilities, or design choices that are inappropriate for a particular project.

Developers should therefore:

- review every suggestion;
- run appropriate tests;
- verify security and performance;
- follow the project’s coding standards;
- avoid including confidential information in prompts;
- apply professional judgment before accepting generated code.

GitHub Copilot should be considered an assistant—not an automatically reliable source.



Learn GitHub Copilot with Doussou Formation

Doussou Formation's **GitHub Copilot: AI-Powered Coding Training** provides practical instruction on integrating this assistant into a developer's daily workflow.

Participants learn how to:

- install and configure GitHub Copilot;
- use code suggestions in Visual Studio Code;
- generate and improve code;
- work with Copilot Chat;
- write more effective prompts;
- review and correct generated suggestions;
- adopt responsible development practices.

View the GitHub Copilot training program and discover how AI can help you code more efficiently while maintaining control over your software.