

Hands-on session with VS Code Copilot

TECH2: Introduction to Programming, Data, and Information Technology

Richard Foltyn

NHH Norwegian School of Economics

October 31, 2025

Contents

- 1 Introduction to GitHub Copilot
- 2 Hands-on session with Copilot

INTRODUCTION TO GITHUB COPILOT

Vibe coding = LLM writes the code based on non-technical instructions



Andrej Karpathy ✓

@karpathy

...

There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.

12:17 AM · Feb 3, 2025 · **5.1M** Views

What is it?

- One of the first (maybe the first) serious AI assistants for programming
- Only vaguely related to the Windows / MS Office Copilot
 - Microsoft owns GitHub and rolled out the “Copilot” branding across its products
- Can use LLMs from many vendors (OpenAI, Google Gemini, Anthropic’s Claude)
 - Interface works the same regardless of which LLM backend is used
- GitHub offers various Copilot plans that differ in terms of limits (how many completions/queries/etc. you can run per month)
 - Limits vary across vendors/models (unlimited for “cheaper” models such as GPT-4o)

How to set it up

- Install the GitHub Copilot and GitHub Copilot Chat extensions
- Sign into Copilot using your GitHub account

Pitfalls

- 1 LLMs are not deterministic (not even for the same prompt)
 - Your suggestions will be different from mine
 - You will get different suggestions each time
 - Suggestions depend on the context (existing code in your workspace)
 - The complexity of the generated code varies (sometimes the solution uses functions, sometimes it doesn't, etc.)
- 2 The quality of generated code depends on how much public code exists for a given language
 - Better results with widely used languages like Python and JavaScript
 - Poor results with more exotic/niche/obsolete languages
- 3 Code suggestions might be outright wrong:
 - The proposed algorithm may not solve the problem
 - Suggestions may use functions or APIs that don't actually exist
- 4 Code suggestions might use functions or APIs that are deprecated
 - Can be a problem with pandas, which frequently changes APIs

⇒ Don't use AI for tasks you can't solve yourself

- LLMs may infringe on copyright: the corpus of training data may impose license restrictions
- VS Code Copilot **does not** run on your computer
 - Everything you enter & code files will be uploaded to servers operated by Microsoft/Google/OpenAI/Anthropic/Amazon/etc.
 - You may not have permission to share your code or data with third parties

HANDS-ON SESSION WITH COPILOT

Hands-on session with Copilot

Explore how to use Copilot to solve:

- Assignment 1
- Assignment 2

Experiment with different approaches:

- 1 Write the function name, let Copilot complete the function & use inline chat to modify generated code
- 2 Write the function header with comments documenting behavior, let Copilot generate the function
- 3 Use `Edit` mode in Copilot chat
 - 1 Use a markdown file with instructions
 - 2 Attach a screenshot of instructions
- 4 Use `Agent` mode to find issues in the codebase

Types of interactions with Copilot

Autocompletion `Tab` or `Ctrl` + `>`

- Generates one- or multi-line suggestions in a single file
- Unlike traditional autocompletion, not limited to functions or variables that *actually exist*

Copilot Inline Chat `Ctrl` + `i`

- Invoked directly in the editor window


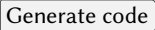

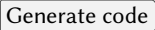

Copilot Chat `Ctrl` + `Alt` + `i`

- Three modes of operation:
 - 1 **Ask:** Quick questions about code or concepts; get explanations and fixes
 - 2 **Edit:** Apply edits across files (refactor, comments, small fixes)
 - 3 **Agent:** Complete high-level tasks & run automated workflows
 - Might trigger multiple premium requests
- Add context (files, selections, etc.) with `#`

Other interactions

Code editor

The editor context menu contains smart actions for common tasks, e.g.:

-  Generates a plain-language description of the selected code
-   Suggests corrections to resolve errors or issues
-   Gives review feedback and improvement suggestions

Terminal

- Use  +  in the terminal to get help with commands and run them

Source control tab

- Use the AI symbol to generate commit messages

Project-specific instructions

Use the file `.github>copilot-instructions.md` to provide additional [context](#):

- Technologies or libraries (e.g., pandas) that should be used
- List files/folders/etc. to be ignored

Instructions should be provided in [Markdown](#) format

VS Code Copilot keeps changing fast — don't use tutorials more than a few months old!

- *GitHub Copilot 101: essential features* [[YouTube Link](#)]
Demonstrates Copilot basics using a simple Python application.
- *Ask, Edit, & Agent - In-depth Overview of GitHub Copilot Chat Modes* [[YouTube Link](#)]
Detailed demo of Copilot Chat modes (using a web application written in TypeScript)