EKON 29

AI Assisted Coding

Der Code-Editor wird zum Partner!

Sebastian Gingter
Developer Consultant
sebastian.gingter@thinktecture.com

AI Assisted Coding:
Der Code-Editor wird zum Partner!

- What to EXPECT
  - Overview of coding assistance powered by Large Language Models (LLMs)
  - Pragmatic use cases
  - Demos (mostly language independent)

- What not NOT TO EXPECT
  - Claude / Codex / Copilot deep dives
  - ML & AI fundamentals

think tecture

AI Assisted Coding

Der Code-Editor wird zum Partner!

Sebastian Gingter
Developer Consultant
sebastian.gingter@thinktecture.com

Sebastian Gingter

Developer Consultant @ Thinktecture AG

- Generative AI in business settings
- AI driven Developer Productivity & Software Quality
- All things .NET

- Microsoft MVP for .NET & Azure AI Services

- sebastian.gingter@thinktecture.com
- https://www.thinktecture.com

think tecture

AI Assisted Coding

Why does it work?

- Language Models are great at languages
- Programming languages: very simple syntax & grammar

- LLMs usually are trained to follow instructions

When does it not work?

- Being cheap.
- Good reasoning and good code output requires using good (= large / expensive / non-local) models
- You want (need) Sonnet 4.5, GPT-5 (Codex), Gemini 2 Pro…

Different Tools

- IDEs
- IDE Plugins
- CLIs

- Subscription-Based
- “Bring your own Key (ByoK)”
- Cloud and local LLMs

Demo

Let’s start Coding…

IDEs

- VS Code /w Copilot (Microsoft / GitHub)
- Cursor
- Windsurf (previous Codeium)
- Kiro (AWS)
- Void (Open-Source Cursor Alternative)
- JetBrains (all)

IDE Plugins

- GitHub Copilot (VS, JetBrains, VS Code)
- Kilo Code (VS Code, JetBrains, Cursor)
- Cline (previous Claude Code) (VS Code)
- Tabby (VS Code)
- Roo (VS Code)
- Augment Code (VS Code, JetBrains, Cursor)

CLIs

- Codex CLI (OpenAI)
- Claude Code (Anthropic)
- GitHub Copilot CLI (GitHub)
- Gemini CLI (Google)
- Cline CLI
- ForgeCode
- Aider
- Plandex
- GPT Engineer
- Smol Developer

think tecture

Behind the scenes

Why does it work?

- Language Models are great at languages
- Programming languages: very simple syntax & grammar

- LLMs usually are trained to follow instructions

Pair Programming?

- Driver
  - writes code
- Observer / Navigator
  - reviews on the go
  - considers “strategic” decisions as they happen

- Similar to aviation: Pilot Flying & Pilot Monitoring

Pair Programming?

- Roles are switched regularly

- Different tools for different roles
  - CLI -> AI is Driver
  - IDE Integration / Chat: AI is Navigator

Demo

Let’s Code…

think tecture

Context is King

Context is King

- LLMs love Markdown
  - README.md, AGENTS.md, CLAUDE.md, Contributing.md etc.
- Memory
  - Tasks, Todos, Guidelines etc. need to be memorized
- Dependencies
  - Model needs to know the APIs or else…
    - it can / will hallucinate calls

https://github.com/josix/awesome-claude-md

Context is King

- Provide a Manifesto / Constitution for a project
  - Defines rules that must be followed always

- Idea: Spec-driven-development?
  - maybe try GitHub Spec Kit

https://github.com/github/spec-kit

Context is King

- MCP (Model Context Protocol) provides access to information (context) as tool calls

Context ist King: MCP

- Some Examples:
  - Context7
  - Playwright
  - GitHub (issues, source)
  - Atlassian MCP Server (Jira, Confluence)
  - Apidog
  - Sequential Thinking MCP Server (well…)
  - …

Context is King

- Claude now supports “Skills”

- Skill:
  - Selective context / instructions as Markdown file in a separate folder, supported by scripts (tools) where needed
  - e.g. Awesome Thinking https://github.com/PawelGerr/ai-plugins

https://docs.claude.com/en/docs/agents-and-tools/agent-skills/overview

Context is King

- Sub-Agents help with context management

- Sub-Task does not clutter context window of main agent
- Sub-Agents can be instructed much more specifically
  - Code-Reviewer
  - Backend- and / Frontend-Specialists
  - Test-Engineer

https://anthropic.mintlify.app/en/docs/claude-code/sub-agents

Demo

Ready when you are…

Thank you!

think tecture

Demos:

https://github.com/thinktecture-labs/ekon-2025-ai-coding

Sebastian Gingter
sebastian.gingter@thinktecture.com
https://thinktecture.com/sebastian-gingter