10 GitHub repositories to help you master Claude Code and AI agent coding.

Discover 10 useful GitHub repositories for learning and optimizing Claude Code, from prompts and subagents to professional AI agent workflows.

Claude Code is rapidly becoming one of the most talked-about 'agentic' programming tools today. Its distinguishing feature is that it not only generates code, but can also read entire codebases, edit files, run terminal commands, and work seamlessly across multiple environments such as IDEs, desktops, and web browsers.

In many cases, you only need to describe the requirements, and Claude Code will handle the rest. However, using only the default settings will only unlock a small fraction of this tool's power. To fully utilize it, you need to understand its ecosystem: from skills, subagents, hooks, and integrations to workflows and project organization.

That's also why there are increasingly more repositories, documentation, and community tools centered around Claude Code. Below are 10 notable repositories to help you delve deeper into how to use and optimize this system.

1. everything-claude-code

If you want to see how Claude Code can be 'upgraded' into a complete agent system, this is a very good starting point.

This repository is not just a collection of prompts or configurations, but a performance optimization system for AI agents, including agents, skills, hooks, rules, MCP configuration, memory optimization, security scanning, and research workflows.

Notably, this system was built on over 10 months of real-world use and won an Anthropic hackathon, so it's often considered a serious 'reference' for advanced workflows.

2. Asystem-prompts-and-models-of-ai-tools

This repository helps you understand the bigger picture of the AI ??ecosystem, not just Claude Code. It compiles system prompts, tool definitions, and model information from various AI products such as Claude Code, Cursor, Devin, Replit, and Perplexity.

If you're interested in how AI tools work behind the scenes, how prompts are designed, or agent behavior, this is a very valuable resource.

3. gstack

gstack demonstrates an interesting approach: transforming Claude Code into an 'AI team' rather than a single assistant. The system clearly defines roles such as CEO, Designer, Engineering Manager, QA, etc., and organizes them through skills and commands instead of random prompts.

This approach is particularly useful if you want to build a structured workflow with clearly defined roles that is easily scalable for large projects.

4. get-shit-done

This repository focuses on helping you work with Claude Code in a more organized way, especially when projects become complex. Instead of lengthy chats, get-shit-done breaks the process down into clear steps: discussion, planning, execution, testing, and deployment.

This approach helps reduce the 'getting lost' of the model during prolonged use and is suitable for those interested in spec-driven development.

5. learn-class-code

If you want to understand the fundamentals of how Claude Code works, this is one of the most worthwhile repositories to learn from. `learn-claude-code` not only provides usage instructions but also teaches you how to build an agent system from scratch: from basic loops, adding tools and subagents, to managing context and task systems.

This is a suitable option if you want to go beyond simply 'using AI' and move towards understanding and designing AI systems.

6. awesome-claude-code

This is a large, comprehensive repository that serves as a 'directory' for the entire Claude Code ecosystem, listing all the relevant skills, hooks, commands, frameworks, apps, and plugins, allowing you to quickly discover the tools currently used by the community.

The main value of awesome-claude-code lies in its discoverability, not in a specific workflow.

7. claude-code-templates

If you don't want to spend time setting everything up from scratch, this repository is a very useful shortcut. It provides pre-configured settings for agents, commands, hooks, integrations, and project templates, allowing you

to quickly deploy and test various workflows.

Claude-Code-Templates is particularly suitable for those who want to optimize speed and repeatability in their work.

8. claude-code-best-practice

This repository isn't a framework or a tool; it's more like a 'guidebook' for using Claude Code effectively. It provides practical guidance on using commands, skills, subagents, and project organization.

This is a good resource for building good work habits and understanding why some methods are more effective than others.

9. awesome-claude-code-subagents

If you're interested in subagents, this repository is almost a complete sample library.

It compiles various subagent definitions for different tasks, helping you understand how roles are divided and agents are specialized in practice.

This is very useful if you want to build a system with multiple agents working together.

10. claude-code-system-prompts

This repository is for those who want to delve deep into how Claude Code works internally, with the ability to view system prompts, tool descriptions, subagent prompts, and version changes.

For developers, researchers, or agent builders, this is a valuable data source for understanding how the system evolves over time.

Claude Code is not just a coding tool, but a platform for building real AI agent systems.

These repositories show a clear trend: developers are no longer just interested in prompts, but are looking for ways to organize agents, optimize workflows, and integrate AI into their software development systems.

If used correctly, Claude Code can not only help you code faster, but also completely change the way you build and run projects.