Getting Started With AI Coding

Jeremy Earle

Two Important Aspects About AI

It will amaze you

It will scare you how stupid it is

So, what to take away from that?

Healthy skepticism

Take care in what you allow AI to do for you

Prerequisites

Access to an AI model

An application to interact with that model

A basic understanding of markdown

Terminology

Model	An algorithm trained on data to recognize patterns, generate text, or make predictions.
Agent	Software running on top of a model that adds instructions, tools, memory etc…
Skill	A set of instructions that can be injected into a conversation with an agent at any moment that tells the agent how to perform a certain task.
Hook	A set of instructions that will be executed whenever a certain event occurs in your AI session.
Prompt	A message or instruction sent to an AI agent.
Token	The fundamental small chunks of data that language models read, process, and generate

About Models

Packaged AI programs. Are given names like “Sonnet”, “Opus”, “GPT-5.6”

Trained on large amounts of data collected from various places (internet, books, image sources, etc…)

Trained for a specific purpose (text generation, image generation, etc…)

Frozen once training is complete, **will not learn anything new**

Non-deterministic. Will not give you the same response every time (a mixture of probability scoring and randomness)

So how does AI remember our conversation?

Context

Every prompt you send to the AI will include your previous chat history, preferences, file system info, etc…

AI reads all of that before reading your latest prompt.

Has limits, depends on the model.

Caution - Large context can lead to AI drift.

What AI model should I be using?

First Choice:

- An AI Service provider (OpenAI, Anthropic, Google, etc…)
  - Costly, but best performance out-of-the-box.
  - Pay a monthly/yearly fee to get a set number of “tokens”
  - Interactions with the AI model are going to use up tokens.
- A Local AI
  - Runs on your computer (or your server)
  - Not as powerful
  - Requires the right hardware and fine tuning.

Second Choice:

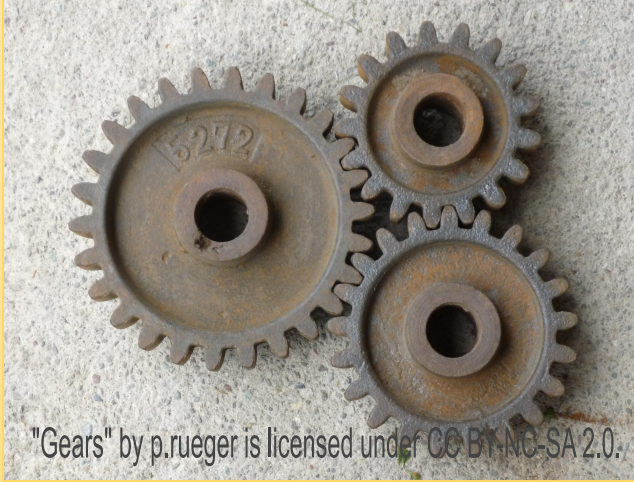
- What model do I use?
  - Depends on the task and the cost

Ok, I have my AI model picked out, how what?

You'll need an application to interact with the model.

- Command-line tools
  - Claude Code (Anthropic)
  - Codex (OpenAI)
  - Gemini CLI (Google)
  - GitHub Copilot (Pick a model from the above providers)
- Desktop Apps
  - Claude Desktop (Just called "Claude")
  - Copilot
  - ChatGPT
- Your IDE
  - VS Code and Jetbrain products have AI plugins
  - Some IDEs are built with AI in mind
    - Cursor

All setup, now what?



1. Personal Settings
    a. General instructions in the form of markdown
    b. Display settings
    c. Trusted directories and tools
2. Projects & directories
    a. Navigate to your project or directory and launch your AI tool
    b. Optional “context” file (markdown)
        i. Claude Code - claude.md
        ii. Codex, Copilot - agents.md
        iii. Symlink is an option
3. Prompting
    a. May need to run commands
        i. By default, should ask for permission for each command
        ii. Optionally can whitelist commands
        iii. Or could allow all commands (Copilot calls this “yolo” mode)

What can we do?

- Knowing when your AI model was trained can help
  - May be knowledgeable about things prior to it release date
    - Libraries, services, applications in the public eye

- Capable of reading and understanding code
  - Can be helped with context files in projects and libraries
  - May need additional context to fully grasp

Planning

- Not necessary for small changes to your code or when just investigating an issue
- Much more important when multiple changes need to be made to the code
- Planning mode sometimes available
  - Forces the AI to only read files, run scripts, and write planning files
  - If not available, try something like “I don’t want you to write any code right now, I only want you to create a plan on how we can do <*xyz*>”
- Once the AI has generated a plan
  - Review, Iterate, Improve
  - Optional: Add pauses/breaks to the plan for review
- Provides a roadmap so you know the what/how/why
- Start small, take on larger tasks as your comfort grows

Priming

- AI can be over-eager
  - Skims over code and documents
  - Jumps to conclusions
  - Makes mistakes
- Instruct it to read a document or piece of code, but do nothing else and wait for further instruction
  - Forces the AI to read the entire document or code
  - Has a better picture of what you might ask of it next

Plugins

- A way to install pre-packaged agents, skills, hooks, or instructions into your AI coding application.
- Usually distributed via “Plugin Marketplaces”
  - Simple git repos with a specific layout/format
  - Agents, skills, hooks, etc… are just markdown files and supporting code/scripts
  - Subscribe to a marketplace, then download the plugins you want
- Marketplaces can be setup by anyone
  - Claude and Github run their own marketplaces
  - Setup just requires creating a git repo, adding some config yaml/json files, and the markdown files.

Skills

- A set of instructions that can be injected into a conversation with an agent at any moment that tells the agent how to perform a certain task.
- Can come with scripts or reference markdown files used to help it complete its task
- Can either explicitly ask the AI to use the skill or the AI may decide on its own to use it
- Skill markdown
  - Provides a name
  - Provides a description
  - Provides instructions

What else should I be doing?

- Code reviews
  - Github copilot
  - Coderabbit
- Teaching/Learning
  - Ask the AI to help you understand a concept…a piece of documentation…
- Help Write Documentation
  - “Listen to what I have to say. I’ll tell you when I’m done.”
- Use it to find bugs
  - Remember: healthy skepticism
- Look at other repos
  - People store AI markdown files, skills, etc… in their library and project repos
  - See what they thought of that you haven’t

Good luck and have fun!