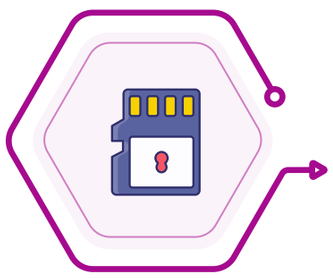
Run Your Own Coding AI:
7 Local Models Worth Installing in 2026

Local coding models have finally grown up. After years of being interesting toys, the latest wave of open large language models, and the community GGUF releases that make them easy to run on everyday hardware, has crossed a real threshold. Several of these models now run on a single consumer GPU like an RTX 3090, generate fast enough to feel genuinely useful, and solve actual coding and agentic programming problems. Not demos. Not party tricks. Real work.



If you want a fully local coding setup and have at least 16GB of video memory (VRAM), the models below can start to free you from depending solely on hosted assistants like Claude Code or Gemini. They are fast, capable, private, and good enough for serious development workflows. You can watch this shift play out in the local AI community, where developers are running coding agents on their own machines, testing GGUF builds, spinning up OpenAI-compatible local servers, and wiring these models straight into their editors and terminals. Here are seven worth your time.

1. Qwen3.6 27B MTP

This is arguably the best all-round local coding model available right now. Across different setups it consistently lands the sweet spot between size, speed, and genuine coding ability. The real advantage is that the GGUF quantized builds let you run it on consumer hardware instead of a cloud rig. Even on a 16GB to 24GB VRAM card, the 4-bit versions make local use realistic.

Qwen models tend to excel at coding because they fold together reasoning, instruction following, multilingual understanding, tool use, and long-context support. That mix makes Qwen3.6 27B MTP a dependable workhorse for coding assistants, repo chat, debugging, shell commands, and agentic workflows, which is exactly why the local community keeps gravitating toward it for fast inference and llama.cpp setups.

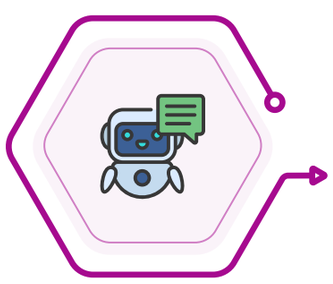
2. Gemma 4 31B IT QAT

Google's open Gemma models have long been a favorite for running capable systems locally, and this quantization-aware training (QAT) GGUF build makes the 31B version surprisingly practical. You get a large model in a 4-bit format that loads far more easily on consumer hardware while holding onto strong quality, landing very close to the Qwen series for local coding and reasoning.

What sets it apart is that it is not only a coding model. It is multimodal, so it can help with screenshots, UI problems, diagrams, documentation images, and web layouts alongside ordinary code generation, debugging, and planning. The benchmark numbers back it up, with strong showings on LiveCodeBench and Codeforces. For coding plus visual development tasks, this is one of the best options to try.

3. DiffusionGemma 26B A4B

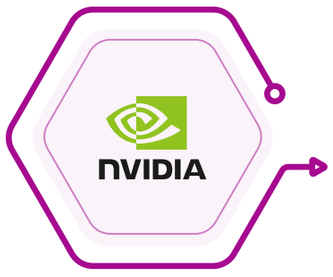
This is one of the newest and most intriguing models on the list, built differently from the usual token-by-token systems. Instead of generating text autoregressively, it uses a block-diffusion approach that denoises blocks of tokens in parallel, a design aimed squarely at faster generation.



That makes it genuinely exciting for local coding, since the architecture could make local assistants much quicker at code generation, structured outputs, and rapid reasoning. The other draw is efficiency: it carries roughly 25B total parameters but only around 3.8B active, giving you the benefits of a larger Mixture-of-Experts design without paying the full inference cost of a dense 26B model.

4. Nemotron Cascade 2 30B A3B

On paper this one looks unusual, but it makes a lot of sense for local coding. It is a 30B Mixture-of-Experts model with only about 3B parameters active during inference, so you skip the full cost of a dense 30B model on every call. That is the ideal profile for a local setup: big enough to reason properly, efficient enough to actually run and test on your own machine.



It behaves more like a reasoning model than a simple autocomplete. NVIDIA positions it as strong for reasoning and agentic tasks, with both thinking and instruct modes, and even claims gold-medal-level performance on the International Mathematical Olympiad 2025 and the International Olympiad in Informatics 2025. That matters because modern coding is rarely just writing functions. You want a model that can debug, plan, review code, and reason through multi-step implementation details.

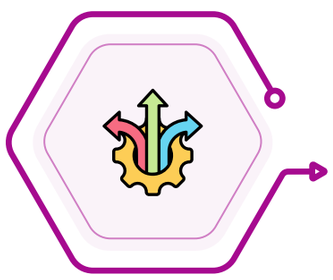
5. Qwen3.5 9B MTP

The smallest model here, but do not underestimate it. For its weight class it punches well above expectations, delivering a proper modern Qwen-style coding assistant without demanding a heavyweight workstation. If your local setup is modest, this one is a gem: fast, practical, and far easier to run than the 27B or 31B options.

The GGUF build makes it even more useful for everyday developers. There is no complicated rig or expensive cloud instance required to test it. Run it locally, connect it to your editor or terminal, and treat it as a private coding assistant. It will not match the larger models on complex reasoning, but for daily tasks like small scripts, debugging, code explanations, and shell commands it is more than enough, and probably the safest first pick for anyone new to local coding models.

6. EXAONE 4.5 33B

LG AI Research's open-weight multimodal model deserves attention, especially if your work goes beyond plain code. Because it handles both text and visual input, it shines in workflows that also involve screenshots, PDFs, diagrams, documentation, and UI layouts.



That versatility is the point. Much of today's coding work is not just writing Python functions. You are reading docs, diagnosing errors from screenshots, parsing architecture diagrams, and wrangling messy project files. A model that understands both code and visuals becomes far more useful. For local code-plus-documents and enterprise-style workflows, EXAONE 4.5 33B is a strong choice.

7. North Mini Code 1.0

One of the freshest entries here, and a welcome sign of Cohere stepping seriously into the local coding space. This is not a general chatbot that happens to write code. It is purpose-built for code generation, agentic software engineering, and terminal-based tasks, which makes it compelling for repo edits, command-line help, code review, and coding-agent workflows.

It is also a 30B-A3B model, carrying 30B total parameters but only around 3B active at inference, so you again get that appealing balance: stronger reasoning than small models, more efficient than a dense 30B. It may not be as broad as Qwen3.6 27B or Gemma 4 31B, but for coding-specific work it looks like a very practical model to try.

Quick Comparison

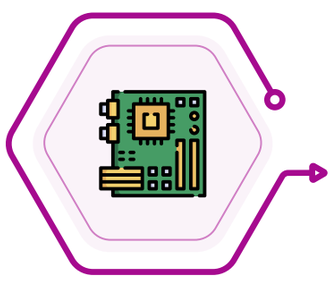
Use this at-a-glance guide to match a model to your hardware, workflow, and use case:

Model	Size / Type	Best Use Case	Why Pick It
Qwen3.6 27B MTP	27B MTP	Strong local coding, reasoning, and agentic workflows	Best all-round local coding model
Gemma 4 31B IT QAT	31B, 4-bit QAT, multimodal	Coding plus screenshots, UI bugs, diagrams, long context	Strong benchmarks and multimodal support
DiffusionGemma 26B A4B	26B / ~4B active	Fast, experimental local coding and reasoning	New architecture built for efficient generation

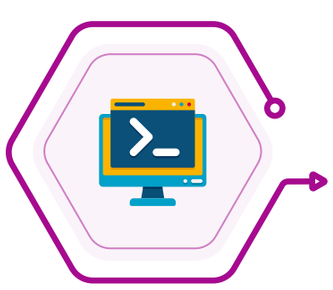
Model	Size / Type	Best Use Case	Why Pick It
Nemotron Cascade 2 30B A3B	30B / ~3B active	Agentic coding, debugging, planning, reasoning-heavy work	Feels more like a reasoning agent than autocomplete
Qwen3.5 9B MTP	9B MTP	Smaller machines and daily coding help	Fast, practical, excellent for its weight class
EXAONE 4.5 33B	33B multimodal	Code, documents, screenshots, PDFs, diagrams	Best for document-heavy, visual coding workflows
North Mini Code 1.0	30B / ~3B active	Local coding agents, repo edits, terminal tasks, review	Most coding-specific model on the list

Which One Should You Actually Install?

Local coding models are now good enough for real development, not just tinkering. If you have a capable GPU like an RTX 3090 or 4090, start with Qwen3.6 27B MTP in 4-bit. It is the best all-round option for local coding, reasoning, and agentic workflows, and it is worth committing to before you waste time hopping between models.



If raw speed is your priority on similar hardware, keep an eye on DiffusionGemma 26B A4B. It is newer and more experimental, but its architecture makes it genuinely interesting for anyone who cares about fast, efficient inference. If you want multimodal understanding and stronger reasoning across code plus screenshots, UI layouts, diagrams, and documentation, Gemma 4 31B IT QAT is a great pick that does far more than write code.



Working with a smaller GPU? Qwen3.5 9B MTP is the best in its weight class, holding up well as a daily assistant for explanations, debugging, scripts, and shell commands even on a simpler setup with enough system RAM. The rest each earn their place too: Nemotron Cascade 2 30B A3B for reasoning-heavy agentic coding and structured problem solving, EXAONE 4.5 33B for document and screenshot-driven enterprise workflows, and North Mini Code 1.0 as the most coding-focused option for agents, repo edits, and code review. They may not all be everyone's first pick, but each has a clear reason to exist.