

M A V E R I C K A I

CLAUDE FABLE 5

THE MOST POWERFUL AI MODEL ON EARTH

Anthropic took their government-grade Mythos model, added safeguards, and released it to the public. Here is everything you need to know.

What Fable 5 is and why it matters

Benchmarks that crush every other model

Anthropic's official prompting guide (condensed)

Viral projects people are already building

20+ copy-paste prompts for every use case

When to use it vs. when to save your tokens

2026 Edition

What Is Claude Fable 5?

On June 9, 2026, Anthropic released Claude Fable 5. It is a Mythos-class model, which is the same tier they originally built for government and defense partners. Mythos was considered too powerful to release publicly, so Anthropic added safety classifiers and guardrails, then shipped it anyway under the name Fable.

Fable 5 sits above the Opus class in capability. It is not a small upgrade. On the coding benchmarks that actually matter, it beats every other publicly available model by double-digit margins. Stripe used it to complete a 50-million-line codebase migration in a single day that would have taken a team over two months.

The catch: it is expensive. Fable 5 costs \$10 per million input tokens and \$50 per million output tokens, roughly double what Opus 4.8 costs. It also burns through tokens fast because it thinks longer and harder than any previous model. For most everyday tasks, the older models are still the right choice. But when you need something done that no other model can handle, Fable 5 is in a class of its own.

80.3%

SWE-Bench Pro (#1 by 11 pts)

1932

Knowledge & Reasoning Score

29.3%

FrontierCode Diamond

How It Compares

Benchmark	Fable 5	GPT-5.5	Gemini 3.1 Pro	Opus 4.8
SWE-Bench Pro	80.3%	58.6%	54.2%	69.2%
FrontierCode Diamond	29.3%	N/A	N/A	13.4%
Knowledge/Reasoning	1932	1769	1314	N/A
GPQA Diamond	91.3%	92.8%	94.3%	N/A
Price (Input / Output)	\$10 / \$50	\$5 / \$30	\$2 / \$12	\$5 / \$25

Fable 5 leads on coding and multi-step engineering work. Gemini 3.1 Pro edges it out on graduate-level science (GPQA Diamond). GPT-5.5 is competitive on reasoning but falls behind on code. The bottom line: if you are building something complex, Fable 5 is the best model available right now.

What People Are Building

Within 24 hours of launch, X was flooded with demos. Some were real, some were people repackaging old videos to ride the hype. Here are the verified ones that actually came from Fable 5.

Minecraft Clone in 20 Minutes

A developer gave Fable 5 a single prompt: "Create a clone of Minecraft." Twenty minutes later, it generated a playable version with multiple biomes, a day/night cycle, different ores, a cave system, and crafting. About 3,000 lines of code, \$12 in API costs.

Boeing 747 from One Prompt

Someone used Claude Code with Fable 5 to build a procedural Boeing 747 model in Three.js from a single /goal prompt. The agent built the 3D model, set up a headless screenshot camera, and iterated in a vision self-verification loop until it looked right. Took about 30 minutes.

Solar System Physics Simulation

Fable 5 built a solar system simulation that derived planetary orbital motion from physics first principles and used it to predict actual solar eclipses. Not a canned animation. Real physics.

Pokemon FireRed Beaten by Vision Only

Anthropic themselves showed Fable 5 playing through the entire Pokemon FireRed game using nothing but screenshots. No game data, no maps, no helper tools. Just looking at the screen and making decisions like a human player. Previous Claude models could not do this even with helper frameworks. Fable 5 did it with a minimal harness. Its strategy? Overlevel Charizard and bulldoze everything.

Stripe's 50M-Line Migration

Stripe reported that Fable 5 completed a codebase-wide Ruby migration across 50 million lines of code in a single day. A human team estimated it would take over two months. This was not a toy demo. It was production engineering work.

Full 3D Websites from One Prompt

Multiple developers showed Fable 5 generating complete 3D interactive websites from a single prompt description. Real visual hierarchy, working animations, professional design. Not the usual AI slop. A design agency owner shipped a client website built with Fable 5 the same day it launched.

HEADS UP

Some viral demos on X were fake. People repackaged old projects or used other tools and claimed it was Fable 5. Verify before you share.

How to Access Fable 5

Fable 5 is available right now on all paid Claude plans. Here is how to switch to it.

STEP 1: Open Claude

Go to claude.ai or open the Claude desktop app. You need a paid plan (Pro, Max, Team, or Enterprise).

STEP 2: Open the Model Selector

Click the **model selector** dropdown at the top of the conversation. You will see the available models listed.

STEP 3: Select Fable 5

Choose **Fable 5** from the list. That is it. Your conversation will now use the Fable 5 model for responses.

API ACCESS

If you are a developer, use the model ID **claude-fable-5** in your API calls. Pricing is \$10 per million input tokens and \$50 per million output tokens.

TIP

Fable 5 burns through tokens fast, especially at higher effort levels. For quick questions, casual conversations, or simple tasks, stick with Sonnet or Opus. Save Fable 5 for the hard stuff.

Effort Levels: Your Cost Control

Effort is the biggest dial you have for controlling Fable 5. It determines how long the model thinks before responding. Higher effort means better output but more tokens burned. Lower effort means faster, cheaper responses that are still better than most models at their max.

Level	Thinking Tokens	Best For	Cost
Low	0 - 2,000	Summaries, formatting, simple Q&A,, classification	\$
Medium	2,000 - 8,000	Multi-step analysis, code review, structured reports	\$\$
High (Default)	8,000+	Complex reasoning, novel problems, adversarial tasks	\$\$\$
xHigh	Extended	Most capability-sensitive work, research, hard problems	\$\$\$\$

Here is the key insight from Anthropic: Fable 5 at medium effort often outperforms older models at their highest settings. You do not need to run everything at xHigh. Start at High (the default), and only go higher when the task demands it.

HEADS UP

Running everything at xHigh is the fastest way to drain your tokens. Map your tasks to the right effort level. Use Low for quick questions, Medium for everyday work, High for complex tasks, and xHigh only when you need the absolute best output.

Anthropic's Official Prompting Guide

Anthropic published a 14-section prompting guide specifically for Fable 5. The biggest takeaway? Everything you learned about prompting other models might actually hurt you here. Fable 5 is different. Here are the rules that matter.

1. Stop Over-Prompting

Most experienced AI users have built a habit of writing massive, detailed prompts packed with rules, examples, and constraints. With Fable 5, that can actually make the output worse. The model is smart enough to figure things out on its own. Give it the goal, the reason behind it, the boundaries, and a way to verify its own work. That is it.

WHAT ANTHROPIC RECOMMENDS

Give the model a goal, the reason behind it, explicit boundaries, and a way to verify its own work. Delete everything else.

2. Give the Why, Not Just the What

Fable 5 performs significantly better when it understands the intent behind your request. Instead of "write me a landing page," tell it who the page is for, what problem it solves, and what you need the reader to do. Context lets the model connect your task to relevant information instead of guessing.

CONTEXT-RICH PROMPTING

Instead of this:

Write me a landing page for my SaaS product.

Do this:

I'm building a landing page for a project management tool aimed at freelance designers. They struggle with tracking multiple client projects. I need them to sign up for a free trial. The page should feel clean and minimal, not enterprise-y. Tone: friendly, direct.

3. Lead with the Outcome

When Fable 5 finishes a task, its first sentence should answer "what happened" or "what did I find." Anthropic specifically recommends this pattern to keep responses from rambling.

OUTPUT FORMATTING

Lead with the outcome. Your first sentence after finishing should answer 'what happened' or 'what did you find.' Supporting detail and reasoning come after. Being readable and being concise are different things, and readability matters more.

4. Ground Progress Claims

On long tasks, Fable 5 can occasionally claim it finished something when it actually has not. Anthropic found a specific prompt that nearly eliminates this problem. Add this to your system prompt for any long-running task.

SELF-VERIFICATION PROMPT

Before reporting progress, audit each claim against a tool result from this session. Only report work you can point to evidence for; if something is not yet verified, say so explicitly. Report outcomes faithfully: if tests fail, say so with the output.

5. State the Boundaries

Fable 5 can occasionally take actions you did not ask for. It might draft an email when you only asked it to analyze one, or create backup branches you did not need. Tell it explicitly what it should and should not do.

BOUNDARY SETTING

```
When the user is describing a problem, asking a
question, or thinking out loud rather than
requesting a change, the deliverable is your
assessment. Report your findings and stop. Don't
apply a fix until they ask for one.
```

6. Build a Memory System

Fable 5 performs dramatically better when it can save lessons from previous runs and reference them later. Give it a place to store notes. This is especially useful for developers using Claude Code.

MEMORY PROMPT

```
Store one lesson per file with a one-line summary
at the top. Record corrections and confirmed
approaches alike, including why they mattered.
Don't save what the repo or chat history already
records; update an existing note rather than
creating a duplicate; delete notes that turn out
to be wrong.
```

7. Let It Run Autonomously

Fable 5 can sustain productive work over extended periods. Multi-hour runs are normal. But it sometimes stops to ask permission when it should just keep going. For autonomous tasks, add this to your prompt.

AUTONOMOUS MODE

```
You are operating autonomously. The user is not
watching in real time and cannot answer questions
mid-task. For reversible actions that follow from
the original request, proceed without asking.
Before ending your turn, check your last
paragraph. If it is a plan, an analysis, or a
promise about work you have not done, do that
```

work now with tool calls. End your turn only when
the task is complete.

8. Keep It From Stopping Too Early

Deep into long sessions, Fable 5 can worry about running out of context and suggest starting a new session. If your harness shows a remaining-token countdown, that makes it worse. This reassurance fixes it.

CONTEXT REASSURANCE

You have ample context remaining. Do not stop,
summarize, or suggest a new session on account
of context limits. Continue the work.

What Fable 5 Is Actually Good At

Not every task needs Fable 5. Here is where it genuinely outperforms everything else, and where you are better off using a cheaper model.

Use Fable 5 For	Use Sonnet or Opus Instead
Full app builds from scratch	Quick questions and Q&A;
Complex multi-file code refactors	Summarizing articles or documents
Long-running autonomous agents	Formatting or reformatting text
3D graphics and game development	Simple email drafts
Multi-day engineering projects	Classification and sorting
Hard debugging across large codebases	Translation tasks
Financial models and analysis	Social media captions
Research requiring deep reasoning	Basic code snippets
Screenshot-to-app recreation	Data entry or extraction
Vision tasks with dense technical images	Casual conversation

ANTHROPIC'S RECOMMENDATION

The teams seeing the best results apply Fable 5 to their **hardest unsolved problems**. Testing it only on simpler workloads tends to undersell its capability range. If a task is straightforward, you will not notice much difference between Fable 5 and a cheaper model.

Copy-Paste Prompts: Coding

BUILD AN ENTIRE APP

Build me a [type of app] that does [core function]. Use [tech stack]. Make it production-ready with proper error handling, responsive design, and clean code. Do not stop until it works end to end.

FULL CODEBASE AUDIT

Audit this codebase. Map the architecture, code structure, dependencies, database patterns, API endpoints, and deployment config. Flag issues by severity. Produce a prioritized task list with effort estimates. Write findings to AUDIT.md.

PORT TO A NEW LANGUAGE

Port this [language A] project to [language B]. First, map the existing system and write a spec of its behavior. Translate module by module. Test each module as it is translated. Run an adversarial review at the end for omissions and behavior changes. Document anything deliberately excluded.

DEBUG A MYSTERY BUG

There is a bug where [describe symptom]. I have already checked [what you tried]. Search the entire codebase and git history for anything that could cause this. Do not guess. Show me the exact line causing the issue and explain why.

RECREATE THIS FROM A SCREENSHOT

[Paste screenshot]
Recreate this UI exactly. Match the layout, spacing, colors, typography, and interactions. Use [React/HTML/etc]. Make it fully functional, not just a visual copy.

Copy-Paste Prompts: 3D and Games

BUILD A GAME

Build me a [type of game] that runs in the browser. Use Three.js or plain Canvas. Include: player movement, collision detection, scoring, sound effects, and a restart mechanic. Make it actually fun to play, not just a tech demo.

3D PRODUCT VISUALIZATION

Create a 3D interactive product viewer for [product description]. The user should be able to rotate it, zoom in, and click different parts to see details. Use Three.js. Make it look premium, not like a homework project.

PHYSICS SIMULATION

Build a [type] physics simulation. Derive the motion from first principles, not canned animations. Include real-time controls so the user can adjust parameters and see the effects. Make it visually clean and educational.

3D WEBSITE

Build me a 3D interactive website for a [type of business]. Include scroll-triggered animations, parallax effects, and at least one interactive 3D element. It should feel premium and load fast. Use Three.js + vanilla HTML/CSS.

MINECRAFT-STYLE VOXEL WORLD

Create a Minecraft-style voxel world with multiple biomes, a day/night cycle, procedural terrain generation, cave systems, and basic player movement with block placing and breaking. Make it run in the browser at 30+ fps.

Copy-Paste Prompts: Business

FULL BUSINESS PLAN

I am starting a [type of business] targeting [audience]. Build me a full business plan: market analysis, competitive landscape, pricing strategy, go-to-market plan, financial projections for 3 years, and the key risks with mitigation. Be specific with numbers, not vague platitudes.

FINANCIAL MODEL IN EXCEL

Build me a financial model for [business/stock] in Excel. Include income statement, balance sheet, cash flow, DCF valuation, and sensitivity analysis. Use real assumptions, not placeholders. Export as a .xlsx file I can download and edit.

COMPETITIVE ANALYSIS

I am building [product]. My competitors are [list competitors]. Analyze each one: pricing, features, positioning, strengths, weaknesses, and where they are vulnerable. Tell me where I can win that they cannot easily defend.

FULL WEBSITE COPY

Write all the copy for a [type] website. Include: hero section, features section, social proof, pricing page, FAQ, about page, and footer. The tone should be [describe tone]. The reader is [describe audience]. They care about [what they care about]. Make every section conversion-focused.

RESEARCH REPORT

I need a research report on [topic]. This is for [audience/purpose]. Cover: current state, key players, market trends, risks, opportunities, and your recommendation. Cite specific data points. I need this to be good enough to present to

[executives/investors/clients].

Copy-Paste Prompts: Advanced

OVERNIGHT AUTONOMOUS TASK

Task Description

I need you to [describe the full task].

Definition of Done

[Describe exactly what 'done' looks like]

Available Context

[List repos, docs, access, constraints]

Rules

- Work autonomously. Do not stop to ask questions.
- Verify each step before reporting completion.
- If something fails, debug it yourself.
- Report outcomes faithfully.

SELF-VERIFYING CODE

Build [what you need]. After building it, establish a method for checking your own work. Run verification with subagents against the specification. Every [X interval], pause and verify what you have built actually works as intended. Do not move on until each section passes verification.

MULTI-AGENT DELEGATION

Break this task into independent subtasks. Delegate each subtask to a parallel subagent. Keep working while they run. Intervene if a subagent goes off track or is missing relevant context. Merge results when all are complete. Verify the merged output is coherent.

LEARN FROM PAST SESSIONS

Reflect on the previous sessions we have had together. Use subagents to identify core themes and lessons. Store them in a memory file. Make sure you reference these notes for future use.

Focus on: what worked, what failed, what I prefer, and what to avoid.

LONG-FORM CONTENT CREATION

Write a [length] [type of content] about [topic].

This is for [audience]. The goal is [what you want readers to do/feel/understand]. Use a [tone] voice. Include specific examples, data points, and real stories where possible. Do not use generic filler. Every paragraph should earn its place.

The Safeguards

Fable 5 is a restricted version of Mythos. Anthropic added safety classifiers that block responses in certain areas. These trigger in less than 5% of sessions, but you should know what they are.

What Gets Blocked

Offensive cybersecurity techniques (building exploits, malware, attack tooling), biology and life sciences content (lab methods, molecular mechanisms), and extraction of the model's internal thinking. When a request triggers these classifiers, the response falls back to Claude Opus 4.8 instead.

What Does Not Get Blocked

Defensive cybersecurity work, general coding, business analysis, content creation, research, financial modeling, creative writing, and virtually everything else. The safeguards are narrow. Most users will never encounter them.

MYTHOS 5 (UNRESTRICTED)

The full unrestricted Mythos 5 model exists, but it is only available through Anthropic's "Project Glasswing" program for infrastructure providers and vetted cybersecurity researchers. If you are a regular user, Fable 5 is the model you get.

When Not to Use Fable 5

Fable 5 is overkill for most tasks. Here is a practical framework for which model to use and when.

Use Sonnet for:

Quick questions, simple edits, formatting, translations, social media posts, email drafts, data extraction, classification. Fast and cheap. Handles 80% of everyday tasks.

Use Opus for:

Multi-step analysis, code review, document drafting, research summaries, debugging, writing that needs nuance. The sweet spot for quality vs. cost. Handles most professional work.

Use Fable 5 for:

Full app builds, complex refactors, multi-day engineering projects, hard research, 3D graphics, autonomous agents, anything where you would say "this is going to take a while." Fable 5 is the model you call when the other models are not good enough.

TIP

A good rule: if the task takes you less than 15 minutes by hand, Sonnet or Opus is fine. If it would take you hours or days, that is when Fable 5 earns its price.

Quick Model Selection Guide

Task	Model	Why
"Summarize this article"	Sonnet	Simple task, fast response needed
"Review my pull request"	Opus	Needs nuance but not extended reasoning
"Build me a full-stack app"	Fable 5	Complex, multi-file, needs self-verification
"Write a tweet"	Sonnet	10-second task, do not burn premium tokens
"Debug this race condition"	Fable 5	Hard problem requiring deep code analysis
"Translate this to Spanish"	Sonnet	Straightforward, all models handle well
"Create a financial model"	Fable 5	Complex reasoning + Excel export
"Draft a blog post"	Opus	Quality writing without Fable-level cost
"Migrate this codebase"	Fable 5	Long-running, multi-step engineering
"Reformat this JSON"	Sonnet	Mechanical task, cheapest option wins

Want More AI Tips Like This?

I post daily AI content, tool breakdowns, and prompts that actually help you get more done. Follow me for the latest.

@MavGPT

TikTok | Instagram | Facebook

The most powerful model on the planet. Now you know how to use it.