Claude Context Window — Every Question Answered (Complete FAQ 2026)

Edition of November 13, 2026 — facts verified November 13, 2026. Current edition always at https://tygartmedia.com/claude-context-window-faq-complete-2026/.

Direct Answer (November 13, 2026): All four current Claude models — **Fable 5.2** ($8/$40), **Opus 5.5** ($4/$20), **Sonnet 5.5** ($2/$10), and **Haiku 5.5** ($0.10/$0.50 ≤100K) — feature a **1,000,000-token context window** and **128,000-token max output** at standard per-token pricing. Only legacy `claude-haiku-4-5-20251001` remains at 200K input / 64K output.

Last refreshed: November 13, 2026 · Sources: Anthropic models overview & API pricing.

Tygart Media · Claude Context Window Reference

Updated November 13, 2026 · Sizes verified from Anthropic's official models page · Based on production use

Context window questions answered from someone who actually uses the 1M token window in production — not from a spec sheet alone.

Covers window sizes by model, what 1M tokens holds, the memory vs context distinction, performance at long context, and API-specific details. Full explainer: Claude Context Window Size 2026

Size Questions

Size questions.

What is Claude's context window size in 2026?

Lineup currency (November 13, 2026, verified): Current API models are **Fable 5.2** ($8/$40), **Opus 5.5** ($4/$20), **Sonnet 5.5** ($2/$10), and **Haiku 5.5** ($0.10/$0.50 ≤100K; $0.50/$2.50 >100K) — all **1M context / 128K output**. Legacy models (Fable 5.1 $10/$50, Opus 5/4.8 $5/$25, Sonnet 5 $2/$10, Sonnet 4.6 $3/$15, Haiku 4.5 $1/$5) remain listed during transition windows.

Model	API String	Context Window	Max Output
Claude Fable 5.2	`claude-fable-5-2`	**1,000,000 tokens**	128,000 tokens
Claude Opus 5.5	`claude-opus-5-5`	**1,000,000 tokens**	128,000 tokens
Claude Sonnet 5.5	`claude-sonnet-5-5`	**1,000,000 tokens**	128,000 tokens
Claude Haiku 5.5	`claude-haiku-5-5`	**1,000,000 tokens**	128,000 tokens
Claude Haiku 4.5 (legacy)	`claude-haiku-4-5-20251001`	200,000 tokens	64,000 tokens

Source: Anthropic's official models page & API pricing, verified November 13, 2026.

What does 1 million tokens actually hold?

- **~750,000 words of English text** — roughly 10 full-length novels, or 1,500 average blog posts
- **A full mid-size codebase** — a 50,000-line Python project with comments
- **~60–100 research PDFs** at 20–30 pages each, all simultaneously
- **Hours of meeting transcripts** — a full workday of recorded calls, transcribed
- **Our full WordPress site audit** — 200+ posts worth of content loaded in one session for comprehensive SEO analysis

The shift from 200K to 1M wasn't just "more room." It changed what we could ask Claude to do in a single session — whole-codebase reasoning, multi-document synthesis, full-history context.

How many pages can Claude read at once?

A typical 20-page PDF is roughly 10,000–15,000 tokens, so at 1M tokens you could load 60–100 such documents simultaneously. A 300-page book runs roughly 150,000–200,000 tokens — Claude can hold 5–6 full books in context at once. In practice, the constraint is usually time to upload and your session structure, not the window ceiling.

What's the difference between context window and memory?

Three distinct things that get conflated:

- **Context window:** Everything Claude can see right now in this session. Temporary — disappears when the session ends.
- **claude.ai memory:** Facts extracted from past conversations and injected as a summary into new sessions. Persistent but compressed — a small snippet in the context, not the full history.
- **Managed Agents memory stores / Dreaming:** Developer-layer knowledge graphs that agents build and refine between sessions. More structured than consumer memory, requires API implementation.

The 1M context window is your working memory for one session. Memory systems are what carry information across sessions — they work by injecting a summary into the new session's context, not by giving Claude access to the full prior history.

Performance Questions

Performance questions.

Does performance degrade at very long context lengths?

The honest answer: yes, somewhat, and it depends on the task. The "lost in the middle" pattern is real — models tend to weight the beginning and end of very long contexts more heavily than the middle. For tasks that require pinpointing specific information buried deep in a 500-page document, performance is lower than for shorter contexts. For tasks that benefit from broad synthesis across a large body of material — architectural review, theme identification, cross-document comparison — long context is a net positive. Structure important information at natural reference points rather than burying it in the middle of a large document.

How does Opus 5.5's context window compare to Sonnet 5.5 (and legacy Opus 4.8 / Sonnet 4.6)?

In Anthropic's current 5.x lineup, Fable 5.2, Opus 5.5, Sonnet 5.5, and Haiku 5.5 all share both a **1M input context window** and **128,000-token max output**. On legacy models (Opus 4.8 vs. Sonnet 4.6 or Sonnet 5), Opus supported 128K output while Sonnet 4.6/5 capped at 64,000 tokens.

What happens when I hit the context window limit?

Earlier messages begin dropping out of the active context. Claude can no longer reference information from those dropped messages — it effectively forgets that part of the conversation. In the claude.ai interface, you'll see a notification as you approach the limit. In API usage, the context window limit is enforced hard — requests exceeding it return an error.

API and Technical Questions

API and technical questions.

Is the 1M context window available on the free plan?

The model available to free plan users supports the 1M window technically, but free plan rate limits mean sustained heavy long-context use hits limits quickly. The window is available; using it intensively for extended periods is more practical on paid tiers.

What's the extended output option on the Batch API?

Current 5.x models (Fable 5.2, Opus 5.5, Sonnet 5.5, Haiku 5.5) list 128,000 max output tokens at 50% off via the Message Batches API (Fable 5.2 $4/$20, Opus 5.5 $2/$10, Sonnet 5.5 $1/$5, Haiku 5.5 $0.05/$0.25 ≤100K). Earlier 2026 releases (Fable 5, Opus 4.8, Sonnet 4.6) tested an `output-300k-2026-03-24` batch beta header for async outputs above 128K.

Can I query context window limits programmatically?

Yes. The Models API returns `max_input_tokens`, `max_tokens`, and a `capabilities` object for every available model. If you're building systems that need to programmatically enforce context limits or route by capability, this is the right way to get current values rather than hardcoding from documentation.

Does context window size affect API cost?

Only indirectly — you pay for tokens consumed, not for context window capacity. A 1M token window doesn't cost more than a 200K window. You pay for the tokens you actually send and receive. Loading a 500K-token document into context costs the same per token regardless of whether the model has a 200K or 1M window. The window size determines whether the request is *possible* at all — not what it costs per token.

Related on Tygart Media: extended thinking · how to use Claude · tokens to words.

Primary & Reference Links Cited on This Page

- Anthropic models overview — `https://platform.claude.com/docs/en/about-claude/models/overview`
- API pricing — `https://platform.claude.com/docs/en/about-claude/pricing`
- Message Batches API — `https://platform.claude.com/docs/en/build-with-claude/batch-processing`
- Tygart Media: Claude Context Window Size 2026
- Tygart Media: extended thinking
- Tygart Media: how to use Claude
- Tygart Media: tokens to words