

Which AI provider should you choose?

OpenAI, Grok, Anthropic, Google and Microsoft compared

Start with the job you want done. A developer repairing a large codebase, a creator researching an article and a business sorting thousands of short messages have different needs. The provider with the most expensive flagship is not automatically the right purchase for all three.

The useful comparison has three parts: what the model can do, how you access it, and what it costs to reach an acceptable result. Company value is interesting background, but it cannot tell you whether a model will follow your brief or fix your bug.

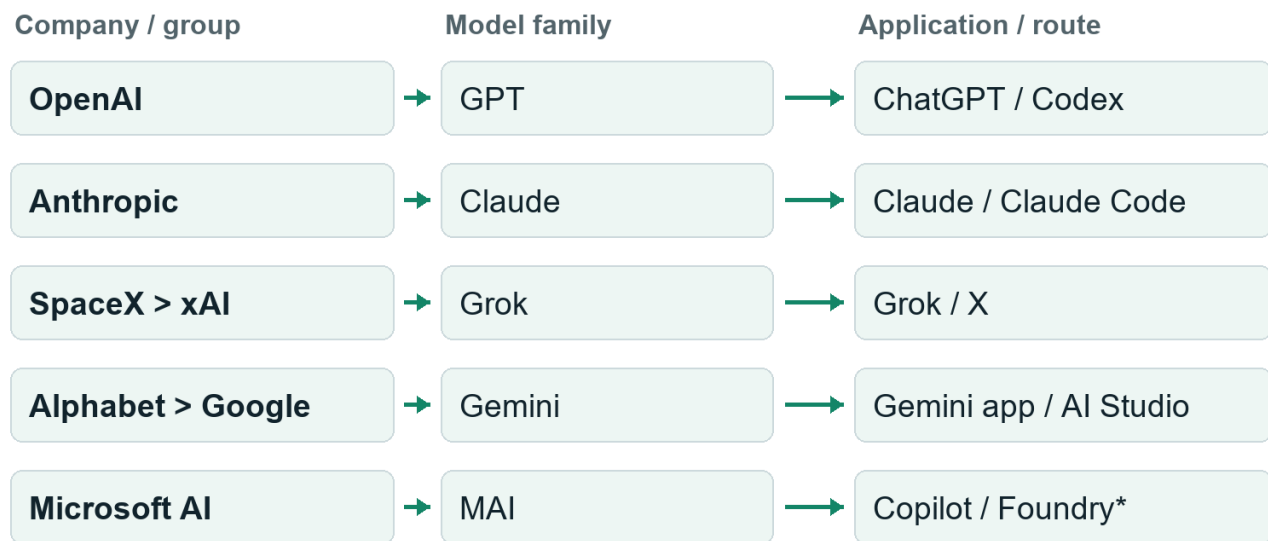
This study examines current company disclosures, model documentation, prices, original benchmark methods and product policies checked on 13 October 2026. The price examples are calculations from published rates; no inference tests were run.

Are we comparing companies, models or apps?

All three, but they answer different questions. OpenAI is a provider, GPT is a model family, and ChatGPT or Codex is a product you use. A product adds its own interface, tools, quotas and settings. Those additions can change your experience as much as switching the underlying model.

Who builds it, and where do you use it?

Follow the company, model and application layers.



* Microsoft products can also host partner models. Identify the selected model.

The company, model and application layers. Microsoft products can also use partner models; always identify the model selected inside the product. [3] [10] [17]

That distinction matters especially for Microsoft. Its native MAI models are one subject; Microsoft 365 Copilot, GitHub Copilot and Foundry are product or deployment choices. Choosing a Claude model through a Microsoft service does not make that model a Microsoft-developed Claude. The current Claude catalogue lists Microsoft Foundry access alongside other routes. [10]

And what do we mean by X? Here we mean Grok and its developer, xAI. SpaceX's June-quarter filing records that xAI became its wholly owned subsidiary on 2 February 2026, having already brought X and X.AI under the same holding company. SpaceX completed its IPO in June and trades as SPCX. Calling the whole arrangement simply a private social-media company would miss the current structure. [3]

So the first purchasing question is simple: which model will actually answer my request, and under whose product terms? Keep that answer in view while reading the rest of the comparison.

How much are the businesses behind these models worth?

The numbers are substantial, but they describe different things. A listed company's market capitalisation reflects the market value of its outstanding equity. A private financing valuation records the equity value assigned in a particular funding transaction. Neither figure is the cash a company has available to spend.

What size of business sits behind the model?

Keep public group values and private financing values distinct.

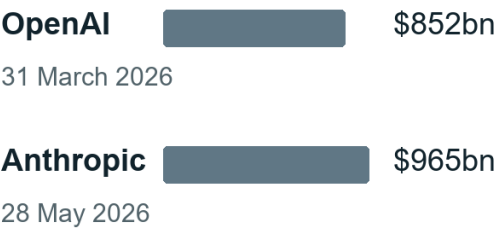
Public market cap

Whole groups | 12 October 2026 close



Private post-money valuation

Latest disclosed closed rounds in this research



Parent values include non-AI businesses. SpaceX estimates vary. These are equity values, not available cash or model-quality scores.

Public whole-group market caps at the 12 October 2026 US close are shown separately from dated private post-money financing valuations. USD retained. SpaceX estimates vary across data providers; its displayed value is approximate. [1] [2] [4] [5] [6]

Business	Figure	Measure	Evidence date
Alphabet	~\$4.31tn	Whole-group market cap [4]	12 Oct 2026
Microsoft	~\$4.02tn	Whole-group market cap [5]	12 Oct 2026
SpaceX parent	~\$2.2tn	Whole-group market cap [6]	12 Oct 2026
OpenAI	\$852bn	Private post-money financing [1]	31 Mar 2026
Anthropic	\$965bn	Private post-money financing [2]	28 May 2026

All figures USD. Whole-parent market caps and private financing valuations have different bases and dates.

OpenAI announced a closed funding round on 31 March at an \$852 billion post-money valuation, with \$122 billion in committed capital. Anthropic announced a \$65 billion Series H on 28 May at a \$965 billion post-money valuation. These are dated financing disclosures, not live exchange prices or a measure of money already collected in a bank account. [1] [2]

For the listed businesses, the previous US-session snapshots put Alphabet at approximately \$4.31 trillion, Microsoft at \$4.02 trillion and SpaceX at about \$2.2 trillion. These cover the whole groups. Alphabet's value is not a standalone Gemini valuation; SpaceX's includes businesses beyond Grok. The market-cap estimates come from dated secondary market-data snapshots, and SpaceX sources differ at roughly \$2.15–\$2.22 trillion. [4] [5] [6]

Does the biggest business therefore have the best model? That conclusion would need performance evidence. A large diversified parent and a specialist AI company have different businesses, costs and

revenue sources. Their financial scale is useful context for ownership and business continuity; your own task still needs a separate capability and cost assessment.

Which models are worth paying attention to?

Think in tiers. Frontier models target demanding work, balanced models aim to make capable assistance practical, and smaller models target workloads where speed and price matter heavily. A specialised coding model is a further category. These roles are more useful than assuming the largest version number wins.

Provider	Current choices	Documented role and status
OpenAI	Astra / Sol / Luna	Demanding / balanced / high volume [7] [8] [9]
Anthropic	Fable / Opus / Sonnet / Haiku	Reasoning / long coding / balanced / volume [10]
Grok / xAI	Grok 4.7	Text/code with configurable reasoning [12]
Google	3.1 Pro / 3.8 Flash / 3.5 Flash-Lite	Pro preview; Flash and Flash-Lite stable [13]
Microsoft	MAI-Thinking-1 / MAI-Code-1.1-Flash	Thinking preview; specialised coding product [16] [17]

Model roles are provider descriptions. Availability depends on the account, model and access route.

OpenAI currently positions Astra for demanding reasoning and coding, Sol as the balance of capability and cost, and Luna for focused high-volume tasks. Anthropic offers Fable for demanding reasoning, Opus for long-running coding and knowledge work, Sonnet for a speed/intelligence balance and Haiku for high-volume tasks. These are provider-defined roles, not our own measured quality ranking. [\[7\]](#) [\[8\]](#) [\[9\]](#) [\[10\]](#)

Grok's current text/code catalogue centres on Grok 4.7 with configurable reasoning. Google's lineup needs two labels: Gemini 3.8 Flash is stable, while Gemini 3.1 Pro remains a preview. MAI-Thinking-1 is also a preview; MAI-Code-1.1-Flash is a specialised coding option integrated into GitHub Copilot. [\[12\]](#) [\[13\]](#) [\[16\]](#) [\[17\]](#)

What should you check before choosing? The exact model ID, the access route and whether the version is stable or a preview. A provider may list a model that your account or chosen product cannot yet select. A coding product's default can also differ from the provider's advertised flagship.

Which provider is best at coding?

A useful answer starts with the kind of coding. Completing a short function, repairing a real repository, translating a design into a page and maintaining a multi-step agent are different tasks. A model that writes plausible code has not finished the job until the relevant checks pass and the change fits the existing project.

Published benchmarks help narrow a shortlist. SWE-bench Verified is a human-filtered set of 500 software issues. Its default Bash Only leaderboard uses the same mini-SWE-agent environment across

models. That shared harness is valuable because it reduces one source of variation: differences in the surrounding agent tools. [18]

But a benchmark name alone is not enough. Terminal-Bench 4.0 changed resources and tasks and requires fresh runs; its authors document an eight-hour agent timeout. Scores from an earlier version cannot simply be moved into the new table. You also need the model version, effort setting, tools and repair allowance to understand a result. [19]

Microsoft's October MAI-Code-1.1-Flash announcement illustrates the distinction. It reports 25% fewer tokens per task, 25% faster token streaming and a quarter of the preceding model's price. Its Terminal-Bench comparison uses version 2.1 in the Copilot CLI. That is evidence of a provider-reported improvement over its predecessor, rather than a matched five-provider result on version 4.0. [17]

For your shortlist, compare the coding product as well as the model: can it inspect the right files, preserve unrelated work, execute tests and show a reviewable diff? Then try the kinds of tasks you repeat. First-attempt quality, the number of corrections and the time you spend reviewing are all part of the result.

This evidence supports comparing Astra/Sol, Claude's coding tiers, Grok 4.7, Gemini's current tiers and the native MAI coding workflow. It does not support declaring an overall coding winner from a mixture of differently configured published scores.

Will the answer be useful, accurate and well written?

Those are separate checks. An answer can be fluent and still cite a source that does not support its conclusion. It can reach the correct result while explaining the reasoning poorly. For research, ask whether the evidence is real, current and connected to the claim; for writing, ask whether the reader can understand and use it.

Current information also depends on the tool route. Grok's own documentation says its model has no access to real-time events unless search tools are enabled. The model's knowledge and a browsing-equipped application are different resources. A similar distinction is worth checking in any provider's product. [12]

A practical comparison would give each model the same source packet, then check dates, numbers, citations and unsupported claims. A second condition could permit browsing, with the same question and a clear time budget. Keeping those conditions separate tells you whether a difference came from interpreting evidence or obtaining it.

For prose, a clear brief is more useful than asking for impressive writing. Specify the audience, the point that must be explained, the facts that must stay, and the intended tone.

Can we name the best writer here? The available documentation cannot settle a subjective preference for your work. The useful next step is to compare short, anonymised outputs against the same brief, checking factual preservation and instruction following alongside your editorial preference.

Does a million-token limit mean it understands a million tokens?

It describes capacity, not guaranteed understanding. Being able to accept a long request is different from finding a buried fact, reconciling contradictions or drawing the right conclusion across many files. The context budget can also include generated work, depending on the model's accounting.

How much material can a request carry?

Documented API ceilings | definitions matter

GPT-6 Astra / Sol	1,050,000	Context window
Current Claude tiers	1,000,000	Context window
Gemini 3.8 Flash	1,048,576	Input-token limit; output separate
Grok 4.7	500,000	Context window
MAI-Thinking-1	256K	Total request token budget

Capacity is a specification. Retrieval accuracy and product memory need separate checks.

Published API ceilings, with their definitions retained: OpenAI/Claude/Grok/Microsoft describe context or total budgets; the Google examples show an input-token limit. These are specifications, not measured long-document comprehension. [\[7\]](#) [\[10\]](#) [\[12\]](#) [\[15\]](#) [\[16\]](#)

Astra and Sol list a 1,050,000-token context window; the current Claude tiers list 1 million. Grok 4.7 lists 500,000. MAI-Thinking-1 documents a 256K total token budget. Gemini 3.8 Flash lists a 1,048,576-token input limit and a separate 65,536-token output limit. The different definitions are a reason to read the detail rather than rank one headline number. [\[7\]](#) [\[8\]](#) [\[10\]](#) [\[12\]](#) [\[15\]](#) [\[16\]](#)

Is context the same as memory? No. Context is the material available for the current request. Product memory can store selected information for later conversations. Microsoft's current Copilot controls, for example, treat saved memories and chat history as separate settings; turning memory off does not automatically remove memories already saved. [\[34\]](#)

For a document-heavy job, use the actual collection of files you need to understand. Check extraction from the beginning, middle and end; ask for evidence locations; and include a contradiction whose handling you can verify. That gives you a more useful answer than a promise that the model can fit an entire library.

Where do my tokens go?

They are spent on more than the sentence you type. Inputs can include instructions, conversation history, documents and tool results. Outputs can include the answer and billable reasoning. A multi-step job may send updated context through the model several times before you see a finished result.

Where does the bill come from?

Follow the work needed for one accepted result.

- 1 Brief, documents and instructions**
Material supplied to the model
- 2 History, context and tool results**
More input can arrive as the job progresses
- 3 Reasoning and answer**
Billed output may include internal work
- 4 Review and possible repair**
New attempts can repeat earlier context
- 5 Accepted result**
Count the cost of every attempt needed

A request's bill can include context, internal reasoning, tools and repeated attempts. Use the provider's reported billing categories, avoiding double-counting reasoning already included in output. [\[20\]](#) [\[21\]](#) [\[22\]](#) [\[23\]](#)

Does a short answer therefore mean a small bill? Not necessarily. OpenAI documents that internal reasoning consumes context and is billed as output even though it is not visible in the API answer. Anthropic bills thinking even when the visible account is summarised. Google's pricing includes thinking tokens, and Grok also documents reasoning-token consumption. [\[20\]](#) [\[21\]](#) [\[22\]](#) [\[23\]](#)

Tokens are not a universal word count. Each model uses its own tokenisation rules, so the same text can produce different totals. Anthropic says its newer tokenizer can produce approximately 30% more tokens than its earlier tokenizer, depending on the content and workload. That is a comparison within Claude's generations, not a 30% comparison against every rival. [\[11\]](#)

For a real budget, collect the usage fields and invoice basis for each attempt. If a product exposes only messages or credits, keep that unit: converting it into a guessed API token figure would give the appearance of precision without the evidence. Separate new input, cache usage, billed output and tools according to that provider's definitions.

What does a request actually cost?

We can calculate the token component without buying an API call. The table uses one hypothetical request with 10,000 uncached input tokens and 2,000 billed output tokens, including any reasoning within that stated output allowance. It assumes standard direct-provider text pricing, a short prompt, and no tool, cache, tax, regional or speed-tier extras. The same token counts are an arithmetic scenario, not an identical-text or quality benchmark.

Model	Input \$/1M	Output \$/1M	Example \$
GPT-6 Astra [7]	10	50	\$0.2
GPT-6.1 Sol [8]	2	10	\$0.04
GPT-6 Luna [9]	0.1	0.5	\$0.002
Claude Fable 5.1 [11]	10	50	\$0.2
Claude Opus 5.5 [11]	4	20	\$0.08
Claude Sonnet 5.5 [11]	2	10	\$0.04
Claude Haiku 5.5 [11]	0.1	0.5	\$0.002
Grok 4.7 [12]	2	6	\$0.032
Gemini 3.1 Pro [14]	2	12	\$0.044
Gemini 3.8 Flash [14]	0.75	3.75	\$0.015
Gemini 3.5 Flash-Lite [14]	0.3	2.5	\$0.008
MAI-Thinking-1	Not verified	Not verified	Not calculated

10,000 uncached input + 2,000 billed output tokens. Standard direct-provider text rates; see chart notes and conditions in the accompanying text.

What do the stated token quantities cost?

10,000 uncached input + 2,000 billed output tokens | standard USD rates

GPT-6 Astra		\$0.2
Claude Fable 5.1		\$0.2
Claude Opus 5.5		\$0.08
Gemini 3.1 Pro		\$0.044
GPT-6.1 Sol		\$0.04
Claude Sonnet 5.5		\$0.04
Grok 4.7		\$0.032
Gemini 3.8 Flash		\$0.015
Gemini 3.5 Flash-Lite		\$0.008
GPT-6 Luna		\$0.002
Claude Haiku 5.5		\$0.002

A rate calculation, not a quality benchmark. Flash promotional rates end 31 December 2026. MAI direct rate not verified; no zero-price assumption.

Published-rate calculation for 10,000 uncached input plus 2,000 billed output tokens. Same billing quantities; no inference run or quality ranking. Microsoft omitted from the bars because a comparable direct rate was not verified. [\[7\]](#) [\[8\]](#) [\[9\]](#) [\[11\]](#) [\[12\]](#) [\[14\]](#)

For example, Astra's standard \$10 input/\$50 output rates per million produce $\$0.10 + \$0.10 = \$0.20$. Sol or Sonnet's \$2/\$10 rates produce $\$0.02 + \$0.02 = \$0.04$. Gemini 3.8 Flash currently produces $\$0.0075 + \$0.0075 = \$0.015$. These amounts retain the providers' US-dollar billing currency; no exchange rate is assumed. [\[7\]](#) [\[8\]](#) [\[11\]](#) [\[14\]](#)

Google's Flash rate has an important date attached. The listed \$0.75 input/\$3.75 output rates apply through 31 December 2026; the announced standard rates become \$1.50/\$7.50 from 1 January 2027. That same hypothetical request would then cost \$0.03 in token charges. A current low price can have a known expiry. [\[14\]](#)

Can I extend the short-request calculation to a giant document? Check the thresholds first. For a 300,000-token uncached prompt and 2,000 billed output tokens, Astra's documented long-context rates give \$6.15. Opus 5.5's standard \$4/\$20 rates give \$1.24. These calculations use each provider's published billing rules and the stated counts; they do not establish which would answer the document correctly. [\[7\]](#) [\[11\]](#)

What if I use a subscription instead? Then compare the plan's included capacity, available models and usage rules. A chat or coding-product subscription is not the same billing arrangement as a direct API request. Check its actual entitlements before assuming that a monthly payment supplies unlimited calls or API credit. The better route depends on whether you want a ready-made product or inference inside your own software.

Does the cheapest model produce the cheapest finished job?

Only when it reaches an acceptable result economically. Imagine five attempts at 1p each versus one accepted attempt at 4p. The second route costs less despite its higher price per attempt. Those are hypothetical amounts, but the accounting rule is useful: include every attempt needed to finish the work.

Cost per accepted result = total cost of all attempts divided by the number of accepted results. When nothing is accepted, record the failure and spend rather than inventing a finite cost per success.

Measure time the same way. Fast token streaming is attractive, but it is only part of the wait. Tool calls, repository inspection, tests, corrections and your own review all contribute to the time until the job is ready. Time to first token and time to an accepted result answer different questions.

Caching and batching can change the calculation again. Reusing eligible context may reduce repeated input charges; batch processing trades a different delivery arrangement for lower rates where supported. Storage, cache writes and tools can still add costs. Use the actual billing categories rather than treating every discounted token as a complete free repeat.

And what about energy efficiency? We do not have a matched independent measurement of joules or carbon per accepted task for these current models. Token price, parameter count and stream speed cannot substitute for one. The meaningful efficiency comparison available here is the documented billing structure and the practical method for measuring a finished job.

Will the provider train on my work?

Check the exact product, plan, region and settings. The company name by itself is not a privacy policy. Training use, safety retention, saved conversation state and feedback are different categories, and a change in one does not automatically remove the others.

Route	Documented data-use distinction
OpenAI API	No training by default; safety and feature retention are separate [24]
Claude	Commercial no-training default; consumer permission/settings and covered-model retention differ [25] [26] [27]
Gemini developer tools	UK/EEA/Swiss users receive Paid Services data-use terms even on free quota [28]
Grok	API no-training default; consumer Grok uses separate settings [33] [35]
Microsoft products	Updated personal Copilot, GitHub Copilot and Microsoft 365 Copilot have distinct rules [30] [31] [32]

These are selected policy distinctions, not complete contractual/privacy assessments or zero-retention promises.

For a UK developer, Google's exception is especially relevant. Its Gemini API terms apply the Paid Services data-use protections to users in the UK, EEA and Switzerland even when they use free AI Studio or unpaid API quota. Tool-specific logging still needs attention. That is different from the consumer Gemini app, whose activity controls have their own rules. [28] [29]

Microsoft also needs a product-level reading. Its updated personal-account Copilot app says prompts, responses and file contents are not used to train foundation models. GitHub Copilot's individual plans have a separate training opt-out, while Microsoft 365 Copilot follows organisational data and permission rules. A statement about one cannot be copied onto all the others. [30] [31] [32]

Does no training mean no storage? No. OpenAI and Grok API documentation separates default no-training rules from abuse-monitoring retention. Anthropic's covered-model policy adds retention/review requirements around the underlying Fable/Mythos models, with specific eligible exceptions. Before using confidential client material, check the intended model and route, permitted data, storage features and available controls together. [24] [27] [33]

Can I depend on this model staying available?

You need both a service plan and a version plan. A request can fail because of a rate limit, tool problem, timeout or service incident. A working application also needs a way to cope when a model alias changes or a model is retired. Those risks are practical even when the model itself produces excellent answers.

Grok's documentation distinguishes moving aliases from dated model releases. Claude's current catalogue lists retirement dates no earlier than specified dates in 2027. Neither feature means every product will behave identically forever. Preserve the exact version/configuration used for an important workflow and rerun the relevant acceptance checks after a change. [12] [10]

Which provider has the best uptime? We have not assembled a common observation window that supports that ranking. The useful questions for a business are whether the chosen service exposes status information, supplies adequate limits and support, and allows recovery or a suitable fallback. A backup route must also meet your data and quality requirements.

Which provider would I shortlist for my work?

Start with the route that fits your workflow, then check the model tier. The suggestions below follow documented roles, prices and integration choices. They are starting points for a decision, rather than a claim that we measured a universal winner.

Your main task	A sensible starting comparison	What to decide
Sustained coding	Astra/Sol, Claude coding tiers, Grok, Gemini and MAI workflow	Working changes, review effort and the product tools
High-volume extraction/routing	Luna, Haiku and Flash-Lite	Acceptance threshold, usage and escalation cost

Your main task	A sensible starting comparison	What to decide
Long documents	Larger-context options that accept your file types	Verified retrieval and billing thresholds
Office work	Available Microsoft/Google product integrations	Actual model, file access and organisational permissions
Research or writing	A route with the sources/tools and editorial fit you need	Citations, current evidence and brief compliance

Task-based shortlists from documentation and pricing; measured comparative performance remains untested in this study.

For a developer, the first useful comparison is often between a capable balanced model and a frontier tier on the same recurring jobs. For a high-volume system, begin with a small model on tightly defined tasks and decide in advance what must trigger escalation. For office work, the product's access to the documents you actually use can matter more than an isolated model leaderboard.

Do I need to subscribe to all five? Begin with one route that fits your main task. Add another when it solves a specific gap, provides a suitable fallback or handles a clearly different workload. Keep the comparison anchored to your own acceptance conditions: correct information, working code, readable prose, appropriate data handling and an affordable finished result.

How was this comparison researched?

Research checked on 13 October 2026. Financial transactions retain their announcement dates; listed market caps use the 12 October US close. Sources are official company/model/policy documents and original benchmark authors, with clearly identified secondary market-cap snapshots. All diagrams are original; cost charts calculate stated token quantities from published rates. No inference tests, provider performance measurements, account purchases or public publication were performed. Numerical costs exclude unstated taxes, exchange conversion, tools, cache/storage, speed tiers and retries. Review exact terms and prices again before a later purchasing or publication decision.

Sources and further reading

[1] [OpenAI financing announcement](https://openai.com/index/accelerating-the-next-phase-ai/)

<https://openai.com/index/accelerating-the-next-phase-ai/>

[2] [Anthropic Series H announcement](https://www.anthropic.com/news/series-h)

<https://www.anthropic.com/news/series-h>

[3] [SpaceX June 2026 SEC filing](https://www.sec.gov/Archives/edgar/data/1181412/000162828026052535/spcx-20260630.htm)

<https://www.sec.gov/Archives/edgar/data/1181412/000162828026052535/spcx-20260630.htm>

[4] [Alphabet market capitalisation snapshot](https://stockanalysis.com/stocks/googl/market-cap/)

<https://stockanalysis.com/stocks/googl/market-cap/>

[5] [Microsoft market capitalisation snapshot](https://stockanalysis.com/stocks/msft/market-cap/)

<https://stockanalysis.com/stocks/msft/market-cap/>

[6] [SpaceX market capitalisation snapshot](https://stockanalysis.com/stocks/spcx/market-cap/)

<https://stockanalysis.com/stocks/spcx/market-cap/>

- [7] [GPT 6 Astra specifications and pricing](https://developers.openai.com/api/docs/models/gpt-6-astra)
<https://developers.openai.com/api/docs/models/gpt-6-astra>
- [8] [GPT 6.1 Sol specifications and pricing](https://developers.openai.com/api/docs/models/gpt-6.1-sol)
<https://developers.openai.com/api/docs/models/gpt-6.1-sol>
- [9] [GPT 6 Luna specifications and pricing](https://developers.openai.com/api/docs/models/gpt-6-luna)
<https://developers.openai.com/api/docs/models/gpt-6-luna>
- [10] [Claude current model catalogue](https://platform.claude.com/docs/en/models/overview)
<https://platform.claude.com/docs/en/models/overview>
- [11] [Claude pricing and billing conditions](https://platform.claude.com/docs/en/about-claude/pricing)
<https://platform.claude.com/docs/en/about-claude/pricing>
- [12] [Grok current models and pricing](https://docs.x.ai/developers/models)
<https://docs.x.ai/developers/models>
- [13] [Gemini current model catalogue](https://ai.google.dev/gemini-api/docs/models)
<https://ai.google.dev/gemini-api/docs/models>
- [14] [Gemini Developer API pricing](https://ai.google.dev/gemini-api/docs/pricing)
<https://ai.google.dev/gemini-api/docs/pricing>
- [15] [Gemini 3.8 Flash specifications](https://ai.google.dev/gemini-api/docs/models/gemini-3.8-flash)
<https://ai.google.dev/gemini-api/docs/models/gemini-3.8-flash>
- [16] [MAI Thinking 1 deployment and limits](https://learn.microsoft.com/en-us/azure/foundry/foundry-models/how-to/use-foundry-models-mai-thinking)
<https://learn.microsoft.com/en-us/azure/foundry/foundry-models/how-to/use-foundry-models-mai-thinking>
- [17] [MAI Code 1.1 Flash announcement](https://microsoft.ai/news/mai-code-1-1-flash-br-better-faster-at-a-quarter-of-the-cost/)
<https://microsoft.ai/news/mai-code-1-1-flash-br-better-faster-at-a-quarter-of-the-cost/>
- [18] [Original SWE bench authors and leaderboard](https://www.swebench.com/)
<https://www.swebench.com/>
- [19] [Original Terminal Bench 4.0 announcement](https://www.tbench.ai/news/terminal-bench-4-0)
<https://www.tbench.ai/news/terminal-bench-4-0>
- [20] [OpenAI reasoning token accounting](https://developers.openai.com/api/docs/guides/reasoning)
<https://developers.openai.com/api/docs/guides/reasoning>
- [21] [Claude thinking token accounting](https://platform.claude.com/docs/en/build-with-claude/thinking)
<https://platform.claude.com/docs/en/build-with-claude/thinking>
- [22] [Gemini thinking token accounting](https://ai.google.dev/gemini-api/docs/thinking)
<https://ai.google.dev/gemini-api/docs/thinking>
- [23] [Grok reasoning token accounting](https://docs.x.ai/developers/model-capabilities/text/reasoning)
<https://docs.x.ai/developers/model-capabilities/text/reasoning>
- [24] [OpenAI API data controls](https://developers.openai.com/api/docs/guides/your-data)
<https://developers.openai.com/api/docs/guides/your-data>
- [25] [Claude commercial training policy](https://privacy.claude.com/en/articles/7996868-is-my-data-used-for-model-training)
<https://privacy.claude.com/en/articles/7996868-is-my-data-used-for-model-training>
- [26] [Claude consumer training policy](https://privacy.claude.com/en/articles/10023580-is-my-data-used-for-model-training)
<https://privacy.claude.com/en/articles/10023580-is-my-data-used-for-model-training>
- [27] [Claude covered model retention policy](https://privacy.claude.com/en/articles/15425996-data-retention-practices-for-covered-models)
<https://privacy.claude.com/en/articles/15425996-data-retention-practices-for-covered-models>
- [28] [Gemini API terms including regional data rules](https://ai.google.dev/gemini-api/terms?hl=en)
<https://ai.google.dev/gemini-api/terms?hl=en>

[29] [Consumer Gemini privacy hub](#)

<https://support.google.com/gemini/answer/13594961?hl=en>

[30] [Updated consumer Copilot activity policy](#)

<https://support.microsoft.com/en-us/privacy/microsoft-copilot/activity-history>

[31] [GitHub Copilot account and training policies](#)

<https://docs.github.com/en/copilot/how-tos/manage-your-account/manage-policies>

[32] [Microsoft 365 Copilot privacy](#)

<https://learn.microsoft.com/en-us/microsoft-365/copilot/microsoft-365-copilot-privacy>

[33] [Grok API security and retention](#)

<https://docs.x.ai/developers/faq/security>

[34] [Copilot memory and privacy controls](#)

<https://support.microsoft.com/en-us/privacy/microsoft-copilot/privacy-controls>

[35] [Consumer Grok privacy settings](#)

<https://x.ai/legal/faq>