KIMI K3

PROMPTING GUIDE

MASTER PROMPT ENGINEERING FOR MOONSHOT AI'S 2.8T-PARAMETER FRONTIER MODEL

- UNDERSTAND KIMI K3 ARCHITECTURE
- MASTER ADVANCED PROMPTING TECHNIQUES
- NAVIGATE STRENGTHS AND LIMITATIONS
- BUILD RELIABLE, SCALABLE, AND COST-EFFICIENT WORKFLOWS

TONY SAN MARTIN

Kimi K3 Prompting Guide

Master Prompt Engineering for Moonshot AI's 2.8T-Parameter Frontier Model

Steve Publications

This book is available at https://leanpub.com/kimik3promptingguide

This version was published on 2026-07-22

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

Master Prompt Engineering for Moonshot AI's 2.8T-Parameter Frontier Model

Kimi K3 is not just another large language model. Released in July 2026 as the world's first open 3-trillion-parameter-class model, it represents a fundamental shift in what frontier intelligence looks like: always-on reasoning, native multimodality, a million-token context window, and architecture designed for long-horizon agentic work. This book teaches you how to prompt Kimi K3 effectively across every major use case, ranging from software engineering and research to enterprise knowledge work and autonomous agent systems. You will learn the model's architecture so you understand why it behaves as it does, master foundational and advanced prompting techniques with annotated examples, navigate its strengths and limitations honestly, and build production-ready workflows that are reliable, cost-efficient, and scalable. Whether you are encountering K3 for the first time or optimizing a deployed system at scale, this guide gives you the complete toolkit.

Introduction: The Kimi K3 Moment

On July 16, 2026, Moonshot AI released Kimi K3. It was not just another model launch in an increasingly crowded field. At 2.8 trillion parameters with a one-million-token context window, native visual understanding, and always-on reasoning mode, K3 claimed the title of the world's first open model in the three-trillion-parameter class [1]. Within days it had taken the top spot on LMArena's Frontend Code Arena at 1,679 points, surpassing Claude Fable 5 at 1,631 [2]. It scored 57 on the Artificial Analysis Intelligence Index v4.1, placing fourth globally behind only the two most expensive proprietary models and narrowly ahead of Claude Opus 4.8 [3]. On the GDPval-AA v2 agentic benchmark, it achieved an Elo rating of approximately 1,668 to 1,687 depending on the source, placing it third overall behind only Fable 5 and GPT-5.6 Sol [4].

This is the backdrop against which you are about to learn how to prompt Kimi K3. The numbers matter because they tell you something about what you are dealing with: this is not a mid-tier model that happens to have a large context window. This is a frontier-class reasoning engine, built from the ground up for long-horizon tasks, and it requires a different prompting discipline than the models most developers are used to.

Why This Book Exists

Most prompt engineering guides are written for generic large language models or for the previous generation of systems. They teach you patterns that work adequately on simple tasks but fall apart when you need reliable, production-grade performance from a model like K3. There are three reasons for this.

First, K3's architecture is unusual. It uses Kimi Delta Attention, a hybrid linear attention mechanism that interleaves efficient recurrent-style layers with full-attention layers in a 3:1 ratio [5]. It uses Attention Residuals, which replace standard identity skip connections with learned, input-dependent weighting across layer depth [6]. It uses a Mixture-of-Experts design with 896 expert subnetworks but activates only 16 per token, making it a sparse model whose total parameter count tells you nothing about its per-token compute

cost [7]. These architectural choices shape how K3 reads your prompts, what information it retains, and where it is likely to fail.

Second, K3 reasons by default. Every response carries a reasoning trace generated before the final answer appears. You cannot turn thinking off. You can only control how much effort the model invests through the `reasoning_effort` parameter, which currently supports "low", "high", and "max" settings, with "max" as the default [8]. This changes everything about prompt design. You do not need to coax K3 into showing its work. Instead, you need to structure prompts that give the model the right information to reason over, in the right order, with the right constraints.

Third, K3 is built for endurance. Its stated purpose is long-horizon coding and end-to-end knowledge work: tasks that span dozens of tool calls, hundreds of files, or thousands of pages of documents [9]. Prompting for such workloads requires thinking about context management, caching behavior, multi-turn consistency, and failure recovery in ways that single-shot prompting guides never address.

What You Will Learn

This book takes you from understanding the model to deploying it in production. Here is the journey:

We begin by explaining K3's architecture so you understand why it behaves as it does. You will learn about Kimi Delta Attention and how it enables efficient million-token contexts, Attention Residuals and how they improve information flow through deep networks, the Stable LatentMoE expert routing system, and the model's native multimodal capabilities.

We then give you an honest assessment of K3's strengths and limitations. It excels at coding endurance tasks like SWE Marathon, where it scored 42.0 ahead of both Opus 4.8 and GPT-5.6 Sol [10]. It dominates frontend code generation on LMArena [2]. But its hallucination rate, independently measured at 51% by Artificial Analysis, is higher than its predecessor K2.6's 39%, despite improved accuracy [11]. It is sensitive to how thinking history is preserved across conversation turns, and it can be excessively proactive when executing agentic tasks without clear boundaries [12]. You need to know all of this before building anything important.

The middle chapters cover prompting technique in depth. We start with foundational principles: how K3 reads your prompt, the role of system prompts, instruction specificity, delimiters, and output control. Then we move to advanced patterns: structured JSON outputs with schema validation, XML-based prompt organization, dynamic tool loading for large function inventories, iterative refinement loops, partial continuations for prefix-driven generation, and more.

We dedicate substantial attention to long-context mastery because K3's million-token window is its killer feature for many workflows. You will learn how context caching works and why it can reduce your effective input cost from $3 per million tokens to $0.30 per million tokens on stable prefixes [13]. You will learn about positional effects: how information at the beginning, middle, and end of a prompt is weighted differently, and what to do about it. You will learn when long context beats traditional retrieval-augmented generation and when RAG still wins.

Agentic workflows get their own deep dive because K3 was built for them. We cover tool-calling patterns, error handling, multi-step task decomposition, integration with Kimi Code and AGENTS.md configuration files, and the Agent Swarm system that can coordinate up to 300 sub-agents in parallel [14].

We dedicate a chapter to multilingual prompting because K3 is developed by a Chinese company with asymmetric language strengths. You will learn how to work with K3 across Chinese, English, and other languages; how tokenization efficiency varies by language and affects your costs; how to handle code-switching in mixed-language prompts; and what language-specific hallucination risks to watch for.

Industry-specific chapters show you how to apply K3 to real work: software engineering workflows for large codebases, research synthesis across hundreds of documents, financial analysis of quarterly reports, legal contract review, healthcare document processing, marketing and content creation pipelines, personalized education and tutoring, customer support automation, and data analysis with SQL generation and statistical interpretation.

We then turn to the engineering side: prompt debugging methodologies with real-world failure case studies, evaluation harnesses, hallucination mitigation strategies given K3's elevated fabrication rate, cost optimization, API integration patterns, rate limit management, self-hosting considerations with vLLM, and multi-model routing architectures.

Finally, we look ahead at what happens when the promised open weights arrive by July 27, 2026 [15], and how fine-tuning, distillation, and customization will change the prompting landscape.

How to Use This Book

Read sequentially if you are new to K3. Each chapter builds on the previous ones: you need to understand the architecture before you can appreciate why certain prompting patterns work, and you need the foundational techniques before the advanced ones make sense.

If you are already experienced with large language models, use this book as a reference. Jump to the chapters most relevant to your immediate needs: long-context strategies if you are processing documents, agentic workflows if you are building autonomous systems, production deployment if you are engineering a service around K3.

Every chapter contains annotated prompts that you can adapt for your own use. Treat them as starting points, not templates to copy blindly. The best prompt for K3 depends on your specific task, your constraints, and your tolerance for risk.

A Note on Honesty

This book does not hype Kimi K3. It is an excellent model with genuine strengths that make it a compelling choice for many workloads. But it also has real limitations: a high hallucination rate, sensitivity to thinking history handling, excessive proactiveness in agent mode, and a current UX gap compared to the strongest proprietary models [12]. We will address each of these directly and give you strategies for working around them.

We will also be honest about what we do not know. At the time of writing, Moonshot AI's full technical report for K3 has not yet been published, though it is promised alongside the open weights release [15]. Some architectural details are based on official announcements, community analysis, and inference from the model's behavior rather than peer-reviewed documentation. Where information is uncertain or incomplete, we will say so explicitly.

The Promise

By the end of this book, you will be able to:

- Explain K3's architecture in enough detail to predict how it will handle different types of prompts
- Design effective system prompts and user prompts for coding, reasoning, knowledge work, and creative tasks
- Control K3's reasoning depth appropriately for each task type
- Structure long-context inputs that leverage the million-token window efficiently
- Build agentic workflows with reliable tool use and error handling
- Mitigate hallucinations through prompt design and verification pipelines
- Deploy K3-powered systems in production with proper cost, latency, and reliability management
- Evaluate whether K3 is the right choice for your specific workload

Let us begin.

Chapter 1: Understanding Kimi K3, Architecture and Capabilities

To prompt Kimi K3 well, you need to understand what it is under the hood. Architecture shapes behavior. A model that uses linear attention processes long contexts differently than one that relies purely on full quadratic attention. A Mixture-of-Experts design routes different tokens through different subnetworks, which affects both capability and consistency. A model trained with preserved thinking history expects its reasoning traces to be fed back across turns, and it degrades when they are not.

This chapter explains K3's architecture in accessible but technically accurate terms. We will cover the five core components that define how K3 works: its Mixture-of-Experts structure, Kimi Delta Attention, Attention Residuals, native multimodality, and its benchmark positioning relative to other frontier models. By the end, you will understand not just what K3 can do but why it does what it does.

The 2.8 Trillion Parameter Claim

The number 2.8 trillion is Kimi K3's headline specification, and it requires careful interpretation. It describes total stored parameters, not per-token compute cost, and the distinction matters enormously for anyone thinking about inference efficiency or self-hosting requirements.

K3 uses a sparse Mixture-of-Experts architecture. The model contains 896 specialized "expert" subnetworks, but for any given token, only 16 of those experts are activated [7]. This is the Stable LatentMoE routing framework that Moonshot developed to scale beyond the trillion-parameter regime while keeping inference tractable. Each token is routed through a small subset of the total capacity, selected by a router network that evaluates which experts are most relevant for that particular input.

The practical implication is straightforward: when you send a prompt to K3, you are not paying for all 2.8 trillion parameters to process every token. You

are paying for approximately 16 out of 896 experts, plus the shared components they draw from, to handle each position in your sequence. This sparsity is what makes a model of this scale economically viable at inference time, despite the enormous total parameter count.

For prompt engineers, this architecture has several consequences. First, different types of input may route to different expert subsets. A coding prompt and a creative writing prompt could activate substantially different portions of the model, which can explain variations in quality across domains. Second, the routing mechanism adds a layer of decision-making that is invisible to you but affects the model's behavior. Third, the total parameter count represents capacity and specialization breadth, not raw processing power per token.

The MoE design also means K3 benefits from quantization-aware training starting at the supervised fine-tuning stage. Moonshot uses MXFP4 weights with MXFP8 activations, which enables deployment across a range of hardware configurations while preserving quality [16]. For self-hosting after the promised July 27, 2026 weight release, this means you will need serious infrastructure: vLLM has announced day-zero support, but expect requirements in the range of 64 or more accelerators for full-precision serving, with quantized deployments requiring fewer resources [17].

Kimi Delta Attention: Linear Efficiency at Scale

Kimi Delta Attention, abbreviated as KDA, is one of two architectural innovations that Moonshot highlights as foundational to K3's design. It addresses a fundamental problem: how to maintain high-quality attention over extremely long sequences without the quadratic compute cost of standard transformer attention.

Standard multi-head attention compares every token with every other token, which means memory and compute scale as O(n^2) where n is sequence length. At one million tokens, that is computationally prohibitive. Linear attention mechanisms like KDA reduce this to approximately O(n) by maintaining a compressed recurrent state that summarizes past information, rather than storing all pairwise relationships explicitly.

KDA is not a pure linear mechanism. It is hybrid: three KDA layers are interleaved with one full-attention layer in a 3:1 ratio throughout the model [5]. This design choice reflects a practical insight. Pure linear attention can

struggle with exact retrieval of specific tokens from far back in the sequence, because the recurrent state compresses information and can lose precise details. Full attention layers, inserted periodically, recover this capability by providing direct access to all previous positions at those points.

The measured benefit is substantial. Moonshot reports up to 6.3 times faster decoding at one-million-token contexts compared with full-attention baselines, along with approximately 75 percent reduction in KV cache memory requirements [18]. These are not marginal improvements; they are what make K3's context window practically deployable rather than theoretically possible.

For prompting, KDA has implications for how information flows through long sequences. The recurrent-state mechanism means that earlier tokens are summarized and compressed as the model processes forward. This can lead to degradation in recall for very old information, especially if it was not structurally emphasized in the prompt. You cannot assume that K3 will remember every detail from token position 50,000 when generating at position 900,000 with perfect fidelity. Strategic placement of critical information, repetition of key constraints, and structured formatting all become more important than they would be in a shorter-context model.

Attention Residuals: Information Flow Through Depth

If Kimi Delta Attention addresses the sequence-length dimension, Attention Residuals, abbreviated as AttnRes, address the depth dimension. They solve a different but equally fundamental problem: how information flows through very deep networks.

In a standard transformer, each layer takes its input, applies transformations, and adds the result back to the original input through an identity skip connection. This is the residual connection that has been central to deep network design since 2015. But as networks grow deeper, this simple addition begins to fail. The accumulated sum of all previous layers' outputs grows in magnitude, drowning out contributions from later layers. Gradients weaken as they propagate backward. Information from early layers becomes difficult for late layers to access selectively. This is what the Kimi team calls "PreNorm dilution" [19].

AttnRes replaces this fixed accumulation with learned, input-dependent weighting. Instead of uniformly adding every layer's output, each layer forms

its input by attending over all preceding layer outputs, selecting which earlier representations are most relevant for its current computation. Think of it as applying the same attention mechanism that operates across sequence positions, but rotated 90 degrees to operate across depth instead [20].

The full version of AttnRes is computationally expensive at scale, so K3 uses Block AttnRes: layers are grouped into approximately eight blocks, with ordinary residual connections within each block and cross-layer attention only between blocks. This reduces the memory cost from O(Ld), where L is depth and d is hidden dimension, to O(Nd), where N is the number of blocks [21].

The empirical payoff is clear. On the 48-billion-parameter Kimi Linear model, Block AttnRes delivered a 7.5-point improvement on GPQA-Diamond and matched the performance of a baseline trained with 1.25 times more compute [22]. It enables deeper, narrower architectures that fully utilize layer depth rather than suffering from progressive dilution.

For prompt engineers, AttnRes matters indirectly but importantly. A model that can selectively retrieve information across its own depth is better equipped to maintain coherence in complex reasoning tasks, to remember constraints stated early in a prompt while processing detailed instructions later, and to integrate information from different parts of a long input. The deeper the network and the more selective its internal information routing, the less likely it is to lose track of your original intent as processing continues.

Stable LatentMoE and Expert Routing

The Stable LatentMoE framework is Moonshot's solution to a persistent problem in Mixture-of-Expert models: load balancing. When you have hundreds of experts and a router deciding which ones to activate for each token, there is a natural tendency for some experts to become overloaded while others sit idle. This imbalance degrades training efficiency and can hurt inference quality.

Traditional MoE systems use heuristic load-balancing rules: penalties on popular experts, capacity limits, random dropping of overflow tokens. These approaches work but introduce hyperparameters that are sensitive and difficult to tune at scale. Stable LatentMoE takes a different approach: Quantile Balancing, which allocates experts based on router-score quantiles rather than heuristic rules [23]. The system derives expert allocation directly from where

each token's router scores fall in the overall distribution, eliminating the need for separate balancing hyperparameters.

K3 also uses Per-Head Muon, an extension of the Muon optimizer that adjusts attention heads independently rather than applying a single learning rate schedule across all heads [24]. This allows different attention mechanisms to learn at different paces, which is important in a model with thousands of attention heads distributed across many layers.

Additional components include the Sigmoid Tanh Unit (SiTU), a custom activation function that replaces standard GeLU or SwiGLU, and Gated Multi-head Latent Attention (Gated MLA) for more selective and memory-efficient KV-cache management [25]. Together, these innovations enable stable training at the 2.8-trillion-parameter scale, which is no small engineering achievement.

Native Multimodality: Vision, Video, and Documents

Kimi K3 processes text, images, and video within a single unified architecture rather than bolting on a separate vision encoder. This native multimodal design means that visual and textual information are represented in the same neural space from the earliest layers, which supports richer cross-modal reasoning [26].

For prompt engineers, this has practical implications. You can mix text and images in a single request without special formatting or separate API calls. The model understands screenshots, diagrams, charts, documents, and video frames alongside code and natural language instructions. It can reference specific regions within an image, compare multiple images, track changes across video frames, and combine visual observations with textual reasoning.

K3's vision capabilities are particularly strong in software engineering contexts. Moonshot highlights screenshot-driven development workflows where the model iterates between code generation and visual feedback: generating frontend code, taking a screenshot of the rendered result, analyzing the visual output, and refining the code accordingly [27]. This "vision in the loop" pattern is enabled by native multimodality and is difficult to replicate with models that treat vision as an afterthought.

Technical details matter for implementation. K3 requires images to be encoded as Base64 data URLs; it does not support public image URLs directly [28]. Video files must be uploaded through the Files API and referenced

with special ms:// identifiers. These constraints affect how you design your prompts and workflows, especially in production systems.

Benchmark Positioning: Where K3 Stands in the Frontier Landscape

Understanding where K3 sits relative to other models helps set realistic expectations for what it can achieve. The benchmark landscape is noisy and often misleading, but several independent evaluations provide a reasonably clear picture.

On the Artificial Analysis Intelligence Index v4.1, which aggregates performance across reasoning, knowledge, mathematics, and coding tasks, K3 scores 57 out of a maximum of approximately 60-65 [3]. This places it fourth globally: behind Claude Fable 5 at approximately 60 and GPT-5.6 Sol at approximately 59, but ahead of Claude Opus 4.8 at approximately 56. The margin between K3 and the top two models is small enough that task-specific performance matters more than overall ranking.

On coding benchmarks, K3's positioning is more nuanced. It leads on SWE Marathon at 42.0, a test of sustained multi-step engineering tasks, ahead of Opus 4.8 (40.0), GPT-5.6 Sol (39.0), and Fable 5 (35.0) [10]. On Program Bench, it scores 77.8, edging both Sol (77.6) and Fable 5 (76.8) [29]. On Terminal-Bench 2.1, it achieves 88.3, trailing only Sol by half a point at 88.8 [30]. However, on DeepSWE, which tests deep repository-level comprehension, GPT-5.6 Sol leads at 73.0 versus K3's 67.5 [31]. This pattern suggests that K3 excels at endurance and breadth but lags slightly on the deepest, most complex single tasks.

On LMArena's Frontend Code Arena, where humans vote on which model builds better frontend code in blind head-to-head comparisons, K3 entered at number one with 1,679 points, ahead of Fable 5's 1,631 [2]. This is a particularly meaningful benchmark because it reflects actual human preference for a concrete engineering task rather than automated scoring on synthetic problems.

On the GDPval-AA v2 agentic benchmark, which measures real-world task completion across 44 occupations and nine industries, K3 scores an Elo of approximately 1,668 to 1,687 depending on the source [4]. This places it third overall, behind Fable 5 (approximately 1,815) and Sol (approximately 1,748), but ahead of Opus 4.8 (approximately 1,600). The agentic benchmark is important

because it tests tool use, planning, and multi-step execution rather than isolated reasoning or generation.

One caution: K3's benchmark results are currently reported primarily by Moonshot AI and a limited set of third-party evaluators. Independent reproduction at the weight level will not be possible until the open weights release, which is scheduled for July 27, 2026 [15]. Until then, treat official numbers as credible but unverified.

Summary

Kimi K3 is a carefully engineered system designed to push frontier intelligence into the open-weight domain. Its architecture, with Kimi Delta Attention for efficient long contexts, Attention Residuals for deep-network coherence, Stable LatentMoE for scalable expert routing, and native multimodality for unified visual-textual reasoning, is not a minor variation on existing designs. It represents a coherent vision of how to build a model that can handle extended, complex, tool-using workloads at scale.

Understanding this architecture helps you prompt K3 better because it tells you where the model's strengths come from and where its vulnerabilities lie. The linear attention mechanism means long-context recall is good but not perfect. The MoE design means different inputs may activate different expert subsets. The always-on reasoning mode means you work with thinking traces, not around them. The agentic training means K3 wants to take initiative, sometimes too much so.

The next chapter examines those strengths and vulnerabilities in detail, giving you an honest portrait of what K3 can reliably deliver and where you need to build safeguards.

Chapter 2: Capabilities, Strengths, and Limitations

Coding Excellence: Long-Horizon Engineering and Frontend Dominance

Reasoning Strengths: Math, Logic, and Agentic Intelligence

The Hallucination Problem: Why Accuracy Rose But Fabrication Did Too

Thinking History Sensitivity and Multi-Turn Fragility

Excessive Proactiveness: When K3 Decides Without Asking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Cybersecurity Gaps and Guardrail Tradeoffs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Chapter 3: Foundational Prompting Principles for K3

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

How K3 Reads Your Prompt: Tokenization, Context Positioning, and Attention Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

The Role of System Prompts in K3 Workflows

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Instruction Specificity and Unambiguous Task Definition

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Delimiters, Structure, and Input Segmentation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Few-Shot vs Zero-Shot: When Examples Matter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Output Control: Length, Format, and Constrained Responses

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Chapter 4: Prompt Anatomy, Building Effective K3 Prompts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

The Six Layers of a Production Prompt

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Role Prompting: Persona Assignment and Expertise Calibration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Instruction Hierarchy: Primary Directives vs Secondary Preferences

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Context Engineering: What Goes In, Where, and Why It Matters

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Annotated Prompt Walkthroughs: Before and After Comparisons

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Common Structural Anti-Patterns to Avoid

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Chapter 5: Reasoning Mode and Thinking Effort Control

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Always-On Thinking: How K3 Reasons Before Responding

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

The reasoning_effort Parameter: Low, High, and Max Explained

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Matching Reasoning Depth to Task Complexity

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Preserving Thinking History Across Turns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Cost Implications of Reasoning Tokens

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

When to Reduce Effort Without Sacrificing Quality

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Chapter 6: Advanced Prompting Techniques

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Chain-of-Thought Alternatives and Self-Directed Reasoning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Structured Outputs with JSON Schema and Strict Mode

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

XML and Tag-Based Prompting Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Dynamic Tool Loading and Function Calling Prompts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Iterative Refinement and Self-Correction Loops

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Partial Continuations and Prefix-Driven Generation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Chapter 7: Long Context Mastery, Working with 1M Tokens

The Reality of Million-Token Context: Opportunities and Pitfalls

Context Caching: Architecture, Cost Optimization, and Best Practices

Document Chunking vs Whole-Document Loading Strategies

Positional Effects: Beginning, Middle, and End Phenomena

Multi-Document Reasoning and Cross-Reference Tasks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

RAG vs Long Context: When Each Approach Wins

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Chapter 8: Agentic Workflows and Tool Use

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Agent-Centric Prompt Design: Planning, Execution, and Verification

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Tool Calling Patterns: Definitions, Selection, and Error Handling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Dynamic Tool Injection and Inventory Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Multi-Step Task Decomposition Prompts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Kimi Code Integration and AGENTS.md Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Swarm Agents: Parallel Execution and Coordination

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Chapter 9: Multimodal Prompting, Vision and Video

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Native Multimodality vs Bolt-On OCR Approaches

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Image Input Formats, Resolution Guidelines, and Encoding

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Screenshot-Driven Development and Visual Debugging

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Document Understanding: PDFs, Spreadsheets, and Charts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Video Reasoning and Temporal Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Multi-Image Comparison and Layout Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Chapter 10: Multilingual Prompting with Kimi K3

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Language Strength Profile: Where K3 Excels and Where Caution Is Needed

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Tokenization Efficiency Across Languages: Cost and Context Implications

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Cross-Lingual Reasoning and Translation Workflows

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Code-Switching and Mixed-Language Prompts in the 1M Context Window

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Language-Specific Hallucination Risks and Mitigation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Practical Patterns for Global Teams Using K3

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Chapter 11: Industry-Specific Applications

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Software Engineering: Codebases, Refactoring, and Testing Workflows

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Research and Academic Work: Literature Synthesis and Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Finance and Quantitative Analysis: Reports, Models, and Risk Assessment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Legal and Compliance: Contract Review, Clause Extraction, and Redlining

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Healthcare and Life Sciences: Document Processing and Regulatory Contexts

Marketing, Content, and Creative Workflows

Education: Personalized Tutoring, Curriculum Generation, and Assessment

Customer Support: Ticket Triage, Sentiment Analysis, and Live Agent Assist

Data Analysis: SQL Generation, Statistical Interpretation, and Visualization

Summary

Chapter 12: Prompt Debugging, Testing, and Evaluation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Diagnosing Common Failure Modes: Refusals, Hallucinations, and Drift

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Case Study: How a Subtle Delimiter Error Caused Costly API Misbehavior

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Case Study: Prompt Injection in a Customer-Facing Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Systematic Prompt Testing Methodologies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Building Evaluation Harnesses and Quality Gates

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

A/B Testing Prompts in Production Environments

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Monitoring for Degradation and Distribution Shifts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Using Official Benchmarks vs Custom Evaluations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Chapter 13: Hallucination Mitigation and Safety

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Understanding K3's Hallucination Profile (51% Rate Explained)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Case Study: Fabricated Legal Citations in a Contract Review Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Case Study: Medical Hallucination in a Clinical Decision Support Tool

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Document Grounding and Source-Constrained Generation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Confidence Calibration and Uncertainty Signaling Prompts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Multi-Pass Verification and Self-Correction Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Output Validation Pipelines and Post-Processing Checks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Safety Considerations: Guardrails, Content Filters, and Compliance

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Chapter 14: Production Deployment and Optimization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

API Integration: Authentication, Rate Limits, and Error Handling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Cost Optimization Strategies: Caching, Tiering, and Token Efficiency

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Case Study: When Context Caching Failed and How We Fixed It

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Latency Management and Streaming Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Self-Hosting Considerations: Hardware Requirements and vLLM Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Multi-Model Routing and Fallback Architectures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Observability, Logging, and Debugging in Production

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Chapter 15: The Future of K3 Prompting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Open Weights Implications: Fine-Tuning, Distillation, and Customization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Evolving Reasoning Modes and Effort Granularity

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Integration with Agent Frameworks and Orchestration Platforms

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

The Prompting-to-Context-Engineering Shift

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Long-Term Trajectory: K3 in the Broader AI Ecosystem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Conclusion: Delivering on the Promise

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

The Core Framework

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

When to Use K3

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

The Practitioner's Checklist

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

Looking Forward

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.

References

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/kimik3promptingguide.