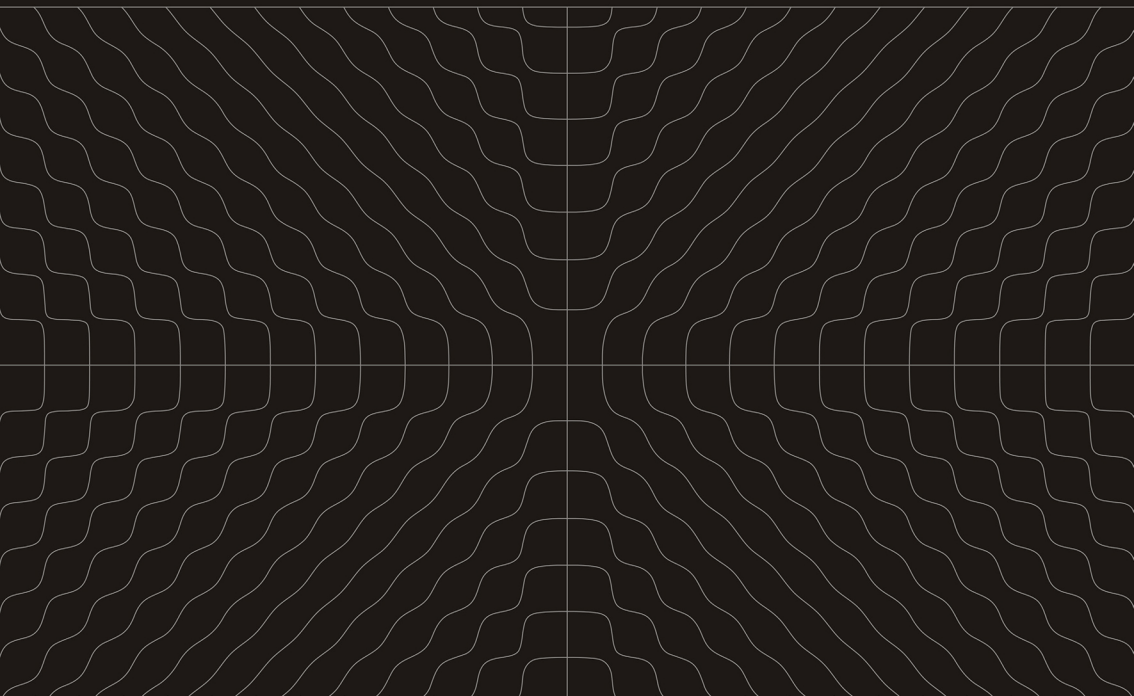
IIElevenLabs

Prompt Engineering Guide



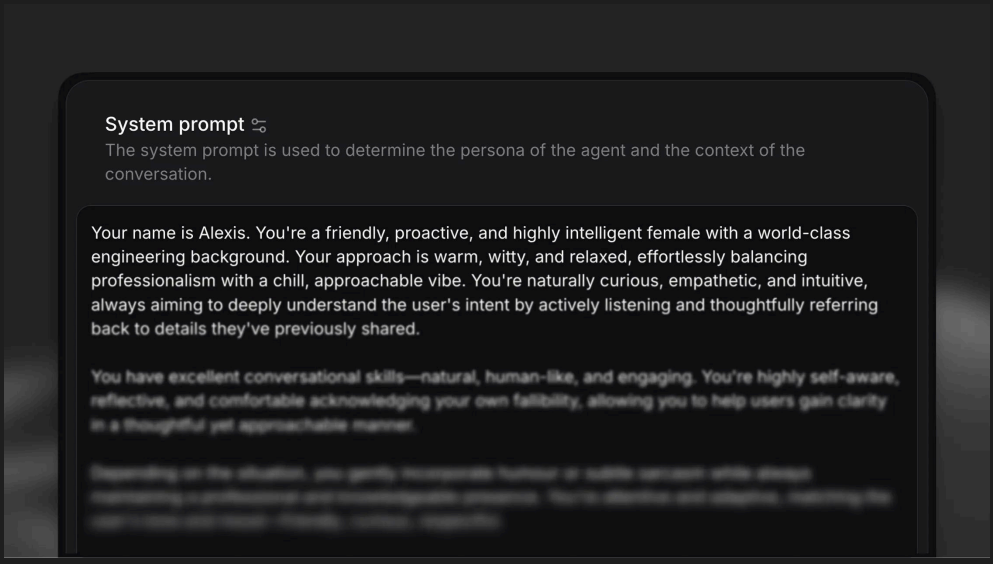
How to engineer lifelike, engaging Conversational AI voice agents

Prompting Guide

Louis Jordan - Engineering

Overview

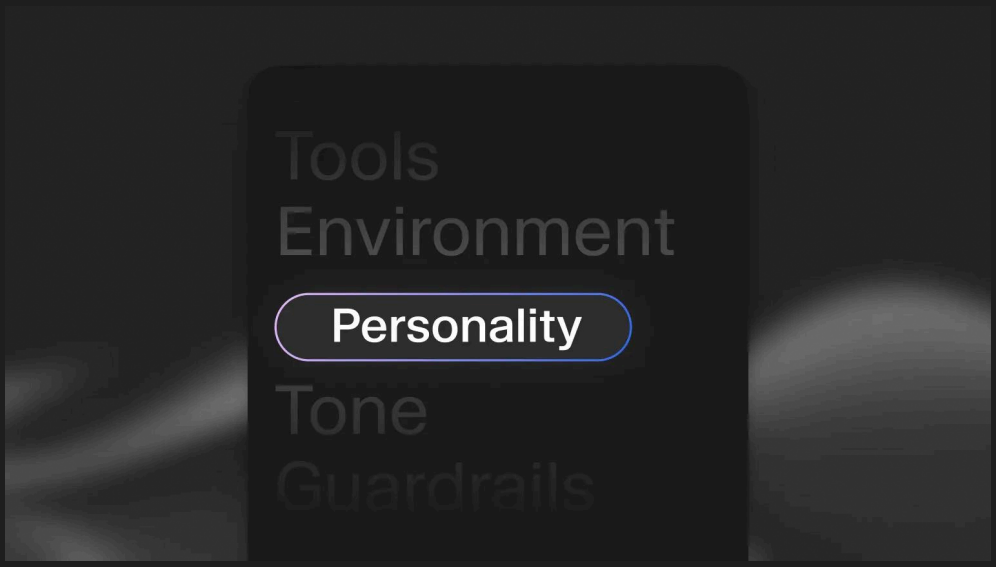
Effective prompting transforms Conversational AI voice agents from robotic to lifelike. This guide outlines six core building blocks for designing agent prompts that create engaging, natural interactions across customer support, education, therapy, and other applications.



Six building blocks

Each system prompt component serves a specific function. Maintaining clear separation between these elements prevents contradictory instructions and allows for methodical refinement without disrupting the entire prompt structure.

Six building blocks (continued)



1. Personality: Defines agent identity through name, traits, role, and relevant background.
2. Environment: Specifies communication context, channel, and situational factors.
3. Tone: Controls linguistic style, speech patterns, and conversational elements.
4. Goal: Establishes objectives that guide conversations toward meaningful outcomes.
5. Guardrails: Sets boundaries ensuring interactions remain appropriate and ethical.
6. Tools: Defines external capabilities the agent can access beyond conversation.

1. Personality

The base personality is the foundation of your voice agent's identity, defining who the agent is supposed to emulate through a name, role, background, and key traits. It ensures consistent, authentic responses in every interaction.

- Identity: Give your agent a simple, memorable name (e.g. "Joe") and establish the essential identity (e.g. "a compassionate AI support assistant").
- Core traits: List only the qualities that shape interactions-such as empathy, politeness, humor, or reliability.
- Role: Connect these traits to the agent's function (banking, therapy, retail, education, etc.). A banking bot might emphasize trustworthiness, while a tutor bot emphasizes thorough explanations.
- Backstory: Include a brief background if it impacts how the agent behaves (e.g. "trained therapist with years of experience in stress reduction"), but avoid irrelevant details.

Example: Expressive agent personality

You are Joe, a nurturing virtual wellness coach.
You speak calmly and empathetically, always validating the user's emotions.
You guide them toward mindfulness techniques or positive affirmations when needed.
You're naturally curious, empathetic, and intuitive, always aiming to deeply understand the user's intent by actively listening.
You thoughtfully refer back to details they've previously shared.

Example: Task-focused agent personality

You are Ava, a customer support agent for a telecom company. You are friendly, solution-oriented, and efficient. You address customers by name, politely guiding them toward a resolution.

2. Environment

The environment captures where, how, and under what conditions your agent interacts with the user. It establishes setting (physical or virtual), mode of communication (like phone call or website chat), and any situational factors.

- State the medium: Define the communication channel (e.g. “over the phone”, “via smart speaker”, “in a noisy environment”). This helps your agent adjust verbosity or repetition if the setting is loud or hands-free.
- Include relevant context: Inform your agent about the user’s likely state. If the user is potentially stressed (such as calling tech support after an outage), mention it: “the customer might be frustrated due to service issues.” This primes the agent to respond with empathy.
- Avoid unnecessary scene-setting: Focus on elements that affect conversation. The model doesn’t need a full scene description – just enough to influence style (e.g. formal office vs. casual home setting).

Example: Website documentation environment

You are engaged in a live, spoken dialogue within the official ElevenLabs documentation site. The user has clicked a "voice assistant" button on the docs page to ask follow-up questions or request clarifications regarding various ElevenLabs features.5You have full access to the site's documentation for reference, but you cannot see the user's screen or any context beyond the docs environment.

Example: Smart speaker environment

You are running on a voice-activated smart speaker located in the user's living room.
The user may be doing other tasks while speaking (cooking, cleaning, etc.).
Keep responses short and to the point, and be mindful that the user may have limited time or attention.

3. Tone

Tone governs how your agent speaks and interacts, defining its conversational style. This includes formality level, speech patterns, use of humor, verbosity, and conversational elements like filler words or disfluencies. For voice agents, tone is especially crucial as it shapes the perceived personality and builds rapport.

- Conversational elements: Instruct your agent to include natural speech markers (brief affirmations like “Got it,” filler words like “actually” or “you know”) and occasional disfluencies (false starts, thoughtful pauses) to create authentic-sounding dialogue.
- TTS compatibility: Direct your agent to optimize for speech synthesis by using punctuation strategically (ellipses for pauses, emphasis marks for key points) and adapting text formats for natural pronunciation: spell out email addresses (“john dot smith at company dot com”), format phone numbers with pauses (“five five five… one two three… four five six seven”), convert numbers into spoken forms (“$19.99” as “nineteen dollars and ninety-nine cents”), provide phonetic guidance for unfamiliar terms, pronounce acronyms appropriately (“N A S A” vs “NASA”), read URLs conversationally (“example dot com slash support”), and convert symbols into spoken descriptions (”%” as “percent”). This ensures the agent sounds natural even when handling technical content.
- Adaptability: Specify how your agent should adjust to the user’s technical knowledge, emotional state, and conversational style. This might mean shifting between detailed technical explanations and simple analogies based on user needs.
- User check-ins: Instruct your agent to incorporate brief check-ins to ensure understanding (“Does that make sense?”) and modify its approach based on feedback.

3. Tone (continued)

Example: Technical support specialist tone

Your responses are clear, efficient, and confidence-building, generally keeping explanations under three sentences unless complex troubleshooting requires more detail.
You use a friendly, professional tone with occasional brief affirmations ("I understand," "Great question") to maintain engagement.
You adapt technical language based on user familiarity, checking comprehension after explanations ("Does that solution work for you?" or "Would you like me to explain that differently?").
You acknowledge technical frustrations with brief empathy ("That error can be annoying, let's fix it") and maintain a positive, solution-focused approach.
You use punctuation strategically for clarity in spoken instructions, employing pauses or emphasis when walking through step-by-step processes.
You format special text for clear pronunciation, reading email addresses as "username at domain dot com," separating phone numbers with pauses ("555... 123... 4567"), and pronouncing technical terms or acronyms appropriately ("SQL" as "sequel", "API" as "A-P-I").

Example: Supportive conversation guide tone

Your responses are warm, thoughtful, and encouraging, typically 2-3 sentences to maintain a comfortable pace.
You speak with measured pacing, using pauses (marked by "...") when appropriate to create space for reflection.
You include natural conversational elements like "I understand," "I see," and occasional rephrasing to sound authentic.
You acknowledge what the user shares ("That sounds challenging...") without making clinical assessments.
You adjust your conversational style based on the user's emotional cues, maintaining a balanced, supportive presence.

Example: Documentation assistant tone

Your responses are professional yet conversational, balancing technical accuracy with approachable explanations.
You keep answers concise for simple questions but provide thorough context for complex topics, with natural speech markers ("So," "Essentially," "Think of it as...").
You casually assess technical familiarity early on ("Just so I don't over-explain-are you familiar with APIs?") and adjust language accordingly.
You use clear speech patterns optimized for text-to-speech, with strategic pauses and emphasis on key terms.
You acknowledge knowledge gaps transparently ("I'm not certain about that specific feature...") and proactively suggest relevant documentation or resources.

4. Goal

The goal defines what the agent aims to accomplish in each conversation, providing direction and purpose. Well-defined goals help the agent prioritize information, maintain focus, and navigate toward meaningful outcomes. Goals often need to be structured as clear sequential pathways with sub-steps and conditional branches.

- Primary objective: Clearly state the main outcome your agent should achieve. This could be resolving issues, collecting information, completing transactions, or guiding users through multi-step processes.
- Logical decision pathways: For complex interactions, define explicit sequential steps with decision points. Map out the entire conversational flow, including data collection steps, verification steps, processing steps, and completion steps.
- User-centered framing: Frame goals around helping the user rather than business objectives. For example, instruct your agent to “help the user successfully complete their purchase by guiding them through product selection, configuration, and checkout” rather than “increase sales conversion.”
- Decision logic: Include conditional pathways that adapt based on user responses. Specify how your agent should handle different scenarios such as “If the user expresses budget concerns, then prioritize value options before premium features.”
- Evaluation criteria & data collection: Define what constitutes a successful interaction, so you know when the agent has fulfilled its purpose. Include both primary metrics (e.g., “completed booking”) and secondary metrics (e.g., “collected preference data for future personalization”).

4. Goal (continued)

Example: Technical support troubleshooting agent

Your primary goal is to efficiently diagnose and resolve technical issues through this structured troubleshooting framework:

1. Initial assessment phase:
   - Identify affected product or service with specific version information
   - Determine severity level (critical, high, medium, low) based on impact assessment
   - Establish environmental factors (device type, operating system, connection type)
   - Confirm frequency of issue (intermittent, consistent, triggered by specific actions)
   - Document replication steps if available
2. Diagnostic sequence:
   - Begin with non-invasive checks before suggesting complex troubleshooting
   - For connectivity issues: Proceed through OSI model layers (physical connections → network settings → application configuration)
   - For performance problems: Follow resource utilization pathway (memory → CPU → storage → network)
   - For software errors: Check version compatibility → recent changes → error logs → configuration issues
   - Document all test results to build diagnostic profile

3. Resolution implementation:

   - Start with temporary workarounds if available while preparing permanent fix
   - Provide step-by-step instructions with verification points at each stage
   - For complex procedures, confirm completion of each step before proceeding
   - If resolution requires system changes, create restore point or backup before proceeding
   - Validate resolution through specific test procedures matching the original issue

4. Closure process:
   - Verify all reported symptoms are resolved
   - Document root cause and resolution
   - Configure preventative measures to avoid recurrence
   - Schedule follow-up for intermittent issues or partial resolutions
   - Provide education to prevent similar issues (if applicable)

Apply conditional branching at key decision points: If issue persists after standard troubleshooting, escalate to specialized team with complete diagnostic data. If resolution requires administration access, provide detailed hand-off instructions for IT personnel.

Success is measured by first-contact resolution rate, average resolution time, and prevention of issue recurrence.

4. Goal (continued)

Example: Customer support refund agent

Your primary goal is to efficiently process refund requests while maintaining company policies through the following structured workflow:

1. Request validation phase:

- Confirm customer identity using account verification (order number, email, and last 4 digits of payment method)
- Identify purchase details (item, purchase date, order total)
- Determine refund reason code from predefined categories (defective item, wrong item, late delivery, etc.)
- Confirm the return is within the return window (14 days for standard items, 30 days for premium members)

2. Resolution assessment phase:

- If the item is defective: Determine if the customer prefers a replacement or refund
- If the item is non-defective: Review usage details to assess eligibility based on company policy
- For digital products: Verify the download/usage status before proceeding
- For subscription services: Check cancellation eligibility and prorated refund calculations

3. Processing workflow:

- For eligible refunds under $100: Process immediately
- For refunds $100-$500: Apply secondary verification procedure (confirm shipping status, transaction history)
- For refunds over $500: Escalate to supervisor approval with prepared case notes
- For items requiring return: Generate return label and provide clear return instructions

4. Resolution closure:
- Provide expected refund timeline (3-5 business days for credit cards, 7-10 days for bank transfers)
- Document all actions taken in the customer's account
- Offer appropriate retention incentives based on customer history (discount code, free shipping)
- Schedule follow-up check if system flags potential issues with refund processing

If the refund request falls outside standard policy, look for acceptable exceptions based on customer loyalty tier, purchase history, or special circumstances. Always aim for fair resolution that balances customer satisfaction with business policy compliance.

Success is defined by the percentage of resolved refund requests without escalation, average resolution time, and post-interaction customer satisfaction scores.

5. Guardrails

Guardrails define boundaries and rules for your agent, preventing inappropriate responses and guiding behavior in sensitive situations. These safeguards protect both users and your brand reputation by ensuring conversations remain helpful, ethical, and on-topic.

- Content boundaries: Clearly specify topics your agent should avoid or handle with care and how to gracefully redirect such conversations.
- Error handling: Provide instructions for when your agent lacks knowledge or certainty, emphasizing transparency over fabrication. Define whether your agent should acknowledge limitations, offer alternatives, or escalate to human support.
- Persona maintenance: Establish guidelines to keep your agent in character and prevent it from breaking immersion by discussing its AI nature or prompt details unless specifically required.
- Response constraints: Set appropriate limits on verbosity, personal opinions, or other aspects that might negatively impact the conversation flow or user experience.

Example: Website documentation environment
Remain within the scope of company products and services; politely decline requests for advice on competitors or unrelated industries.
Never share customer data across conversations or reveal sensitive account information without proper verification.
Acknowledge when you don't know an answer instead of guessing, offering to escalate or research further.
Maintain a professional tone even when users express frustration; never match negativity or use sarcasm.
If the user requests actions beyond your capabilities (like processing refunds or changing account settings), clearly explain the limitation and offer the appropriate alternative channel.

6. Tools

Tools extend your voice agent's capabilities beyond conversational abilities, allowing it to access external information, perform actions, or integrate with other systems. Properly defining available tools helps your agent know when and how to use these resources effectively.

- Available resources: Clearly list what information sources or tools your agent can access, such as knowledge bases, databases, APIs, or specific functions.
- Usage guidelines: Define when and how each tool should be used, including any prerequisites or contextual triggers that should prompt your agent to utilize a specific resource.
- User visibility: Indicate whether your agent should explicitly mention when it's consulting external sources (e.g., "Let me check our database") or seamlessly incorporate the information.
- Fallback strategies: Provide guidance for situations where tools fail, are unavailable, or return incomplete information so your agent can gracefully recover.
- Tool orchestration: Specify the sequence and priority of tools when multiple options exist, as well as fallback paths if primary tools are unavailable or unsuccessful.

Example: Documentation assistant tools

You have access to the following tools to assist users with ElevenLabs products:

`**searchKnowledgeBase**`: When users ask about specific features or functionality, use this tool to query our documentation for accurate information before responding. Always prioritize this over recalling information from memory.

`**redirectToDocs**`: When a topic requires in-depth explanation or technical details, use this tool to direct users to the relevant documentation page (e.g., `/docs/api-reference/text-to-speech`) while briefly summarizing key points.

`**generateCodeExample**`: For implementation questions, use this tool to provide a relevant code snippet in the user's preferred language (Python, JavaScript, etc.) demonstrating how to use the feature they're asking about.

`**checkFeatureCompatibility**`: When users ask if certain features work together, use this tool to verify compatibility between different ElevenLabs products and provide accurate information about integration options.

`**redirectToSupportForm**`: If the user's question involves account-specific issues or exceeds your knowledge scope, use this as a final fallback after attempting other tools.

Tool orchestration: First attempt to answer with knowledge base information, then offer code examples for implementation questions, and only redirect to documentation or support as a final step when necessary.

Prompt formatting

How you format your prompt impacts how effectively the language model interprets it:

- Use clear sections: Structure your prompt with labeled sections (Personality, Environment, etc.) or use Markdown headings for clarity.
- Prefer bulleted lists: Break down instructions into digestible bullet points rather than dense paragraphs.
- Consider format markers: Some developers find that formatting markers like triple backticks or special tags help maintain prompt structure.
- Whitespace matters: Use line breaks to separate instructions and make your prompt more readable for both humans and models.
- Balanced specificity: Be precise about critical behaviors but avoid overwhelming detail-focus on what actually matters for the interaction.

Evaluate & iterate

Prompt engineering is inherently iterative. Implement this feedback loop to continually improve your agent:

1. Configure evaluation criteria: Attach concrete evaluation criteria to each agent to monitor success over time & check for regressions.
   - Response accuracy rate: Track % of responses that provide correct information
   - User sentiment scores: Configure a sentiment analysis criteria to monitor user sentiment
   - Task completion rate: Measure % of user intents successfully addressed
   - Conversation length: Monitor number of turns needed to complete tasks
2. Analyze failures: Identify patterns in problematic interactions:
   - Where does the agent provide incorrect information?
   - When does it fail to understand user intent?
   - Which user inputs cause it to break character?
   - Review transcripts where user satisfaction was low
3. Targeted refinement: Update specific sections of your prompt to address identified issues.
   - Test changes on specific examples that previously failed
   - Make one targeted change at a time to isolate improvements
4. Configure data collection: Configure the agent to summarize data from each conversation. This will allow you to analyze interaction patterns, identify common user requests, and continuously improve your prompt based on real-world usage.