Message Format & Conversation Roles

Building Effective Conversations with LLMs

🤖 System Messages

Purpose: Set context, behavior, and instructions for the AI.
Position: Typically first in the conversation.
Content: Instructions, persona, guidelines, constraints.

> *"You are a helpful programming tutor. Explain concepts clearly and provide code examples."*

👤 User Messages

Purpose: User input, questions, and requests.
Position: Throughout the conversation.
Content: Queries, prompts, follow-up questions.

> *"How do I create a Python function that calculates the factorial?"*

🤖 Assistant Messages

Purpose: AI responses and generated content.
Position: Following user messages.
Content: Answers, explanations, generated text.

> *"Here's a Python function to calculate factorial: def factorial(n): ..."*

```
[
  {
    "role": "system",
    "content": "You are a helpful coding assistant specializing in Python
  },
  {
    "role": "user",
    "content": "I need help with loops in Python."
  },
  {
    "role": "assistant",
    "content": "I'd be happy to help! Python has two main types of loops:
  },
  {
    "role": "user",
    "content": "Can you show me a for loop example?"
  }
]
```

📝 System Message Tips

- Be specific about desired behavior and output format.
- Include relevant context and constraints.
- Set the tone and expertise level.
- Keep it concise but comprehensive for clarity.

💬 Conversation Management

- Include relevant conversation history for context.
- Manage context window limits effectively.
- Use assistant messages to maintain conversational flow.
- Implement summarization for long chats.

✨ Expert Tips

- System messages are more influential than user instructions.
- Include examples in system messages for better results.
- Test different system message variations for optimal results.