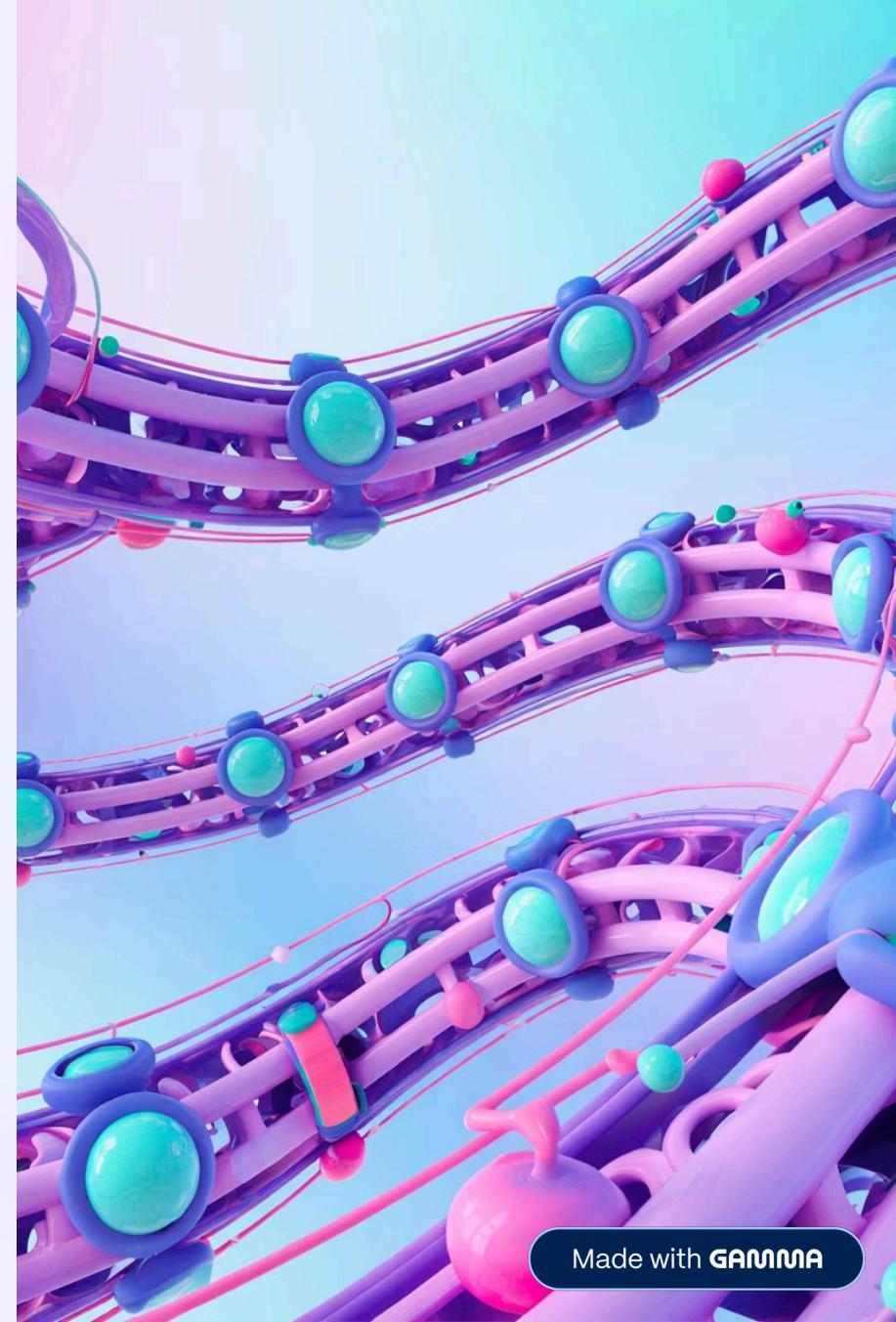


Building Intelligent Workflows

Chains in LangChain

Understand how to link multiple LLM steps into a single, intelligent workflow. Master simple, sequential, and parallel chains for multi-step reasoning and prompt transformation.

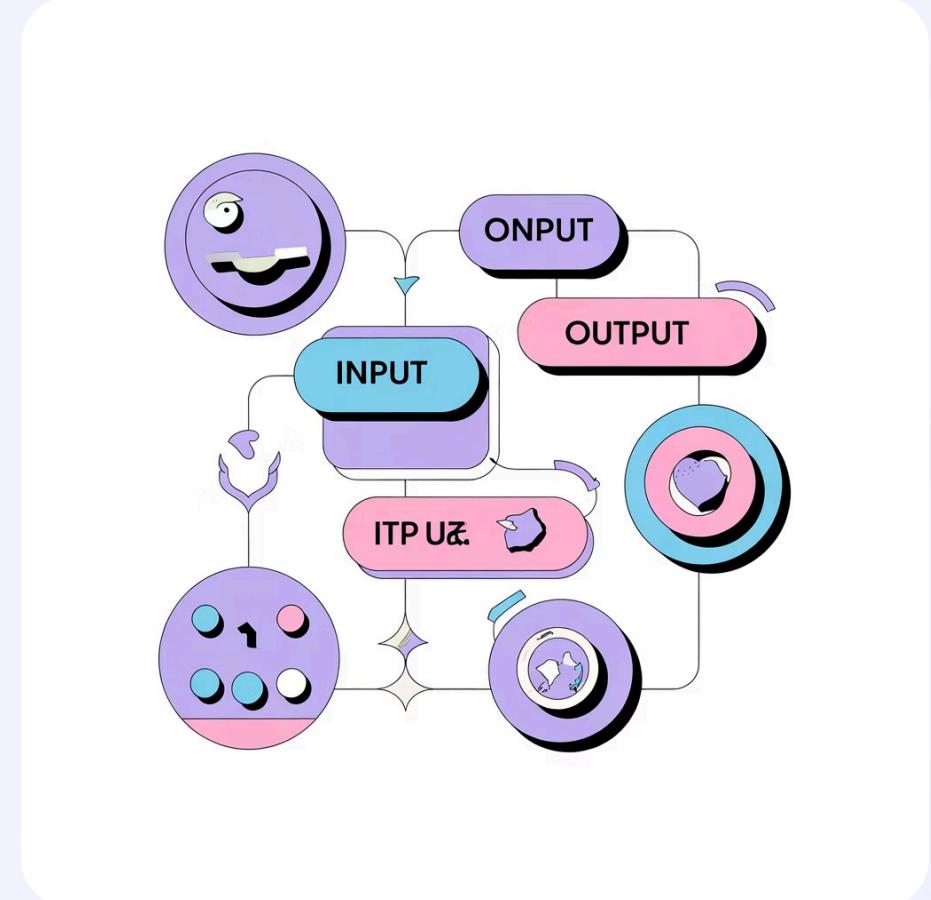


What are Chains in LangChain?

A **Chain** in LangChain is a **pipeline that connects multiple components** – such as LLMs, prompts, tools, or memory – enabling data to flow step-by-step between them in an automated fashion.

- ❑  Think of a Chain as a "workflow": Input → Processing → Output → (used as next input)

Why Needed: LLMs often require multiple reasoning steps – not just one prompt-response. Chains automate that entire process, making complex workflows manageable and repeatable.



Why Use Chains?



One LLM call too simple

Combine multiple logic steps together to handle complex tasks that require layered reasoning.



Reuse of outputs

Pass results automatically to the next step without manual intervention or copying.



Modularity

Each step is independent and reusable, making your workflows easier to maintain and test.



Real-world workflows

Enable multi-stage reasoning and decision-making that mirrors actual business processes.



Example Use Case: Step 1: Generate a research summary from an article. Step 2: Create an interactive quiz from that summary. Step 3: Format the quiz for different platforms.

Components of a Chain

A typical chain can include several interconnected components working together:

01

Prompt Template

Defines how user input is formatted and structures the context for the LLM to process effectively.

02

LLM

The language model that generates intelligent responses based on the formatted prompt input.

03

Memory (optional)

Stores conversation context and maintains state across multiple interactions in the workflow.

04

Output Parser

Structures and transforms the raw result into a usable format for downstream processes or users.

💡 These pieces are linked together in a logical sequence to create a powerful **Chain** that executes complex workflows automatically.



Type 1 — Simple LLM Chain

Single-Step Workflow

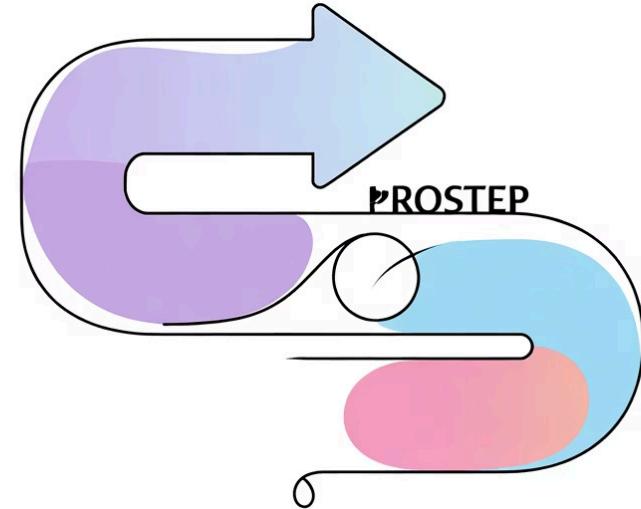
Definition: A single-step chain with one prompt → one LLM → one output. Used for straightforward tasks that don't require complex reasoning.

Structure:

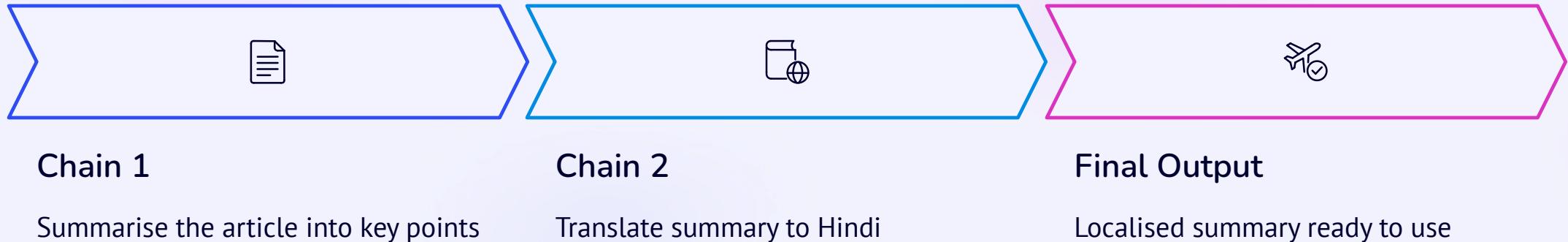
User Input → Prompt Template → LLM → Output

- ❑ **Example:** Generate a compelling product description for "wireless earbuds" including features and benefits.

When to Use: When only one reasoning or transformation step is needed, and the task can be accomplished with a single prompt-response cycle.



Type 2 — Sequential Chain

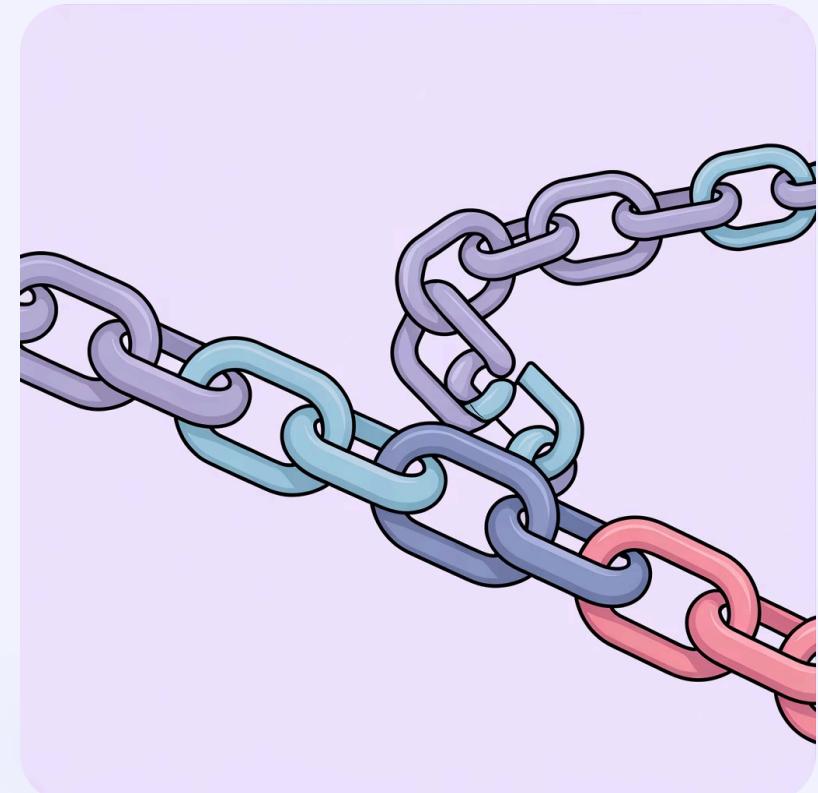


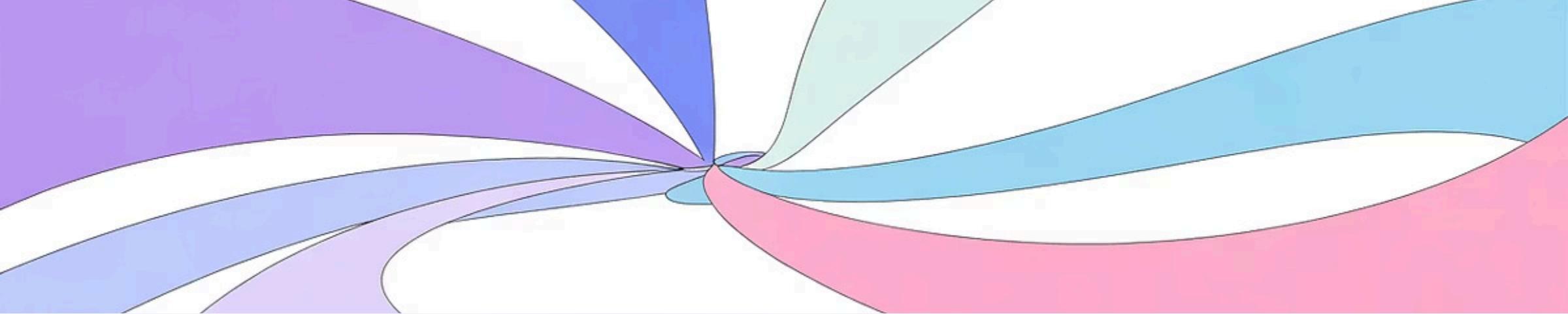
Definition: Output of one chain becomes input for the next. Used when multiple logical steps are needed in a specific order.

How it works: input → step1 → step2 → final output

When to Use:

- Multi-step text workflows
- Reasoning pipelines requiring sequential logic
- Sequential data transformations
- Tasks where order of operations matters





Type 3 — Parallel Chain

Definition: Runs multiple chains **simultaneously** on the same input and merges their outputs efficiently.



When to Use: When tasks are independent but related – this approach saves time and improves overall efficiency. Perfect for content generation workflows that need multiple variations or formats simultaneously.

Practical Use Case — Prompt Transformation

Goal: Automate a complete text generation pipeline with intelligent reuse of outputs.



Step 1: Generate Summary

Extract key information and create a concise summary of the source material.



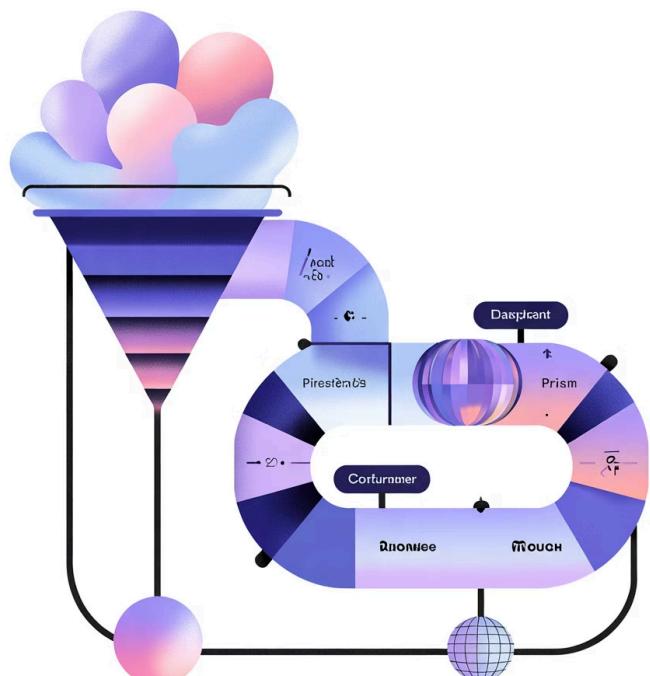
Step 2: Rewrite for Children

Transform the summary into simple language suitable for young readers with age-appropriate vocabulary.



Step 3: Create Quiz

Generate an interactive quiz based on the rewritten text to test comprehension.



💡 Why this matters: Chains let you reuse outputs intelligently – just like a workflow engine. Each step builds upon the previous one, creating sophisticated automation without manual intervention.

Summary — Chains in LangChain

Simple Chain

One LLM step

Single prompt-task workflow

Example: Generate email response

Sequential Chain

Step-by-step

Output becomes next input

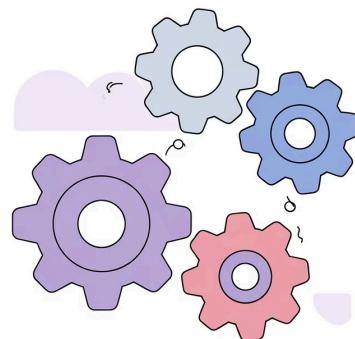
Example: Summarise → Translate

Parallel Chain

Multi-task

Independent outputs simultaneously

Example: Caption + Hashtag + Ad text



✓ **Chains = automation layer** that makes LLMs useful beyond single prompts. They enable you to build sophisticated, production-ready applications with multi-step reasoning, parallel processing, and intelligent output reuse.