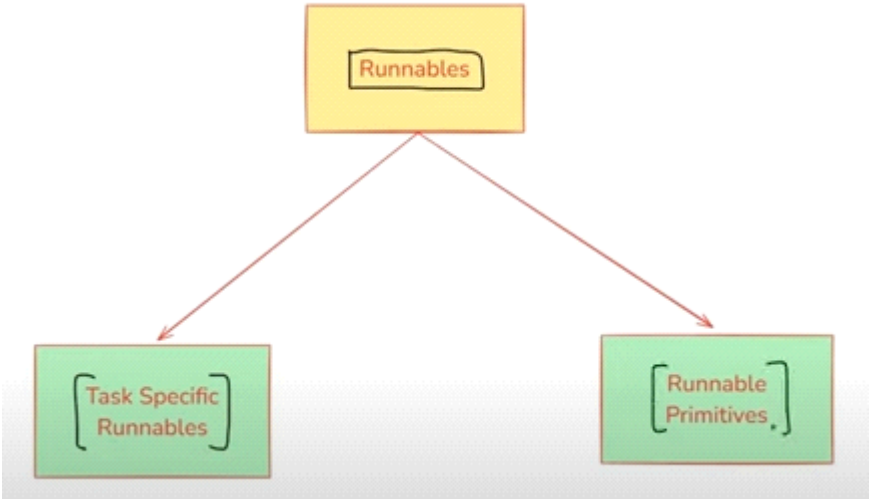
Langchain Runnables

23 May 2025 23:07

Chain Name	Description
LLMChain	Basic chain that calls an LLM with a prompt template.
SequentialChain	Chains multiple LLM calls in a specific sequence.
SimpleSequentialChain	A simplified version of SequentialChain for easier use.
ConversationalRetrievalChain	Handles conversational Q&A with memory and retrieval.
RetrievalQA	Fetches relevant documents and uses an LLM for question-answering.
RouterChain	Directs user queries to different chains based on intent.
MultiPromptChain	Uses different prompts for different user intents dynamically.
HydeChain (Hypothetical Document Embeddings)	Generates hypothetical answers to improve document retrieval.
AgentExecutorChain	Orchestrates different tools and actions dynamically using an agent.
SQLDatabaseChain	Connects to SQL databases and answers natural language queries.

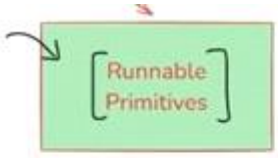


✅ **Definition**: These are **core LangChain components** that have been converted into Runnables so they can be used in pipelines.

✅ **Purpose**: Perform **task-specific** operations like LLM calls, prompting, retrieval, etc.

✅ **Examples**:

- `ChatOpenAI` → Runs an LLM model.
- `PromptTemplate` → Formats prompts dynamically.
- `Retriever` → Retrieves relevant documents.



- [x] **Definition**: These are **fundamental building blocks** for structuring execution logic in AI workflows.
- [x] **Purpose**: They **help orchestrate execution** by defining how different Runnables interact (sequentially, in parallel, conditionally, etc.).
- [x] **Examples**:
  - `RunnableSequence` → Runs steps **in order** (`|` operator).
  - `RunnableParallel` → Runs multiple steps **simultaneously**.
  - `RunnableMap` → Maps the same input across multiple functions.
  - `RunnableBranch` → Implements **conditional execution** (if-else logic).
  - `RunnableLambda` → Wraps **custom Python functions** into Runnables.
  - `RunnablePassthrough` → Just forwards input as output (acts as a placeholder).

1. RunnableSequence

20 March 2025 12:10

RunnableSequence is a sequential chain of runnables in LangChain that executes each step one after another, passing the output of one step as the input to the next.

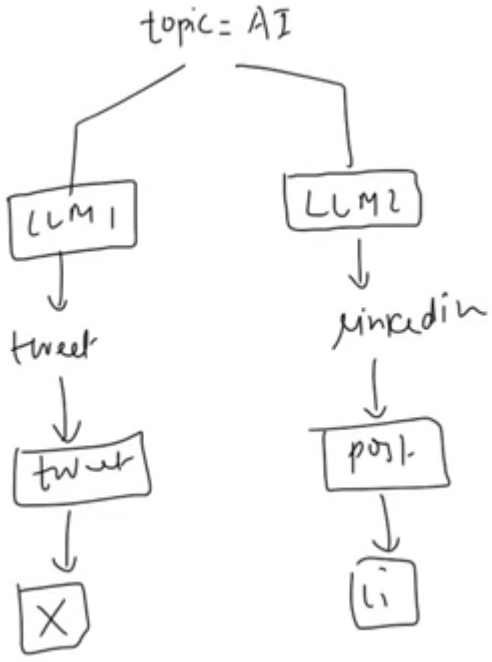
It is useful when you need to compose multiple runnables together in a structured workflow.

2. RunnableParallel

20 March 2025 18:33

RunnableParallel is a runnable primitive that allows multiple runnables to execute in parallel.

Each runnable receives the same input and processes it independently, producing a dictionary of outputs.



3. RunnablePassthrough

20 March 2025 22:34

RunnablePassthrough is a special Runnable primitive that simply **returns the input as output** without modifying it.

4. RunnableLambda

20 March 2025 23:18

RunnableLambda is a runnable primitive that allows you to **apply custom Python functions** within an AI pipeline.

It acts as a middleware between different AI components, **enabling preprocessing, transformation, API calls, filtering, and post-processing** in a LangChain workflow.

5. RunnableBranch → cond:

21 March 2025 08:02

RunnableBranch is a control flow component in LangChain that allows you to conditionally route input data to different chains or runnables based on custom logic.

It functions like an if/elif/else block for chains — where you define a set of condition functions, each associated with a runnable (e.g., LLM call, prompt chain, or tool). The first matching condition is executed. If no condition matches, a default runnable is used (if provided).