Welcome to LangChain

Contents

LangChain is a framework for developing applications powered by language models. We believe that the most powerful and differentiated applications will not only call out to a language model, but will also be:

1. *Data-aware*: connect a language model to other sources of data
2. *Agentic*: allow a language model to interact with its environment

The LangChain framework is designed around these principles.

This is the Python specific portion of the documentation. For a purely conceptual guide to LangChain, see here. For the JavaScript documentation, see here.

Getting Started

How to get started using LangChain to create an Language Model application.

- Quickstart Guide

Concepts and terminology.

- Concepts and terminology

Tutorials created by community experts and presented on YouTube.

🦜🔗

⌘ + K

Skip to main content

Modules

These modules are the core abstractions which we view as the building blocks of any LLM-powered application.

For each module LangChain provides standard, extendable interfaces. LangChain also provides external integrations and even end-to-end implementations for off-the-shelf use.

The docs for each module contain quickstart examples, how-to guides, reference docs, and conceptual guides.

The modules are (from least to most complex):

- Models: Supported model types and integrations.
- Prompts: Prompt management, optimization, and serialization.
- Memory: Memory refers to state that is persisted between calls of a chain/agent.
- Indexes: Language models become much more powerful when combined with application-specific data - this module contains interfaces and integrations for loading, querying and updating external data.
- Chains: Chains are structured sequences of calls (to an LLM or to a different utility).
- Agents: An agent is a Chain in which an LLM, given a high-level directive and a set of tools, repeatedly decides an action, executes the action and observes the outcome until the high-level directive is complete.
- Callbacks: Callbacks let you log and stream the intermediate steps of any chain, making it easy to observe, debug, and evaluate the internals of an application.

Use Cases

Best practices and built-in implementations for common LangChain use cases:

- Autonomous Agents: Autonomous agents are long-running agents that take many steps in an attempt to accomplish an objective. Examples include AutoGPT and BabyAGI.
- Agent Simulations: Putting agents in a sandbox and observing how they interact with each other and react to events can be an effective way to evaluate their lon 🦜🔗 reasoning and planning abilities.
- Personal Assistants: One of the primary LangChain use cases. Personal ass

⌘ + K

Skip to main content

- Question Answering: Another common LangChain use case. Answering questions over specific documents, only utilizing the information in those documents to construct an answer.
- Chatbots: Language models love to chat, making this a very natural use of them.
- Querying Tabular Data: Recommended reading if you want to use language models to query structured data (CSVs, SQL, dataframes, etc).
- Code Understanding: Recommended reading if you want to use language models to analyze code.
- Interacting with APIs: Enabling language models to interact with APIs is extremely powerful. It gives them access to up-to-date information and allows them to take actions.
- Extraction: Extract structured information from text.
- Summarization: Compressing longer documents. A type of Data-Augmented Generation.
- Evaluation: Generative models are hard to evaluate with traditional metrics. One promising approach is to use language models themselves to do the evaluation.

Reference Docs

Full documentation on all methods, classes, installation methods, and integration setups for LangChain.

- LangChain Installation
- Reference Documentation

Ecosystem

LangChain integrates a lot of different LLMs, systems, and products.
From the other side, many systems and products depend on LangChain.
It creates a vibrant and thriving ecosystem.

- Integrations: Guides for how other products can be used with LangChain.
- Dependents: List of repositories that use LangChain.
- Deployments: A collection of instructions, code snippets, and template rep
for deploying LangChain apps.

🦜🔗

⌘ + K

Skip to main content

Additional Resources

Additional resources we think may be useful as you develop your application!

- LangChainHub: The LangChainHub is a place to share and explore other prompts, chains, and agents.
- Gallery: A collection of great projects that use Langchain, compiled by the folks at Kyrolabs. Useful for finding inspiration and example implementations.
- Tracing: A guide on using tracing in LangChain to visualize the execution of chains and agents.
- Model Laboratory: Experimenting with different prompts, models, and chains is a big part of developing the best possible application. The ModelLaboratory makes it easy to do so.
- Discord: Join us on our Discord to discuss all things LangChain!
- YouTube: A collection of the LangChain tutorials and videos.
- Production Support: As you move your LangChains into production, we'd love to offer more comprehensive support. Please fill out this form and we'll set up a dedicated support Slack channel.

🦜🔗

⌘ + K