Introduction:

LangChain is a framework for developing applications powered by large language models (LLMs).

LangChain simplifies every stage of the LLM application lifecycle:

- DevelopmentBuild your applications using LangChain's open-source [components](/docs/concepts) and [third-party integrations](/docs/integrations/providers/).

Use [LangGraph](/docs/concepts/architecture/#langgraph) to build stateful agents with first-class streaming and human-in-the-loop support.

- ProductionizationUse [LangSmith](https://docs.smith.langchain.com/) to inspect, monitor and evaluate your applications, so that you can continuously optimize and deploy with confidence.

- Deployment: Turn your LangGraph applications into production-ready APIs and Assistants with [LangGraph Platform](https://langchain-ai.github.io/langgraph/cloud/).

LangChain implements a standard interface for large language models and related technologies, such as embedding models and vector stores, and integrates with hundreds of providers. See the integrations page for more.

```
pip install -qU "langchain[openai]"
```

```
import getpass
import os

if not os.environ.get("OPENAI_API_KEY"):
  os.environ["OPENAI_API_KEY"] = getpass.getpass("Enter API key for OpenAI: ")

from langchain.chat_models import init_chat_model

model = init_chat_model("gpt-4o-mini", model_provider="openai")

model.invoke("Hello, world!")
```

Architecture:

The LangChain framework consists of multiple open-source libraries. Read more in the

[Architecture](/docs/concepts/architecture/) page.

- langchain-core: Base abstractions for chat models and other components.

- Integration packages (e.g. langchain-openai, langchain-anthropic, etc.): Important integrations have been split into lightweight packages that are co-maintained by the LangChain team and the integration developers.

- langchain: Chains, agents, and retrieval strategies that make up an application's cognitive architecture.

- langchain-community: Third-party integrations that are community maintained.

- langgraph: Orchestration framework for combining LangChain components into production-ready applications with persistence, streaming, and other key features. See [LangGraph documentation](https://langchain-ai.github.io/langgraph/).

Guides

[Tutorials](/docs/tutorials)

If you're looking to build something specific or are more of a hands-on learner, check out our [tutorials section](/docs/tutorials).

This is the best place to get started.

These are the best ones to get started with:

- [Build a Simple LLM Application](/docs/tutorials/llm_chain)

- [Build a Chatbot](/docs/tutorials/chatbot)

- [Build an Agent](/docs/tutorials/agents)

- [Introduction to LangGraph](https://langchain-ai.github.io/langgraph/tutorials/introduction/)

Explore the full list of LangChain tutorials [here](/docs/tutorials), and check out other [LangGraph tutorials here](https://langchain-ai.github.io/langgraph/tutorials/). To learn more about LangGraph, check out our first LangChain Academy course, *Introduction to LangGraph*, available [here](https://academy.langchain.com/courses/intro-to-langgraph).

[How-to guides](/docs/how_to)

[Here](/docs/how_to) you'll find short answers to "How do I....?" types of questions.

These how-to guides don’t cover topics in depth – you’ll find that material in the [Tutorials](/docs/tutorials) and the [API Reference](https://python.langchain.com/api_reference/).

However, these guides will help you quickly accomplish common tasks using [chat models](/docs/how_to/#chat-models),

[vector stores](/docs/how_to/#vector-stores), and other common LangChain components.

Check out [LangGraph-specific how-tos here](https://langchain-ai.github.io/langgraph/how-tos/).

[Conceptual guide](/docs/concepts)

Introductions to all the key parts of LangChain you’ll need to know! [Here](/docs/concepts) you'll find high level explanations of all LangChain concepts.

For a deeper dive into LangGraph concepts, check out [this page](https://langchain-ai.github.io/langgraph/concepts/).

[Integrations](integrations/providers/index.mdx)

LangChain is part of a rich ecosystem of tools that integrate with our framework and build on top of it.

If you're looking to get up and running quickly with [chat models](/docs/integrations/chat/), [vector stores](/docs/integrations/vectorstores/),

or other LangChain components from a specific provider, check out our growing list of [integrations](/docs/integrations/providers/).

[API reference](https://python.langchain.com/api_reference/)

Head to the reference section for full documentation of all classes and methods in the LangChain Python packages.

Ecosystem

LangSmith](https://docs.smith.langchain.com)

Trace and evaluate your language model applications and intelligent agents to help you move from prototype to production.

LangGraph](https://langchain-ai.github.io/langgraph)

Build stateful, multi-actor applications with LLMs. Integrates smoothly with LangChain, but can be used without it. LangGraph powers production-grade agents, trusted by Linkedin, Uber, Klarna, GitLab, and many more.

Additional resources

[Versions](/docs/versions/v0_3/)

See what changed in v0.3, learn how to migrate legacy code, read up on our versioning policies, and more.

[Security](/docs/security)

Read up on [security](/docs/security) best practices to make sure you're developing safely with LangChain.

[Contributing](contributing/index.mdx)

Check out the developer's guide for guidelines on contributing and help getting your dev environment set up.