



# Introduction

**LangChain** is a framework for developing applications powered by language models. It enables applications that are:

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

The main value props of LangChain are:

1. **Components:** abstractions for working with language models, along with a collection of implementations for each abstraction. Components are modular and easy-to-use, whether you are using the rest of the LangChain framework or not
2. **Off-the-shelf chains:** a structured assembly of components for accomplishing specific higher-level tasks

Off-the-shelf chains make it easy to get started. For more complex applications and nuanced use-cases, components make it easy to customize existing chains or build new ones.

## Get started

---

Here's how to install LangChain, set up your environment, and start building.

We recommend following our [Quickstart](#) guide to familiarize yourself with the framework by building your first LangChain application.

**Note:** These docs are for the LangChain *Python package*. For documentation on *LangChain.js*, the JS/TS version, [head here](#).

## Modules

---

LangChain provides standard, extendable interfaces and external integrations for the following modules, listed from least to most complex:

### Model I/O

Interface with language models

## Data connection

Interface with application-specific data

## Chains

Construct sequences of calls

## Agents

Let chains choose which tools to use given high-level directives

## Memory

Persist application state between runs of a chain

## Callbacks

Log and stream intermediate steps of any chain

# Examples, ecosystem, and resources

---

## Use cases

Walkthroughs and best-practices for common end-to-end use cases, like:

- [Chatbots](#)
- [Answering questions using sources](#)
- [Analyzing structured data](#)
- and much more...

## Guides

Learn best practices for developing with LangChain.

## Ecosystem

LangChain is part of a rich ecosystem of tools that integrate with our framework and build on top of it. Check out our growing list of [integrations](#) and [dependent repos](#).

## Additional resources

Our community is full of prolific developers, creative builders, and fantastic teachers. Check out [YouTube tutorials](#) for great tutorials from folks in the community, and [Gallery](#) for a list of awesome LangChain projects, compiled by the folks at [KyroLabs](#).

## Support

Join us on [GitHub](#) or [Discord](#) to ask questions, share feedback, meet other developers building with LangChain, and dream about the future of LLM's.

## API reference

---

Head to the [reference](#) section for full documentation of all classes and methods in the LangChain Python package.