# LangChain: Indexes Component

*Connecting Applications to External Knowledge Sources*

**Overview**

Indexes in LangChain provide a bridge between your application and external knowledge sources—such as PDFs, websites, or databases. They enable fast semantic retrieval and context-aware querying by structuring and embedding the documents into searchable formats.

**Key Benefits**

- **Unified Access:** Connect various data sources seamlessly.
- **Semantic Search:** Retrieve relevant information via embeddings.
- **Pipeline Integration:** Work directly with LLMs for context-aware answers.
- **Scalable:** Support large document collections efficiently.

## 1. Core Components of Indexes

**1. Document Loader (DocLoader)**

- Reads and imports documents from different sources (PDFs, Word files, websites, databases).
- Normalizes and standardizes content for downstream processing.

**2. Text Splitter**

- Breaks documents into smaller chunks for efficient embedding and retrieval.
- Supports overlapping chunks for better context retention.
- Examples: RecursiveCharacterTextSplitter, TokenTextSplitter.

**3. Vector Databases**

- Stores document embeddings in a structured format.
- Supports fast similarity search (nearest-neighbor queries).
- Popular options: FAISS, Chroma, Pinecone, Weaviate.

**4. Retrievers**

- Queries the vector database to fetch relevant documents.
- Can rank results based on similarity scores or filters.
- Used as input to LLMs for retrieval-augmented generation (RAG).

## 2. Indexes Workflow Diagram

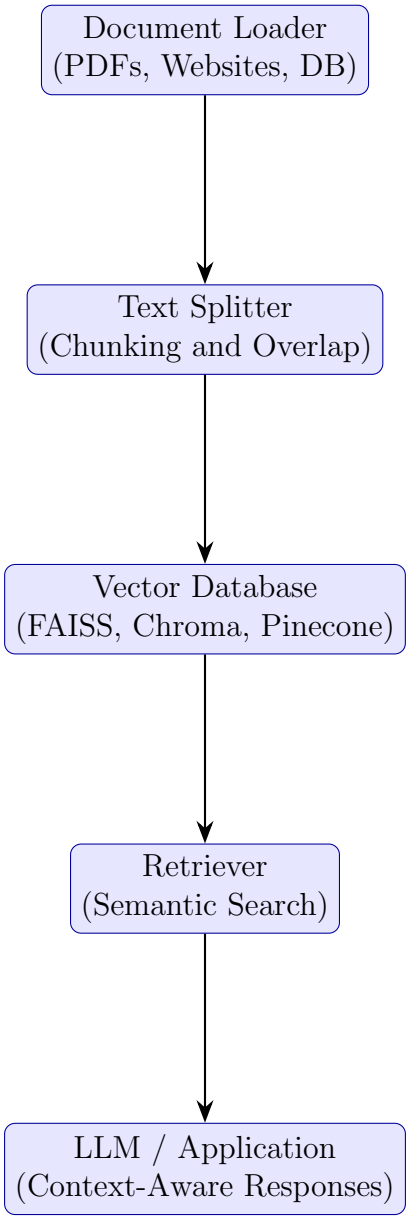


Figure 1: LangChain Indexes Pipeline (Vertical Layout): From Documents to Contextual Responses

## 3. Summary

LangChain's **Indexes** component allows applications to efficiently access and query external knowledge sources. By combining **Doc Loaders, Text Splitters, Vector Databases,** and **Retrievers**, developers can implement **retrieval-augmented generation (RAG) pipelines** for context-aware, intelligent outputs.