

LangChain Cheatsheet

Installation

```
pip install langchain
Install LangChain library

pip install langchain-openai
Install LangChain OpenAI integration
```

Basic LangChain Setup

```
from langchain.llms import OpenAI
Import OpenAI LLM

llm = OpenAI(openai_api_key='your-api-key')
Initialize OpenAI LLM

llm('your prompt here')
Generate simple completion
```

Chains

```
from langchain.chains import LLMChain
Import basic LLMChain

from langchain.prompts import PromptTemplate
Import PromptTemplate

chain = LLMChain(llm=llm, prompt=prompt)
Create a chain

chain.run(input_variables)
Execute a chain
```

Agents

```
from langchain.agents import initialize_agent, Tool
Import agent components

agent = initialize_agent(tools, llm, agent='zero-shot-react-description')
Create an agent

agent.run('your task here')
Run an agent
```

Memory

```
from langchain.memory import ConversationBufferMemory
Import conversation memory

memory = ConversationBufferMemory()
Initialize memory

chain = LLMChain(llm=llm, prompt=prompt, memory=memory)
Add memory to chain
```

Document Loading

```
from langchain.document_loaders import TextLoader
Import document loader

loader = TextLoader('path/to/file.txt')
Create a loader

documents = loader.load()
Load documents
```

Embeddings

```
from langchain.embeddings import OpenAIEmbeddings
Import embeddings

embeddings = OpenAIEmbeddings()
Initialize embeddings

vector = embeddings.embed_query('your text')
Create embedding for text
```

Vector Stores

```
from langchain.vectorstores import Chroma
Import vector store

db = Chroma.from_documents(documents, embeddings)
Create vector store from documents

docs = db.similarity_search('query')
Search for similar documents
```

Debugging

```
import langchain
Import LangChain

langchain.debug = True
Enable debug mode

langchain.verbose = True
Enable verbose mode
```