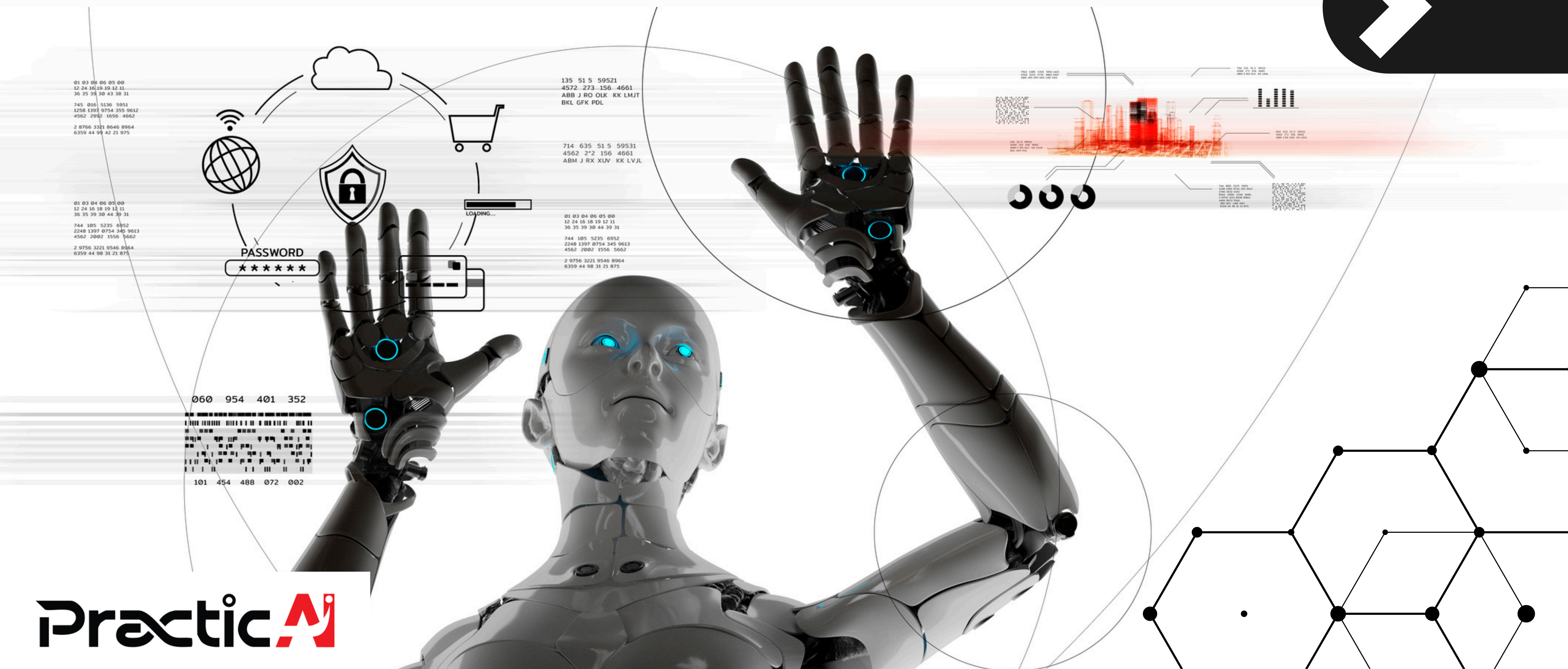
PractiAI

Run LLMs Locally with Ollama

Run powerful LLMs locally and integrate with frameworks like **LangChain & LlamaIndex** easily!



PractiAI

What is Ollama?

Your Local AI Powerhouse Ollama is a lightweight, extensible framework that lets you run large language models locally on your machine. No cloud dependencies, no API costs, completeprivacy!

Key Features

- Run models like Mistral, Llama, CodeLlama locally
- Easy installation and setup
- Multiple framework integrations
- Complete data privacy
- Offline capabilities

Perfect for

- Building RAG applications
- Creating AI agents
- Prototyping without API costs
- Privacy-sensitive projects

Installation Process

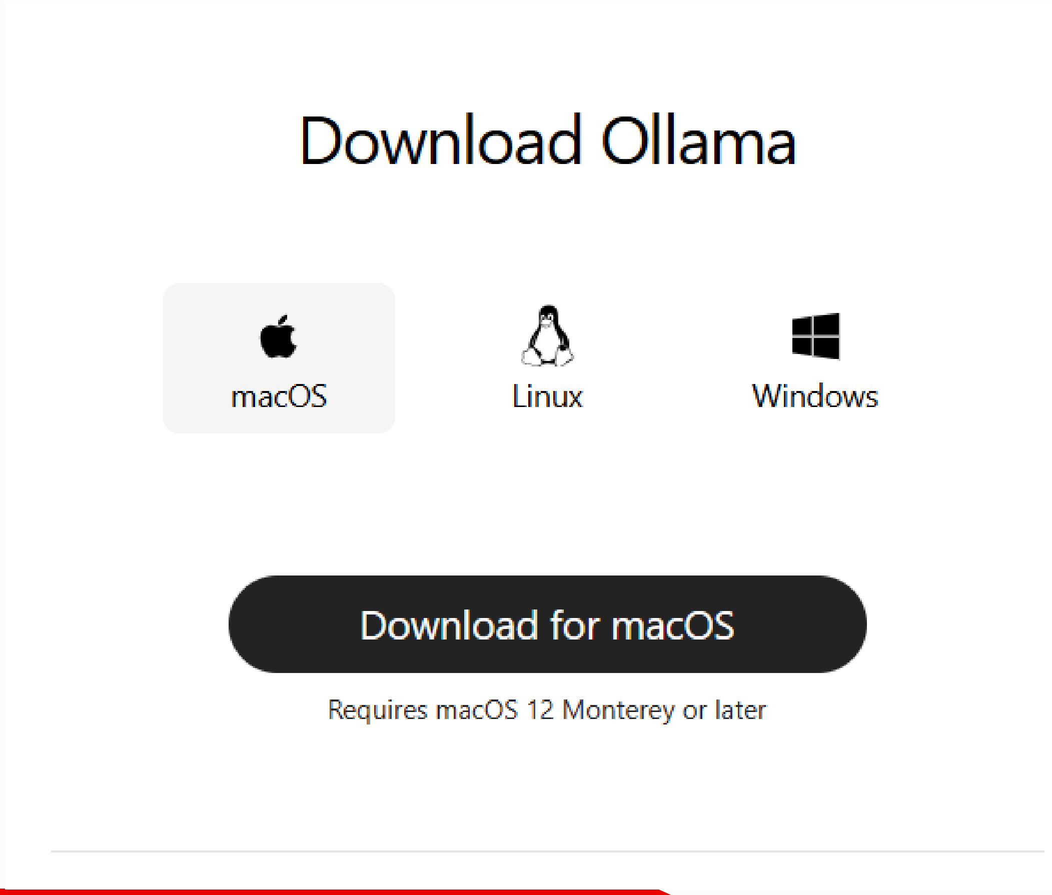
Quick Installation Steps

Linux:

curl -fsSL https://ollama.com/install.sh | sh

Download installer from ollama.com

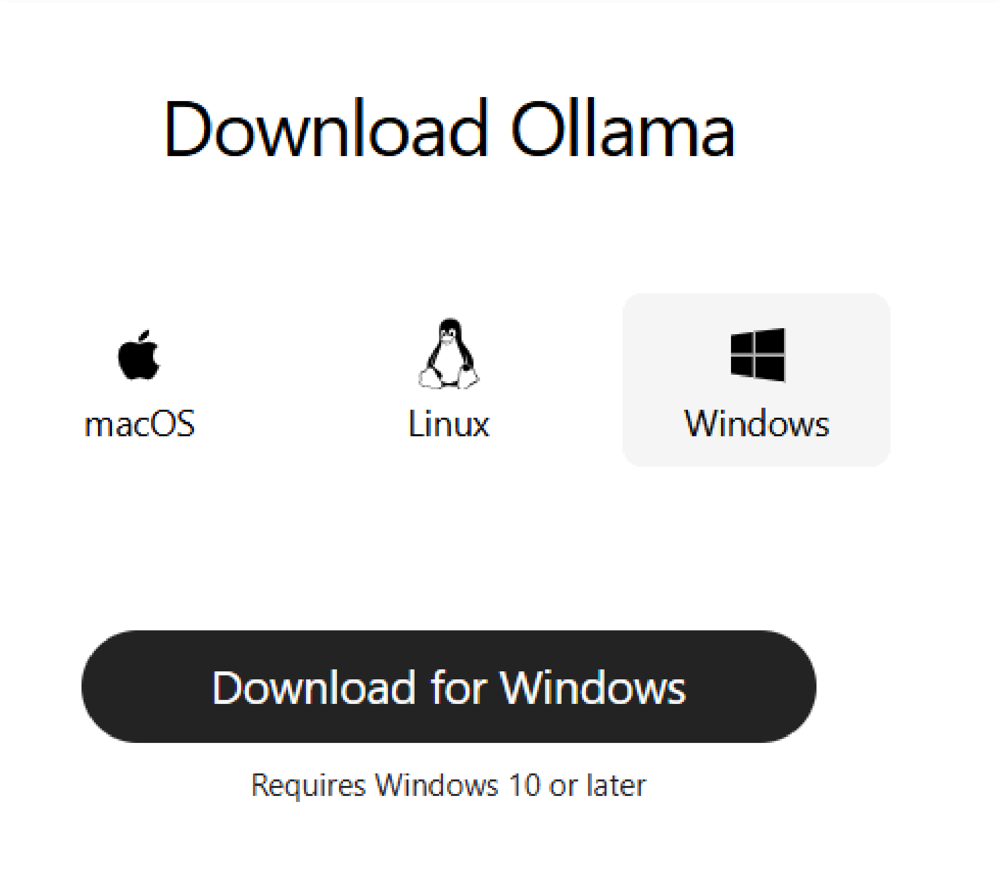
macOS



Download Ollama

Requires macOS 12 Monterey or later

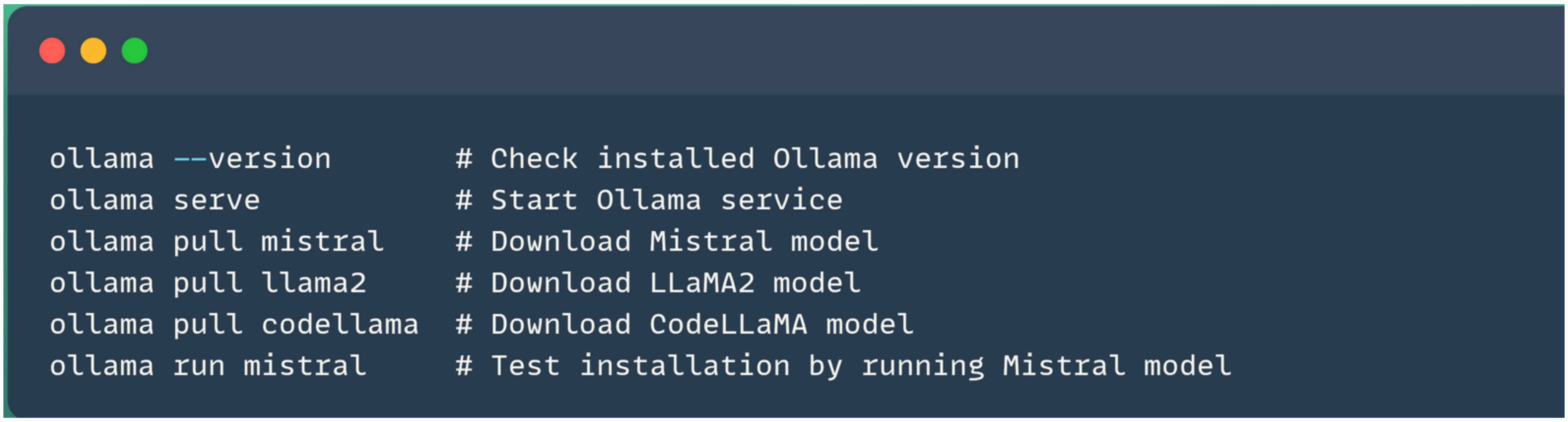
Windows



Download Ollama

Requires Windows 10 or later

Commands



```
ollama --version       # Check installed Ollama version
ollama serve           # Start Ollama service
ollama pull mistral    # Download Mistral model
ollama pull llama2     # Download LLaMA2 model
ollama pull codellama  # Download CodeLLaMA model
ollama run mistral     # Test installation by running Mistral model
```

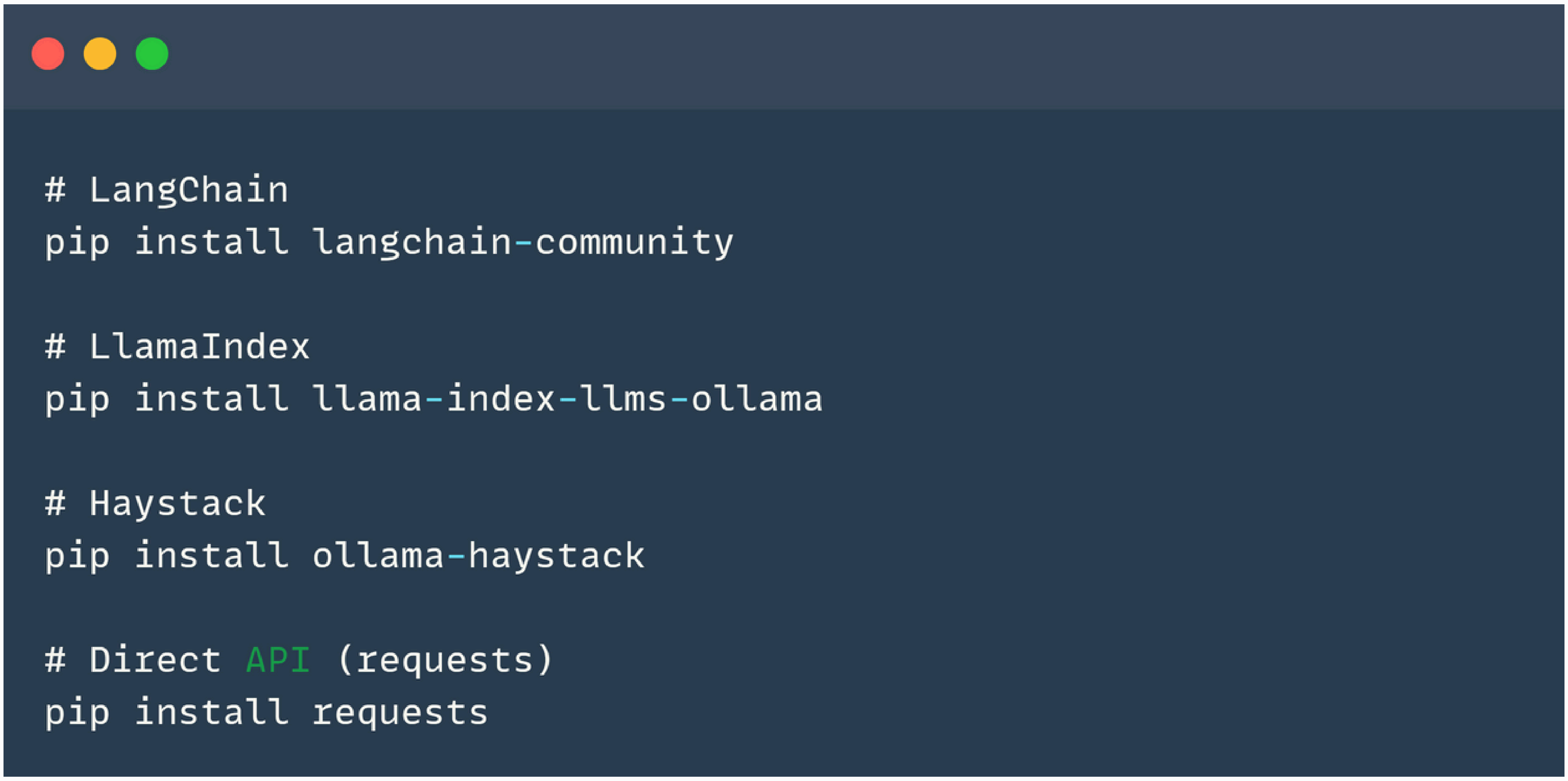
Framework Integration

Multiple Integration Options

Supported Frameworks:

- LangChain - Full-featured LLM framework
- LlamaIndex - Document indexing and retrieval
- Haystack - Enterprise-grade NLP pipelines
- Direct API - Pure HTTP requests

Installation Commands:

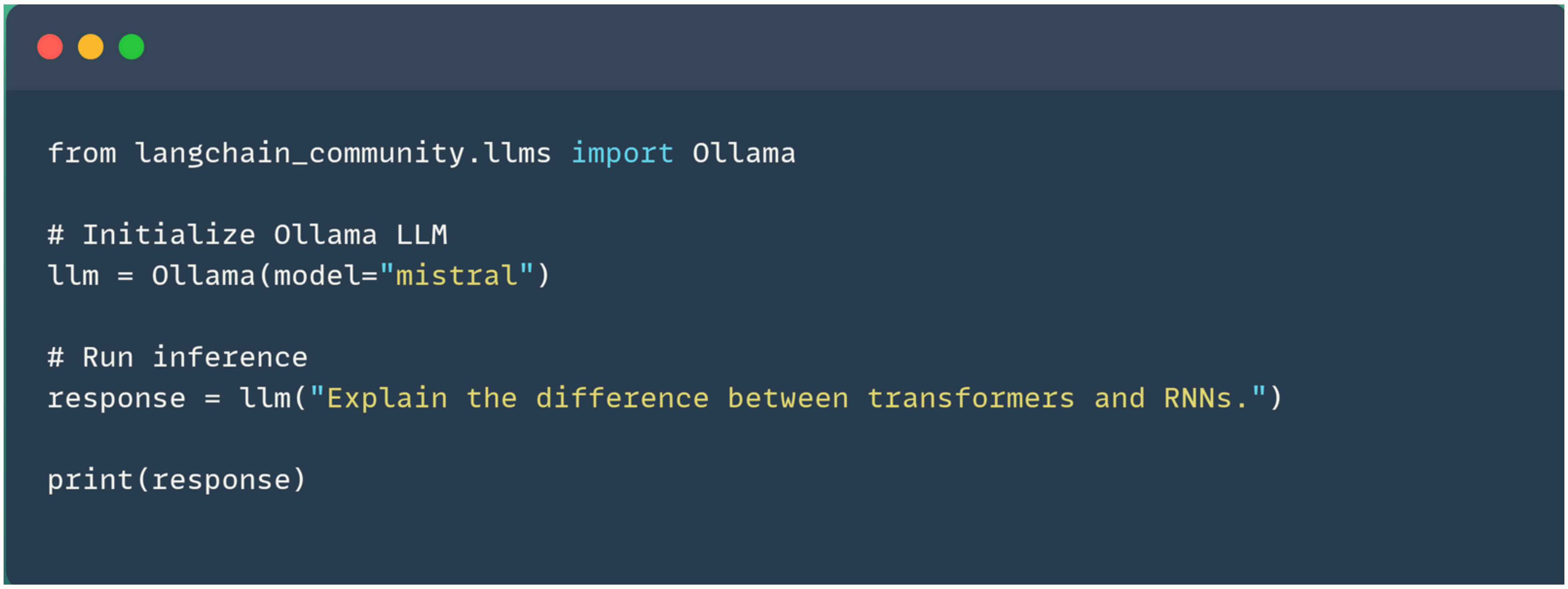


```
# LangChain
pip install langchain-community

# LlamaIndex
pip install llama-index-llms-ollama

# Haystack
pip install ollama-haystack

# Direct API (requests)
pip install requests
```

LangChain Integration

🦜 LangChain + Ollama

Code Example:



```
from langchain_community.llms import Ollama

# Initialize Ollama LLM
llm = Ollama(model="mistral")

# Run inference
response = llm("Explain the difference between transformers and RNNs.")

print(response)
```

Benefits

- Seamless integration with LangChain ecosystem
- Access to chains, agents, and tools
- Perfect for RAG applications
- Memory management built-in

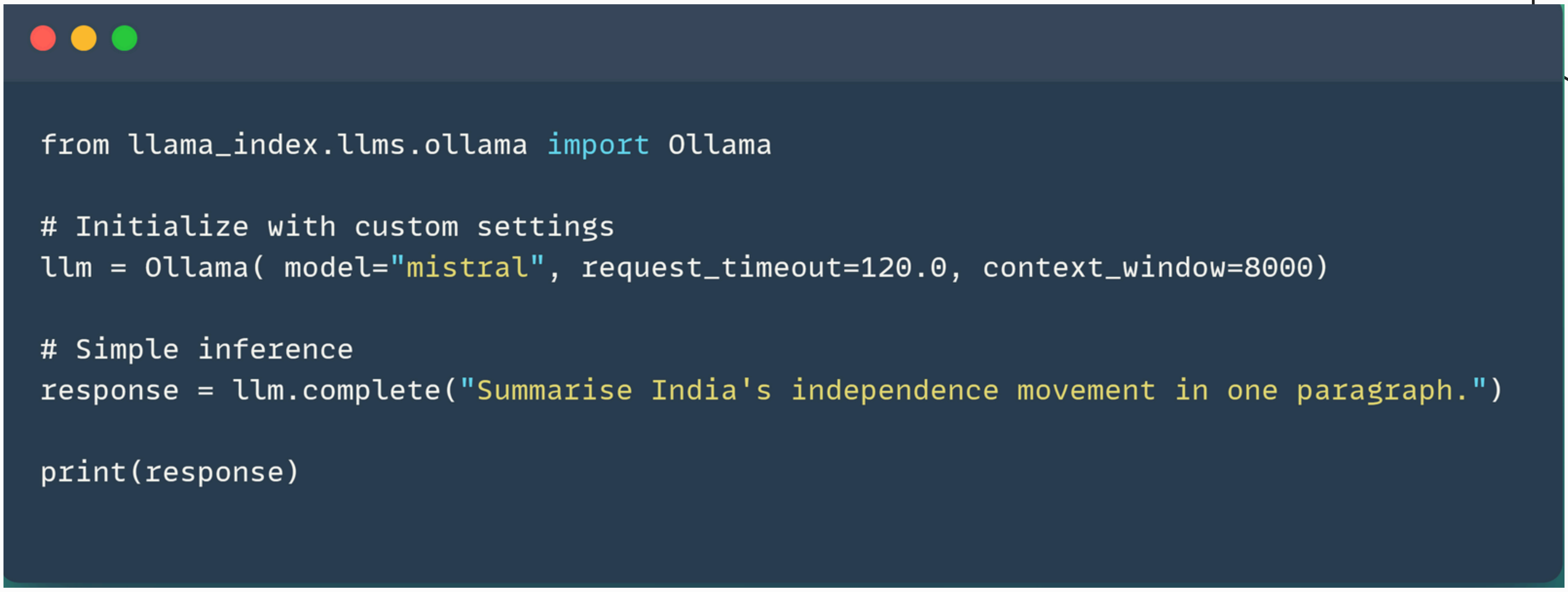
Use Cases

- Chatbots with memory
- Document Q&A systems
- Multi-step reasoning tasks

LlamaIndex Integration

🦙 LlamaIndex + Ollama

Code Example:



```
from llama_index.llms.ollama import Ollama

# Initialize with custom settings
llm = Ollama( model="mistral", request_timeout=120.0, context_window=8000)

# Simple inference
response = llm.complete("Summarise India's independence movement in one paragraph.")

print(response)
```

Benefits

- Optimized for document indexing
- Built-in RAG capabilities
- Efficient context management
- Easy data ingestion

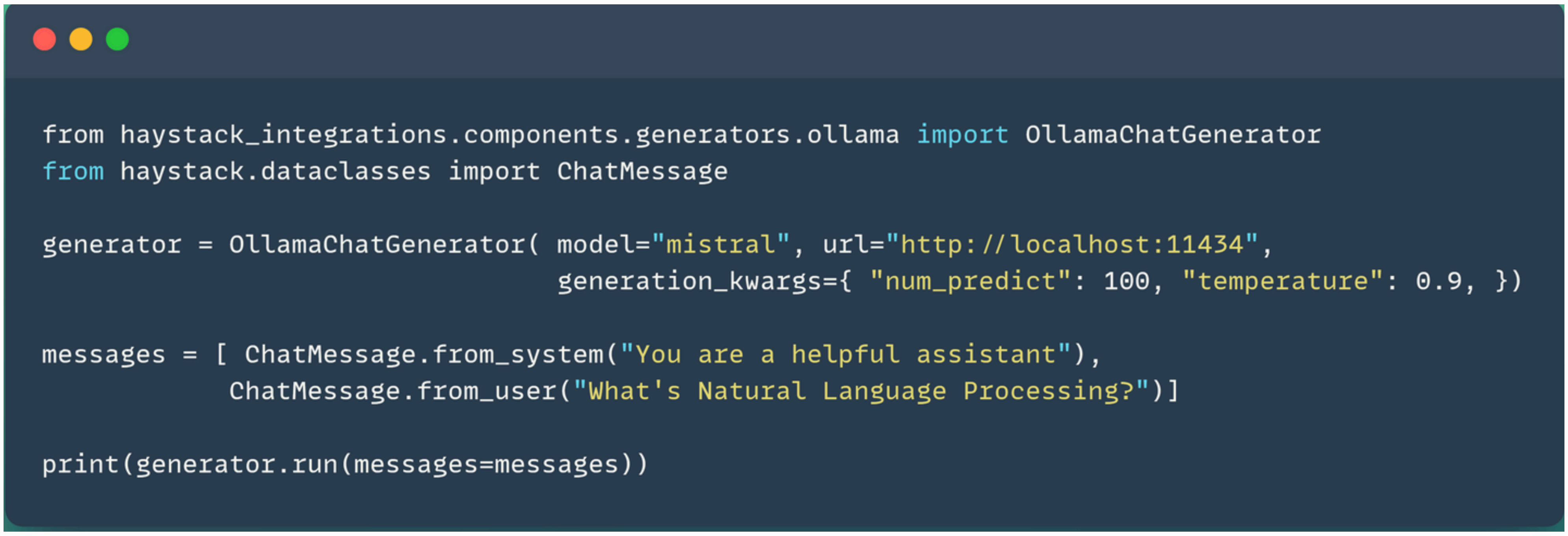
Use Cases

- Knowledge base systems
- Document search and retrieval
- Research assistants

Haystack Integration

🌾 Haystack + Ollama

Code Example:



```
from haystack_integrations.components.generators.ollama import OllamaChatGenerator
from haystack.dataclasses import ChatMessage

generator = OllamaChatGenerator( model="mistral", url="http://localhost:11434",
                                 generation_kwargs={ "num_predict": 100, "temperature": 0.9, })

messages = [ ChatMessage.from_system("You are a helpful assistant"),
             ChatMessage.from_user("What's Natural Language Processing?")]

print(generator.run(messages=messages))
```

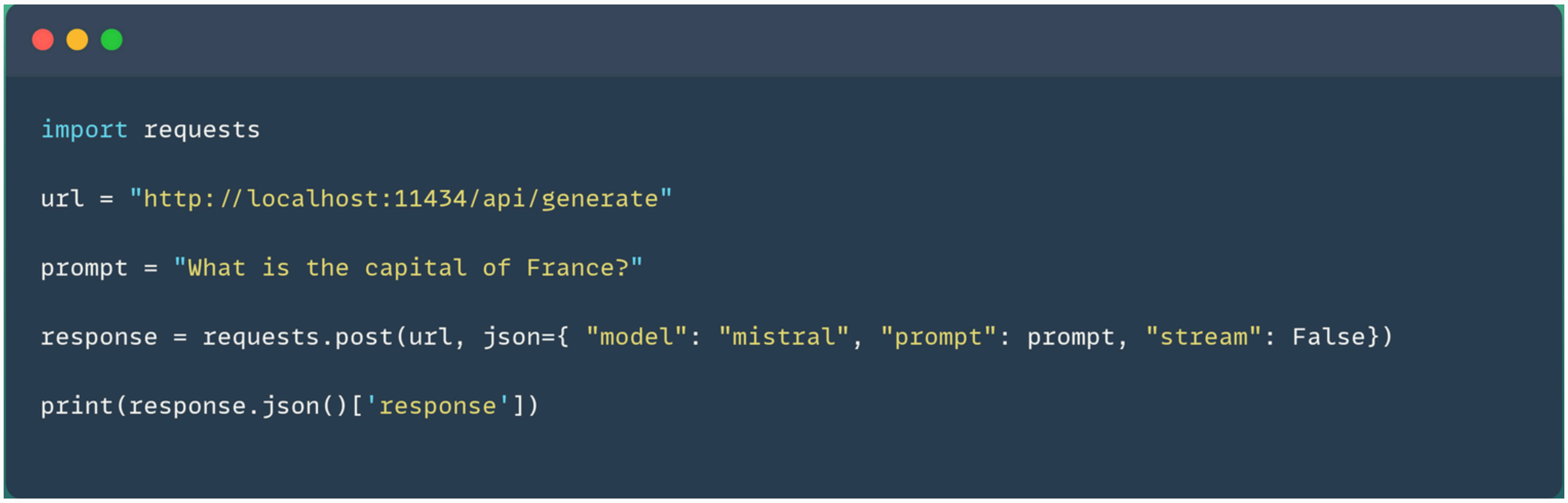
Benefits

- Enterprise-grade pipelines
- Advanced preprocessing
- Scalable architecture
- Production-ready

Direct API Integration

🔌 Direct API Access

Code Example:



```
import requests

url = "http://localhost:11434/api/generate"

prompt = "What is the capital of France?"

response = requests.post(url, json={ "model": "mistral", "prompt": prompt, "stream": False})

print(response.json()['response'])
```

Benefits

- Maximum control and flexibility
- Lightweight integration
- Custom request handling
- No additional dependencies

Use Cases

- Custom applications
- Microservices
- Simple integrations

Benefits & Advantages

✅ Why Choose Ollama?

Privacy & Security:

- Data never leaves your machine
- No cloud dependencies
- Complete control over your data
- GDPR/compliance friendly

Cost Efficiency:

- No API fees or usage limits
- One-time setup cost
- Unlimited inference
- Perfect for experimentation

Performance:

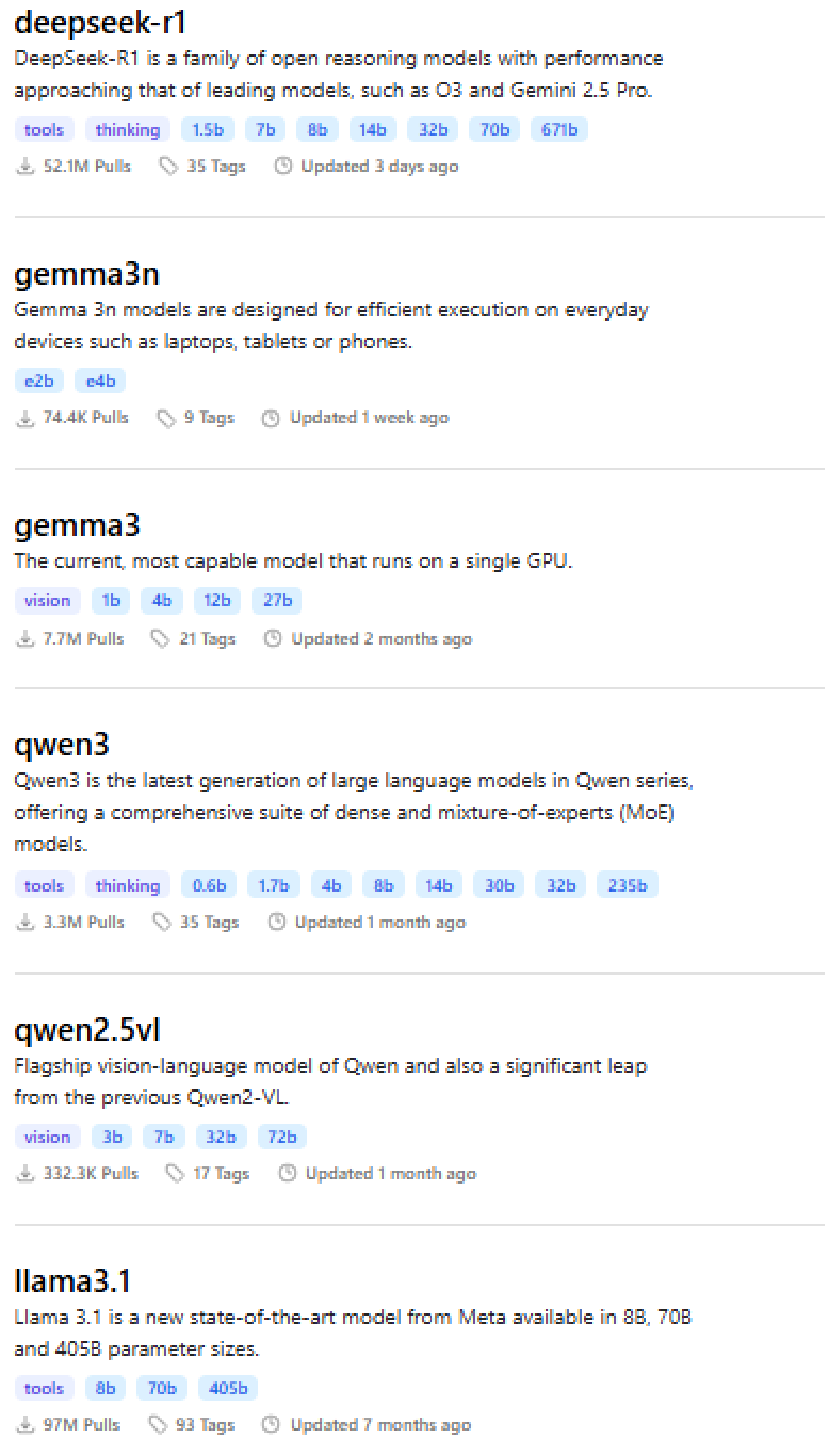
- Low latency responses
- Offline capabilities
- Consistent performance
- Hardware optimization

Flexibility:

- Multiple model choices
- Custom model fine-tuning
- Various framework integrations
- Open-source ecosystem

Popular Models

🎯 Choose the Right Model



deepseek-r1

DeepSeek-R1 is a family of open reasoning models with performance approaching that of leading models, such as O3 and Gemini 2.5 Pro.

tools thinking 1.5b 7b 8b 14b 32b 70b 671b

52.1M Pulls 35 Tags Updated 3 days ago

gemma3n

Gemma 3n models are designed for efficient execution on everyday devices such as laptops, tablets or phones.

e2b e4b

74.4K Pulls 9 Tags Updated 1 week ago

gemma3

The current, most capable model that runs on a single GPU.

vision 1b 4b 12b 27b

7.7M Pulls 21 Tags Updated 2 months ago

qwen3

Qwen3 is the latest generation of large language models in Qwen series, offering a comprehensive suite of dense and mixture-of-experts (MoE) models.

tools thinking 0.6b 1.7b 4b 8b 14b 30b 32b 235b

3.3M Pulls 35 Tags Updated 1 month ago

qwen2.5vl

Flagship vision-language model of Qwen and also a significant leap from the previous Qwen2-VL.

vision 3b 7b 32b 72b

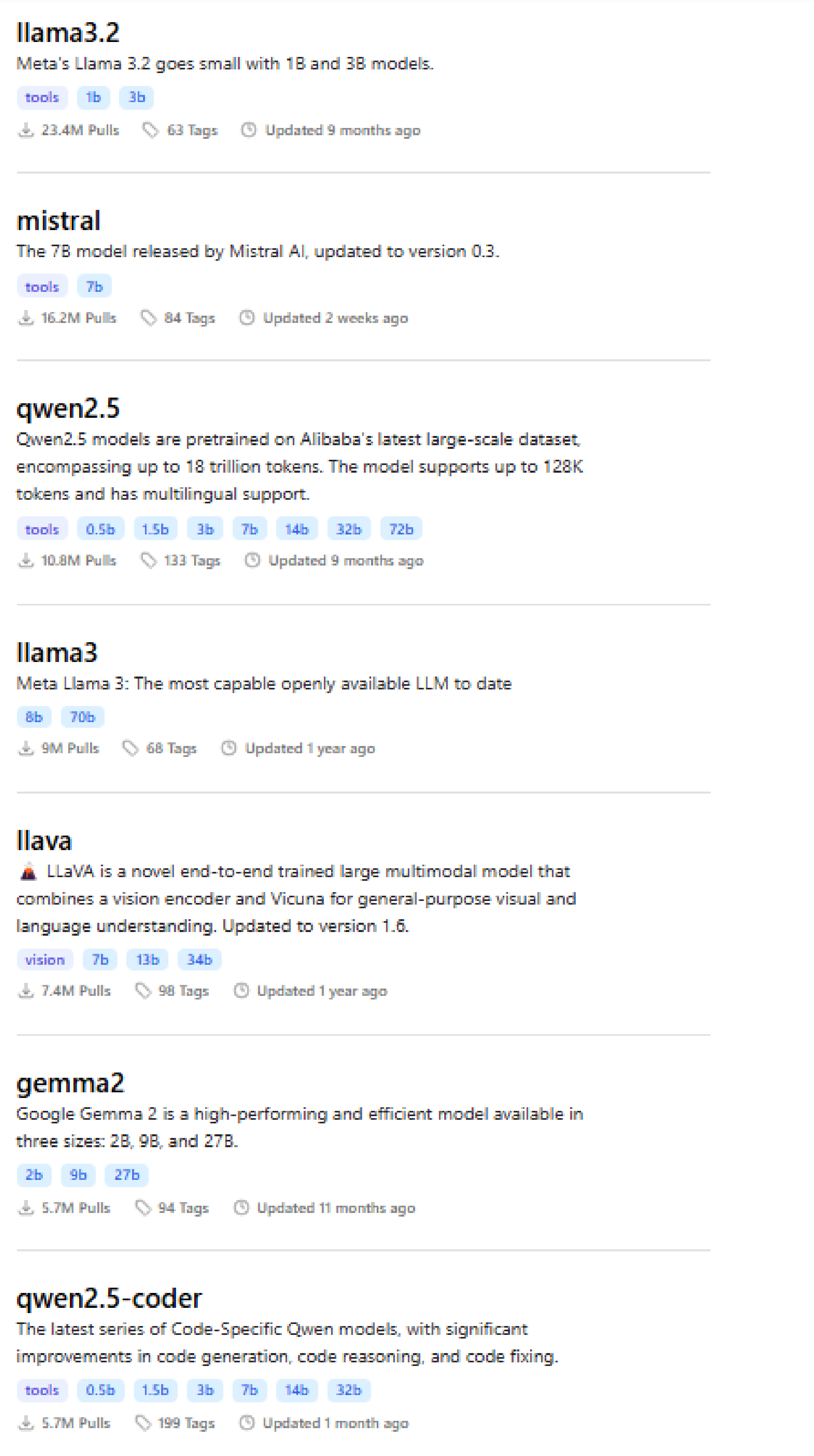
332.3K Pulls 17 Tags Updated 1 month ago

llama3.1

Llama 3.1 is a new state-of-the-art model from Meta available in 8B, 70B and 405B parameter sizes.

tools 8b 70b 405b

97M Pulls 93 Tags Updated 7 months ago



llama3.2

Meta's Llama 3.2 goes small with 1B and 3B models.

tools 1b 3b

23.4M Pulls 63 Tags Updated 9 months ago

mistral

The 7B model released by Mistral AI, updated to version 0.3.

tools 7b

16.2M Pulls 84 Tags Updated 2 weeks ago

qwen2.5

Qwen2.5 models are pretrained on Alibaba's latest large-scale dataset, encompassing up to 18 trillion tokens. The model supports up to 128K tokens and has multilingual support.

tools 0.5b 1.5b 3b 7b 14b 32b 72b

10.8M Pulls 133 Tags Updated 9 months ago

llama3

Meta Llama 3: The most capable openly available LLM to date

8b 70b

9M Pulls 68 Tags Updated 1 year ago

llava

🌋 LLaVA is a novel end-to-end trained large multimodal model that combines a vision encoder and Vicuna for general-purpose visual and language understanding. Updated to version 1.6.

vision 7b 13b 34b

7.4M Pulls 98 Tags Updated 1 year ago

gemma2

Google Gemma 2 is a high-performing and efficient model available in three sizes: 2B, 9B, and 27B.

2b 9b 27b

5.7M Pulls 94 Tags Updated 11 months ago

qwen2.5-coder

The latest series of Code-Specific Qwen models, with significant improvements in code generation, code reasoning, and code fixing.

tools 0.5b 1.5b 3b 7b 14b 32b

5.7M Pulls 199 Tags Updated 1 month ago