```
import torch
# If it returns True, GPU is active.
torch.cuda.is_available()
```

```
True
```

```
import os
print(os.getcwd())
```

```
/content
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import sys

notebook_dir = "/content/drive/MyDrive/Colab Notebooks/CS4100_project-main"
sys.path.append(str(notebook_dir))
```

```
from agent_system import ReActAgent, AgentConfig
from language_model import LLM
from knowledge_base import TOOLS
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/s
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
tokenizer_config.json:       7.30k/? [00:00<00:00, 650kB/s]
vocab.json:       2.78M/? [00:00<00:00, 65.2MB/s]
merges.txt:       1.67M/? [00:00<00:00, 52.5MB/s]
tokenizer.json:       7.03M/? [00:00<00:00, 113MB/s]
config.json: 100%                                                 659/659 [00:00<00:00, 48.8kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors: 100%                                           988M/988M [00:09<00:00, 77.1MB/s]
generation_config.json: 100%                                      242/242 [00:00<00:00, 25.1kB/s]
```

```
agent = ReActAgent(LLM, TOOLS, AgentConfig(max_steps=6, verbose=True))

demo_q = "Who painted The Starry Night and where was it painted?"
result = agent.run(demo_q)

print("Question:", result["question"])
print("\nFinal Answer:", result["final_answer"])
print("\nTrajectory:")
for i, s in enumerate(result["steps"], 1):
    print(f"\nStep {i}")
    print("Thought:", s["thought"])
    print("Action:", s["action"])
    print("Observation:", s["observation"][:500] + ("..." if len(s["observation"])>500 else ""))
```

```
Question: Who painted The Starry Night and where was it painted?

Final Answer: The Starry Night is located in Saint-Rémy-de-Provence, France.

Trajectory:
```

```
Step 1
Thought: I need to find out who painted "The Starry Night" and where it was painted.
Action: search[query="The Starry Night artist", k=1]
Observation: {"tool": "search", "query": "The Starry Night artist", "results": [{"id": "doc2", "title":

Step 2
Thought: I should search for key facts related to the question.
Action: search[query="The Starry Night location", k=1]
Observation: {"tool": "search", "query": "The Starry Night location", "results": [{"id": "doc2", "title"

Step 3
Thought: I have found the answer. The Starry Night is located in Saint-Rémy-de-Provence, France.
Action: finish[answer="The Starry Night is located in Saint-Rémy-de-Provence, France."]
Observation: done
```

I've updated the corpus to include relevant documentation for legal assistance

```
demo_q1 = "What rights does the First Amendment protect?"
demo_q2 = "When are police required to give Miranda warnings?"
demo_q3 = "What are the basic elements required to form a valid contract?"
demo_q4 = "What should a tenant do if a landlord refuses to make necessary repairs?"
demo_q5 = "What happens if someone does not file their federal income taxes on time?"
```

```
for q in [demo_q1]:
    result = agent.run(q)
    print("Question:", result["question"])
    print("\nFinal Answer:", result["final_answer"])
    print("\nTrajectory:")
for i, s in enumerate(result["steps"], 1):
    print(f"\nStep {i}")
    print("Thought:", s["thought"])
    print("Action:", s["action"])
    print("Observation:", s["observation"][:500] + ("..." if len(s["observation"])>500 else ""))
```

```
Question: What rights does the First Amendment protect?

Final Answer: The First Amendment protects freedom of speech, religion, press, assembly, and petition.

Trajectory:

Step 1
Thought: To find out what rights the First Amendment protects, I need to search for information about th
Action: search[query="First Amendment rights"]
Observation: {"tool": "search", "query": "First Amendment rights", "results": [{"id": "doc6", "title": "

Step 2
Thought: The First Amendment protects freedom of speech, religion, press, assembly, and petition.
Action: search[query="First Amendment rights"]
Observation: {"tool": "search", "query": "First Amendment rights", "results": [{"id": "doc6", "title": "

Step 3
Thought: The First Amendment protects freedom of speech, religion, press, assembly, and petition.
Action: finish[answer="The First Amendment protects freedom of speech, religion, press, assembly, and pe
Observation: done
```

```
for q in [demo_q2]:
    result = agent.run(q)
    print("Question:", result["question"])
    print("\nFinal Answer:", result["final_answer"])
    print("\nTrajectory:")
for i, s in enumerate(result["steps"], 1):
    print(f"\nStep {i}")
    print("Thought:", s["thought"])
    print("Action:", s["action"])
    print("Observation:", s["observation"][:500] + ("..." if len(s["observation"])>500 else ""))
```

```
Question: When are police required to give Miranda warnings?
```

```
Final Answer: Police are required to give Miranda warnings when arresting a suspect.

Trajectory:

Step 1
Thought: I should search for key facts related to the question.
Action: search[query="(auto) refine the user question", k=3]
Observation: {"tool": "search", "query": "(auto) refine the user question", "results": [{"id": "doc4", "

Step 2
Thought: To find when police are required to give Miranda warnings, I'll need to search for relevant inf
Action: search[query="(auto) refine the user question", k=3]
Observation: {"tool": "search", "query": "(auto) refine the user question", "results": [{"id": "doc4", "

Step 3
Thought: I should search for key facts related to the question.
Action: search[query="(auto) refine the user question", k=3]
Observation: {"tool": "search", "query": "(auto) refine the user question", "results": [{"id": "doc4", "

Step 4
Thought: I should search for key facts related to the question.
Action: search[query="(auto) refine the user question", k=3]
Observation: {"tool": "search", "query": "(auto) refine the user question", "results": [{"id": "doc4", "

Step 5
Thought: Police are required to give Miranda warnings when arresting a suspect.
Action: finish[answer="Police are required to give Miranda warnings when arresting a suspect."]
Observation: done
```

```
for q in [demo_q3]:
    result = agent.run(q)
    print("Question:", result["question"])
    print("\nFinal Answer:", result["final_answer"])
```

```
Question: What are the basic elements required to form a valid contract?

Final Answer: The essential elements required to form a valid contract include: a clear and unambiguous
```