Agents in LangChain

DESIGNING AGENTIC SYSTEMS WITH LANGCHAIN

Dilini K. Sumanapala, PhD
Founder & AI Engineer, Genverv Ltd.

datacamp

Meet your instructor

- Dilini K. Sumanapala, PhD
- AI Engineer
- Cognitive Neuroscience
- Natural Language Applications
- Founder, Genverv Ltd.

An overview of agents and tools

- **Agents**

  Autonomous systems that make decisions and take actions

- **Tools**

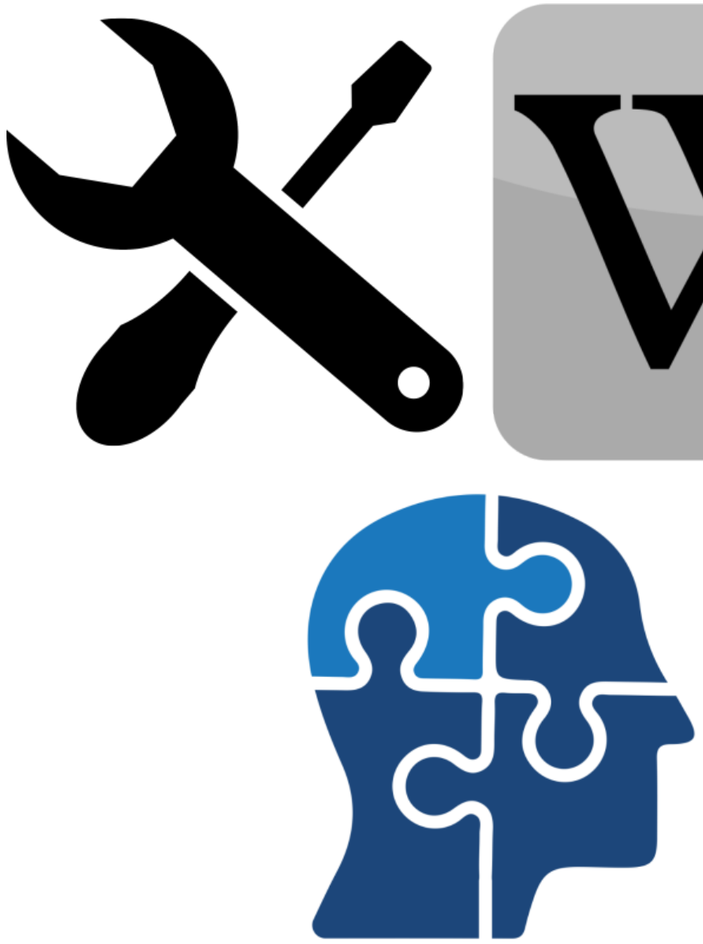
  Functions agents use to perform specific tasks

  - Data query
  - Research reports
  - Data analysis

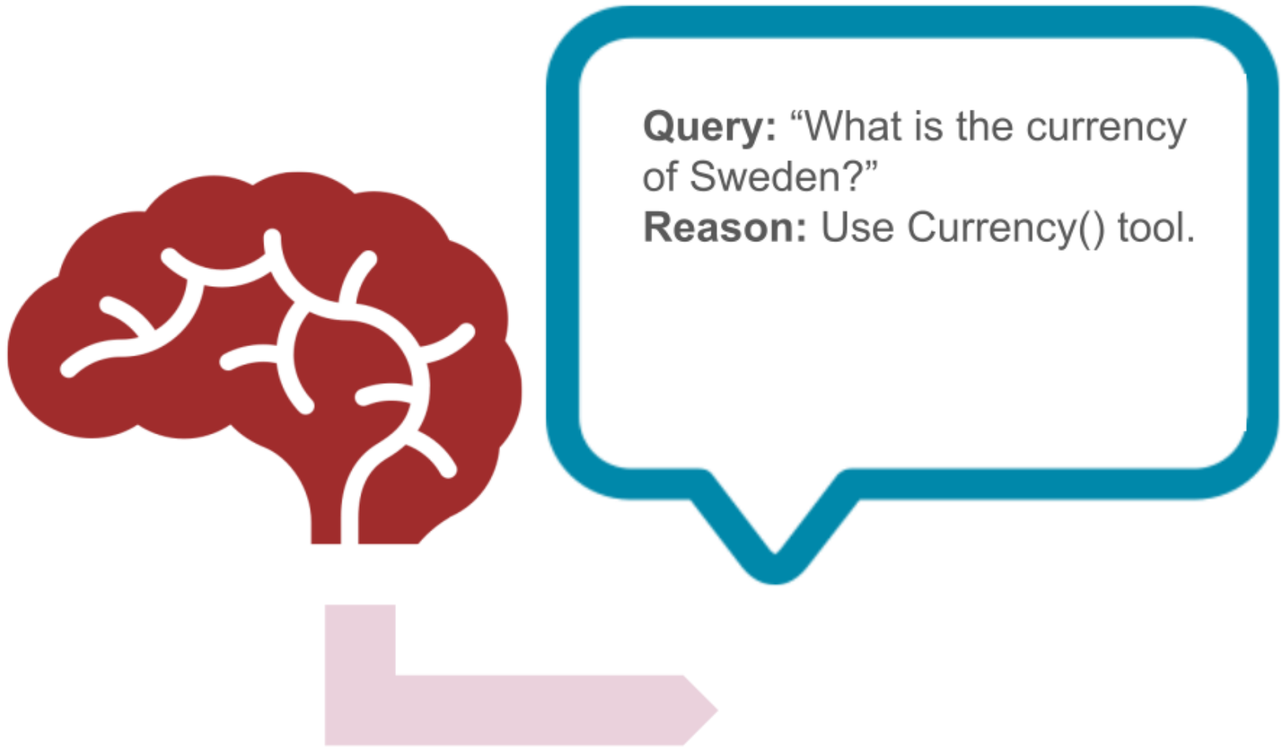
Basic concepts

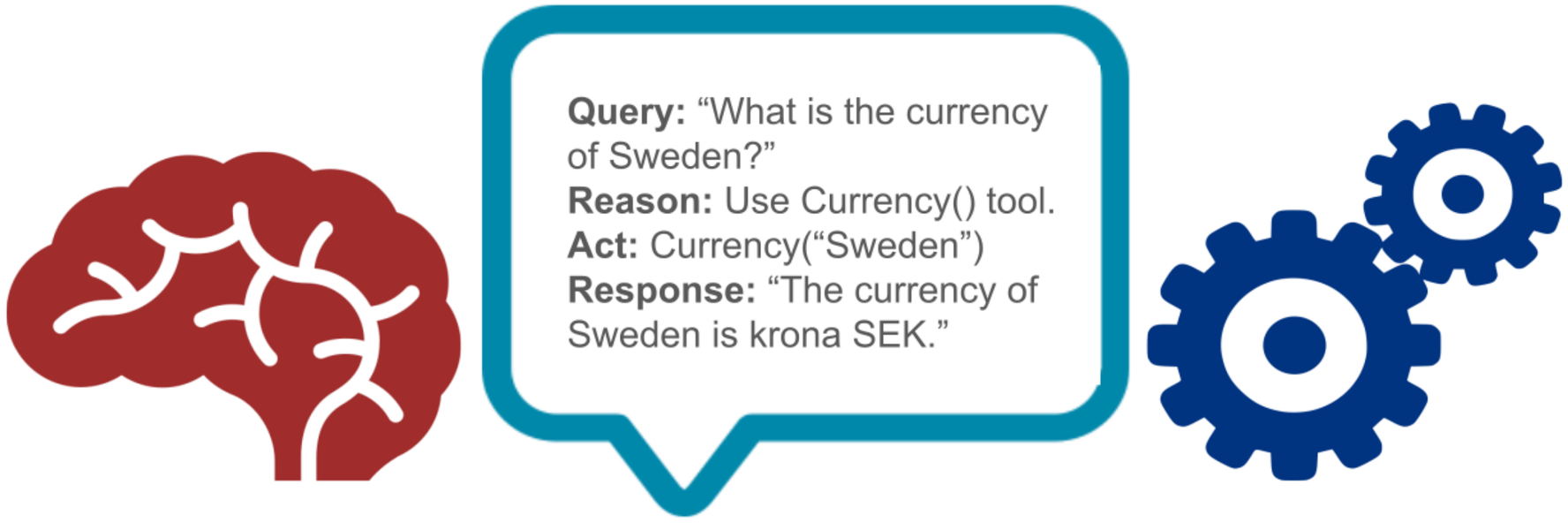
- LLMs (e.g., ChatGPT)
- Prompts
- Tools
- API
- LangChain
  - Building AI agents

Course overview

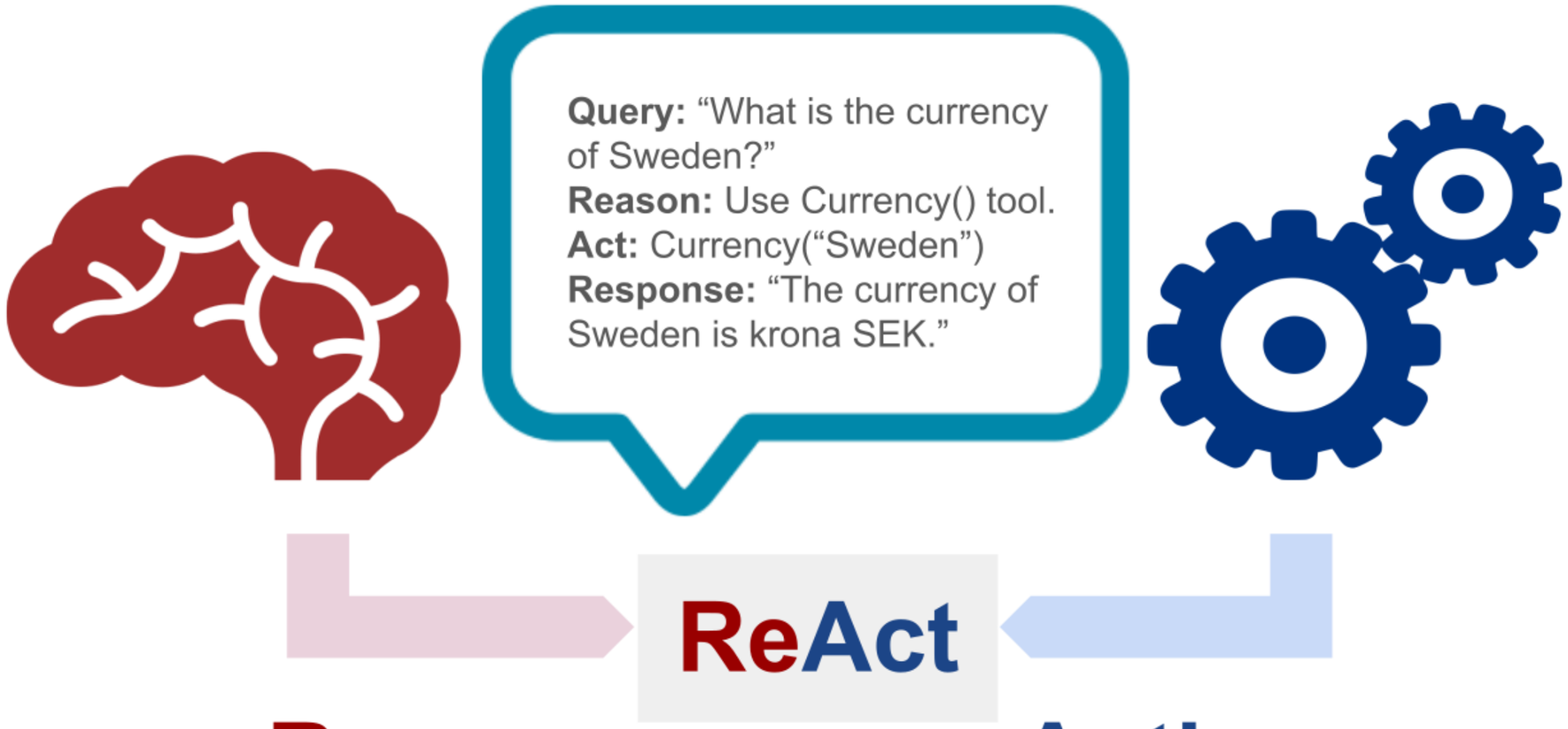


- Math problems
- Wikipedia search
- Switch between tools and LLMs


Query: “What is the currency of Sweden?”
Reason: Use Currency() tool.
Reason



Reason **Action**


Query: “What is the currency of Sweden?”
Reason: Use Currency() tool.
Act: Currency(“Sweden”)
Response: “The currency of Sweden is krona SEK.”
ReAct
Reason
Action

Improving response accuracy

241 - (-241) + 1

241 - (-241) + 1 is equivalent to 241 + 241 + 1, which simplifies to 483 +
241 - (-241) + 1 is equal to 484.

Correct Answer: 483

- Coding
- Math

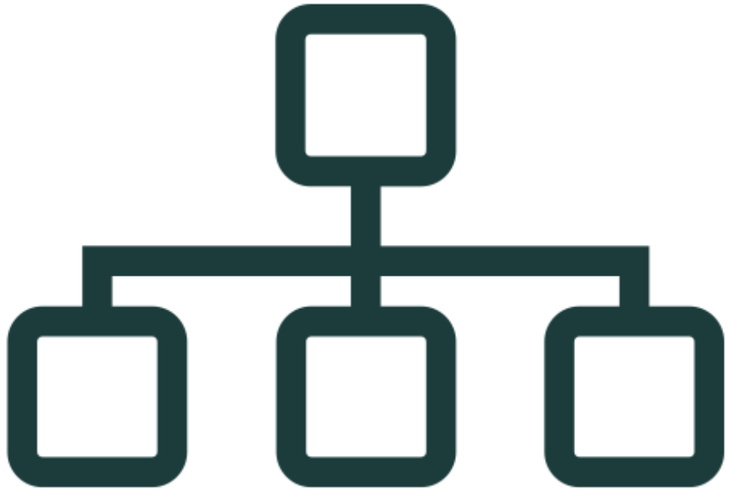
[1] https://community.openai.com/t/chatgpt-simple-math-calculation-mistake/62780

Breaking up problems

Order of Math Operations

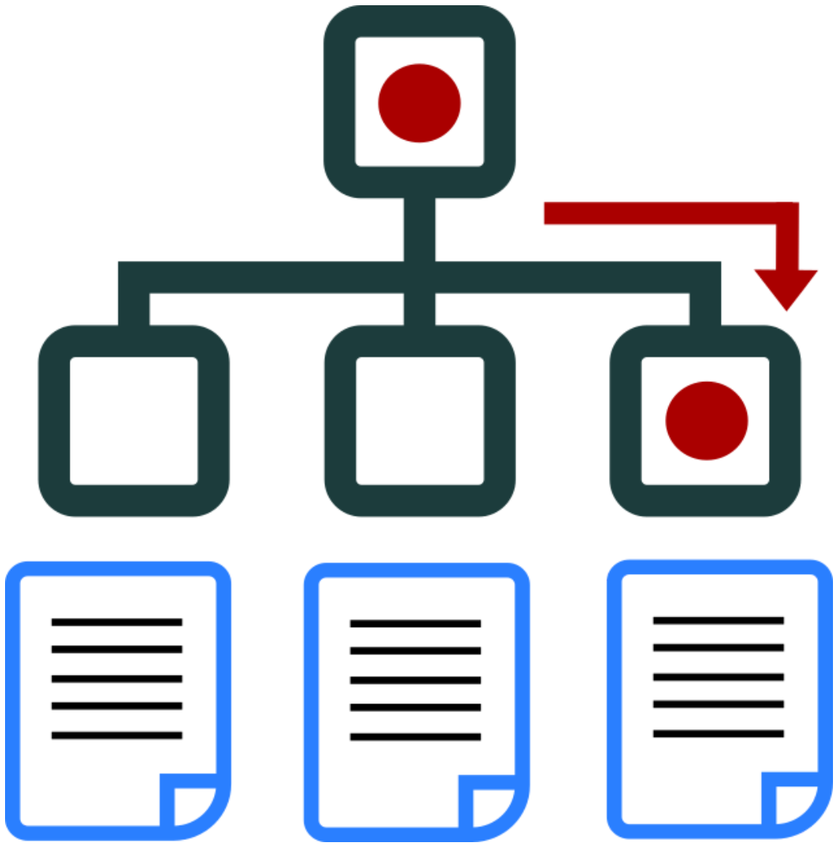
1. Parentheses
2. Exponents
3. Multiplication/Division
4. Addition/Subtraction

Expanding agents with LangGraph



LangGraph

Graph structures



Nodes

- Query the Database
- Return the Document

Edges

Rules connecting nodes

Create a ReAct agent

```
# Module imports
from langchain_core.tools import tool
from langchain_openai import ChatOpenAI
from langgraph.prebuilt import create_react_agent
import math

# LLM Setup
model = ChatOpenAI(openai_api_key="<OPENAI_API_TOKEN">, model="gpt-4o-mini")
```

Create a ReAct agent

```
# Create the agent
agent = create_react_agent(model, tools)

# Create a query
query = "What is (2+8) multiplied by 9?"

# Invoke the agent and print the response
response = agent.invoke({"messages": [("human", query)]})

# Print the agent's response
print(response['messages'][-1].content)
```

```
<script.py> output:
    The result of (2 + 8) multiplied by 9 is 90.
```

Let's practice!

DESIGNING AGENTIC SYSTEMS WITH LANGCHAIN

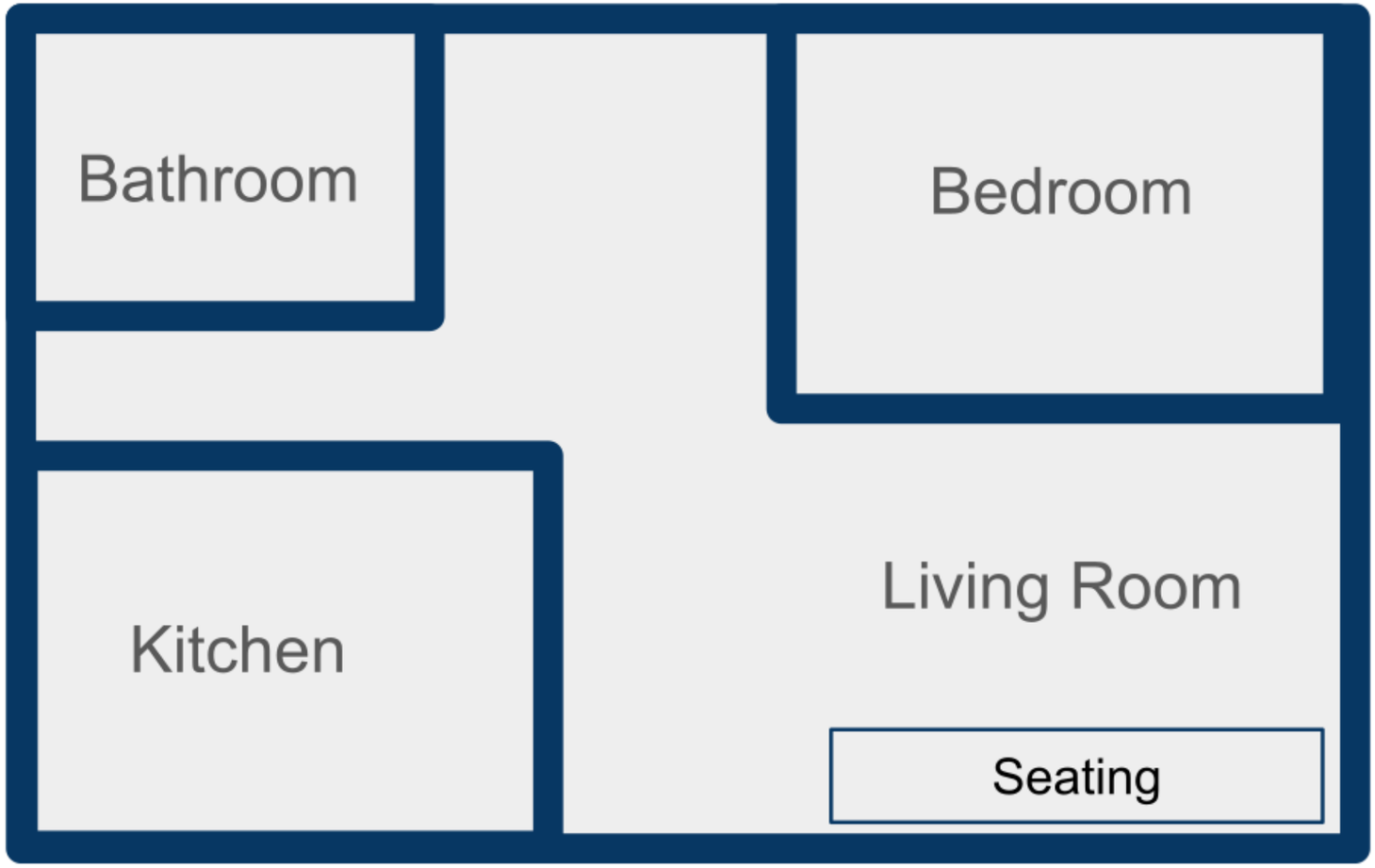
Building custom tools

DESIGNING AGENTIC SYSTEMS WITH LANGCHAIN

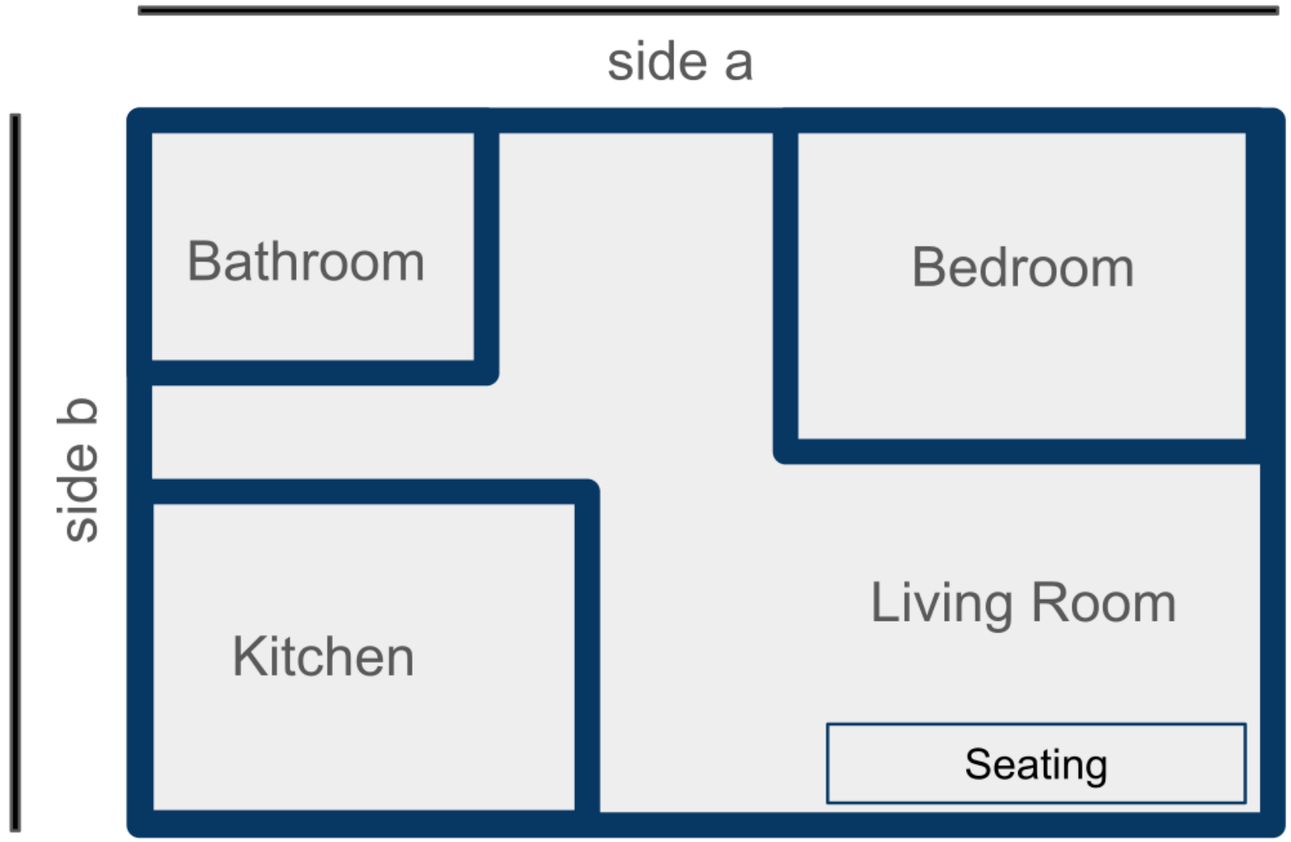
Dilini K. Sumanapala, PhD
Founder & AI Engineer, Genverv, Ltd.

datacamp

Calculating square footage



Calculating square footage



Creating a math tool

LangChain's internal query handling

```
"What is the area of a rectangle with
sides 5 and 7?"
input = " 5, 7"
```

- **Natural language input**
- **Extract numeric values as strings**

Creating a math tool

Define your tool function

```
@tool
def rectangle_area(input: str) -> float:
    """Calculates the area of a
    rectangle given the lengths of
    sides a and b."""
    sides = input.split(',')
    a = float(sides[0].strip())
    b = float(sides[1].strip())
    return a * b
```

- Use `@tool` decorator
- Name the function
- Create a docstring
- Split the input using `.split()`
- Strip whitespace using `.strip()` and convert to float
- Multiply `a` and `b` and return the answer

Tools and query setup

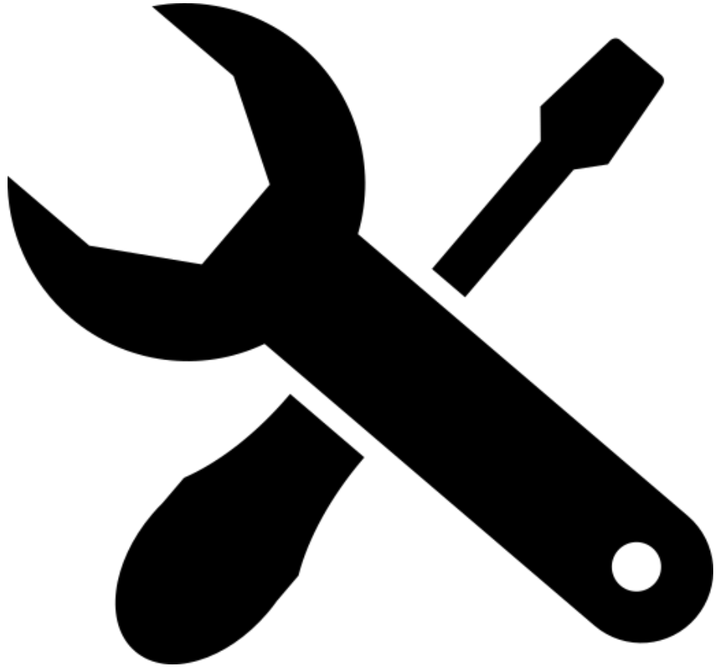
```
# Define the tools that the agent can access
tools = [rectangle_area]

# Create a query using natural language
query = "What is the area of a rectangle with sides 5 and 7?"

# Pass in the hypotenuse length tool and invoke the agent
app = create_react_agent(model, tools)
```

Tools and query setup

```
# Invoke the agent and print the response
response = app.invoke({"messages": [("human", query)]})
print(response['messages'][-1].content)
```

```
The area of the rectangle with sides 5 and 7 is 35 square units.
```

Pre-built and custom tools



- Database Querying
- Web Scraping
- Image Generation

- Pre-built tools API guide
- Custom tool guide

Let's practice!

DESIGNING AGENTIC SYSTEMS WITH LANGCHAIN

Conversation with a ReAct agent

DESIGNING AGENTIC SYSTEMS WITH LANGCHAIN

Dilini K. Sumanapala, PhD
Founder & AI Engineer, Genverv, Ltd.

Conversation

```
The area of a rectangle with sides 5 and 7 is 35 square units.
```

- **Validating answers**
- **User:** "What is the area of a rectangle with sides 5 and 7?"
- **Agent:** "The area of a rectangle with sides 5 and 7 is 35 square units."

Conversation

```
tools = [rectangle_area]
query = "What is the area of a rectangle with sides 14 and 4?"

# Create the ReAct agent
app = create_react_agent(model, tools)

# Invoke the agent with a query and store the messages
response = app.invoke({"messages": [("human", query)]})

# Define and print the input and output messages
print({
    "user_input": query,
    "agent_output": response["messages"][-1].content})
```

Conversation output

```
{'user_input': 'What is the area of a rectangle with sides 14 and 4?',
 'agent_output': 'The area of a rectangle with sides 14 and 4 is 56
 square units.'}
```

Follow-up questions

- **Follow-up:**
  - **User**: "What about one with sides 12 and 14?"
- **Conversation history:**
  - **User**: "What is the area of a rectangle with sides 5 and 7?"
  - **Agent**: "The area of a rectangle with sides 5 and 7 is 35 square units."
  - **User**: "What about one with sides 12 and 14?"
  - **Agent**: "The area of a rectangle with sides 12 and 14 is 168 square units."
- **Output**
  - **User**: "What about one with sides 12 and 14?"
  - **Agent**: "The area of a rectangle with sides 12 and 14 is 168 square units."

Follow-up questions

```
{'user_input': 'What about one with sides 12 and 14?',
'agent_output': ['HumanMessage: What is the area of a rectangle with sides
5 and 7?', 'AIMessage: The area of a rectangle with sides 5 and 7 is 35
square units.',
'HumanMessage: What about one with sides 12 and 14?',
'AIMessage: The area of a rectangle with sides 12 and 14 is 168 square
units.',
'HumanMessage: What about one with sides 12 and 14?',
'AIMessage: The area of a rectangle with sides 12 and 14 is 168 square
units.']}
```

Conversation history

```
from langchain_core.messages import
HumanMessage, AIMessage
```

Conversation history

```
from langchain_core.messages import
HumanMessage, AIMessage

message_history = messages["messages"]
```

Conversation history

```
from langchain_core.messages import
HumanMessage, AIMessage

message_history = messages["messages"]
new_query = "What about one with sides
4 and 3?"
```

Conversation history

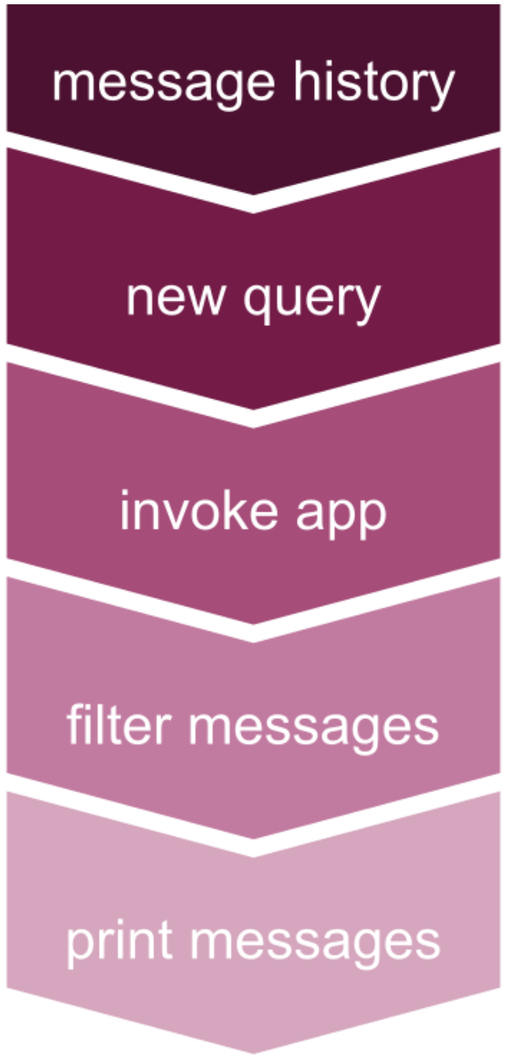
```
from langchain_core.messages import
HumanMessage, AIMessage

message_history = messages["messages"]
new_query = "What about one with sides
4 and 3?"

# Invoke the app with the full message history
messages = app.invoke({"messages":
           message_history + [("human",
           new_query)]})
```

Conversation history

```
# Extract the human and AI messages
filtered_messages = [msg for msg in
                     messages["messages"] if
                     isinstance(msg,
                     (HumanMessage,
                     AIMessage))
                     and msg.content.strip()]
```

Conversation history

```
# Extract the human and AI messages
filtered_messages = [msg for msg in
                     messages["messages"] if
                     isinstance(msg,
                     (HumanMessage,
                     AIMessage))
                     and msg.content.strip()]

# Format and print the final result
print({
    "user_input": new_query, "agent_output":
    [f"{msg.__class__.__name__}:
    {msg.content}" for msg in
    filtered_messages]})
```



Conversation history output

```
{'user_input': 'What about one with sides 4 and 3?',
'agent_output': ['HumanMessage: What is the area of a rectangle with sides
14 and 4?', 'AIMessage: The area of a rectangle with sides 14 and 4 is 56
square units.',
'HumanMessage: What about one with sides 4 and 3?',
'AIMessage: The area of a rectangle with sides 4 and 3 is 12 square
units.',
'HumanMessage: What about one with sides 4 and 3?',
'AIMessage: The area of a rectangle with sides 4 and 3 is 12 square
units.']}
```

Let's practice!

DESIGNING AGENTIC SYSTEMS WITH LANGCHAIN