```
<ipython-input-7-601339460>:25: LangChainDeprecationWarning: LangChain agents will continue to be supported, but it is recommended for new use cases to be built with LangGraph. LangGraph offers a more flexible and full-featured framework for building agents, including support for tool-calling, persistence of state, and human-in-the-loop workflows. For details, refer to the `LangGraph documentation <https://langchain-ai.github.io/langgraph/>`_ as well as guides for `Migrating from AgentExecutor <https://python.langchain.com/docs/how_to/migrate_agent/>`_ and LangGraph's `Pre-built ReAct agent <https://langchain-ai.github.io/langgraph/how-tos/create-react-agent/>`_.

  agent = initialize_agent(

<ipython-input-7-601339460>:34: LangChainDeprecationWarning: The method `Chain.run` was deprecated in langchain 0.1.0 and will be removed in 1.0. Use :meth:`~invoke` instead.

  response = agent.run(

=== Final Answer ===

[chain/start] [chain:AgentExecutor] Entering Chain run with input:

{

  "input": "What's the population of Germany divided by 100?"

}




> Entering new AgentExecutor chain...

[chain/start] [chain:AgentExecutor > chain:LLMChain] Entering Chain run with input:

{

  "input": "What's the population of Germany divided by 100?",

  "agent_scratchpad": "",

  "stop": [

    "\nObservation:",

    "\n\tObservation:"
  ]
}
[llm/start] [chain:AgentExecutor > chain:LLMChain > llm:ChatGoogleGenerativeAI]
Entering LLM run with input:
{
  "prompts": [
    "Human: Answer the following questions as best you can. You have access to the
following tools:\n\nSearch(query: str, **kwargs: Any) -> str - A search engine. Useful for
when you need to answer questions about current events. Input should be a search
query.\nCalculator(*args: Any, callbacks:
Union[list[langchain_core.callbacks.base.BaseCallbackHandler],
langchain_core.callbacks.base.BaseCallbackManager, NoneType] = None, tags:
Optional[list[str]] = None, metadata: Optional[dict[str, Any]] = None, **kwargs: Any) -> Any -
Useful for when you need to answer questions about math.\n\nUse the following
format:\n\nQuestion: the input question you must answer\nThought: you should always
think about what to do\nAction: the action to take, should be one of [Search,
Calculator]\nAction Input: the input to the action\nObservation: the result of the action\n...
(this Thought/Action/Action Input/Observation can repeat N times)\nThought: I now know
the final answer\nFinal Answer: the final answer to the original input
question\n\nBegin!\n\nQuestion: What's the population of Germany divided by
100?\nThought:"
  ]
}
[llm/end] [chain:AgentExecutor > chain:LLMChain > llm:ChatGoogleGenerativeAI] [754ms]
Exiting LLM run with output:
{
  "generations": [
    [
      {
        "text": "I need to find the population of Germany and then divide that number by
100.\nAction: Search\nAction Input: \"population of Germany\"",
      "generation_info": {
       "finish_reason": "STOP",
       "safety_ratings": []
      },
      "type": "ChatGeneration",
      "message": {
       "lc": 1,
       "type": "constructor",
       "id": [
        "langchain",
        "schema",
        "messages",
        "AIMessage"
       ],
       "kwargs": {
        "content": "I need to find the population of Germany and then divide that number by 100.\nAction: Search\nAction Input: \"population of Germany\"",
        "response_metadata": {
         "prompt_feedback": {
          "block_reason": 0,
          "safety_ratings": []
         },
         "finish_reason": "STOP",
         "safety_ratings": []
        },
        "type": "ai",
      "id": "run--fd11457b-d2d4-4c09-b035-1cd5b323a663-0",
      "usage_metadata": {
       "input_tokens": 264,
       "output_tokens": 32,
       "total_tokens": 296,
       "input_token_details": {
        "cache_read": 0
       }
      },
      "tool_calls": [],
      "invalid_tool_calls": []
     }
    }
   }
  ]
 ],
 "llm_output": {
  "prompt_feedback": {
   "block_reason": 0,
   "safety_ratings": []
  }
 },
 "run": null,
 "type": "LLMResult"
}
```

```
[chain/end] [chain:AgentExecutor > chain:LLMChain] [755ms] Exiting Chain run with output:
{
  "text": "I need to find the population of Germany and then divide that number by 100.\nAction: Search\nAction Input: \"population of Germany\""
}
I need to find the population of Germany and then divide that number by 100.
Action: Search
Action Input: "population of Germany"[tool/start] [chain:AgentExecutor > tool:Search] Entering Tool run with input:
"population of Germany"
[tool/end] [chain:AgentExecutor > tool:Search] [1.79s] Exiting Tool run with output:
"{'type': 'population_result', 'population': '83.28 million', 'year': '2023'}"


Observation: {'type': 'population_result', 'population': '83.28 million', 'year': '2023'}
Thought:[chain/start] [chain:AgentExecutor > chain:LLMChain] Entering Chain run with input:
{
  "input": "What's the population of Germany divided by 100?",
  "agent_scratchpad": "I need to find the population of Germany and then divide that number by 100.\nAction: Search\nAction Input: \"population of Germany\"\nObservation: {'type': 'population_result', 'population': '83.28 million', 'year': '2023'}\nThought:",
  "stop": [
    "\nObservation:",
    "\n\tObservation:"
  ]
}
```

```
[llm/start] [chain:AgentExecutor > chain:LLMChain > llm:ChatGoogleGenerativeAI]
Entering LLM run with input:

{

 "prompts": [

  "Human: Answer the following questions as best you can. You have access to the
following tools:\n\nSearch(query: str, **kwargs: Any) -> str - A search engine. Useful for
when you need to answer questions about current events. Input should be a search
query.\nCalculator(*args: Any, callbacks:
Union[list[langchain_core.callbacks.base.BaseCallbackHandler],
langchain_core.callbacks.base.BaseCallbackManager, NoneType] = None, tags:
Optional[list[str]] = None, metadata: Optional[dict[str, Any]] = None, **kwargs: Any) -> Any -
Useful for when you need to answer questions about math.\n\nUse the following
format:\n\nQuestion: the input question you must answer\nThought: you should always
think about what to do\nAction: the action to take, should be one of [Search,
Calculator]\nAction Input: the input to the action\nObservation: the result of the action\n...
(this Thought/Action/Action Input/Observation can repeat N times)\nThought: I now know
the final answer\nFinal Answer: the final answer to the original input
question\n\nBegin!\n\nQuestion: What's the population of Germany divided by
100?\nThought:I need to find the population of Germany and then divide that number by
100.\nAction: Search\nAction Input: \"population of Germany\"\nObservation: {'type':
'population_result', 'population': '83.28 million', 'year': '2023'}\nThought:"

 ]

}

[llm/end] [chain:AgentExecutor > chain:LLMChain > llm:ChatGoogleGenerativeAI] [693ms]
Exiting LLM run with output:

{

 "generations": [

  [

   {

    "text": "I now know the population of Germany is 83.28 million. I need to divide that by
100.\nAction: Calculator\nAction Input: 83280000 / 100",

    "generation_info": {
```

```
  "finish_reason": "STOP",
  "safety_ratings": []
},
"type": "ChatGeneration",
"message": {
  "lc": 1,
  "type": "constructor",
  "id": [
    "langchain",
    "schema",
    "messages",
    "AIMessage"
  ],
  "kwargs": {
    "content": "I now know the population of Germany is 83.28 million. I need to divide that by 100.\nAction: Calculator\nAction Input: 83280000 / 100",
    "response_metadata": {
      "prompt_feedback": {
        "block_reason": 0,
        "safety_ratings": []
      },
      "finish_reason": "STOP",
      "safety_ratings": []
    },
    "type": "ai",
    "id": "run--3455c1cb-2995-487f-9f1d-e715cd7252f5-0",
      "usage_metadata": {
       "input_tokens": 330,
       "output_tokens": 49,
       "total_tokens": 379,
       "input_token_details": {
        "cache_read": 0
       }
      },
      "tool_calls": [],
      "invalid_tool_calls": []
     }
    }
   }
  ]
 ],
 "llm_output": {
  "prompt_feedback": {
   "block_reason": 0,
   "safety_ratings": []
  }
 },
 "run": null,
 "type": "LLMResult"
}
```

```
[chain/end] [chain:AgentExecutor > chain:LLMChain] [694ms] Exiting Chain run with
output:
```

````
{
  "text": "I now know the population of Germany is 83.28 million. I need to divide that by 100.\nAction: Calculator\nAction Input: 83280000 / 100"
}
I now know the population of Germany is 83.28 million. I need to divide that by 100.
Action: Calculator
Action Input: 83280000 / 100[tool/start] [chain:AgentExecutor > tool:Calculator] Entering Tool run with input:
"83280000 / 100"
[chain/start] [chain:AgentExecutor > tool:Calculator > chain:LLMMathChain] Entering Chain run with input:
{
  "question": "83280000 / 100"
}
[chain/start] [chain:AgentExecutor > tool:Calculator > chain:LLMMathChain > chain:LLMChain] Entering Chain run with input:
{
  "question": "83280000 / 100",
  "stop": [
    "```output"
  ]
}
[llm/start] [chain:AgentExecutor > tool:Calculator > chain:LLMMathChain > chain:LLMChain > llm:ChatGoogleGenerativeAI] Entering LLM run with input:
{
  "prompts": [
    "Human: Translate a math problem into a expression that can be executed using Python's numexpr library. Use the output of running this code to answer the question.\n\nQuestion:
````

````
${Question with math problem.}\n```text\n${single line mathematical expression that solves the problem}\n```\n...numexpr.evaluate(text)...\n```output\n${Output of running the code}\n```\nAnswer: ${Answer}\n\nBegin.\n\nQuestion: What is 37593 * 67?\n```text\n37593 * 67\n```\n...numexpr.evaluate(\"37593 * 67\")...\n```output\n2518731\n```\nAnswer: 2518731\n\nQuestion: 37593^(1/5)\n```text\n37593**(1/5)\n```\n...numexpr.evaluate(\"37593**(1/5)\")...\n```output\n8.222831614237718\n```\nAnswer: 8.222831614237718\n\nQuestion: 83280000 / 100"

  ]

}

[llm/end] [chain:AgentExecutor > tool:Calculator > chain:LLMMathChain > chain:LLMChain > llm:ChatGoogleGenerativeAI] [508ms] Exiting LLM run with output:

{

  "generations": [

    [

      {

        "text": "```text\n83280000 / 100\n```\n...numexpr.evaluate(\"83280000 / 100\")...",

        "generation_info": {

          "finish_reason": "STOP",

          "safety_ratings": []

        },

        "type": "ChatGeneration",

        "message": {

          "lc": 1,

          "type": "constructor",

          "id": [

            "langchain",

            "schema",
````

````
      "messages",
      "AIMessage"
     ],
     "kwargs": {
      "content": "```text\n83280000 / 100\n```\n...numexpr.evaluate(\"83280000 / 100\")...",
      "response_metadata": {
       "prompt_feedback": {
        "block_reason": 0,
        "safety_ratings": []
       },
       "finish_reason": "STOP",
       "safety_ratings": []
      },
      "type": "ai",
      "id": "run--17ee7168-eab5-44b9-b523-f0ac72f92e5e-0",
      "usage_metadata": {
       "input_tokens": 274,
       "output_tokens": 42,
       "total_tokens": 316,
       "input_token_details": {
        "cache_read": 0
       }
      },
      "tool_calls": [],
      "invalid_tool_calls": []
````

````
     }
    }
   }
  ]
 ],
 "llm_output": {
  "prompt_feedback": {
   "block_reason": 0,
   "safety_ratings": []
  }
 },
 "run": null,
 "type": "LLMResult"
}
[chain/end] [chain:AgentExecutor > tool:Calculator > chain:LLMMathChain > chain:LLMChain] [509ms] Exiting Chain run with output:
{
 "text": "```text\n83280000 / 100\n```\n...numexpr.evaluate(\"83280000 / 100\")..."
}
[chain/end] [chain:AgentExecutor > tool:Calculator > chain:LLMMathChain] [510ms] Exiting Chain run with output:
{
 "answer": "Answer: 832800.0"
}
[tool/end] [chain:AgentExecutor > tool:Calculator] [512ms] Exiting Tool run with output:
"Answer: 832800.0"
````

```
Observation: Answer: 832800.0
Thought:[chain/start] [chain:AgentExecutor > chain:LLMChain] Entering Chain run with input:
{
  "input": "What's the population of Germany divided by 100?",
  "agent_scratchpad": "I need to find the population of Germany and then divide that number by 100.\nAction: Search\nAction Input: \"population of Germany\"\nObservation: {'type': 'population_result', 'population': '83.28 million', 'year': '2023'}\nThought:I now know the population of Germany is 83.28 million. I need to divide that by 100.\nAction: Calculator\nAction Input: 83280000 / 100\nObservation: Answer: 832800.0\nThought:",
  "stop": [
    "\nObservation:",
    "\n\tObservation:"
  ]
}
[llm/start] [chain:AgentExecutor > chain:LLMChain > llm:ChatGoogleGenerativeAI] Entering LLM run with input:
{
  "prompts": [
    "Human: Answer the following questions as best you can. You have access to the following tools:\n\nSearch(query: str, **kwargs: Any) -> str - A search engine. Useful for when you need to answer questions about current events. Input should be a search query.\nCalculator(*args: Any, callbacks: Union[list[langchain_core.callbacks.base.BaseCallbackHandler], langchain_core.callbacks.base.BaseCallbackManager, NoneType] = None, tags: Optional[list[str]] = None, metadata: Optional[dict[str, Any]] = None, **kwargs: Any) -> Any - Useful for when you need to answer questions about math.\n\nUse the following format:\n\nQuestion: the input question you must answer\nThought: you should always think about what to do\nAction: the action to take, should be one of [Search, Calculator]\nAction Input: the input to the action\nObservation: the result of the action\n... (this Thought/Action/Action Input/Observation can repeat N times)\nThought: I now know the final answer\nFinal Answer: the final answer to the original input
```

```
question\n\nBegin!\n\nQuestion: What's the population of Germany divided by
100?\nThought:I need to find the population of Germany and then divide that number by
100.\nAction: Search\nAction Input: \"population of Germany\"\nObservation: {'type':
'population_result', 'population': '83.28 million', 'year': '2023'}\nThought:I now know the
population of Germany is 83.28 million. I need to divide that by 100.\nAction:
Calculator\nAction Input: 83280000 / 100\nObservation: Answer: 832800.0\nThought:"
 ]
}
[llm/end] [chain:AgentExecutor > chain:LLMChain > llm:ChatGoogleGenerativeAI] [488ms]
Exiting LLM run with output:
{
 "generations": [
  [
   {
    "text": "I now know the final answer\nFinal Answer: 832800.0",
    "generation_info": {
     "finish_reason": "STOP",
     "safety_ratings": []
    },
    "type": "ChatGeneration",
    "message": {
     "lc": 1,
     "type": "constructor",
     "id": [
      "langchain",
      "schema",
      "messages",
      "AIMessage"
  ],
  "kwargs": {
   "content": "I now know the final answer\nFinal Answer: 832800.0",
   "response_metadata": {
    "prompt_feedback": {
     "block_reason": 0,
     "safety_ratings": []
    },
    "finish_reason": "STOP",
    "safety_ratings": []
   },
   "type": "ai",
   "id": "run--9ef4a8fe-c06c-4525-bb99-0767fb3a77cf-0",
   "usage_metadata": {
    "input_tokens": 396,
    "output_tokens": 20,
    "total_tokens": 416,
    "input_token_details": {
     "cache_read": 0
    }
   },
   "tool_calls": [],
   "invalid_tool_calls": []
  }
 }
}
```

```
   ]
  ],
  "llm_output": {
   "prompt_feedback": {
    "block_reason": 0,
    "safety_ratings": []
   }
  },
  "run": null,
  "type": "LLMResult"
}
[chain/end] [chain:AgentExecutor > chain:LLMChain] [489ms] Exiting Chain run with
output:
{
  "text": "I now know the final answer\nFinal Answer: 832800.0"
}
I now know the final answer
Final Answer: 832800.0
[chain/end] [chain:AgentExecutor] [4.24s] Exiting Chain run with output:
{
  "output": "832800.0"
}

> Finished chain.
832800.0
```