

# 第06章：LangChain使用之Agents

讲师：尚硅谷-宋红康

官网：[尚硅谷](#)

## 1、理解Agents

通用人工智能（AGI）将是AI的终极形态，几乎已成为业界共识。同样，构建智能体（Agent）则是AI工程应用当下的“终极形态”。

将 AI 和人类协作的程度类比自动驾驶的不同阶段：

| AI 等级<br>( 类比自动驾驶 ) | 名称      | 特点                                                             | 示例                                 |
|---------------------|---------|----------------------------------------------------------------|------------------------------------|
| L1                  | Tool    | 人类完成所有工作，没有任何显性的 AI 辅助                                         | 目前绝大多数软件产品                         |
| L2                  | Chatbot | 人类完成绝大部分工作。人类向 AI 询问意见，了解信息，AI 提供信息和建议但不直接处理工作                 | 初代 ChatGPT 等 Chatbot               |
| L3                  | Copilot | 人类和 AI 进行协作，工作量相当。AI 根据人类 prompt 完成工作初稿，人类进行目标设定、修改调整，最后确认     | GitHub Copilot、Midjourney、Jasper 等 |
| L4                  | Agent   | AI 完成绝大部分工作，人类负责设定目标、提供资源和监督结果。AI 完成任务拆分，工具选择，进度控制，实现目标后自主结束工作 | AutoGPT 等                          |
| L5                  | Species | 完全无需人类监督，AI 自主拆解目标、寻找资源、选择并使用工具、完成全部工作，人类只需给出目标                | 机器人？                               |

### 1.1 Agent与Chain的区别

在Chain中行动序列是 **硬编码的、固定流程的**，像是“线性流水线”，而Agent则采用语言模型作为 **推理引擎**，具备一定的 **自主决策** 能力，来确定以什么样的顺序采取什么样的行动，像是“拥有大脑的机器人”。

它可以根据任务 **动态决定**：

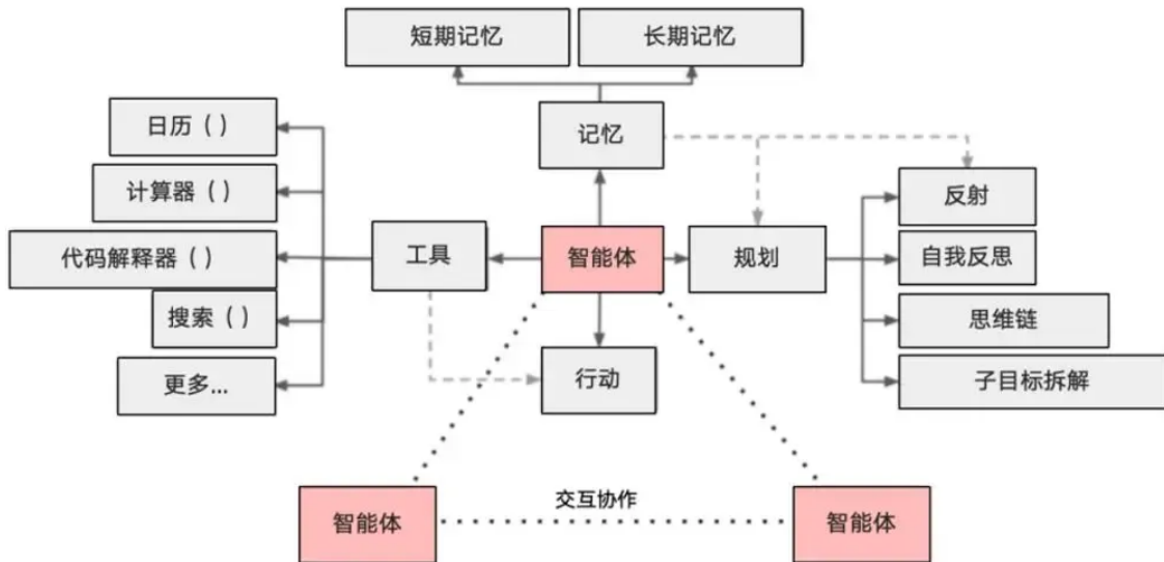
- 如何 **拆解任务**
- 需要 **调用哪些工具**
- 以 **什么顺序调用**
- 如何利用好 **中间结果** 推进任务

## 1.2 什么是Agent

Agent（智能体）是一个通过动态协调 **大语言模型（LLM）** 和 **工具（Tools）** 来完成复杂任务的智能系统。它让LLM充当“决策大脑”，根据用户输入自主选择和执行工具（如搜索、计算、数据库查询等），最终生成精准的响应。

## 1.3 Agent的核心能力/组件

作为一个智能体，需要具备以下核心能力：



1) **大模型(LLM)**：作为大脑，提供推理、规划和知识理解能力。

- 比如：OpenAI()、ChatOpenAI()

2) **记忆(Memory)**：具备短期记忆（上下文）和长期记忆（向量存储），支持快速知识检索。

- 比如：ConversationBufferMemory、ConversationSummaryMemory、ConversationBufferWindowMemory等

3) **工具(Tools)**：调用外部工具（如API、数据库）的执行单元

- 比如：SearchTool、CalculatorTool

4) **规划(Planning)**：任务分解、反思与自省框架实现复杂任务处理

5) **行动(Action)**：实际执行决策的能力

- 比如：检索、推理、编程

6) **协作**：通过与其他智能体交互合作，完成更复杂的任务目标。

**问题**：为什么要调用第三方工具（比如：搜索引擎或者 数据库）或借助第三方库呢？

因为大模型虽然非常强大，但是也具备一定的局限性。比如不能回答 **实时信息**、处理 **复杂数学逻辑** 问题仍然非常初级等等。因此，可以借助第三方工具来辅助大模型的应用。

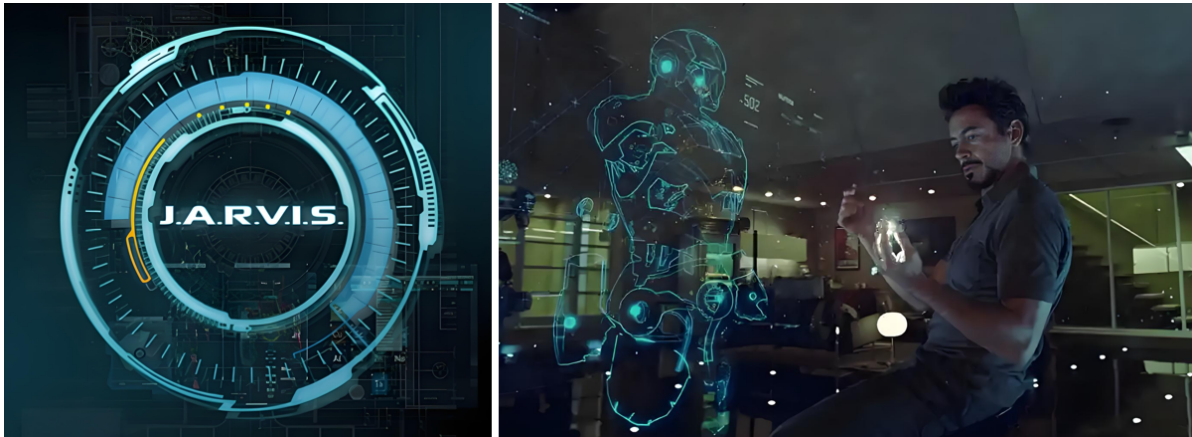
以MCP工具为例说明：<https://bailian.console.aliyun.com/?tab=mcp#/mcp-market>

## 1.4 举例

举例1: [扣子平台][<https://www.coze.cn/home>] 智能体演示

举例2: Manus、纳米AI 使用演示

我们期待它不仅仅是执行任务的工具，而是一个能够思考、自主分析需求、拆解任务并逐步实现目标的智能实体。这种形态的智能体才更接近于人工智能的终极目标-----AGI（通用人工智能），它能让类似于托尼·斯塔克的 **贾维斯** 那样的智能助手成为现实，服务于每个人。



## 1.5 明确几个组件

Agents 模块有几个关键组件：

### 1、工具 Tool

LangChain 提供了广泛的入门工具，但也支持 **自定义工具**，包括自定义描述。

在框架内，每个功能或函数被 **封装成一个工具**（Tools），具有自己的输入、输出及处理方法。

具体使用步骤：

- ① Agent 接收任务后，通过大模型推理选择适合的工具处理任务。
- ② 一旦选定，LangChain将任务输入传递给该工具，工具处理输入生成输出。
- ③ 输出经过大模型推理，可用于其他工具的输入或作为最终结果返回给用户。

### 2、工具集 Toolkits

在构建Agent时，通常提供给LLM的工具不仅仅只有一两个，而是一组可供选择的工具集(Tool列表)，这样可以让 LLM 在完成任务时有更多的选择。

### 3、智能体/代理 Agent

智能体/代理（agent）可以协助我们做出决策，调用相应的 API。底层的实现方式是通过 LLM 来决定下一步执行什么动作。

### 4、代理执行器 AgentExecutor

AgentExecutor本质上是代理的运行时，负责协调智能体的决策和实际的工具执行。

AgentExecutor是一个很好的起点，但是当你开始拥有更多定制化的代理时，它就不够灵活了。为了解决这个问题，我们构建了LangGraph，使其成为这种灵活、高度可控的运行时。

## 2、Agent 入门使用

### 2.1 Agent、AgentExecutor的创建

|          | 环节1：创建Agent                                                                   | 环节2：创建AgentExecutor       |
|----------|-------------------------------------------------------------------------------|---------------------------|
| 方式1：传统方式 | 使用 AgentType 指定                                                               | initialize_agent()        |
| 方式2：通用方式 | create_xxx_agent()<br>比如：create_react_agent()、<br>create_tool_calling_agent() | 调用AgentExecutor()<br>构造方法 |

### 2.2 Agent的类型

顾名思义就是某件事可以由不同的人去完成，最终结果可能是一样的，但是做的过程可能各有千秋。比如一个公司需求，**普通开发** 可以编写，**技术经理** 也可以编写，**CTO** 也可以编写。虽然都能完成最后的需求，但是CTO做的过程可能更加直观，高效。

在LangChain中Agent的类型就是为你提供不同的"问题解决姿势"的。

API说明：[https://python.langchain.com/v0.1/docs/modules/agents/agent\\_types/](https://python.langchain.com/v0.1/docs/modules/agents/agent_types/)

Agents的核心类型有两种模式：

- 方式1：Funcation Call模式
- 方式2：ReAct 模式

#### 2.2.1 FUNCATION\_CALL模式

- 基于 **结构化函数调用** （如 OpenAI Function Calling）
- 直接生成工具调用参数（**JSON 格式**）
- 效率更高，适合工具明确的场景

**典型 AgentType：**

```
1  #第1种:
2  AgentType.OPENAI_FUNCTIONS
3
4  #第2种:
5  AgentType.OPENAI_MULTI_FUNCTIONS
```

**工作流程示例：**

```

1 第1步：找到Search工具：{"tool": "Search", "args": {"query": "LangChain最新版本"}}
2
3 第2步：执行Search工具
4
5 =====
6
7 第1步：找打scrape_website工具：{"tool": "Search", "args": {"target": "LangChain最新版本", "url": "要
  抓取的网站地址"}}
8
9 第2步：执行scrape_website工具

```

## 2.2.2 ReAct 模式

- 基于 **文本推理** 的链式思考 (Reasoning + Acting)，具备反思和自我纠错能力。
  - 推理 (Reasoning)：分析当前状态，决定下一步行动
  - 行动 (Acting)：调用工具并返回结果
- 通过 **自然语言描述决策过程**
- 适合需要明确推理步骤的场景。例如智能客服、问答系统、任务执行等。

**典型 AgentType：**

```

1  #第1种：零样本推理(可以在没有预先训练的情况下尝试解决新的问题)
2  AgentType.ZERO_SHOT_REACT_DESCRIPTION
3
4
5  #第2种：无记忆对话
6  AgentType.STRUCTURED_CHAT_ZERO_SHOT_REACT_DESCRIPTION
7
8
9  #第3种：带记忆对话
10 AgentType.CONVERSATIONAL_REACT_DESCRIPTION

```

**工作流程示例：**

```

1 问题：我想要查询xxx
2 思考：我需要先搜索最新信息 → 行动：调用Search工具 → 观察：获得3个结果 →
3 思考：需要抓取第一个链接 → 行动：调用scrape_website工具... → 观察：获得工具结果
4 最后：获取结果

```

**Agent两种典型类型对比表**

| 特性    | Function Call模式 | ReAct 模式    |
|-------|-----------------|-------------|
| 底层机制  | 结构化函数调用         | 自然语言推理      |
| 输出格式  | JSON/结构化数据      | 自由文本        |
| 适合场景  | 需要高效工具调用        | 需要解释决策过程    |
| 典型延迟  | 较低（直接参数化调用）     | 较高（需生成完整文本） |
| LLM要求 | 需支持函数调用（如gpt-4） | 通用模型即可      |

## 2.3 AgentExecutor创建方式

### 传统方式：initialize\_agent()

- 特点：
  - 内置一些标准化模板（如 ZERO\_SHOT\_REACT\_DESCRIPTION）
  - Agent的创建：使用AgentType
- 优点：快速上手（3行代码完成配置）
- 缺点：定制化能力较弱（如提示词固定）
- 代码片段：

```
1 from langchain.agents import initialize_agent
2
3 #第1步: 创建AgentExecutor
4 agent_executor = initialize_agent(
5     llm=llm,
6     agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION,
7     tools=[search_tool],
8     verbose=True
9 )
10
11 #第2步: 执行
12 agent_executor.invoke({"xxx"})
```

### 通用方式：AgentExecutor构造方法

- 特点：
  - Agent的创建：使用create\_xxx\_agent
- 优点：
  - 可自定义提示词（如从远程hub获取或本地自定义）
  - 清晰分离Agent逻辑与执行逻辑
- 缺点：
  - 需要更多代码
  - 需理解底层组件关系
- 代码片段：

```
1 prompt = hub.pull("hwchase17/react")
2 tools = [search_tool]
```

```

3
4 #第1步: 创建Agent实例
5 agent = create_react_agent(
6     llm=llm,
7     prompt=prompt,
8     tools=tools
9 )
10
11 #第2步: 创建AgentExecutor实例
12 agent_executor = AgentExecutor(
13     agent=agent,
14     tools=tools
15 )
16
17 #第3步: 执行
18 agent_executor.invoke({"input": "xxxxx"})

```

## 2.4 小结创建方式

| 组件                     | 传统方式                                  | 通用方式                                  |
|------------------------|---------------------------------------|---------------------------------------|
| <b>Agent创建</b>         | 通过 <code>AgentType</code> 枚举选择预设      | 通过 <code>create_xxx_agent</code> 显式构建 |
| <b>AgentExecutor创建</b> | 通过 <code>initialize_agent()</code> 创建 | 通过 <code>AgentExecutor()</code> 创建    |
| <b>提示词</b>             | 内置不可见                                 | 可以自定义                                 |
| <b>工具集成</b>            | AgentExecutor中显式传入                    | Agent/AgentExecutor中需显式传入             |

## 3、Agent中工具的使用

### 3.1 传统方式

#### 案例1：单工具使用

- 需求：今天北京的天气怎么样？
- 使用Tavily搜索工具
  - Tavily的搜索API是一个专门为人工智能Agent(或LLM)构建的搜索引擎，可以快速提供实时、准确和真实的结果。
  - LangChain 中有一个内置工具，可以轻松使用 **Tavily 搜索引擎** 作为工具。
  - TAVILY\_API\_KEY申请: <https://tavily.com/>，注册账号并登录，创建 API 密钥。

## 方式1: ReAct模式

- AgentType是 **ZERO\_SHOT\_REACT\_DESCRIPTION**

```
1 from langchain_openai import ChatOpenAI
2 from langchain.agents import initialize_agent, AgentType
3 from langchain.tools import Tool
4 import os
5 import dotenv
6 from langchain_openai import ChatOpenAI
7 from langchain_community.tools.tavily_search import TavilySearchResults
8
9 # 1. 设置 API 密钥
10 os.environ[ "TAVILY_API_KEY" ] = "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"
11
12 # 2. 初始化搜索工具
13 search = TavilySearchResults(max_results=3)
14
15 # 3. 创建Tool的实例 (本步骤可以考虑省略, 直接使用[search]替换[search_tool]。但建议加上
16 search_tool = Tool(
17     name= "Search",
18     func=search.run,
19     description= "用于搜索互联网上的信息"
20 )
21
22 # 4. 初始化 LLM
23 dotenv.load_dotenv()
24
25 os.environ[ 'OPENAI_API_KEY' ] = os.getenv( "OPENAI_API_KEY1" )
26 os.environ[ 'OPENAI_BASE_URL' ] = os.getenv( "OPENAI_BASE_URL" )
27 llm = ChatOpenAI(
28     model= "gpt-4o-mini",
29     temperature=0,
30 )
31
32 # 5. 创建 AgentExecutor
33 agent_executor = initialize_agent(
34     tools=[search_tool],
35     llm=llm,
36     agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION,
37     verbose=True
38 )
39
40 # 5. 测试查询
41 query = "今天北京的天气怎么样? "
42 result = agent_executor.invoke(query)
43 print(f"查询结果: {result}")
```

```
1 > Entering new AgentExecutor chain...
2 我需要查找今天北京的天气信息。
3 Action: Search
4 > Entering new AgentExecutor chain...
```



5 我需要查找今天北京的天气信息。

6 Action: Search

7 Action Input: '今天 北京 天气'

[illegible]

- 9 > Entering new AgentExecutor chain...
- 10 我需要查找今天北京的天气信息。
- 11 Action: Search
- 12 Action Input: '今天 北京 天气'

Observation: [{"title": "中国气象局-天气预报-北京", "url":  
'https://weather.cma.cn/web/weather/54511.html', 'content': '| | | | | |\n| --- | ---  
| --- | --- | --- | --- | --- | --- |\n| æ\x97fë\x97' | 08:00 | 11:00 | 14:00 | 17:00 | 20:00 |  
23:00 | 02:00 | 05:00 |\n| å©æ°\x94 | | | | | | |\n| æ°\x94æ, © | 28.2â\x84\x83 |  
32â\x84\x83 | 33.8â\x84\x83 | 33.5â\x84\x83 | 31â\x84\x83 | 27.8â\x84\x83 |  
25.8â\x84\x83 | 24.2â\x84\x83 |\n| é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97  
é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97  
é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97 é\x99\x8dæ° |\n| éf\x8eé\x80\x9f | 3.3m/s |  
2.3m/s | 3.3m/s | 2.8m/s | 1.5m/s | 0.7m/s | 0.1m/s | 0.8m/s | [...] | | | | | |\n| \n| --- | -  
-- | --- | --- | --- | --- | --- | --- |\n| æ\x97fë\x97' | 08:00 | 11:00 | 14:00 | 17:00 | 20:00 |  
23:00 | 02:00 | 05:00 |\n| å©æ°\x94 | | | | | | |\n| æ°\x94æ, © | 24.9â\x84\x83 |  
28.7â\x84\x83 | 32.3â\x84\x83 | 32.8â\x84\x83 | 29.2â\x84\x83 | 26.7â\x84\x83 |  
24.9â\x84\x83 | 23.2â\x84\x83 |\n| é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97  
é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97  
é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97 é\x99\x8dæ° |\n| éf\x8eé\x80\x9f | 3.3m/s |  
2.2m/s | 2.9m/s | 3.3m/s | 1.7m/s | 2m/s | 3.3m/s | 3m/s | [...] | | | | | |\n| \n| --- | --- |  
--- | --- | --- | --- | --- | --- | --- |\n| æ\x97fë\x97' | 05:00 | 08:00 | 11:00 | 14:00 | 17:00 |  
20:00 | 23:00 | 02:00 |\n| å©æ°\x94 | | | | | | |\n| æ°\x94æ, © | 17.2â\x84\x83 |  
21â\x84\x83 | 26.5â\x84\x83 | 30.8â\x84\x83 | 29.5â\x84\x83 | 25.8â\x84\x83 |  
19.9â\x84\x83 | 16.8â\x84\x83 |\n| é\x99\x8dæ° | 2.3mm | 2.3mm | æ\x97 é\x99\x8dæ° |  
æ\x97 é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97  
é\x99\x8dæ° |\n| éf\x8eé\x80\x9f | 3m/s | 3.1m/s | 5.1m/s | 7.9m/s | 6.9m/s |  
5.2m/s | 3m/s | 2.6m/s | , 'score': 0.74273956}, {'title': '北京市天气预报\_天气查询- 墨迹天气',  
'url': 'https://tianqi.moji.com/weather/china/beijing/beijing', 'content': '【北京市天气】 \_北  
京市天气预报\_天气查询 - 墨迹天气\n\n=====\n\nImage 1: 墨迹天气  
\n\n\_Image 2\_ 随时随地 想查就查\n\n首页天气下载资讯关于墨迹\n\n天气\n中国\n北京市\n北京市\n北京市\n北京市, 北京市, 中国\_\n\n更多城市\n\nImage 3: 72 良 72 良  
\n\n\_Image 4: 阴 阴今天18:37更新\n\n湿度 36% \_南风2级\_\n\n今日  
天气提示\_略微偏热, 注意衣物变化.\_\n\nImage 5: 墨迹天气 小墨哥\n\nImage 6: Windows  
下载Windows 下载\n\n预报\n7天预报\n10天预报\n15天预报\n\n今天\nImage 7: 晴  
晴\n18° / 32°\n西南风1级\n93 良\n\n明天\nImage 8: 晴晴\n19° / 33°\n西  
南风1级\n150 轻度污染 [...]\n\_image 66: 阴\n\n12/22°\n\n北风1级  
\n\n\_image 67: 晴\n\n14/28°\n\n西南风1级\n\n\_image 68: 多  
云\n\n15/28°\n\n东风1级\n\n\_image 69: 多云  
\n\n17/29°\n\n西南风1级\n\n\_image 70: 阴\n\n18/32°\n\n西南风1级  
\n\n\_image 71: 晴\n\n19/33°\n\n西南风1级\n\n\_image 72: 晴  
\n\n22/33°\n\n西南风1级', 'score': 0.7230167}, {'title': '北京-天气预报 - 中  
央气象台', 'url': 'https://www.nmc.cn/publish/forecast/ABJ/beijing.html', 'content': '土壤水  
分监测\n农业干旱综合监测\n关键农时农事\n农业气象周报\n农业气象月报\n农业气象专报\n生态气象监测评估\n作物发育期监测\n数值预报\nCMA  
全球天气模式\nCMA全球集合模式\nCMA区域模式\nCMA区域集合模式\nCMA台风模式\n海浪模式\n\n1. 当前位置: 首页\n2. 北京市\n3. 北京天气预报\n省份:  
城市:\n\n09:50更新\n日出04:45\n北京\n30°C\n日落19:43\n降水量  
0mm\n西南风\n3级\n相对湿度\n43%\n体感温度\n29.9°C\n空气质量:  
良\n舒适度: 温暖, 较舒适\n雷达图\nImage 4\n24小时预报7天预报10天预报11-30  
天预报\n发布时间: 06-12 08:00\n06/12\n周四\nImage 5\n多云\n南风  
3~4级\n35°C\n23°C\nImage 6 [...]\n05:00\nImage 49\n- \n20.2°C  
1.8m/s\n北风\n988.3hPa\n71.3%\n80%\n08:00\nImage 50\n-  
26.2°C\n3.3m/s\n北风\n990hPa\n57.7%\n80.9%\n11:00  
Image 51\n2.3mm\n25.7°C\n3.3m/s\n西风\n990.6hPa\n26.4%\n70%\n14:00\nImage 52\n2.3mm\n28.8°C\n2.2m/s\n西南风\n990.9hPa\n30.3%\n72.1%\n17:00\nImage 53\n2.3mm\n28.2°C\n2.3m/s\n南风\n992.2hPa\n27.2%\n70%\n20:00\nImage 54\n2.3mm  
24.2°C\n0.8m/s\n东南风\n993.9hPa\n56.9%\n70%\n23:00',  
'score': 0.71425}]

```
14 Thought:我现在知道今天北京的天气情况。
15 Final Answer: 今天北京的天气是晴，最高气温32°C，最低气温18°C，湿度为36%，风速为南风2级，空气质量良好。
16
17 > Finished chain.
18 查询结果: 今天北京的天气是晴，最高气温32°C，最低气温18°C，湿度为36%，风速为南风2级，空气质量良好。
```

拓展：上述程序中tool的设置也可以简化为：

```
1  # 初始化搜索工具
2  search = TavilySearchResults(max_results=3)
3
4  # 创建 AgentExecutor
5  agent_executor = initialize_agent(
6      tools=[search],
7      llm=llm,
8      agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION,
9      verbose=True
10 )
```

## 方式2：FUNCATION\_CALL模式

- AgentType是 **OPENAI\_FUNCTIONS**

提示：只需要修改前面代码中的initialize\_agent中的agent参数值。

```
1  from langchain_openai import ChatOpenAI
2  from langchain.agents import initialize_agent, AgentType
3  from langchain.tools import Tool
4  import os
5  import dotenv
6  from langchain_openai import ChatOpenAI
7  from langchain_community.tools.tavily_search import TavilySearchResults
8
9  # 1. 设置 API 密钥
10 os.environ["TAVILY_API_KEY"] = "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"
11
12 # 2. 初始化搜索工具
13 search = TavilySearchResults(max_results=3)
14
15 # 3. 创建Tool的实例
16 search_tool = Tool(
17     name="Search",
18     func=search.run,
19     description="用于搜索互联网上的信息"
20 )
21
22 # 4. 初始化 LLM
23 dotenv.load_dotenv()
24
```

```

25 os.environ[ 'OPENAI_API_KEY' ] = os.getenv( "OPENAI_API_KEY1" )
26 os.environ[ 'OPENAI_BASE_URL' ] = os.getenv( "OPENAI_BASE_URL" )
27 llm = ChatOpenAI(
28     model= "gpt-4o-mini",
29     temperature=0,
30 )
31
32 # 5. 创建 AgentExecutor
33 agent_executor = initialize_agent(
34     tools=[search_tool],
35     llm=llm,
36     agent=AgentType.OPENAI_FUNCTIONS, #唯一变化
37     verbose=True
38 )
39
40 # 5. 测试查询
41 query = "今天北京的天气怎么样? "
42 result = agent_executor.invoke(query)
43 print(f"查询结果: {result}")

```

```

1  > Entering new AgentExecutor chain...
2
3  Invoking: `tavily_search_results_json` with `{'query': '北京天气'}`
4
5
6  > Entering new AgentExecutor chain...
7
8  Invoking: `tavily_search_results_json` with `{'query': '北京天气'}`
9
10
11 > Entering new AgentExecutor chain...
12
13 Invoking: `tavily_search_results_json` with `{'query': '北京天气'}`
14
15

```



27 如果你想查看更详细的天气预报，可以访问[中央气象台]  
(<https://www.nmc.cn/publish/forecast/ABJ/beijing.html>)。

28

29 > Finished chain.

30 查询结果: {'input': '今天北京的天气怎么样? ', 'output': '今天北京的天气情况如下: \n\n- 当前气温: 30°C\n- 天气状况: 多云\n- 风速: 西南风, 约3级\n- 相对湿度: 43%\n- 体感温度: 29.9°C\n- 空气质量: 良\n\n预计今天的最高气温为35°C, 最低气温为23°C, 降水量为0mm, 整体感觉温暖且较为舒适。'\n\n如果你想查看更详细的天气预报，可以访问[中央气象台]  
(<https://www.nmc.cn/publish/forecast/ABJ/beijing.html>)。'} }

二者对比: ZERO\_SHOT\_REACT\_DESCRIPTION和OPENAI\_FUNCTIONS

| 对比维度   | ZERO_SHOT_REACT_DESCRIPTION | OPENAI_FUNCTIONS    |
|--------|-----------------------------|---------------------|
| 底层机制   | 模型生成文本指令，系统解析后调用工具          | 模型直接返回JSON格式工具调用    |
| 执行效率   | 🐢 较低（需多轮文本交互）               | ⚡ 更高（单步完成）          |
| 输出可读性  | 直接显示人类可读的思考过程               | 需查看结构化日志            |
| 工具参数处理 | 依赖模型文本描述准确性                 | 自动匹配工具参数结构          |
| 兼容模型   | 所有文本生成模型                    | 仅GPT-4/Claude 3等新模型 |
| 复杂任务表现 | 可能因文本解析失败出错                 | 更可靠（结构化保证）          |

案例2：多工具使用

- 需求：
  - 计算特斯拉当前股价是多少？
  - 比去年上涨了百分之几？（提示：调用PythonREPL实例的run方法）
- 多个（两个）工具的选择

方式1：ReAct 模式

- AgentType是 ZERO\_SHOT\_REACT\_DESCRIPTION

```
1  # 1.导入相关依赖
2  from langchain_openai import ChatOpenAI
3  from langchain.agents import initialize_agent, AgentType
4  from langchain.tools import Tool
5  from langchain_community.tools.tavily_search import TavilySearchResults
6  from langchain_experimental.utilities.python import PythonREPL
7
8  # 2. 设置 TAVILY_API 密钥
9  os.environ["TAVILY_API_KEY"] = "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" # 需要替换
   为你的 Tavily API 密钥
10
11 # 3.定义搜索工具
12 search = TavilySearchResults(max_results=3)
13
14 search_tool = Tool(
```



```

15     name= "Search",
16     func=search.run,
17     description= "用于搜索互联网上的信息，特别是股票价格和新闻"
18 )
19
20 # 4.定义计算工具
21 python_repl = PythonREPL() # LangChain封装的工具类可以进行数学计算
22
23 calc_tool = Tool(
24     name= "Calculator",
25     func=python_repl.run,
26     description= "用于执行数学计算，例如计算百分比变化"
27 )
28
29 # 5. 定义LLM
30 llm = ChatOpenAI(
31     model= "gpt-4o-mini",
32     temperature=0,
33 )
34
35 # 6. 创建AgentExecutor执行器对象
36 agent_executor = initialize_agent(
37     tools=[search_tool, calc_tool],
38     llm=llm,
39     agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION,
40     verbose=True
41 )
42
43
44 # 7. 测试股票价格查询
45 query = "特斯拉当前股价是多少？比去年上涨了百分之几？"
46 result=agent_executor.invoke(query)
47 print(f"查询结果: {result}")

```

Entering new AgentExecutor chain...

我需要先查找特斯拉当前的股价，然后再查找去年同期的股价，最后计算百分比变化。

#### Action: Search

Action Input: "特斯拉当前股价"

Observation: [{'title': '特斯拉(TSLA) 股票价格、股票新闻及分析师预测 - eToro', 'url': '<http://www.eto.com/zh/markets/tsla>', 'content': '"今日特斯拉股价为323.63，反映过去24小时变动\u200e-0.66\u200e%，过去一周变动\u200e0.72\u200e%。目前TSLA 的市值为1.04T，过去三个月平均成交量为123.32M。该股票市盈率为177.98，股息率为0%。'; 'score': 0.918663}], {'title': 'TSLA股價— 特斯拉圖表 - TradingView', 'url': '<https://tw.tradingview.com/symbols/NASDAQ-TSLA/>', 'content': 'Image 94NIO 股票價格 R休市 NIO Inc.\n\n3.97USD+0.25%\n\nImage 95ZK 股票價格 R休市 ZEEKR Intelligent Technology Holding Limited\n\n28.09USD+7.79%\n\nImage 96OSK 股票價格 R休市 Oshkosh Corporation (Holding Company)\n\n92.02USD+0.93%\n\nImage 97HOG 股票價格 R休市 Harley-Davidson, Inc.\n\n23.62USD+0.94%\n\nImage 98PSNY 股票價格 R休市 Polestar Automotive Holding UK Limited\n\n1.09USD-5.22%\n\n常見問題\n----\uffeff\n\n今天 Tesla的股價是多少？\n\nTSLA目前的價格是298.26 USD — 在過去24小時內上漲了4.72%。在圖表上更密切地關注Tesla股票價格表現。 \n\nTesla股票代碼是什麼？ [...] 1.77ZM93 20.9a.1.1 0 0 1-.1.1h-2.8a.1.1 0 0 1-.1V10.1c0-.06.04-.1.1-.1h2.8c.06 0



<http://stock.finance.sina.com.cn/usstock/quotes/TSLA.html>, 'content': '| 详细行情 | 基本面摘要 | 开盘: | 247.61 | 前收盘: | 254.11 | 市盈率: | 108.32 | 市值: | 7769.50 亿 | 成交量: | 1.12亿 | 区间: | 233.89-251.97 | 每股收益: | 2.23 | 股本: | 32.17亿 | 10 日均量: | 1.60亿 | 52周区间: | 138.80-488.54 | 贝塔系数: | | 股息/收益率: | | 汽车制造: | -1.82% | 领涨股: WNC 9.91(+8.07%) | 领跌股: 特斯拉 241.55(-4.94%) | 行情对比: | 大盘指数 | 纳斯达克 | 道琼斯 | 标普500 | 当日 | 5日 | 周K | 月K | 季K | 年K | 年线 | 不复权 | 前复权 | 年YTD | 日K | ... | Image 56 | 新浪证券关注 | 新浪财经 | 证券微博 | Image 57 | Tom孙 | Chinese | 美股专家 | Image 58 | 沈东军关注 | 美股专家 | Image 59 | 张荣楠关注 | 国家信息中心世界经济副研究员 | 关注 TSLA的好友 | 名称 | 最新价 | 涨跌幅 | 苹果 | 194.27 | -3.89% | 奇虎360 | 76.92 | 0.00% | 谷歌 | 155.50 | -2.00% | Meta Platforms | 196.64 | 0.00% | 百度 | 82.50 | -2.42% | 新浪 | 43.26 | 0.00% | 普拉格能源 | 0.92 | -7.89% | 欢聚集团 | 41.55 | 0.00% | 唯品会 | 12.39 | -1.43% | 去哪儿网 | 30.41 | 0.00% |', 'score': 0.7686816}]

Thought:从搜索结果中，我发现去年同期特斯拉的股价大约在298.26美元左右。现在我可以计算当前股价与去年同期股价的百分比变化。

Action: Calculator

Action Input: "(323.63 - 298.26) / 298.26 \* 100"

Observation:

Thought:抱歉，我需要重新计算百分比变化。让我再次尝试。

Action: Calculator

Action Input: "(323.63 - 298.26) / 298.26 \* 100"

Observation:

Thought:Action: Calculator

Action Input: "(323.63 - 298.26) / 298.26 \* 100"

Observation:

Thought:It seems there was an issue with the calculation tool. Let me try again to calculate the percentage change manually.

Action: Calculator

...

Final Answer: 特斯拉当前股价为323.63美元，比去年同期上涨了约8.51%。

Finished chain.

查询结果: {'input': '特斯拉当前股价是多少？比去年上涨了百分之几？', 'output': '特斯拉当前股价为323.63美元，比去年同期上涨了约8.51%。'}

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

注意：可能会失败（查询超时）。

## 方式2：FUNCATION\_CALL模式

- AgentType是 **FUNCATION\_CALL**

```
1 # 1.导入相关依赖
2 from langchain_openai import ChatOpenAI
3 from langchain.agents import initialize_agent, AgentType
```

```

4 from langchain.tools import Tool
5 from langchain_community.tools.tavily_search import TavilySearchResults
6 from langchain_experimental.utilities.python import PythonREPL
7
8 # 2. 设置 TAVILY_API 密钥
9 os.environ["TAVILY_API_KEY"] = "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" # 需要替换
  为你的 Tavily API 密钥
10
11 # 3. 定义搜索工具
12 search = TavilySearchResults(max_results=3)
13 search_tool = Tool(
14     name="Search",
15     func=search.run,
16     description="用于搜索互联网上的信息，特别是股票价格和新闻"
17 )
18
19 # 4. 定义计算工具
20 python_repl = PythonREPL() # LangChain封装的工具类可以进行数学计算
21 calc_tool = Tool(
22     name="Calculator",
23     func=python_repl.run,
24     description="用于执行数学计算，例如计算百分比变化"
25 )
26
27 # 5. 定义LLM
28 llm = ChatOpenAI(
29     model="gpt-4",
30     temperature=0,
31 )
32
33 # 6. 创建AgentExecutor执行器对象
34 agent_executor = initialize_agent(
35     tools=[search_tool, calc_tool],
36     llm=llm,
37     agent=AgentType.OPENAI_FUNCTIONS, # 唯一变化的位置
38     verbose=True
39 )
40
41
42 # 7. 测试股票价格查询
43 query = "特斯拉当前股价是多少？比去年上涨了百分之几？"
44 result=agent_executor.invoke(query)
45 print(f"查询结果: {result}")

```

Entering new AgentExecutor chain...

Invoking: **Search** with 特斯拉当前股价

Entering new AgentExecutor chain...

Invoking: **Search** with 特斯拉当前股价

Entering new AgentExecutor chain...

Invoking: **Search** with **特斯拉当前股价**

[{'title': '特斯拉(TSLA)股票历史数据 - 英为财经', 'url': 'https://cn.investing.com/equities/tesla-motors-historical-data', 'content': '| 日期 | 收盘 | 开盘 | 高 | 低 | 交易量 | 涨跌幅 |\n| --- | --- | --- | --- | --- | --- | --- |\n| 5月 26, 2025 | 362.89 | 347.35 | 363.79 | 347.32 | 120.15M | +6.94% |\n| 5月 22, 2025 | 339.34 | 337.92 | 343.18 | 333.21 | 84.65M | -0.50% |\n| 5月 21, 2025 | 341.04 | 331.90 | 347.27 | 331.39 | 97.11M | +1.92% |\n| 5月 20, 2025 | 334.62 | 344.43 | 347.35 | 332.20 | 102.35M | -2.68% |\n| 5月 19, 2025 | 343.82 | 347.87 | 354.99 | 341.63 | 131.72M | +0.51% | [...] | 4月 28, 2025 | 292.03 | 285.50 | 293.32 | 279.47 | 108.91M | +2.15% |\n| 4月 27, 2025 | 285.88 | 288.98 | 294.86 | 272.42 | 151.73M | +0.33% | [...] | 名称 | 最新价 | 看涨 | 公允价值 |\n| --- | --- | --- | --- |\n| Aaaaaaaaa | 12.18 | +60.67% | 19.57 |\n| Aaaaaa Aaaa | 25.34 | +59.67% | 40.46 |\n| Aaaaaa Aa | 6.86 | +58.94% | 10.90 |\n| Aaaaaa Aa Aaaaaa | 14.46 | +57.31% | 22.75 |\n| Aaaaaaaaaaaaaa | 22.38 | +55.41% | 34.78 |\n| Aaaaaa Aa Aaaaaaaaaa | 19.08 | +54.75% | 29.53 |\n| Aaaaaaaaaaaaa A A A A | 17.07 | +54.60% | 26.39 |', 'score': 0.8649622}, {'title': '特斯拉(TSLA)股票走势图\_K线图 - 英为财经', 'url': 'https://cn.investing.com/equities/tesla-motors-candlestick', 'content': '取消关注此评论\n\n保存;\n\n已保存。查看收藏列表。);\n\n此评论已保存至您的收藏列表;\n\n屏蔽该用户;\n\n120 能撑住不\n\n回复\n\n- [x] 0- [x] 0\n\n报告;\n\nImage 17: Peter Li\n\nPeter Li2025年3月22日 0:01\n\n;\n\n分享;\n\n关注此评论\n\n取消关注此评论\n\n保存;\n\n已保存。查看收藏列表。);\n\n此评论已保存至您的收藏列表;\n\n屏蔽该用户;\n\n卖掉\n\n回复\n\n- [x] 1- [x] 0\n\n报告;\n\nImage 18: Íker Santiago\n\n先生 樊2025年3月19日 7:12\n\n;\n\n分享;\n\n关注此评论\n\n取消关注此评论\n\n保存;\n\n已保存。查看收藏列表。);\n\n此评论已保存至您的收藏列表;\n\n屏蔽该用户;\n\n现在是3月18日, 股价223, 大胆买进, 加仓到预期的56%, 下一次加仓点位是200左右, 后续是180, 150, 120。在120左右打光所有子弹\n\n回复\n\n- [x] 0- [x] 1\n\n报告;\n\nImage 19: Íker Santiago [...] - 实时数据\n\n类型:股票\n\n市场:美国\n\n量:84,654,818\n\n卖价/买价:0.00 / 0.00\n\n当日幅度:333.21 - 343.18\n\n特斯拉 339.34-1.70-0.50%\n\n总况\n\n概览\n\n简介\n\n历史数据\n\n相关品种\n\n期权\n\n所属股指\n\n图表\n\n流图\n\n互动图表\n\n资讯和分析\n\n资讯\n\n分析评论\n\n财务状况\n\n财务概要\n\n利润表\n\n资产负债表\n\n现金流量\n\n比率\n\n股息\n\n财报\n\n技术\n\n论坛\n\n所有评论\n\n最近观点\n\n用户排名\n\n技术分析\n\nK线型态\n\n分析师目标价\n\nTSLAK线型态\n\n----- [...] 创建提醒;\n\n创建提醒;);\n\n我如何收到这些提醒? )\n\nNew!\n\n创建提醒;\n\n我如何收到这些提醒? )\n\n网站\n\n作为提醒通知\n\n您需要登录至您的账户才能使用此功能\n\n移动App\n\n涨跌额;\n\n量;\n\n财报;\n\n提醒频率\n\n一次 %\n\n提醒频率\n\n循环一次\n\n提醒频率\n\n循环一次\n\n关注特斯拉的财务报告\n\n所有未来的发布 仅限下次发布 - [x] 提前1个交易日给我发送提醒\n\n通知方式\n\n网站弹出窗口\n\n手机App通知\n\n电子邮件提醒\n\n状态\n\n- [x]\n\n创建;)管理我的提醒\n\n返回;\n\n从投资组合添加/删除添加至投资组合;\n\n);\n\n添加至自选组合\n\n添加头寸\n\n头寸已成功添加至: \n\n- [x]\n\n请给您的持仓投资组合命名\n\n类型: \n\n日期: \n\n数量: \n\n价格\n\n基点值: \n\n杠杆: \n\n佣金: \n\n创建新自选组合)创建;)创建新持仓投资组合)添加;)创建;)+ 添加其他头寸)结束;\n\n339.34-1.70-0.50%\n\n- 收盘. USD 货币\n\n收盘后\n\n339.90\n\n+0.56\n\n+0.17%', 'score': 0.8482825}, {'title': '特斯拉(TSLA)股价、新闻、报价和图表 - Moomoo', 'url': 'https://www.moomoo.com/hans/stock/TSLA-US', 'content': 'downloadimg\n\nappLogo\n\n## TSLA 特斯拉\n\n暂无内容\n\n暂无数据\n\n## 资金分布\n\n## 资金流向\n\n暂无数据\n\n暂无数据\n\n## 新闻\n\n快讯 | 美国上诉法院恢复特朗普关税, 同时听取法律论据\n\nWSJ05/29 15:42 (美东)\n\n快讯 | 截至2025年5月30日, WallStreetBets上最受欢迎的十大股票 (通过Swaggy Stocks) \n\nBenzinga05/30 09:16 (美东)\n\n特斯拉的Optimus可能是第一款实现高成交量和技术规模的人形机器人, 英伟达CEO黄仁勋表示: '...可能成为下一个万亿级行业。'\n\nYahoo Finance05/30 07:42 (美东)\n\n摩根士丹利维持特斯拉(TSLA.US)买入评级, 维持目标价410美元 [...] 摩根士丹利分析师Adam Jonas维持Tesla(TSLA.US)买入评级, 维持目标价410美元。根据TipRanks数据显示, 该分析师近一年总胜率为52.5%, 总平均回报率为4.4%。提示: TipRanks为独立第三方, 提供金融分析



师的分析数据，并计算分析师推荐的平均回报率和胜率。提供的信息并非投资建议，仅供参考。本文不对评级数据和报告的完整性与准确性做出认可、声明或保证。

\n\nmoomoo资讯05/30 07:31 (美东) · 评级/大行评级\n\n为什么埃隆·马斯克比以往更需要特斯拉？人形机器人、比亚迪以及特斯拉的更多内容\n\nYahoo Finance05/30 06:59 (美东)\n\n社交热议：周五盘前Wallstreetbets股票走势不一；Gap将下跌，戴尔科技将上涨\n\n在Reddit的Wallstreetbets论坛上，备受关注的股票在周五开盘前几个小时的表现各不相同。Gap（GAP）在盘前下跌了17.7%，继前一交易日下跌1%之后。

\n\nMT Newswires05/30 06:48 (美东)\n\n## 评论", 'score': 0.73868835}]

Invoking: **Search** with **特斯拉去年股价**

Entering new AgentExecutor chain...

Invoking: **Search** with **特斯拉当前股价**

[{'title': '特斯拉(TSLA)股票历史数据 - 英为财经', 'url': 'https://cn.investing.com/equities/tesla-motors-historical-data', 'content': '| 日期 | 收盘 | 开盘 | 高 | 低 | 交易量 | 涨跌幅 |\n| --- | --- | --- | --- | --- | --- | --- |\n| 5月 26, 2025 | 362.89 | 347.35 | 363.79 | 347.32 | 120.15M | +6.94% |\n| 5月 22, 2025 | 339.34 | 337.92 | 343.18 | 333.21 | 84.65M | -0.50% |\n| 5月 21, 2025 | 341.04 | 331.90 | 347.27 | 331.39 | 97.11M | +1.92% |\n| 5月 20, 2025 | 334.62 | 344.43 | 347.35 | 332.20 | 102.35M | -2.68% |\n| 5月 19, 2025 | 343.82 | 347.87 | 354.99 | 341.63 | 131.72M | +0.51% | [...] | 4月 28, 2025 | 292.03 | 285.50 | 293.32 | 279.47 | 108.91M | +2.15% |\n| 4月 27, 2025 | 285.88 | 288.98 | 294.86 | 272.42 | 151.73M | +0.33% | [...] | 名称 | 最新价 | 看涨 | 公允价值 |\n| --- | --- | --- | --- |\n| Aaaaaaaaaa | 12.18 | +60.67% | 19.57 |\n| Aaaaaa Aaaa | 25.34 | +59.67% | 40.46 |\n| Aaaaaa Aaaaaa Aa | 6.86 | +58.94% | 10.90 |\n| Aaaaaa Aa Aaaaaa | 14.46 | +57.31% | 22.75 |\n| Aaaaaaaaaaaaaa | 22.38 | +55.41% | 34.78 |\n| Aaaaaa Aa Aaaaaaaaaa | 19.08 | +54.75% | 29.53 |\n| Aaaaaaaaaaaaaa A A A A | 17.07 | +54.60% | 26.39 |', 'score': 0.8649622}, {'title': '特斯拉(TSLA)股票走势图\_K线图 - 英为财经', 'url': 'https://cn.investing.com/equities/tesla-motors-candlestick', 'content': '取消关注此评论\n\n保存;)\n\n已保存。查看收藏列表。;)\n\n此评论已保存至您的收藏列表;)\n\n屏蔽该用户;)\n\n120 能撑住不\n\n回复)\n\n- [x] 0- [x] 0\n\n报告;)\n\nImage 17: Peter Li\n\nPeter Li2025年3月22日 0:01\n\n;)\n\n分享;)\n\n关注此评论\n\n取消关注此评论\n\n保存;)\n\n已保存。查看收藏列表。;)\n\n此评论已保存至您的收藏列表;)\n\n屏蔽该用户;)\n\n卖掉\n\n回复)\n\n- [x] 1- [x] 0\n\n报告;)\n\nImage 18: Íker Santiago\n\n先生 樊2025年3月19日 7:12\n\n;)\n\n分享;)\n\n关注此评论\n\n取消关注此评论\n\n保存;)\n\n已保存。查看收藏列表。;)\n\n此评论已保存至您的收藏列表;)\n\n屏蔽该用户;)\n\n现在是3月18日，股价223，大胆买进，加仓到预期的56%，下一次加仓点位是200左右，后续是180，150，120。在120左右打光所有子弹\n\n回复)\n\n- [x] 0- [x] 1\n\n报告;)\n\nImage 19: Íker Santiago [...] - 实时数据\n\n类型:股票\n\n市场:美国\n\n量:84,654,818\n\n卖价/买价:0.00 / 0.00\n\n当日幅度:333.21 - 343.18\n\n特斯拉 339.34-1.70-0.50%\n\n总况\n\n概览\n\n简介\n\n历史数据\n\n相关品种\n\n期权\n\n所属股指\n\n图表\n\n流图\n\n互动图表\n\n资讯和分析\n\n资讯\n\n分析评论\n\n财务状况\n\n财务概要\n\n利润表\n\n资产负债表\n\n现金流量\n\n比率\n\n股息\n\n财报\n\n技术\n\n论坛\n\n所有评论\n\n最近观点\n\n用户排名\n\n技术分析\n\nK线型态\n\n分析师目标价\n\nTSLAK 线型态\n\n----- [...] 创建提醒;)\n\n创建提醒;); "我如何收到这些提醒? ") \n\nNew! \n\n创建提醒; "我如何收到这些提醒? ") \n\n网站\n\n作为提醒通知\n\n您需要登录至您的账户才能使用此功能\n\n移动App\n\n涨跌额;)\n\n量;)\n\n财报;)\n\n提醒频率\n\n一次 % \n\n提醒频率\n\n循环一次 \n\n提醒频率\n\n循环一次 \n\n关注特斯拉的财务报告\n\n所有未来的发布 仅限下次发布 - [x] 提前1个交易日给我发送提醒\n\n通知方式\n\n网站弹出窗口\n\n手机App通知\n\n电子邮件提醒\n\n状态\n\n- [x] \n\n创建;)\n\n管理我的提醒\n\n返回;)\n\n从投资组合添加/删除添加至投资组合;)\n\n;)\n\n添加至自选组合\n\n添加头寸\n\n头寸已成功添加至: \n\n- [x] \n\n请给您的持仓投资组合命名\n\n类型: \n\n日期: \n\n数量: \n\n价格\n\n基点值: \n\n杠杆: \n\n佣金: \n\n创建新自选组合)创建;)创建新持仓投资组合)添加;)创建;)+ 添加其他头寸)结束;)\n\n339.34-1.70-0.50%\n\n- 收盘. USD 货币\n\n盘后\n\n339.90\n\n+0.56\n\n+0.17%', 'score': 0.8482825}, {'title': '特斯拉(TSLA)股价、新闻、报价和图表 - Moomoo', 'url': 'https://www.moomoo.com/h

ans/stock/TSLA-US', 'content': "donwloadimg\nappLogo\n\n## TSLA 特斯拉\n\n暂无内容\n\n暂无数据\n\n## 资金分布\n\n## 资金流向\n\n暂无数据\n\n暂无数据\n\n## 新闻\n\n快讯 | 美国上诉法院恢复特朗普关税，同时听取法律论据\n\nWSJ05/29 15:42 (美东)\n\n快讯 | 截至2025年5月30日，WallStreetBets上最受欢迎的十大股票（通过Swaggy Stocks）\n\nBenzinga05/30 09:16 (美东)\n\n特斯拉的Optimus可能是第一款实现高成交量和技术规模的人形机器人，英伟达CEO黄仁勋表示：'...可能成为下一个万亿级行业。'\n\nYahoo Finance05/30 07:42 (美东)\n\n摩根士丹利维持特斯拉(TSLA.US)买入评级，维持目标价410美元 [...] 摩根士丹利分析师Adam Jonas维持Tesla(TSLA.US)买入评级，维持目标价410美元。根据TipRanks数据显示，该分析师近一年总胜率为52.5%，总平均回报率为4.4%。提示: TipRanks为独立第三方，提供金融分析师的分析数据，并计算分析师推荐的平均回报率和胜率。提供的信息并非投资建议，仅供参考。本文不对评级数据和报告的完整性与准确性做出认可、声明或保证。 \n\nmoomoo资讯05/30 07:31 (美东) · 评级/大行评级\n\n为什么埃隆·马斯克比以往更需要特斯拉？人形机器人、比亚迪以及特斯拉的更多内容\n\nYahoo Finance05/30 06:59 (美东)\n\n社交热议：周五盘前Wallstreetbets股票走势不一；Gap将下跌，戴尔科技将上涨\n\n在Reddit的Wallstreetbets论坛上，备受关注的股票在周五开盘前几个小时的表现各不相同。Gap（GAP）在盘前下跌了17.7%，继前一交易日下跌1%之后。 \n\nMT Newswires05/30 06:48 (美东)\n\n## 评论", 'score': 0.73868835}]

Invoking: **Search** with **特斯拉去年股价**

Entering new AgentExecutor chain...

Invoking: **Search** with **特斯拉当前股价**

[{'title': '特斯拉(TSLA)股票历史数据 - 英为财情', 'url': 'https://cn.investing.com/equities/tesla-motors-historical-data', 'content': '| 日期 | 收盘 | 开盘 | 高 | 低 | 交易量 | 涨跌幅 |\n| --- | --- | --- | --- | --- | --- | --- |\n| 5月 26, 2025 | 362.89 | 347.35 | 363.79 | 347.32 | 120.15M | +6.94% |\n| 5月 22, 2025 | 339.34 | 337.92 | 343.18 | 333.21 | 84.65M | -0.50% |\n| 5月 21, 2025 | 341.04 | 331.90 | 347.27 | 331.39 | 97.11M | +1.92% |\n| 5月 20, 2025 | 334.62 | 344.43 | 347.35 | 332.20 | 102.35M | -2.68% |\n| 5月 19, 2025 | 343.82 | 347.87 | 354.99 | 341.63 | 131.72M | +0.51% | [...] | 4月 28, 2025 | 292.03 | 285.50 | 293.32 | 279.47 | 108.91M | +2.15% |\n| 4月 27, 2025 | 285.88 | 288.98 | 294.86 | 272.42 | 151.73M | +0.33% | [...] | 名称 | 最新价 | 看涨 | 公允价值 |\n| --- | --- | --- | --- |\n| Aaaaaaaaa | 12.18 | +60.67% | 19.57 |\n| Aaaaaa Aaaa | 25.34 | +59.67% | 40.46 |\n| Aaaaaa Aaaaaaaa Aa | 6.86 | +58.94% | 10.90 |\n| Aaaaaa Aa Aaaaaa | 14.46 | +57.31% | 22.75 |\n| Aaaaaaaaaaaaaa | 22.38 | +55.41% | 34.78 |\n| Aaaaaa Aa Aaaaaaaaaa | 19.08 | +54.75% | 29.53 |\n| Aaaaaaaaaaaaa A A A A | 17.07 | +54.60% | 26.39 |', 'score': 0.8649622}, {'title': '特斯拉(TSLA)股票走势图\_K线图 - 英为财情', 'url': 'https://cn.investing.com/equities/tesla-motors-candlestick', 'content': '取消关注此评论\n\n保存;)\n\n已保存。查看收藏列表。;)\n\n此评论已保存至您的收藏列表;)\n\n屏蔽该用户;)\n\n120 能撑住不\n\n回复)\n\n- [x] 0- [x] 0\n\n报告;)\n\nImage 17: Peter Li\n\nPeter Li2025年3月22日 0:01\n\n分享;)\n\n关注此评论\n\n取消关注此评论\n\n保存;)\n\n已保存。查看收藏列表。;)\n\n此评论已保存至您的收藏列表;)\n\n屏蔽该用户;)\n\n卖掉\n\n回复)\n\n- [x] 1- [x] 0\n\n报告;)\n\nImage 18: Íker Santiago\n\n先生 樊2025年3月19日 7:12\n\n分享;)\n\n关注此评论\n\n取消关注此评论\n\n保存;)\n\n已保存。查看收藏列表。;)\n\n此评论已保存至您的收藏列表;)\n\n屏蔽该用户;)\n\n现在是3月18日，股价223，大胆买进，加仓到预期的56%，下一次加仓点位是200左右，后续是180，150，120。在120左右打光所有子弹\n\n回复)\n\n- [x] 0- [x] 1\n\n报告;)\n\nImage 19: Íker Santiago [...] - 实时数据\n\n类型:股票\n\n市场:美国\n\n量:84,654,818\n\n卖价/买价:0.00 / 0.00\n\n当日幅度:333.21 - 343.18\n\n特斯拉 339.34-1.70-0.50%\n\n总况\n\n概览\n\n简介\n\n历史数据\n\n相关品种\n\n期权\n\n所属股指\n\n图表\n\n流图\n\n互动图表\n\n资讯和分析\n\n资讯\n\n分析评论\n\n财务状况\n\n财务概要\n\n利润表\n\n资产负债表\n\n现金流量\n\n比率\n\n股息\n\n财报\n\n技术\n\n论坛\n\n所有评论\n\n最近观点\n\n用户排名\n\n技术分析\n\nK线型态\n\n分析师目标价\n\nTSLAK线型态\n\n----- [...] 创建提醒;)\n\n创建提醒;)\n\n我如何收到这些提醒?")\n\nNew!\n\n创建提醒;)\n\n我如何收到这些提醒?")\n\n网站\n\n作为提醒通知\n\n您需要登录至您的账户才能使用此功能

\n\n移动App\n\n\n 涨跌额);\n 量);\n 财报);\n\n提醒频率\n\n一次 % \n提醒频率\n\n循环 一次\n提醒频率\n\n循环 一次 \n关注特斯拉的财务报告\n\n所有未来的发布 仅限下次发布 - [x] 提前1个交易日给我发送提醒 \n\n通知方式\n\n网站弹出窗口\n\n手机App通知\n\n电子邮件提醒\n\n\n状态\n\n- [x] \n\n创建;)管理我的提醒\n\n返回;) \n\n从投资组合添加/删除添加至投资组合;) \n\n);\n\n添加至自选组合 \n\n添加头寸 \n\n头寸已成功添加至: \n\n- [x] \n\n请给您的持仓投资组合命名\n\n类型: \n\n日期: \n\n数量: \n\n价格 \n\n基点值: \n\n杠杆: \n\n佣金: \n\n创建新自选组合)创建;)创建新持仓投资组合)添加;)创建;)+ 添加其他头寸)结束;)\n\n339.34-1.70-0.50%\n\n - 收盘. USD 货币 \n\n盘后 \n\n339.90\n\n+0.56\n\n+0.17%', 'score': 0.8482825}, {'title': '特斯拉(TSLA)股价、新闻、报价和图表 - Moomoo', 'url': 'https://www.moomoo.com/hans/stock/TSLA-US', 'content': "donwloadimg\nappLogo\n\n## TSLA 特斯拉\n\n暂无内容\n\n暂无数据\n\n## 资金分布\n\n## 资金流向\n\n暂无数据\n\n暂无数据\n\n## 新闻\n\n快讯 | 美国上诉法院恢复特朗普关税，同时听取法律论据\n\nWSJ05/29 15:42 (美东)\n\n快讯 | 截至2025年5月30日，WallStreetBets上最受欢迎的十大股票（通过Swaggy Stocks） \n\nBenzinga05/30 09:16 (美东)\n\n特斯拉的Optimus可能是第一款实现高成交量和技术规模的人形机器人，英伟达CEO黄仁勋表示: '...可能成为下一个万亿级行业。'\n\nYahoo Finance05/30 07:42 (美东)\n\n摩根士丹利维持特斯拉(TSLA.US)买入评级，维持目标价410美元 [...] 摩根士丹利分析师Adam Jonas维持Tesla(TSLA.US)买入评级，维持目标价410美元。根据TipRanks数据显示，该分析师近一年总胜率为52.5%，总平均回报率为4.4%。提示: TipRanks为独立第三方，提供金融分析师的分析数据，并计算分析师推荐的平均回报率和胜率。提供的信息并非投资建议，仅供参考。本文不对评级数据和报告的完整性与准确性做出认可、声明或保证。 \n\nmoomoo资讯05/30 07:31 (美东) · 评级/大行评级\n\n为什么埃隆·马斯克比以往更需要特斯拉？人形机器人、比亚迪以及特斯拉的更多内容\n\nYahoo Finance05/30 06:59 (美东)\n\n社交热议: 周五盘前Wallstreetbets股票走势不一; Gap将下跌，戴尔科技将上涨\n\n在Reddit的Wallstreetbets论坛上，备受关注的股票在周五开盘前几个小时的表现各不相同。Gap（GAP）在盘前下跌了17.7%，继前一交易日下跌1%之后。 \n\nMT Newswires05/30 06:48 (美东)\n\n## 评论", 'score': 0.73868835}]

Invoking: **Search** with **特斯拉去年股价**

[{'title': '特斯拉(TSLA)股票历史数据 - 英为财情', 'url': 'https://cn.investing.com/equities/tesla-motors-historical-data', 'content': '| 名称 | 最新价 | 看涨 | 公允价值 |\n| --- | --- | --- | --- |\n| Aaaaaaaa | 12.18 | +60.67% | 19.57 |\n| Aaaaaa Aaaa | 25.34 | +59.67% | 40.46 |\n| Aaaaaa Aa Aaaaaa | 14.46 | +57.31% | 22.75 |\n| Aaaaaaaaaaaaaa | 22.38 | +55.41% | 34.78 |\n| Aaaaaa Aa Aaaaaaaaaa | 19.08 | +54.75% | 29.53 |\n| |\n| Aaaaaaaaaa A A A A | 17.07 | +54.60% | 26.39 | [...]| 名称 | 最新价 | 看涨 | 公允价值 |\n| --- | --- | --- | --- |\n| Aaaaaaaa | 12.18 | +60.67% | 19.57 |\n| Aaaaaa Aaaa | 25.34 | +59.67% | 40.46 |\n| |\n| Aaaaaa Aaaaaaaa Aa | 6.86 | +58.94% | 10.90 |\n| Aaaaaa Aa Aaaaaa | 14.46 | +57.31% | 22.75 |\n| |\n| Aaaaaaaaaaaaaa | 22.38 | +55.41% | 34.78 |\n| Aaaaaa Aa Aaaaaaaaaa | 19.08 | +54.75% | 29.53 |\n| |\n| Aaaaaaaaaa A A A A | 17.07 | +54.60% | 26.39 |\n\n查看完整列表\n\nProPicks AI\n\nAI实力助阵，我们的 优选股票一路领跑，轻松超越标普500\n\n科技巨头\n\n该策略中的股票\n\nnaaa aaaaaa aaaa aaa\n\nnaaa aaaaaa aaaa aaa [...] | 4月 28, 2025 | 292.03 | 285.50 | 293.32 | 279.47 | 108.91M | +2.15% |\n| 4月 27, 2025 | 285.88 | 288.98 | 294.86 | 272.42 | 151.73M | +0.33% |', 'score': 0.81770587}, {'title': 'Tesla股价10年大涨百倍的启示-未来特斯拉股价走势如何 ... - Mittrade', 'url': 'https://www.mittrade.com/cn/insights/shares-analysis/us-stock/how-to-trade-tesla', 'content': '但此后特斯拉股价经历了剧烈波动。2025年初至今，特斯拉股价已较24年末高点跌去40%。这一跌势主要是由于Tesla在欧洲和中国销量下滑，马斯克的个x00x00人争议，及竞争加剧等因素。 \n\n然而，从长期来看，特斯拉仍被视为新能源科技领域的重要参与者。自挂牌以来，其股价在过去十多年间上涨了超过百倍。未来，特斯拉计划在2026年推出自动驾驶出租车（Robotaxi），市场期待下一个数十年的电动车普及浪潮之下，其技术创新仍可能带来新的增长契机! \n\n本文从特斯拉产业链、竞争优势，对TSLA股价进行分析，为大家介绍投资特斯拉股票的方法。在开始之前，我们先来看看特斯拉股价走势。 \n\n## Tesla特斯拉股价趋势图\n\n特斯拉股票代码: TSLA\n\n历史最低价: 5USD\n\n历史最高价: 1243.49USD\n\n▼ 查看Tesla特斯拉股价趋势图表, 在Mittrade交易TSLA\n\nSellBuy\n\n## 特斯拉Tesla股价分析及预测\n\n一、短期挑战（2025-2026年） \n\n市场竞争白热化 [...] 由于竞争加



剧、销量波动、市场不确定性等短期因素，特斯拉的股价在接下来的时间内可能会继续波动，投资者需要保持警惕。

中长期的乐观前景：\n\n预计随着电动车需求的持续增长以及特斯拉的新产品发布，其股价在未来几年内有可能恢复升势。尤其是如果Robotaxi的推出成功，极有可能进一步增强市场信心。

随着更多国家和地区对电动车推广的政策支持，特斯拉在全球市场的表现或将复苏，并推动价格逐渐回升。

## 特斯拉股票值得投资吗？\n\n毫无疑问特斯拉是一只好股票，身边的朋友也有从Tesla上面赚到很多钱的。特斯拉作为全球第一个采用锂离子电池的电动车制造商，业务范围涵盖了新能源汽车的生产和销售，以及充电基础设施的建设和运营。

以下我们总结了TSLA股票的关键特征，大家需要理性分析是否投资。

1. 产业领导地位：不只电动车，更是科技巨头 [...] Tesla引领着新能源和智能电动车技术，Tesla不仅涉足电动车还有太阳能板，电池业务，这几年也涉足加密货币领域。特斯拉电动汽车日益发展迅速是引领其股票价值攀升的主要原因。相较于10年前，Tesla股价如今累计涨幅已经有百倍以上。当今，欧美中国都在大力发展新能源，这是未来的大势所趋。

### 第二，特斯拉创始人马斯克是一个有远见的人\n\n马斯克为公司制定的战略完备，执行力强，公司发展步步为营，成绩斐然。早在2015年，特斯拉创始人就订下每年维持双位数成长，2025年追赶苹果市值的战略。

随着特斯拉的成长，营收逐年增加，特斯拉从市场占有到公司形象都稳健向上。

### 第三，特斯拉产品得到市场认可，销量逐年攀升\n\n虽然特斯拉推出的车型并不多，但是每一个都是热销款。特斯拉在上海的工厂在2021年交付60万辆，并且每个季度都在增长。瑞银分析师认为，特斯拉在2022年会仍然主导该市场，预计，2022年特斯拉汽车销量会达到140万辆，随之利润也会更高。

也就是说，在特斯拉股价上涨的背后，其产品也是在逐年获得更多消费者认可，企业销售额持续增长。

### 第四，特斯拉市值得到专业认可', 'score': 0.65835214}, {'title': 'TSLA股價— 特斯拉圖表 - TradingView', 'url': 'https://tw.tradingview.com/symbols/NASDAQ-TSLA/', 'content': '1.3213.71 1.49Zm-9 16.2a2.1 2.1 0 0 0 1.95-1.32l-3.71-1.49A1.9 1.9 0 0 1 25.93 19v4Zm-6.78-4a1.9 1.9 0 0 1 1.76 2.6l-3.71-1.48A2.1 2.1 0 0 0 19.15 23v-4Zm-5.25 4a2.1 2.1 0 0 0 2.1-2.1h-4c0-1.05.85-1.9 1.9-1.9v4ZM6 20.9c0 1.16.94 2.1 2.1 2.1v-4c1.05 0 1.9.85 1.9 1.9H6ZM7.9 14A1.9 1.9 0 0 1 6 12.1h4A2.1 2.1 0 0 0 7.9 10v4ZM0 11.9C0 13.06.94 14 2.1 14v-4c1.05 0 1.9.85 1.9 1.9H0ZM13.9 8A1.9 1.9 0 0 1 12 6.1h4A2.1 2.1 0 0 0 13.9 4v4ZM2.1 4A2.1 2.1 0 0 0 6.1h4A1.9 1.9 0 0 1 2.1 8V4Z" [...] 2.89v5.67ZM82.93 21a.1.1 0 0 0 .1-.06l5.92-14.8a.1.1 0 0 0-.1-.14h-3.28a.1.1 0 0 0-.1.06l-3.88 10.2a.1.1 0 0 1-.19.01L77.03 6.06a.1.1 0 0 0-.1-.06h-3.28a.1.1 0 0 0-.09.14l6.41 14.8a.1.1 0 0 0 .1.06h2.86ZM119.47 20.93a.1.1 0 0 1-.1.07h-2.77a.1.1 0 0 1-.1-.07l-1.9-5.88c-.03-.1-.16-.1-.2 0l-1.88 5.88a.1.1 0 0 1-.1.07h-2.78a.1.1 0 0 1-.09-.07l-3.5-10.8a.1.1 0 0 1 .09-.13h2.93a.1.1 0 0 1 .1.07l1.92 6.65c.03.1.16.1.2 0l2.05-6.65a.1.1 0 0 1 .1-.07h2.1a.1.1 0 0 1 .1.07l2.08 6.65c.02.1.16.1.19 [...] 20.9v-8.8H6v8.8h4Zm3.9-1.9H8.1v4h5.8v-4ZM12 6.1v14.8h4V6.1h-4ZM19.15 23h6.78v-4h-6.78v4Zm3.97-17.68-5.92 14.8 3.71 1.49 5.92-14.8-3.71-1.49ZM31.85 4h-6.78v4h6.78V4Zm-3.97 17.68 5.92-14.8-3.71-1.49-5.92 14.8 3.71 1.49ZM20 9a1 1 0 0 1-1 1v4a5 5 0 0 0 5-5h-4Zm-1-1a1 1 0 0 1 1 1h4a5 5 0 0 0-5-5v4Zm-1 1a1 1 0 0 1 1-1V4a5 5 0 0 0-5 5h4Zm1 1a1 1 0 0 1-1-1h-4a5 5 0 0 0 5 5v-4Zm12.85-2a1.9 1.9 0 0 1-1.76-2.6l3.71 1.48A2.1 2.1 0 0 0 31.85 4v4Zm-5.02-1.2A1.9 1.9 0 0 1 25.07 8V4a2.1 2.1 0 0 0-1.95', 'score': 0.6565047}}]

Invoking: **Calculator** with **(362.89-292.03)/292.03\*100**

Entering new AgentExecutor chain...

Invoking: **Search** with **特斯拉当前股价**

[{'title': '特斯拉(TSLA)股票历史数据 - 英为财经', 'url': 'https://cn.investing.com/equities/tesla-motors-historical-data', 'content': '| 日期 | 收盘 | 开盘 | 高 | 低 | 交易量 | 涨跌幅 |\n| --- | --- | --- | --- | --- | --- | --- |\n| 5月 26, 2025 | 362.89 | 347.35 | 363.79 | 347.32 | 120.15M | +6.94% |\n| 5月 22, 2025 | 339.34 | 337.92 | 343.18 | 333.21 | 84.65M | -0.50% |\n| 5月 21, 2025 | 341.04 | 331.90 | 347.27 | 331.39 | 97.11M | +1.92% |\n| 5月 20, 2025 | 334.62 | 344.43 | 347.35 | 332.20 | 102.35M | -2.68% |\n| 5月 19, 2025 | 343.82 | 347.87 | 354.99 | 341.63 | 131.72M | +0.51% | [...] | 4月 28, 2025 | 292.03 | 285.50 | 293.32 | 279.47 | 108.91M | +2.15% |\n| 4月 27, 2025 | 285.88 | 288.98 | 294.86 | 272.42 | 151.73M | +0.33% | [...] | 名称 | 最新价 | 看涨 | 公允价值 |\n| --- | --- | --- | --- | --- |\n

Aaaaaaaa | 12.18 | +60.67% | 19.57 | Aaaaaa Aaaa | 25.34 | +59.67% | 40.46 | Aaaaaa  
Aaaaaaaa Aa | 6.86 | +58.94% | 10.90 | Aaaaaa Aa Aaaaaa | 14.46 | +57.31% | 22.75 |  
Aaaaaaaaaaaaaa | 22.38 | +55.41% | 34.78 | Aaaaaa Aa Aaaaaaaaaaaaaa | 19.08 | +54.75% | 29.53  
| Aaaaaaaa A A A A | 17.07 | +54.60% | 26.39 |, 'score': 0.8649622}, {'title': '特斯拉  
(TSLA)股票走势图\_K线图 - 英为财经', 'url': 'https://cn.investing.com/equities/tesla-motors-c  
andlestick', 'content': '取消关注此评论\n\n保存;)\n\n已保存。查看收藏列表。;)\n\n此评论已保  
存至您的收藏列表;)\n\n屏蔽该用户;)\n\n120 能撑住不\n\n回复)\n\n- [x] 0- [x] 0\n\n报  
告;)\n\nImage 17: Peter Li\n\nPeter Li2025年3月22日 0:01\n\n;)\n\n分享;)\n\n关注此评论  
\n\n取消关注此评论\n\n保存;)\n\n已保存。查看收藏列表。;)\n\n此评论已保存至您的收藏列  
表;)\n\n屏蔽该用户;)\n\n卖掉\n\n回复)\n\n- [x] 1- [x] 0\n\n报告;)\n\nImage 18: Íker  
Santiago\n\n先生 樊2025年3月19日 7:12\n\n;)\n\n分享;)\n\n关注此评论\n\n取消关注此评论  
\n\n保存;)\n\n已保存。查看收藏列表。;)\n\n此评论已保存至您的收藏列表;)\n\n屏蔽该用  
户;)\n\n现在是3月18日，股价223，大胆买进，加仓到预期的56%，下一次加仓点位是200左右，  
后续是180，150，120。在120左右打光所有子弹\n\n回复)\n\n- [x] 0- [x] 1\n\n报告;)\n\nImage  
19: Íker Santiago [...] - 实时数据\n\n类型:股票\n\n市场:美国\n\n量:84,654,818\n\n卖价/买  
价:0.00 / 0.00\n\n当日幅度:333.21 - 343.18\n\n特斯拉 339.34-1.70-0.50%\n\n总况\n\n概  
览\n\n简介\n\n历史数据\n\n相关品种\n\n期权\n\n所属股指\n\n图表\n\n流图\n\n互动图表\n\n资讯和分析\n\n资讯\n\n分析评论\n\n财务状况\n\n财务概要\n\n利  
润表\n\n资产负债表\n\n现金流量\n\n比率\n\n股息\n\n财报\n\n技术\n\n论坛\n\n所有评论\n\n最近观点\n\n用户排名\n\n技术分析\n\nK线型态\n\n分析师目标价\n\nTSLAK  
线型态\n\n----- [...] 创建提醒;)\n\n创建提醒;); "我如何收到这些提醒?")\n\nNew!\n\n创建提醒;  
"我如何收到这些提醒?")\n\n网站\n\n作为提醒通知\n\n您需要登录至您的账户才能使用此功能  
\n\n移动App\n\n涨跌额;\n\n量;\n\n财报;)\n\n提醒频率\n\n一次 % \n提醒频率\n\n循环一  
次\n\n提醒频率\n\n循环一次\n\n关注特斯拉的财务报告\n\n所有未来的发布 仅限下次发布 - [x] 提  
前1个交易日给我发送提醒\n\n通知方式\n\n网站弹出窗口\n\n手机App通知\n\n电子邮件提醒  
\n\n状态\n\n- [x] \n创建;)\n\n管理我的提醒\n\n返回;)\n\n从投资组合添加/删除添加至投资组  
合;)\n\n;)\n\n添加至自选组合\n\n添加头寸\n\n头寸已成功添加至: \n- [x] \n请给您的持仓  
投资组合命名\n\n类型: \n\n日期: \n\n数量: \n\n价格\n\n基点值: \n\n杠杆: \n\n佣金: \n\n创建  
新自选组合\n\n创建;)\n\n创建新持仓投资组合\n\n添加;)\n\n创建;)+ 添加其他头寸\n\n结束;)\n\n339.34-1.70-  
0.50%\n\n- 收盘. USD 货币\n\n盘后\n\n339.90\n\n+0.56\n\n+0.17%', 'score': 0.8482825},  
{ 'title': '特斯拉(TSLA)股价、新闻、报价和图表 - Moomoo', 'url': 'https://www.moomoo.com/h  
ans/stock/TSLA-US', 'content': "donwloadimg\n\nappLogo\n\n## TSLA 特斯拉\n\n暂无内容  
\n\n暂无数据\n\n## 资金分布\n\n## 资金流向\n\n暂无数据\n\n暂无数据\n\n## 新闻\n\n快讯 |  
美国上诉法院恢复特朗普关税，同时听取法律论据\n\nWSJ05/29 15:42 (美东)\n\n快讯 | 截至2025  
年5月30日，WallStreetBets上最受欢迎的十大股票（通过Swaggy Stocks）\n\nBenzinga05/30  
09:16 (美东)\n\n特斯拉的Optimus可能是第一款实现高成交量和技术规模的人形机器人，英伟达  
CEO黄仁勋表示: '...可能成为下一个万亿级行业。'\n\nYahoo Finance05/30 07:42 (美东)\n\n摩根  
士丹利维持特斯拉(TSLA.US)买入评级，维持目标价410美元 [...] 摩根士丹利分析师Adam  
Jonas维持Tesla(TSLA.US)买入评级，维持目标价410美元。根据TipRanks数据显示，该分析  
师近一年总胜率为52.5%，总平均回报率为4.4%。提示: TipRanks为独立第三方，提供金融分析  
师的分析数据，并计算分析师推荐的平均回报率和胜率。提供的信息并非投资建议，仅供参考。本  
文不对评级数据和报告的完整性与准确性做出认可、声明或保证。)\n\nmoomoo资讯05/30 07:31  
(美东) · 评级/大行评级\n\n为什么埃隆·马斯克比以往更需要特斯拉？人形机器人、比亚迪以及特  
斯拉的更多内容\n\nYahoo Finance05/30 06:59 (美东)\n\n社交热议：周五盘前Wallstreetbets股  
票走势不一；Gap将下跌，戴尔科技将上涨\n\n在Reddit的Wallstreetbets子论坛上，备受关注的  
股票在周五开盘前几个小时的表现各不相同。Gap（GAP）在盘前下跌了17.7%，继前一交易日下  
跌1%之后。)\n\nMT Newswires05/30 06:48 (美东)\n\n## 评论", 'score': 0.73868835}]

Invoking: Search with 特斯拉去年股价

['title': '特斯拉(TSLA)股票历史数据 - 英为财经', 'url': 'https://cn.investing.com/equities/tesla-motors-historical-data', 'content': '| 名称 | 最新价 | 看涨 | 公允价值 | \n| --- | --- | --- | --- | \n| Aaaaaaaa | 12.18 | +60.67% | 19.57 | \n| Aaaaaa Aaaa | 25.34 | +59.67% | 40.46 | \n| Aaaaaa Aa Aaaaaa | 14.46 | +57.31% | 22.75 | \n| Aaaaaaaaaaaaaa | 22.38 | +55.41% | 34.78 | \n| Aaaaaa Aa Aaaaaaaaaaaaaa | 19.08 | +54.75% | 29.53 | \n| Aaaaaaaaaaaaaa A A A A | 17.07 | +54.60% | 26.39 | [...] | 名称 | 最新价 | 看涨 | 公允价值 | \n| --- | --- | --- | --- | \n| Aaaaaaaa | 12.18 | +60.67% | 19.57 | \n| Aaaaaa Aaaa | 25.34 | +59.67% | 40.46 | \n| Aaaaaa Aaaaaaaa Aa | 6.86 | +58.94% | 10.90 | \n| Aaaaaa Aa Aaaaaa | 14.46 | +57.31% | 22.75 | \n| Aaaaaaaaaaaaaa | 22.38 | +55.41% | 34.78 | \n| Aaaaaa Aa Aaaaaaaaaaaaaa | 19.08 | +54.75% | 29.53 | \n| Aaaaaaaaaaaaaa A A A A | 17.07 | +54.60% | 26.39 | \n\n查看完整列表\n\nPro Picks AI\n\nAI实力助阵，我们的 优选股票一路领跑，轻松超越标普500\n\n科技巨头\n\n该策略中的股票\n\nnaaa aaaaaaa aaaa aaa\n\nnaaa aaaaaaa aaaa aaa [...] | 4月 28, 2025 | 292.03 | 285.50 | 293.32 | 279.47 | 108.91M | +2.15% | \n| 4月 27, 2025 | 285.88 | 288.98 | 294.86 | 272.42 | 151.73M | +0.33% |', 'score': 0.81770587}, {'title': 'Tesla股价10年大涨百倍的启示-未来特斯拉股价走势如何 ... - Mitrade', 'url': 'https://www.mitrade.com/cn/insights/shares-analysis/us-stock/how-to-trade-tesla', 'content': '但此后特斯拉股价经历了剧烈波动。2025年初至今，特斯拉股价已较24年末高点跌去40%。这一跌势主要是由于Tesla在欧洲和中国销量下滑，马斯克的个x00x00人争议，及竞争加剧等因素。然而，从长期来看，特斯拉仍被视为新能源科技领域的重要参与者。自挂牌以来，其股价在过去十多年间上涨了超过百倍。未来，特斯拉计划在2026年推出自动驾驶出租车（Robotaxi），市场期待下一个数十年的电动车普及浪潮之下，其技术创新仍可能带来新的增长契机！本文从特斯拉产业链、竞争优势，对TSLA股价进行分析，为大家介绍投资特斯拉股票的方法。在开始之前，我们先来看看特斯拉股价走势。## Tesla特斯拉股价趋势图\n\n特斯拉股票代码：TSLA\n\n历史最低价：5USD\n\n历史最高价：1243.49USD\n\n查看Tesla特斯拉股价趋势图表，在Mitrade交易TSLA\n\nSellBuy\n\n## 特斯拉Tesla股价分析及预测\n\n一、短期挑战（2025-2026年）\n\n市场竞争白热化 [...] 由于竞争加剧、销量波动、市场不确定性等短期因素，特斯拉的股价在接下来的时间内可能会继续波动，投资者需要保持警惕。中长期的乐观前景：\n\n预计随着电动车需求的持续增长以及特斯拉的新产品发布，其股价在未来几年内有可能恢复升势。尤其是如果Robotaxi的推出成功，极有可能进一步增强市场信心。随着更多国家和地区对电动车推广的政策支持，特斯拉在全球市场的表现或将复苏，并推动价格逐渐回升。## 特斯拉股票值得投资吗？\n\n毫无疑问特斯拉是一只好股票，身边的朋友也有从Tesla上面赚到很多钱的。特斯拉作为全球第一个采用锂离子电池的电动车制造商，业务范围涵盖了新能源汽车的生产和销售，以及充电基础设施的建设和运营。以下我们总结了 TSLA 股票的关键特征，大家需要理性分析是否投资。1. 产业领导地位：不只电动车，更是科技巨头 [...] Tesla引领着新能源和智能电动车技术，Tesla不仅涉足电动车还有太阳能板，电池业务，这几年也涉足加密货币领域。特斯拉电动汽车日益发展迅速是引领其股票价值攀升的主要原因。相较于10年前，Tesla股价如今累计涨幅已经有百倍以上。当今，欧美中国都在大力发展新能源，这是未来的大势所趋。#### 第二，特斯拉创始人马斯克是一个有远见的人\n\n马斯克为公司制定的战略完备，执行力强，公司发展步步为营，成绩斐然。早在2015年，特斯拉创始人就订下每年维持双位数成长，2025年追赶苹果市值的战略。随着特斯拉的成长，营收逐年增加，特斯拉从市场占有到公司形象都稳健向上。#### 第三，特斯拉产品得到市场认可，销量逐年攀升\n\n虽然特斯拉推出的车型并不多，但是每一个都是热销款。特斯拉在上海的工厂在2021年交付60万辆，并且每个季度都在增长。瑞银分析师认为，特斯拉在2022年会仍然主导该市场，预计，2022年特斯拉汽车销量会达到140万辆，随之利润也会更高。也就是说，在特斯拉股价上涨的背后，其产品也是在逐年获得更多消费者认可，企业销售额持续增长。#### 第四，特斯拉市值得到专业认可', 'score': 0.65835214}, {'title': 'TSLA股價— 特斯拉圖表 - TradingView', 'url': 'https://tw.tradingview.com/symbols/NASDAQ-TSLA/', 'content': '1.3213.71 1.49Zm-9 16.2a2.1 2.1 0 0 0 1.95-1.321-3.71-1.49A1.9 1.9 0 0 1 25.93 19v4Zm-6.78-4a1.9 1.9 0 0 1 1.76 2.61-3.71-1.48A2.1 2.1 0 0 0 19.15 23v-4Zm-5.25 4a2.1 2.1 0 0 0 2.1-2.1h-4c0-1.05.85-1.9 1.9-1.9v4ZM6 20.9c0 1.16.94 2.1 2.1 2.1v-4c1.05 0 1.9.85 1.9 1.9H6ZM7.9 14A1.9 1.9 0 0 1 6 12.1h4A2.1 2.1 0 0 0 7.9 10v4ZM0 11.9C0 13.06.94 14 2.1 14v-4c1.05 0 1.9.85 1.9 1.9H0ZM13.9 8A1.9 1.9 0 0 1 12 6.1h4A2.1 2.1 0 0 0 13.9 4v4ZM2.1 4A2.1 2.1 0 0 0 6.1h4A1.9 1.9 0 0 1 2.1 8V4Z" [...] 2.89v5.67ZM82.93 21a.1.1 0 0 0 .1-.0615.92-14.8a.1.1 0 0



0-1-14h-3.28a.1.1 0 0 0-.1.06l-3.88 10.2a.1.1 0 0 1-.19.01L77.03 6.06a.1.1 0 0 0-.1-.06h-3.28a.1.1 0 0 0-.09.14l6.41 14.8a.1.1 0 0 0 .1.06h2.86ZM119.47 20.93a.1.1 0 0 1-.1.07h-2.77a.1.1 0 0 1-.1-.07l-1.9-5.88c-.03-.1-.16-.1-.2 0l-1.88 5.88a.1.1 0 0 1-.1.07h-2.78a.1.1 0 0 1-.09-.07l-3.5-10.8a.1.1 0 0 1 .09-.13h2.93a.1.1 0 0 1 .1.07l1.92 6.65c.03.1.16.1.2 0l2.05-6.65a.1.1 0 0 1 .1-.07h2.1a.1.1 0 0 1 .1.07l2.08 6.65c.02.1.16.1.19 [...] 20.9v-8.8H6v8.8h4Zm3.9-1.9H8.1v4h5.8v-4ZM12 6.1v14.8h4V6.1h-4ZM19.15 23h6.78v-4h-6.78v4Zm3.97-17.68-5.92 14.8 3.71 1.49 5.92-14.8-3.71-1.49ZM31.85 4h-6.78v4h6.78V4Zm-3.97 17.68 5.92-14.8-3.71-1.49-5.92 14.8 3.71 1.49ZM20 9a1 1 0 0 1-1 1v4a5 5 0 0 0 5-5h-4Zm-1-1a1 1 0 0 1 1 1h4a5 5 0 0 0-5-5v4Zm-1 1a1 1 0 0 1 1-1V4a5 5 0 0 0-5 5h4Zm1 1a1 1 0 0 1-1-1h-4a5 5 0 0 0 5 5v-4Zm12.85-2a1.9 1.9 0 0 1-1.76-2.6l3.71 1.48A2.1 2.1 0 0 0 31.85 4v4Zm-5.02-1.2A1.9 1.9 0 0 1 25.07 8V4a2.1 2.1 0 0 0-1.95', 'score': 0.6565047}]

Invoking: **Calculator** with **(362.89-292.03)/292.03\*100**

特斯拉当前股价为362.89美元，去年同期股价为292.03美元。相比去年，特斯拉股价上涨了约24.3%。

Finished chain.

查询结果: {'input': '特斯拉当前股价是多少？比去年上涨了百分之几？', 'output': '特斯拉当前股价为362.89美元，去年同期股价为292.03美元。相比去年，特斯拉股价上涨了约24.3%。'}

注意：也有可能输出失败，但是机会小。

### 案例3：自定义函数与工具

需求：计算3的平方，Agent自动调用工具完成

```
1 from langchain.agents import initialize_agent, AgentType, Tool
2 from langchain_openai import ChatOpenAI
3 import langchain
4
5
6 # 1. 定义工具 - 计算器 (要求字符串输入)
7 def simple_calculator(expression: str) -> str:
8     """
9     基础数学计算工具，支持加减乘除和幂运算
10    参数:
11        expression: 数学表达式字符串，如 "3+5" 或 "2**3"
12    返回:
13        计算结果字符串或错误信息
14    """
15    print(f"\n[工具调用] 计算表达式: {expression}")
16
17    print("只因为在人群中多看了你一眼，确认下你调用了我^_^")
18    return str(eval(expression))
19
20
21 # 2. 创建工具对象
22 math_calculator_tool = Tool(
23     name="Math_Calculator", # 工具名称 (Agent将根据名称选择工具)
24     func=simple_calculator, # 工具调用的函数
25     description="用于数学计算，输入必须是纯数学表达式 (如'3+5'或'3**2'表示平方)。不支持字母或特殊符号" # 关键：明确输入格式要求
26 )
```

```

27
28 # 3. 初始化大模型
29 llm = ChatOpenAI(
30     model="gpt-4o-mini",
31     temperature=0,
32 )
33
34 # 4. 初始化AgentExecutor (使用零样本React模式、增加超时设置)
35 agent_executor = initialize_agent(
36     tools=[math_calculator_tool], # 可用的工具列表
37     llm=llm,
38     agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION, # 简单指令模式
39     verbose=True # 关键参数! 在控制台显示详细的推理过程
40 )
41
42 # 5. 测试工具调用 (添加异常捕获)
43 print("=== 测试: 正常工具调用 ===")
44 response = agent_executor.invoke("计算3的平方") # 向Agent提问
45 print("最终答案:", response)

```

```

1  === 测试: 正常工具调用 ===
2
3
4  > Entering new AgentExecutor chain...
5  我需要计算3的平方，这可以用数学表达式3**2来表示。
6  Action: Math_Calculator
7  Action Input: 3**2
8  [工具调用] 计算表达式: 3**2
9  只因为在人群中多看了你一眼，确认下你调用了我^_^
10
11  Observation: 9
12  > Entering new AgentExecutor chain...
13  我需要计算3的平方，这可以用数学表达式3**2来表示。
14  Action: Math_Calculator
15  Action Input: 3**2
16  [工具调用] 计算表达式: 3**2
17  只因为在人群中多看了你一眼，确认下你调用了我^_^
18
19  Observation: 9
20  Thought:我现在知道最终答案
21  Final Answer: 9
22
23  > Finished chain.
24  最终答案: {'input': '计算3的平方', 'output': '9'}

```

## 3.2 通用方式

需求：今天北京的天气怎么样？？

### 方式1：FUNCATION\_CALL模式

```
1  # 1.导入相关包
2  from langchain.agents import AgentExecutor, create_tool_calling_agent
3  from langchain_community.tools.tavily_search import TavilySearchResults
4  from langchain_core.prompts import ChatPromptTemplate
5
6  # 2.定义搜索化工具
7  # ① 设置 TAVILY_API 密钥
8  os.environ["TAVILY_API_KEY"] = "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" # 需要替换
   为你的 Tavily API 密钥
9
10 # ② 定义搜索工具
11 search = TavilySearchResults(max_results=1)
12
13
14 # 3.自定义提示词模版
15 prompt = ChatPromptTemplate.from_messages([
16     ("system", "您是一位乐于助人的助手，请务必使用 tavily_search_results_json 工具来获取信
   息。"),
17     ("human", "{input}"),
18     ("placeholder", "{agent_scratchpad}"),
19 ])
20
21 # 4.定义LLM
22 llm = ChatOpenAI(
23     model="gpt-4o-mini",
24     temperature=0,
25 )
26
27 # 5.创建Agent对象
28 agent = create_tool_calling_agent(
29     llm = llm,
30     tools = [search],
31     prompt = prompt
32 )
33
34
35 # 6.创建AgentExecutor执行器
36 agent_executor = AgentExecutor(agent=agent, tools=[search], verbose=True)
37
38 # 7.测试
39 agent_executor.invoke({"input": "今天北京的天气怎么样? ?"})
```

注意：agent\_scratchpad必须声明，它用于存储和传递Agent的思考过程。比如，在调用链式工具时（如先搜索天气再推荐行程），`agent_scratchpad` 保留所有历史步骤，避免上下文丢失。format方法会将intermediate\_steps转换为特定格式的字符串，并赋值给agent\_scratchpad变量。如果不传递intermediate\_steps参数，会导致KeyError: 'intermediate\_steps'错误。

```
1 > Entering new AgentExecutor chain...
2
3 Invoking: `tavily_search_results_json` with `{'query': '北京天气'}`
4
5
6 > Entering new AgentExecutor chain...
7
8 Invoking: `tavily_search_results_json` with `{'query': '北京天气'}`
9
10
11 > Entering new AgentExecutor chain...
12
13 Invoking: `tavily_search_results_json` with `{'query': '北京天气'}`
14
15
16 [{ 'title': '北京-天气预报 - 中央气象台', 'url':
    'https://www.nmc.cn/publish/forecast/ABJ/beijing.html', 'content': '土壤水分监测\n 农
    业干旱综合监测\n 关键农时农事\n 农业气象周报\n 农业气象月报\n 农业气象专
    报\n 生态气象监测评估\n 作物发育期监测\n\n 数值预报\n\n CMA全球天气模式\n
    CMA全球集合模式\n CMA区域模式\n CMA区域集合模式\n CMA台风模式\n
    海浪模式\n\n1. 当前位置: 首页\n2. 北京市\n3. 北京天气预报\n\n省份: 城市: \n\n09:50
    更新\n\n日出04:45\n\n 北京 \n\n30°C\n\n日落19:43\n\n 降水量 \n\n0mm\n\n西南风\n\n3
    级\n\n相对湿度 \n\n43%\n\n 体感温度 \n\n29.9°C\n\n空气质量: 良 \n\n舒适度: 温暖, 较
    舒适\n\n 雷达图 \n\nImage 4\n\n24小时预报7天预报10天预报11-30天预报\n\n 发布时间:
    06-12 08:00 \n\n 06/12 \n\n周四 \n\nImage 5\n\n 多云 \n\n 南风 \n\n 3~4级 \n\n 35°C
    \n\n 23°C \n\nImage 6', 'score': 0.78493005}]今天北京的天气情况如下:
17
18 - 当前温度: 30°C
19 - 天气状况: 多云
20 - 风速: 西南风, 3级
21 - 相对湿度: 43%
22 - 体感温度: 29.9°C
23 - 空气质量: 良
24 - 舒适度: 温暖, 较舒适
25
26 日出时间为04:45, 日落时间为19:43。预计降水量为0mm。
27
28 如果需要更详细的信息, 可以查看[中央气象台的天气预报]
    (https://www.nmc.cn/publish/forecast/ABJ/beijing.html)。
29
30 > Finished chain.
31
32 {'input': '今天北京的天气怎么样? ',
33  'output': '今天北京的天气情况如下: \n\n- 当前温度: 30°C\n- 天气状况: 多云\n- 风速: 西南
    风, 3级\n- 相对湿度: 43%\n- 体感温度: 29.9°C\n- 空气质量: 良\n- 舒适度: 温暖, 较舒适
    \n\n日出时间为04:45, 日落时间为19:43。预计降水量为0mm。 \n\n如果需要更详细的信息, 可
    以查看[中央气象台的天气预报](https://www.nmc.cn/publish/forecast/ABJ/beijing.html)。'}
```

## 方式2：ReAct模式

### 体会1：使用PromptTemplate

提示词要体现可以使用的工具、用户输入和agent\_scratchpad。

远程的提示词模版通过<https://smith.langchain.com/hub/hwchase17>获取

- 举例：<https://smith.langchain.com/hub/hwchase17/react>，这个模板是专为ReAct模式设计的提示模板。这个模板中已经有聊天对话键 `tools`、`tool_names`、`agent_scratchpad`

方式1：

```
1  # 1.导入相关包
2  from langchain.agents import AgentExecutor, create_react_agent
3  from langchain_community.tools.tavily_search import TavilySearchResults
4  from langchain_core.prompts import PromptTemplate
5
6  # 2.定义搜索化工具
7  tools = [TavilySearchResults(max_results=1,tavily_api_key= "tvly-dev-
8  T9z5UN2xmiw6XlruXnH2JXbYFZf12JYd")]
9
10 # 3.自定义提示词模版
11 template = "Answer the following questions as best you can. You have access to the following
12 tools:
13
14 {tools}
15
16 Use the following format:
17
18 Question: the input question you must answer
19 Thought: you should always think about what to do
20 Action: the action to take, should be one of [{tool_names}]
21 Action Input: the input to the action
22 Observation: the result of the action
23 ... (this Thought/Action/Action Input/Observation can repeat N times)
24 Thought: I now know the final answer
25 Final Answer: the final answer to the original input question
26
27 Begin!
28
29 Question: {input}
30 Thought:{agent_scratchpad}"
31
32 # template1 = "
33 # 请尽可能准确地回答以下问题。您可以使用以下工具: {tools}
34 # 请使用以下格式:
35 # 问题: 您必须回答的输入问题
36 # 思考: 您应该始终思考要做什么
37 # 行动: 要采取的行动, 应为 [{tool_names}] 中的一个
38 # 行动输入: 行动的输入
39 # 观察: 行动的结果
40 # ..... (此思考/行动/行动输入/观察可重复 N 次)
41 # 思考: 我现在知道最终答案了
42 # 最终答案: 对原始输入问题的最终答案
```



```

40 # 开始!
41 # 问题: {question}
42 # 思考: {agent_scratchpad}
43 # ""
44
45 prompt = PromptTemplate.from_template(template)
46
47 # 4.定义LLM
48 llm = ChatOpenAI(
49     model="gpt-4o-mini",
50     temperature=0,
51 )
52
53 # 5.创建Agent对象
54 agent = create_react_agent(llm, tools, prompt)
55
56 # 6.创建AgentExecutor执行器
57 agent_executor = AgentExecutor(agent=agent, tools=tools,
58     verbose=True,handle_parsing_errors=True)
59
60 # 7.测试
61 agent_executor.invoke({"input": "今天北京的天气怎么样? ?"})

```

```

1  > Entering new AgentExecutor chain...
2  我需要查找今天北京的天气情况。
3  Action: tavily_search_results_json
4  > Entering new AgentExecutor chain...
5  我需要查找今天北京的天气情况。
6  Action: tavily_search_results_json
7  > Entering new AgentExecutor chain...
8  我需要查找今天北京的天气情况。
9  Action: tavily_search_results_json
10 Action Input: "今天 北京 天气" [{'title': '中国气象局-天气预报- 北京', 'url':
    'https://weather.cma.cn/web/weather/54511', 'content': '| | | | | | | | \n| --- | --- | --- |
    --- | --- | --- | --- | --- | --- \n| 时间 | 08:00 | 11:00 | 14:00 | 17:00 | 20:00 | 23:00 | 02:00 |
    05:00 \n| 天气 | | | | | | | | \n| 气温 | 21.4°C | 23.6°C | 26.8°C | 29.8°C | 24.4°C | 23.3°C |
    21.6°C | 20.2°C \n| 降水 | 无降水 | 无降水 | 无降水 | 无降水 | 无降水 | 无降水 | 无降水 | 无降水
    \n| 风速 | 2.9m/s | 6.8m/s | 7.5m/s | 7.9m/s | 7.6m/s | 2.9m/s | 2.7m/s | 3.2m/s \n| 风向 | 东
    北风 | 东南风 | 西南风 | 西北风 | 西北风 | 西北风 | 东北风 | 西北风 | [...] | | | | | | | | \n| ---
    | --- | --- | --- | --- | --- | --- | --- | --- \n| 时间 | 02:00 | 05:00 | 08:00 | 11:00 | 14:00 | 17:00 |
    20:00 | 23:00 \n| 天气 | | | | | | | | \n| 气温 | 23.6°C | 22.2°C | 24°C | 26.8°C | 28.8°C |
    27.3°C | 24.7°C | 24.1°C \n| 降水 | 无降水 | 无降水 | 无降水 | 无降水 | 无降水 | 无降水 | 无降水 |
    无降水 \n| 风速 | 2.9m/s | 3m/s | 3m/s | 3.1m/s | 3.3m/s | 3.1m/s | 2.5m/s | 2.6m/s \n| 风向 |
    东南风 | 东南风 | 东北风 | 东北风 | 东北风 | 东北风 | 东北风 | 东北风 | [...] | | | | | | | | \n|
    --- | --- | --- | --- | --- | --- | --- | --- \n| 时间 | 08:00 | 11:00 | 14:00 | 17:00 | 20:00 |
    23:00 | 02:00 | 05:00 \n| 天气 | | | | | | | | \n| 气温 | 23.7°C | 27°C | 30.2°C | 30.8°C |
    28.1°C | 26.5°C | 21.1°C | 19.2°C \n| 降水 | 无降水 | 无降水 | 无降水 | 无降水 | 无降水 | 无降水 |
    无降水 | 无降水 \n| 风速 | 3.3m/s | 7.4m/s | 7.3m/s | 7m/s | 7.9m/s | 3.3m/s | 2.2m/s |
    1.8m/s \n| 风向 | 西南风 | 西北风 | 西北风 | 西北风 | 东北风 | 西北风 | 西北风 | 西北风 |',
    'score': 0.7859175}]我现在知道今天北京的天气情况。
11 Final Answer: 今天北京的天气是晴天, 气温在21.4°C到30.8°C之间, 全天无降水。
12
13

```

```
14 {'input': '今天北京的天气怎么样? ', 'output': '今天北京的天气是晴天, 气温在21.4°C到30.8°C  
    之间, 全天无降水。'}
```

方式2:

```
1  # 1.导入相关包
2  from langchain.agents import AgentExecutor, create_react_agent
3  from langchain_community.tools.tavily_search import TavilySearchResults
4  from langchain_core.prompts import PromptTemplate
5
6  # 2.定义搜索化工具
7  tools = [TavilySearchResults(max_results=1,tavily_api_key= "tvly-dev-
    T9z5UN2xmiw6XlruXnH2JXbYFZf12JYd")]
8
9  # 3.使用LangChain Hub中的官方ReAct提示模板
10 prompt = hub.pull("hwchase17/react")
11
12 # 4.定义LLM
13 llm = ChatOpenAI(
14     model="gpt-4o-mini",
15     temperature=0,
16 )
17
18 # 5.创建Agent对象
19 agent = create_react_agent(llm, tools, prompt)
20
21 # 6.创建AgentExecutor执行器
22 agent_executor = AgentExecutor(agent=agent, tools=tools,
    verbose=True,handle_parsing_errors=True)
23
24 # 7.测试
25 agent_executor.invoke({"input": "今天北京的天气怎么样? ?"})
```

## 体会2：使用ChatPromptTemplate

提示词中需要体现使用的工具、用户输入和agent\_scratchpad。

```
1  from langchain.agents import create_react_agent
2  from langchain_core.prompts import ChatPromptTemplate
3  # 获取Tavily搜索的实例
4  from langchain_openai import ChatOpenAI
5  from langchain.agents import initialize_agent, AgentType, create_tool_calling_agent,
    AgentExecutor
6  from langchain.tools import Tool
7  import os
8  import dotenv
9  from langchain_openai import ChatOpenAI
10 from langchain_community.tools.tavily_search import TavilySearchResults
11 dotenv.load_dotenv()
12
13 # 读取配置文件的信息
```

```

14 os.environ['TAVILY_API_KEY'] = "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"
15
16 # 获取Tavily搜索工具的实例
17 search = TavilySearchResults(max_results=3)
18
19 # 获取一个搜索的工具
20 # 使用Tool
21 search_tool = Tool(
22     func=search.run,
23     name="Search",
24     description="用于检索互联网上的信息，尤其是天气情况",
25 )
26
27
28 # 获取大语言模型
29 os.environ['OPENAI_API_KEY'] = os.getenv("OPENAI_API_KEY1")
30 os.environ['OPENAI_BASE_URL'] = os.getenv("OPENAI_BASE_URL")
31 llm = ChatOpenAI(
32     model="gpt-4o-mini",
33     temperature=0,
34 )
35
36 # 提供提示词模板 (以ChatPromptTemplate为例)
37 # prompt_template = ChatPromptTemplate.from_messages([
38 #     ("system", "你是一一个人工智能的助手，在用户提出需求以后，必须要调用Search工具进行联网搜索"),
39 #     ("system", ""这是你可以使用的工具列表{tools}，每次Action时调用{tool_names}，并根据工具输出的网络结果回答用户的问题。"),
40 #     # 工具调用格式 (必须严格遵守) :
41 #     # Thought: 分析需要做什么 (例如: 需要查询明天北京的天气)
42 #     # Action: 工具名称 (只能是[{tool_names}]中的一个)
43 #     # ActionInput: 工具的输入内容 (例如: 北京明天天气)
44 #     # 若工具返回内容:
45 #     # Observation: 工具返回的结果
46 #     # 当我知道了足够的信息之后我应该即刻输出:
47 #     # Thought: 我知道答案了
48 #     # FinalAnswer: 经过整理后的最终答案! """),
49 #     ("system", "{agent_scratchpad}"),
50 #     ("human", "我的问题是: {question}"),
51 # ])
52 # ])
53
54 prompt_template = ChatPromptTemplate.from_messages([
55     ("system", "你是一一个人工智能的助手，在用户提出需求以后，必须要调用Search工具进行联网搜索"),
56     ("system", ""Answer the following questions as best you can. You have access to the following tools:
57         {tools}
58
59         Use the following format:
60
61         Question: the input question you must answer
62         Thought: you should always think about what to do
63         Action: the action to take, should be one of [{tool_names}]
64         Action Input: the input to the action

```

```

65 Observation: the result of the action
66 ... (this Thought/Action/Action Input/Observation can repeat N times)
67 Thought: I now know the final answer
68 Final Answer: the final answer to the original input question
69
70 Begin!
71 执行过程建议使用中文
72 ""),
73 ("system", "当前思考: {agent_scratchpad}"),
74 ("human", "我的问题是: {question}"), # 必须在提示词模板中提供agent_scratchpad参数。
75 )
76
77
78 # 获取Agent的实例: create_tool_calling_agent()
79 agent = create_react_agent(
80     llm=llm,
81     prompt=prompt_template,
82     tools=[search_tool]
83 )
84
85 # 获取AgentExecutor的实例
86 agent_executor = AgentExecutor(
87     agent=agent,
88     tools=[search_tool],
89     verbose=True,
90     handle_parsing_errors=True,
91     max_iterations=6 # 可选: 限制最大迭代次数, 防止无限循环
92 )
93
94
95 # 通过AgentExecutor的实例调用invoke(),得到响应
96 result = agent_executor.invoke({ "question": "查询今天北京的天气情况" })
97
98 # 处理响应
99 print(result)
100

```

执行结果：

[illegible]

上述执行可能会报错。

**错误原因：**

- 使用ReAct模式时，要求 LLM 的响应必须遵循严格的格式（如包含 **Thought:**、**Action:** 等标记）。
- 但 LLM 直接返回了自由文本（非结构化），导致解析器无法识别。

修改：

- 任务不变，添加 `handle_parsing_errors=True` 。用于控制 Agent 在解析工具调用或输出时发生错误的容错行为。

### handle\_parsing\_errors=True 的作用

- 自动捕获错误并修复：当解析失败时，Agent 不会直接崩溃，而是将错误信息传递给 LLM，让 LLM **自行修正并重试** 。
- 降级处理：如果重试后仍失败，Agent 会返回一个友好的错误消息（如 `"I couldn't process that request."` ），而不是抛出异常。

小结：

| 场景   | handle_parsing_errors=True | handle_parsing_errors=False |
|------|----------------------------|-----------------------------|
| 解析成功 | 正常执行                       | 正常执行                        |
| 解析失败 | 自动修复或降级响应                  | 直接抛出异常                      |
| 适用场景 | 生产环境（保证鲁棒性）                | 开发调试（快速发现问题）                |

## 4、Agent嵌入记忆组件

### 4.1 传统方式

比如：北京明天的天气怎么样？上海呢？（通过两次对话实现）

举例：以REACT模式为例

```
1  # 1. 导入依赖包
2  from langchain_openai import ChatOpenAI
3  from langchain.agents import initialize_agent, AgentType
4  from langchain_community.tools.tavily_search import TavilySearchResults
5  from langchain.memory import ConversationBufferMemory
6  import os
7  import dotenv
8
9  dotenv.load_dotenv()
10
11 # 2. 设置 API 密钥
12 os.environ["TAVILY_API_KEY"] = "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"
13
14 # 3. 定义搜索工具
15 search_tool = TavilySearchResults(max_results=2)
16
17 # 4. 定义 LLM
18 os.environ["OPENAI_API_KEY"] = os.getenv("OPENAI_API_KEY1")
19 os.environ["OPENAI_BASE_URL"] = os.getenv("OPENAI_BASE_URL")
20 llm = ChatOpenAI(
21     model="gpt-4o-mini",
22     temperature=0
23 )
24
```

```

25 # 5. 定义记忆组件(以ConversationBufferMemory为例)
26 memory = ConversationBufferMemory(
27     memory_key= "chat_history", #必须是此值, 通过initialize_agent()的源码追踪得到
28     return_messages=True
29 )
30
31 # 6. 创建 AgentExecutor
32 agent_executor = initialize_agent(
33     tools=[search_tool],
34     llm=llm,
35     agent=AgentType.CONVERSATIONAL_REACT_DESCRIPTION,
36     memory=memory, #在AgentExecutor中声明
37     verbose=True
38 )
39
40
41 # 7. 测试对话
42 # 第一个查询
43 query1= "北京明天的天气怎么样? "
44 result1 = agent_executor.invoke(query1)
45 print(f"查询结果: {result1}")
46
47 # print("\n=== 继续对话 ===")
48 query2= "上海呢"
49 result2=agent_executor.invoke(query2)
50 print(f"分析结果: {result2}")

```

上述执行可能会报错。

#### 错误原因：

- 使用 ReAct模式时，要求 LLM 的响应必须遵循严格的格式（如包含 **Thought:**、**Action:** 等标记）。
- 但 LLM 直接返回了自由文本（非结构化），导致解析器无法识别。

#### 修改：

- 任务不变，添加 **handle\_parsing\_errors=True**。用于控制 Agent 在解析工具调用或输出时发生错误的容错行为。

#### handle\_parsing\_errors=True 的作用

- 自动捕获错误并修复：当解析失败时，Agent 不会直接崩溃，而是将错误信息传递给 LLM，让 LLM **自行修正并重试**。
- 降级处理：如果重试后仍失败，Agent 会返回一个友好的错误消息（如 **"I couldn't process that request."**），而不是抛出异常。

#### 小结：

| 场景   | handle_parsing_errors=True | handle_parsing_errors=False |
|------|----------------------------|-----------------------------|
| 解析成功 | 正常执行                       | 正常执行                        |
| 解析失败 | 自动修复或降级响应                  | 直接抛出异常                      |
| 适用场景 | 生产环境（保证鲁棒性）                | 开发调试（快速发现问题）                |

代码修改为：

```
1  # 1. 导入依赖包
2  from langchain_openai import ChatOpenAI
3  from langchain.agents import initialize_agent, AgentType
4  from langchain.tools import Tool
5  from langchain_community.tools.tavily_search import TavilySearchResults
6  from langchain_experimental.utilities.python import PythonREPL
7  from langchain.memory import ConversationBufferMemory
8
9  # 2. 设置 API 密钥
10 os.environ["TAVILY_API_KEY"] = "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"
11
12 # 3. 定义搜索工具
13 search_tool = TavilySearchResults(max_results=2)
14
15 # 5. 定义LLM
16 llm = ChatOpenAI(
17     model="gpt-4",
18     temperature=0
19 )
20
21 # 6. 定义记忆组件(以ConversationBufferMemory为例)
22 memory = ConversationBufferMemory(
23     memory_key="chat_history",
24     return_messages=True
25 )
26
27 # 7. 创建 AgentExecutor
28 agent_executor = initialize_agent(
29     tools=[search_tool],
30     llm=llm,
31     agent=AgentType.CONVERSATIONAL_REACT_DESCRIPTION,
32     memory=memory,
33     handle_parsing_errors=True,
34     verbose=True
35 )
36
37
38 # 8. 测试对话
39 # 第一个查询
40 query1 = "北京明天的天气怎么样? "
41 result1 = agent_executor.invoke(query1)
42 print(f"查询结果: {result1}")
43
44 # print("\n=== 继续对话 ===")
45 query2 = "上海呢"
46 result2 = agent_executor.invoke(query2)
47 print(f"分析结果: {result2}")
```

```
1  > Entering new AgentExecutor chain...
2  ...
```





```
15  查询结果: {'input': '北京明天的天气怎么样? ', 'chat_history': [HumanMessage(content='北京
    明天的天气怎么样? ', additional_kwargs={}, response_metadata={}),
    AIMessage(content='北京明天的天气预报显示, 天气将是晴转多云, 最高气温约为33°C, 最低
    气温约为20°C。风速在3级以下。', additional_kwargs={}, response_metadata={})], 'output':
    '北京明天的天气预报显示, 天气将是晴转多云, 最高气温约为33°C, 最低气温约为20°C。风速在
    3级以下。'}
16
17
18  > Entering new AgentExecutor chain...
19  ```
20  Thought: Do I need to use a tool? Yes
21  Action: tavily_search_results_json
22  Action Input: 北京 明天的天气
23  ```
24  Observation: [{'title': '中国气象局-天气预报- 北京', 'url':
25
26  > Finished chain.
27  查询结果: {'input': '北京明天的天气怎么样? ', 'chat_history': [HumanMessage(content='北京
    明天的天气怎么样? ', additional_kwargs={}, response_metadata={}),
    AIMessage(content='北京明天的天气预报显示, 天气将是晴转多云, 最高气温约为33°C, 最低
    气温约为20°C。风速在3级以下。', additional_kwargs={}, response_metadata={})], 'output':
    '北京明天的天气预报显示, 天气将是晴转多云, 最高气温约为33°C, 最低气温约为20°C。风速在
    3级以下。'}
28
29
30  > Entering new AgentExecutor chain...
31  ```
32  Thought: Do I need to use a tool? Yes
33  Action: tavily_search_results_json
34  Action Input: 上海明天的天气
35  ```
```

```
36 Observation: [{'title': '中国气象局-天气预报-城市预报- 上海', 'url':  
'https://weather.cma.cn/web/weather/58367.html', 'content': '| | | | | | | | | | \n| --- | ---  
| --- | --- | --- | --- | --- | --- | --- | --- \n| æ\x97\x97\x97 | 14:00 | 17:00 | 20:00 | 23:00 | 02:00 |  
05:00 | 08:00 | 11:00 \n| å©æ°\x94 | | | | | | | | \n| æ°\x94æ,© | 32.3â\x84\x83 |  
28.5â\x84\x83 | 25.9â\x84\x83 | 23.7â\x84\x83 | 23â\x84\x83 | 22.6â\x84\x83 |  
25.9â\x84\x83 | 30.8â\x84\x83 \n| é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97  
é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97  
é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97 é\x99\x8dæ° \n| é\x8eé\x80\x9f | 7.9m/s |  
7.4m/s | 7.4m/s | 7.8m/s | 7.7m/s | 7.5m/s | 7.9m/s | 7.9m/s | [...] | | | | | | | | \n| --- | --  
- | --- | --- | --- | --- | --- | --- | --- \n| æ\x97\x97\x97 | 08:00 | 11:00 | 14:00 | 17:00 | 20:00 |  
23:00 | 02:00 | 05:00 \n| å©æ°\x94 | | | | | | | | \n| æ°\x94æ,© | 25.9â\x84\x83 |  
30.8â\x84\x83 | 31.5â\x84\x83 | 30.3â\x84\x83 | 28.1â\x84\x83 | 25.2â\x84\x83 |  
23.7â\x84\x83 | 24.1â\x84\x83 \n| é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97  
é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97  
é\x99\x8dæ° | æ\x97 é\x99\x8dæ° | æ\x97 é\x99\x8dæ° \n| é\x8eé\x80\x9f | 7.9m/s |  
7.9m/s | 7.7m/s | 7.5m/s | 7.5m/s | 7.9m/s | 7.7m/s | 7.5m/s | [...] | | | | | | | | \n| --- | --  
- | --- | --- | --- | --- | --- | --- | --- \n| æ\x97\x97\x97 | 08:00 | 11:00 | 14:00 | 17:00 | 20:00 |  
23:00 | 02:00 | 05:00 \n| å©æ°\x94 | | | | | | | | \n| æ°\x94æ,© | 25â\x84\x83 |  
25.9â\x84\x83 | 26.4â\x84\x83 | 25.5â\x84\x83 | 24.8â\x84\x83 | 24â\x84\x83 |  
23.8â\x84\x83 | 23.6â\x84\x83 \n| é\x99\x8dæ° | 0.6mm | 1.6mm | 2.4mm | 1.6mm |  
4.3mm | 3mm | 6.9mm | 7mm \n| é\x8eé\x80\x9f | 7.6m/s | 7.6m/s | 7.8m/s | 7.9m/s |  
7.7m/s | 7.8m/s | 7.7m/s | 7.8m/s |', 'score': 0.7192461}, {'title': '【上海天气预报】上海天气预  
报7天,10天,15天 - 全国天气网', 'url': 'https://tianqi.so.com/weather/101020100', 'content':  
'# 全国天气网\n\n45日天气\n\n15日天气\n\n\n### 当前天气信息\n\n天气: 晴 34°C\n\n\n体感:  
33°C\n\n\n风向: 北风\n\n\n风力: 2级\n\n\n气压: 1007\n\n\n湿度: 45%\n\n\n日出: 04:50\n\n\n日  
落: 19:00\n\n\n### 空气质量\n\n\n良\n\n\n无需戴口罩\n\n\n\n适宜外出\n\n\n\n适宜开窗\n\n\n\n关闭净化器  
\n\n\n### 主要污染物\n\n\nPM2.5\n\n\n21优\n\n\nPM10\n\n\n39优\n\n\nO3\n\n\n166轻度  
\n\n\nNO2\n\n\n14优\n\n\nSO2\n\n\n7优\n\n\n\n14时\n\n\n17时\n\n\n20时\n\n\n23时\n\n\n\n02时\n\n\n05时  
\n\n\n08时\n\n\n11时\n\n\n\n北风\n\n\n2级\n\n\n东南风\n\n\n微风\n\n\n东南风\n\n\n\n微风\n\n\n\n南风\n\n\n\n微风  
\n\n\n南风\n\n\n\n微风\n\n\n\n南风\n\n\n\n微风\n\n\n\n南风\n\n\n\n微风\n\n\n\n3-5级\n\n\n\n### 明日天气信  
息\n\n\n天气: 小雨\n\n\n温度: 26~30°C\n\n\n风向: 东南风\n\n\n风力: 3-5级\n\n\n\n### 最优空气质  
量排行榜\n\n\n查看全国最优排行榜>>\n\n\n### 最差空气质量排行榜\n\n\n查看全国最差排行  
榜>>\n\n\n02时\n\n\n05时\n\n\n08时\n\n\n11时\n\n\n14时\n\n\n17时\n\n\n20时\n\n\n23时\n\n\n\n南风\n\n\n\n微  
风\n\n\n\n南风\n\n\n\n微风\n\n\n\n南风\n\n\n\n微风\n\n\n\n南风\n\n\n\n3-5级\n\n\n\n东南风', 'score': 0.6680367}]
```

```
37 Thought:No  
38 AI: 上海明天的天气预报显示, 天气将是小雨, 气温在26°C到30°C之间, 风向为东南风, 风力为3-  
5级。  
39  
40 > Finished chain.  
41 分析结果: {'input': '上海呢', 'chat_history': [HumanMessage(content='北京明天的天气怎么  
样?', additional_kwargs={}, response_metadata={}), AIMessage(content='北京明天的天气  
预报显示, 天气将是晴转多云, 最高气温约为33°C, 最低气温约为20°C。风速在3级以下。',  
additional_kwargs={}, response_metadata={}), HumanMessage(content='上海呢',  
additional_kwargs={}, response_metadata={}), AIMessage(content='上海明天的天气预报显  
示, 天气将是小雨, 气温在26°C到30°C之间, 风向为东南风, 风力为3-5级。',  
additional_kwargs={}, response_metadata={})], 'output': '上海明天的天气预报显示, 天气将是  
小雨, 气温在26°C到30°C之间, 风向为东南风, 风力为3-5级。'}
```

再例:

- 1 # ... 前面的代码省略 ...
- 2 # 8. 测试对话
- 3 # 第一个查询

```

4  print("=== 查询昨天的新闻 ===")
5  result1 = agent_executor.invoke("昨天的新闻头条是什么? ")
6  print(f"查询结果: {result1}")
7
8  # 第二个查询 (基于上下文)
9  print("\n=== 用中文总结 ===")
10 result2 = agent_executor.invoke("用中文总结一下")
11 print(f"总结结果: {result2}")
12
13 # 可以继续对话
14 print("\n=== 继续对话 ===")
15 result3 = agent_executor.invoke("这些新闻对股市有什么影响? ")
16 print(f"分析结果: {result3}")

```

## 4.2 通用方式

通用方式，相较于传统方式，可以提供自定义的提示词模板

### 举例1：FUNCTION\_CALL模式

如果使用的是FUNCTION\_CALL方式，则创建Agent时，推荐使用ChatPromptTemplate

```

1  # 1. 导入依赖包
2  from langchain_openai import ChatOpenAI
3  from langchain.agents import initialize_agent, AgentType
4  from langchain.tools import Tool
5  from langchain_community.tools.tavily_search import TavilySearchResults
6  from langchain_experimental.utilities.python import PythonREPL
7  from langchain.memory import ConversationBufferMemory
8
9  # 2. 定义 TAVILY_KEY 密钥
10 os.environ["TAVILY_API_KEY"] = "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"
11
12 # 3. 定义搜索工具
13 # search = TavilySearchResults(max_results=2)
14
15 # search_tool = Tool(
16 #     name="search_tool",
17 #     func=search.run,
18 #     description="用于互联网信息的检索"
19 # )
20 # )
21 # tools = [search_tool]
22 #或者
23 search = TavilySearchResults(max_results=2)
24 tools = [search]
25
26 # 4. 定义LLM
27 llm = ChatOpenAI(
28     model="gpt-4",
29     temperature=0
30 )

```

```

31
32 # 5. 定义提示词模板
33 prompt = ChatPromptTemplate.from_messages([
34     ("system", "你是一个有用的助手，可以回答问题并使用工具。"),
35     ("placeholder", "{chat_history}"), # 存储多轮对话的历史记录 如果你没有显式传入 chat_history,
    Agent 会默认将其视为空列表 []
36     ("human", "{input}"),
37     ("placeholder", "{agent_scratchpad}")
38 ])
39
40 # 6. 定义记忆组件(以ConversationBufferMemory为例)
41 memory = ConversationBufferMemory(
42     memory_key="chat_history",
43     return_messages=True
44 )
45
46 # 7. 创建Agent对象
47 agent = create_tool_calling_agent(llm, tools, prompt)
48
49
50 # 8. 创建AgentExecutor执行器对象(通过源码可知, memory参数声明在AgentExecutor父类中)
51 agent_executor = AgentExecutor(agent=agent, memory=memory, tools=tools, verbose=True)
52
53
54 # 9. 测试对话
55 # 第一个查询
56 result1 = agent_executor.invoke({"input": "北京的天气是多少"})
57 print(f"查询结果: {result1}")
58
59 # print("\n=== 继续对话 ===")
60 result2 = agent_executor.invoke({"input": "上海呢"})
61 print(f"分析结果: {result2}")

```

```

1  > Entering new AgentExecutor chain...
2
3  Invoking: `tavily_search_results_json` with `{'query': '北京明天的天气预报'}`
4
5
6  > Entering new AgentExecutor chain...
7
8  Invoking: `tavily_search_results_json` with `{'query': '北京明天的天气预报'}`
9
10
11 > Entering new AgentExecutor chain...
12
13 Invoking: `tavily_search_results_json` with `{'query': '北京明天的天气预报'}`
14
15

```













```
分析结果: {'input': '上海呢', 'chat_history': [HumanMessage(content='北京的天气是多少', additional_kwargs={}, response_metadata={}), AIMessage(content='北京当前的气温是30°C, 体感温度为29.9°C, 相对湿度为43%, 空气质量良好, 天气较舒适。风向为西南风, 风力3级。', additional_kwargs={}, response_metadata={}), HumanMessage(content='上海呢', additional_kwargs={}, response_metadata={}), AIMessage(content='你是想了解上海的什么信息呢? 比如天气、旅游景点、文化活动, 还是其他方面的内容? 请具体说明一下, 我会尽力帮你解答! ', additional_kwargs={}, response_metadata={})], 'output': '你是想了解上海的什么信息呢? 比如天气、旅游景点、文化活动, 还是其他方面的内容? 请具体说明一下, 我会尽力帮你解答! '}
```

拓展2: 如果删除AgentExecutor构造方法中的memory参数, 又会如何呢?

```
查询结果: {'input': '北京的天气是多少', 'output': '北京当前的气温为30°C, 体感温度为29.9°C, 湿度为43%, 空气质量良好, 天气较舒适。风向为西南风, 风力3级。'}
```

```
分析结果: {'input': '上海呢', 'output': '你是想了解上海的什么信息呢? 比如天气、旅游景点、文化活动、还是其他方面的内容? 请具体说明一下, 我会尽力帮你解答! '}
```

## 举例2: ReAct 模式

ReAct模式下, 创建Agent时, 可以使用ChatPromptTemplate、PromptTemplate

```
1  # 1.导入相关包
2  from langchain.agents import AgentExecutor, create_react_agent
3  from langchain_community.tools.tavily_search import TavilySearchResults
4  from langchain_core.prompts import PromptTemplate
5  import os
6
7  # 2.定义搜索化工具
8  # ① 设置 TAVILY_API 密钥
9  os.environ["TAVILY_API_KEY"] = "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" # 需要替换为你的 Tavily API 密钥
10 # ② 定义搜索工具
11 search = TavilySearchResults(max_results=1)
12 # ③ 设置工具集
13 tools = [search]
14
15 # 3.自定义提示词模版
16 template = ("Assistant is a large language model trained by OpenAI.\n\n"
17             "Assistant is designed to be able to assist with a wide range of tasks, from answering simple questions to providing in-depth explanations and discussions on a wide range of topics. As a language model, Assistant is able to generate human-like text based on the input it receives, allowing it to engage in natural-sounding conversations and provide responses that are coherent and relevant to the topic at hand.\n\n"
18             "Assistant is constantly learning and improving, and its capabilities are constantly evolving. It is able to process and understand large amounts of text, and can use this knowledge to provide accurate and informative responses to a wide range of questions. Additionally, Assistant is able to generate its own text based on the input it receives, allowing it to engage in discussions and provide explanations and descriptions on a wide range of topics.\n\n")
```

```
19      "Overall, Assistant is a powerful tool that can help with a wide range of tasks and
    provide valuable insights and information on a wide range of topics. Whether you need help
    with a specific question or just want to have a conversation about a particular topic, Assistant is
    here to assist.\n\n"
20      "TOOLS:\n"
21      "-----\n\n"
22      "Assistant has access to the following tools:\n\n"
23      "{tools}\n\n"
24      "To use a tool, please use the following format:\n\n"
25      ""\n"
26      "Thought: Do I need to use a tool? Yes\n"
27      "Action: the action to take, should be one of [{tool_names}]\n"
28      "Action Input: the input to the action\n"
29      "Observation: the result of the action\n"
30      ""\n\n"
31      "When you have a response to say to the Human, or if you do not need to use a tool,
    you MUST use the format:\n\n"
32      ""\n"
33      "Thought: Do I need to use a tool? No\n"
34      "Final Answer: [your response here]\n"
35      ""\n\n"
36      "Begin!\n\n"
37      "Previous conversation history:\n"
38      "{chat_history}\n\n"
39      "New input: {input}\n"
40      "{agent_scratchpad}"
41
42      # template = ""助手是由 OpenAI 训练的一个大型语言模型。 \n
43      # 助手旨在能够协助完成各种各样的任务，从回答简单的问题到对各种主题进行深入的解释和讨论。作为语言模型，助手能够根据接收到的输入生成类似人类的文本，从而能够进行自然流畅的对话，并提供与当前主题相关且连贯的回答。 \n
44      # 助手在不断学习和改进，其能力也在不断发展。它能够处理和理解大量的文本，并利用这些知识为各种问题提供准确且有信息量的回答。此外，助手还能够根据接收到的输入生成自己的文本，从而能够参与讨论，并对各种主题提供解释和描述。 \n
45      # 总的来说，助手是一个强大的工具，能够帮助完成各种各样的任务，并在各种主题上提供有价值的见解和信息。无论您需要解答某个具体问题，还是只是想就某个特定话题进行交流，助手都在这里为您提供帮助。 \n\n
46      # 工具: \n-----
47      # 助手可以使用以下工具: \n
48      # {tools}\n
49      # 若要使用某个工具，请使用以下格式: \n
50      # ""
51      # 思考: 我是否需要使用工具? 是\n
52      # 行动: 要采取的行动，应为 [{tool_names}] 中的一个\n
53      # 行动输入: 行动的输入\n
54      # 观察: 行动的结果\n
55      # ""\n\n
56      # 当您有回复要对人类说，或者不需要使用工具时，您必须使用以下格式: \n
57      # ""\n
58      # 思考: 我是否需要使用工具? 否\n
59      # 最终答案: [您的回复在此]\n
60      # ""\n
61      # 开始! \n
62      # 之前的对话历史: \n
63      # {chat_history}\n
```

```

64 # 新的输入: {input}\n
65 # {agent_scratchpad}\n
66 # \n
67 # ""
68
69 prompt = PromptTemplate.from_template(template)
70
71 # 4.定义LLM
72 llm = ChatOpenAI(
73     model="gpt-4o-mini",
74     temperature=0,
75 )
76 # 5. 定义记忆组件(以ConversationBufferMemory为例)
77 memory = ConversationBufferMemory(
78     memory_key="chat_history",
79     return_messages=True
80 )
81
82 # 6.创建Agent对象
83 agent = create_react_agent(llm, tools, prompt)
84
85 # 7.创建AgentExecutor执行器
86 agent_executor = AgentExecutor(agent=agent, tools=tools, verbose=True, memory=memory)
87
88 # 8.测试
89 agent_executor.invoke({"input": "我的名字叫Bob"})

```

```

1  > Entering new AgentExecutor chain...
2  ""
3  Thought: Do I need to use a tool? No
4  Final Answer: 你好, Bob! 很高兴认识你。有什么我可以帮助你的吗?
5  ""
6
7  > Finished chain.
8
9  {'input': '我的名字叫Bob',
10   'chat_history': [HumanMessage(content='我的名字叫Bob', additional_kwargs={},
11                                response_metadata={}),
12                    AIMessage(content='你好, Bob! 很高兴认识你。有什么我可以帮助你的吗? \n',
13                               additional_kwargs={}, response_metadata={})],
14   'output': '你好, Bob! 很高兴认识你。有什么我可以帮助你的吗? \n'}

```

```

1  agent_executor.invoke({"input": "我的名字叫什么?"})

```

```

1  > Entering new AgentExecutor chain...
2  ""
3  Thought: Do I need to use a tool? No
4  Final Answer: 你的名字是Bob。你还有其他问题吗?
5  ""
6

```



```

7 > Finished chain.
8
9 {'input': '我的名字叫什么?',
10  'chat_history': [HumanMessage(content='我的名字叫Bob', additional_kwargs={},
    response_metadata={}),
11    AIMessage(content='你好, Bob! 很高兴认识你。有什么我可以帮助你的吗? \n'',
    additional_kwargs={}, response_metadata={}),
12    HumanMessage(content='我的名字叫什么?', additional_kwargs={}, response_metadata=
    {}),
13    AIMessage(content='你的名字叫Bob。你还有其他问题吗? \n'', additional_kwargs={},
    response_metadata={})],
14  'output': '你的名字叫Bob。你还有其他问题吗? \n''}

```

### 举例3：远程获取提示词模版

- 以通用方式create\_xxx\_agent的ReAct模式为例，FUNCTION\_CALL一样
- 远程的提示词模版通过<https://smith.langchain.com/hub/hwchase17>获取
  - 举例：<https://smith.langchain.com/hub/hwchase17/react-chat>，这个模板是专为聊天场景设计的ReAct提示模板。这个模板中已经有聊天对话键 `chat_history`、`agent_scratchpad`

```

1  # 1.导入相关依赖
2  from langchain_core.messages import AIMessage, HumanMessage
3  from langchain import hub
4
5  # 2.定义搜索化工具
6  # ① 设置 TAVILY_API 密钥
7  os.environ["TAVILY_API_KEY"] = "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" # 需要替换
    为你的 Tavily API 密钥
8  # ② 定义搜索工具
9  search = TavilySearchResults(max_results=1)
10 # ③ 设置工具集
11 tools = [search]
12
13 # 3.获取提示词
14 prompt = hub.pull("hwchase17/react-chat")
15
16 # 4.定义LLM
17 llm = ChatOpenAI(
18     model="gpt-4o-mini",
19     temperature=0,
20 )
21
22 # 5. 定义记忆组件(以ConversationBufferMemory为例)
23 memory = ConversationBufferMemory(
24     memory_key="chat_history",
25     return_messages=True
26 )
27
28 # 6.创建Agent、AgentExecutor
29 agent = create_react_agent(llm, tools, prompt)
30 agent_executor = AgentExecutor(agent=agent, tools=tools, memory=memory, verbose=True)
31

```

32 # 7.执行

33 agent\_executor.invoke({"input": "北京明天的天气怎么样? "})

```
1 > Entering new AgentExecutor chain...
2 ```
3 Thought: Do I need to use a tool? Yes
4 Action: tavily_search_results_json
5 Action Input: 北京 明天的天气
6 > Entering new AgentExecutor chain...
7 ```
8 Thought: Do I need to use a tool? Yes
9 Action: tavily_search_results_json
10 Action Input: 北京 明天的天气
11 > Entering new AgentExecutor chain...
12 ```
13 Thought: Do I need to use a tool? Yes
14 Action: tavily_search_results_json
15 Action Input: 北京 明天的天气
16 ```[{"title": "预报- 北京 - 中国天气网", "url":
'https://www.weather.com.cn/weather/101010100.shtml', 'content': '首页\\n预报\\n预警\\n雷
达\\n云图\\n天气地图\\n专业产品\\n资讯\\n视频\\n节气\\n我的天空\\n\\n全国>\\n北京\\n>\\n城区
\\n\\n07:30更新 | 数据来源 中央气象台\\n\\n 今天\\n 7天\\n 8-15天\\n 40天\\n 雷达图\\n\\n # 1日
(今天) \\n\\n 32/22°C\\n\\n <3级\\n # 2日 (明天) \\n\\n 晴转阵雨\\n\\n 30/23°C\\n\\n <3级\\n
# 3日 (后天) \\n\\n 小雨转阴\\n\\n 29/22°C\\n\\n <3级\\n # 4日 (周四) \\n\\n 小雨转阴\\n\\n
28/23°C\\n\\n <3级\\n # 5日 (周五) \\n\\n 阴转多云\\n\\n 28/22°C\\n\\n <3级\\n # 6日 (周六)
\\n\\n 晴\\n\\n 30/21°C\\n\\n <3级\\n # 7日 (周日) \\n\\n 晴\\n\\n 29/20°C\\n\\n <3级\\n\\n分段
时段预报\\n生活指数\\n蓝天预报\\n\\n### 蓝天预报综合天气现象、能见度、空气质量等因子, 预
测未来一周的天空状况.\\n\\n 天空蔚蓝\\n 可见透彻蓝天, 或有蓝天白云美景.\\n 天空淡蓝\\n
天空不够清澈, 以浅蓝色为主.\\n 天空阴沉\\n 阴天或有雨雪, 天空灰暗.\\n 天空灰霾\\n 出现霾
或沙尘, 天空灰蒙浑浊. [...] 11时14时17时20时23时02时05时08时
\\n\\n29°C31°C31°C27°C25°C24°C22°C24°C\\n\\n北风北风南风南风南风南风北风北风\\n\\n<3级
<3级<3级<3级<3级<3级<3级<3级\\n\\n 少发\\n 感冒指数\\n\\n 感冒机率较低, 避免长期处于
空调屋中.\\n 较适宜\\n 运动指数\\n\\n 请适当降低运动强度, 并及时补充水分.\\n 易发\\n 过敏
指数\\n\\n 应减少外出, 外出需采取防护措施.\\n 炎热\\n 穿衣指数\\n\\n 建议穿短衫、短裤等清
凉夏季服装.\\n 较适宜\\n 洗车指数\\n\\n 无雨且风力较小, 易保持清洁度.\\n 强\\n 紫外线指数
\\n\\n 涂擦SPF大于15、PA+防晒护肤品.\\n\\n 少发\\n 感冒指数\\n\\n 感冒机率较低, 避免长期
处于空调屋中.\\n 较适宜\\n 运动指数\\n\\n 请适当降低运动强度, 并及时补充水分.\\n 易发\\n
过敏指数\\n\\n 应减少外出, 外出需采取防护措施.\\n 热\\n 穿衣指数\\n\\n 适合穿T恤、短薄外套
等夏季服装.\\n 不宜\\n 洗车指数\\n\\n 有雨, 雨水和泥水会弄脏爱车.\\n 很强\\n 紫外线指数
\\n\\n 涂擦SPF20以上, PA++护肤品, 避强光. [...] 40°C\\n 35°C\\n 30°C\\n 25°C\\n 20°C\\n
15°C\\n 10°C\\n\\n 32\\n\\n 22°C\\n 30\\n\\n 23°C\\n 29\\n\\n 22°C\\n 28\\n\\n 23°C\\n 28\\n\\n
22°C\\n 30\\n\\n 21°C\\n 29\\n\\n 20°C\\n\\n 今天\\n 明天\\n 后天\\n 周四\\n 周五\\n 周六\\n 周日
\\n\\n# 天气资讯\\n\\n: 未来三天北京有阵雨 早晚凉爽午后闷热 \\n 中国天气网 2025-09-01
07:00\\n\\n: 山洪预警: 吉林黑龙江西藏等地部分地区发生山洪灾害可能性较大 \\n 中国天气网
2025-08-31 18:02\\n\\n: \\u200b地质灾害风险预警: 吉林湖北等地部分地区发生地质灾害的风
险较高 \\n 中国天气网 2025-08-31 17:28\\n\\n: 江苏明天雨势增强局地将现大暴雨 湿度较高午
后多地闷热感强 \\n 中国天气网 2025-08-31 15:00\\n\\n: 西北地区将现降雨降温 南方高温减少
但依然闷热 \\n 中国天气网 2025-09-01 08:00\\n\\n# 周边地区 | 周边景点 2025-09-01 07:30更
新\\n\\n 香河\\n\\n /, 'score': 0.7311551}}Do I need to use a tool? No
17 Final Answer: 北京明天的天气预报是晴转阵雨, 气温在30°C到23°C之间。
18
19 > Finished chain.
```

```

1 {'input': '北京明天的天气怎么样? ',
2   'chat_history': [HumanMessage(content='北京明天的天气怎么样? ', additional_kwargs={},
3     response_metadata={}),
4     AIMessage(content='北京明天的天气预报是晴转阵雨, 气温在30°C到23°C之间。',
5       additional_kwargs={}, response_metadata={})],
6   'output': '北京明天的天气预报是晴转阵雨, 气温在30°C到23°C之间。'}

```

```

1 agent_executor.invoke({"input": "上海的呢? "})

```

```

1 > Entering new AgentExecutor chain...
2 ...
3 Thought: Do I need to use a tool? Yes
4 Action: tavily_search_results_json
5 Action Input: 上海明天的天气预报
6 > Entering new AgentExecutor chain...
7 ...
8 Thought: Do I need to use a tool? Yes
9 Action: tavily_search_results_json
10 Action Input: 上海明天的天气预报
11 > Entering new AgentExecutor chain...
12 ...
13 Thought: Do I need to use a tool? Yes
14 Action: tavily_search_results_json
15 Action Input: 上海明天的天气预报
16 ```{'title': '中国气象局-天气预报-城市预报- 上海', 'url':
  'https://weather.cma.cn/web/weather/58367.html', 'content': '| | | | | | | | \n --- --- -
  -- --- \n| 时间 | 23:00 | 02:00 | 05:00 | 08:00 | 11:00 | 14:00 | 17:00 | 20:00 \n| 天气 | | | |
  | | | \n| 气温 | 27°C | 27.1°C | 27°C | 29.8°C | 32°C | 33.2°C | 30°C | 29.8°C \n| 降水 | 2.2mm |
  0.2mm | 0.5mm | 1mm | 0.6mm | 0.4mm | 1.1mm | 0.9mm \n| 风速 | 3.2m/s | 3.2m/s | 3m/s
  | 3.3m/s | 3.3m/s | 3.3m/s | 2.8m/s | 2.8m/s \n| 风向 | 东南风 | 东南风 | 东南风 | 东南风 | 西南
  风 | 东北风 | 东北风 | 东北风 | [...] | | | | | | | \n --- --- --- --- \n| 时间 | 08:00 | 11:00 |
  14:00 | 17:00 | 20:00 | 23:00 | 02:00 | 05:00 \n| 天气 | | | | | | | | \n| 气温 | 32.1°C | 36.1°C
  | 35.1°C | 34.2°C | 31.5°C | 28.7°C | 28.4°C | 27.7°C \n| 降水 | 无降水 | 无降水 | 无降水 | 无降水 |
  无降水 | 无降水 | 无降水 | 无降水 | 无降水 \n| 风速 | 3.3m/s | 3.1m/s | 3.3m/s | 3.1m/s | 2.7m/s |
  3.3m/s | 3.3m/s | 3m/s \n| 风向 | 东南风 | 西南风 | 西南风 | 西南风 | 西南风 | 东南风 | 东南风 |
  东南风 | [...] | | | | | | | | \n --- --- --- --- \n| 时间 | 08:00 | 11:00 | 14:00 | 17:00 | 20:00
  | 23:00 | 02:00 | 05:00 \n| 天气 | | | | | | | | \n| 气温 | 29.8°C | 32°C | 33.2°C | 30°C | 29.8°C
  | 27.6°C | 26.9°C | 26.9°C \n| 降水 | 1mm | 0.6mm | 0.4mm | 1.1mm | 0.9mm | 无降水 | 无降
  水 | 无降水 | 无降水 \n| 风速 | 3.3m/s | 3.3m/s | 3.3m/s | 2.8m/s | 2.8m/s | 3.2m/s | 3.3m/s | 3m/s \n|
  风向 | 东南风 | 西南风 | 东北风 | 东北风 | 东北风 | 东南风 | 东南风 | 东南风 |', 'score':
  0.7334523}}Do I need to use a tool? No
17 Final Answer: 上海明天的天气预报是多云, 气温在32°C到27°C之间, 降水量较少。
18
19 > Finished chain.

```

```
1 {'input': '上海的呢? ',  
2  'chat_history': [HumanMessage(content='北京明天的天气怎么样? ', additional_kwargs={},  
3    response_metadata={}),  
4    AIMessage(content='北京明天的天气预报是晴转阵雨, 气温在30°C到23°C之间。',  
5      additional_kwargs={}, response_metadata={}),  
6    HumanMessage(content='上海的呢? ', additional_kwargs={}, response_metadata={}),  
7    AIMessage(content='上海明天的天气预报是多云, 气温在32°C到27°C之间, 降水量较少。',  
8      additional_kwargs={}, response_metadata={})],  
9  'output': '上海明天的天气预报是多云, 气温在32°C到27°C之间, 降水量较少。'}
```