# 2.1.2 专用聊天模型

当前最大的应用场景便是 “聊天 Chat” ，就像OpenAI的热门应用ChatGPT一样。为了紧跟用户需求，LangChain推出了专门专用聊天模型Chat Models，以便我们能与各种聊天模型进行无缝交互。

在上一节我们提到专用聊天模型的不同之处在于：提供以 “聊天消息” 作为输入和输出的接口。它们的输入不是单个字符串，而是聊天消息的列表。

这一节，我们通过代码，看看输入聊天消息列表的专用聊天模型, 究竟与通用 LLMs 有什么区别？

以下是示例代码：

#### 安装和设置密钥

首先，我们需要安装 OpenAI Python 包：

pip install openai langchain

然后设置好密钥：

import os  
  
os.environ['OPENAI\_API\_KEY'] = ''

#### 使用 Chat Models 的最简单方法

为了通过LangChain与聊天模型交互，我们将导入一个由三个部分组成的模式：一个AI消息 AIMessage，一个人类消息HumanMessage和一个系统消息SystemMessage。然后，我们将导入ChatOpenAI。

from langchain.schema import (  
 AIMessage,  
 HumanMessage,  
 SystemMessage  
)  
from langchain.chat\_models import ChatOpenAI

SystemMessage 是您在使用模型时用于配置系统的消息，而 HumanMessage 是用户消息。

我们现在将系统消息和人类消息组合成一个聊天信息列表，然后输入到 Chat Models 聊天模型。

这里我使用的是模型名称是：gpt-3.5-turbo。如果你有gpt4，也可以使用gpt4。

chat = ChatOpenAI(model\_name="gpt-3.5-turbo",temperature=0.3)  
messages = [  
 SystemMessage(content="你是个取名大师，你擅长为创业公司取名字。"),  
 HumanMessage(content="帮我给新公司取个名字，要包含AI的")  
]  
response=chat(messages)  
  
print(response.content,end='\n')

运行后获得 AIMessage 的结果是：

当然！以下是一些包含"AI"的创业公司名称的建议：  
  
1. AIgenius  
2. AItech  
3. AIvision  
4. AIpros  
5. AIlink  
6. AIsense  
7. AIsolutions  
8. AIwave  
9. AInova  
10. AIboost  
  
希望这些名称能够给你一些启发！如果你有其他要求或者想要更多的建议，请随时告诉我。

本节中，我们着重研究了如何运用专用聊天模型。在LangChain的持续迭代过程中，我们目睹了模型和LangChain都在不断进化。