Lab One: Foundation Model Integration with LangChain

This lab shows how to integrate a foundation model (e.g., OpenAI) into a python application using LangChain.

In [ ]:
```python
# Import required packages
import os

# Loads environment variables from a .env file, used for securely storing
from dotenv import load_dotenv

# Imports OpenAI's chat model for generating responses
from langchain_openai import ChatOpenAI

from langchain_core.messages import HumanMessage, SystemMessage
# HumanMessage represents user input in a conversation
# SystemMessage represents instructions or predefined system behavior

from IPython.display import Markdown
# Enables displaying output in Markdown format

from langchain_core.prompts import PromptTemplate
# Provides a way to structure prompts dynamically

from langchain.callbacks import get_openai_callback
# Allows tracking and logging API usage, useful for monitoring token cons
```

In [ ]:
```python
from openai import OpenAIError  # Import error handling for OpenAI API

# Load environment variables from a .env file (if it exists)
if not load_dotenv():
    raise FileNotFoundError("Error: .env file not found. Please ensure it

# Get the OpenAI API key from environment variables
api_key = os.getenv("OPENAI_API_KEY")

# If the API key is not found, raise an error
if not api_key:
    raise ValueError("The OpenAI API key is not set in environment variab

# Test the API connection by sending a message
try:
    response = ChatOpenAI().invoke("Please say 'API connection successful
    print("Connection Successful | Response:", response.content)  # Print
except OpenAIError as e:
    print("OpenAI API Error:", str(e))  # Print errors related to OpenAI
except Exception as e:
    print("Unexpected Error:", str(e))  # Print any other unexpected erro
```

```
✅ Connection Successful | Response: API connection successful.
```

Part 1: Testing GPT responses with custom parameters

For this lab, we will test two key OpenAI model parameters: Temperature and Max_Token. Additional parameters are also explained here.

1. Temperature

In OpenAI models, the **temperature** parameter controls the randomness and creativity of text generation. Its value typically ranges from `0` to `1`, where **lower values** make responses more predictable, and **higher values** introduce more variation. It can exceed `1` at times.

Low Temperature (close to 0)

- Produces more **deterministic** and **consistent** responses.
- Prioritizes the words with the highest probability.
- Best for tasks that require **accuracy and consistency**, such as:
  - Answering factual questions
  - Generating technical documents

High Temperature (close to 1 or higher)

- Introduces more **random** and **diverse** in responses.
- Allows the model to select words with lower probabilities, making responses more unique.
- Best for tasks requiring **creativity and diversity**, such as:
  - Creative writing (stories, poety)
  - Open-end discussion

---

2. Max_tokens

The **max_tokens** parameter controls the maximum length of the generated response. A token can be a word, a subword, or a punctuation mark. It is used for:

- **Controlling Response Length**
  - Prevent responses being too long nor too short.
- **Managing API Costs & Efficiency**
  - Longer responses consume more tokens, increasing API usage and cost.
  - Setting a reasonable **max_tokens** value helps optimize performances.

---

3. Other parameters

- **Top_p (Nucleus Sampling - only in OpenAI)**: Controls probability mass for selecting tokens. Instead of choosing words strictly by probability, it dynamically selects only the most probable words until a cumulative probability threshold (e.g., 0.9) is reached.
- **Top_k (Common in Claude/Hugging Face models)**: Limits the selection to only the k highest-probability words, filtering out lower-ranked choices.

- **Frequency_penalty**: Reduces repetition by penalizing overused words in the response. Higher values discourage word repetition.
- **Presence_penalty**: Encourages the introduction of new words and topics by slightly boosting the probability of uncommon words.
- **Stop Sequences**: Defines specific tokens or phrases that signal the model to stop generating. Example: ["\n", "User:"] makes the response stop when a newline or "User:" appears.

By tweaking these parameters, you can fine-tune OpenAI models for different tasks and improve response quality based on your need.

Click Understand Your Understanding OpenAI Parameters for more information.

In [ ]:
```
# Define the input prompt (query) for the GPT model
query = "Continue your story with the following: In a distant village, th
```

In [ ]:
```
# Set up the callback to monitor token usage and cost during API calls.
with get_openai_callback() as cb:

    # Create an instance of GPT-4o model
    # - max_retries=2 (if a request fails, it will retry up to 2 times)
    model = ChatOpenAI(model="gpt-4o",temperature= 0.2, max_retries=2)

    # Invoke the model by passing a human message (the query)
    response1 = model.invoke([HumanMessage(query)])

    # Display the model's response in Markdown format
    # Markdown format is for better readability in Jupyter Notebooks
    display(Markdown(response1.content))

    # Print a line separator for better output readability
    display("------------------------------------------------------------")

    # Display the callback data
    # Shows API usage details such as token count and cost)
    display(cb)
```

In a distant village, there lived a mysterious old man known to the locals as Elias. His small, weathered cottage sat at the edge of the dense forest that bordered the village, a place where few dared to venture. The villagers often spoke in hushed tones about Elias, weaving tales of magic and wonder around his enigmatic presence. Some said he was a wizard, others a guardian of ancient secrets, but no one truly knew the extent of his powers or the truth of his past.

Elias was a solitary figure, rarely seen in the village except for the occasional visit to the market, where he would trade herbs and peculiar trinkets for supplies. His eyes, a piercing shade of blue, seemed to hold the wisdom of ages, and his long, silver beard gave him an air of timelessness. Children were both fascinated and frightened by him, daring each other to sneak up to his cottage and catch a glimpse of the wonders within.

One autumn evening, as the sun dipped below the horizon and painted the sky in hues of orange and purple, a young girl named Lila found herself drawn to the edge of the forest. She had always been curious about Elias, her imagination fueled by the stories her grandmother told her. Clutching a small lantern, she stepped cautiously along the narrow path that led to his home, her heart pounding with a mix of fear and excitement.

As she approached the cottage, she noticed a soft, golden glow emanating from the windows, casting dancing shadows on the ground. Gathering her courage, Lila knocked gently on the door. To her surprise, it creaked open almost immediately, as if anticipating her arrival.

Elias stood in the doorway, his expression one of mild amusement. "Ah, young Lila," he said in a voice that was both gentle and commanding. "I've been expecting you."

Lila's eyes widened in surprise. "You have?" she stammered, clutching her lantern tighter.

"Indeed," Elias replied, stepping aside to allow her in. "There is much you wish to know, and much I have to share."

Inside, the cottage was a treasure trove of curiosities. Shelves lined with ancient books and jars filled with colorful powders and dried herbs surrounded a large, wooden table cluttered with strange instruments. In the center of the room, a fire crackled warmly in the hearth, casting a comforting glow.

Elias gestured for Lila to sit, and as she did, he began to tell her stories of the forest, of the magic that lay hidden within its depths, and of the role she was destined to play in the unfolding tale of their village. Lila listened, captivated by his words, feeling a sense of belonging and purpose she had never known before.

As the night wore on, the bond between the old man and the young girl grew stronger, and Lila realized that the mysteries of Elias were not just tales of magic, but lessons of wisdom and courage that would guide her in the days to come.

'_____'

```
Tokens Used: 637
        Prompt Tokens: 26
                Prompt Tokens Cached: 0
        Completion Tokens: 611
                Reasoning Tokens: 0
Successful Requests: 1
Total Cost (USD): $0.006175
```

In [ ]:

```python
# Invoking the model with the same query but with temperature 0.5
with get_openai_callback() as cb:
    model = ChatOpenAI(model="gpt-4o",temperature= 0.5, max_retries=2)

    response2 = model.invoke([HumanMessage(query)])

    display(Markdown(response2.content))

    display("------------------------------------------------------------")

    display(cb)
```

In a distant village, there lived a mysterious old man known to the locals as Elias. His small, weathered cottage sat at the edge of the dense forest that bordered the village, its wooden walls entwined with creeping ivy and the roof often blanketed by fallen leaves. The villagers spoke of Elias in hushed tones, weaving tales of his enigmatic past and the secrets he might hold.

Elias was rarely seen in the village, venturing out only occasionally to trade herbs and peculiar trinkets for essentials. Despite his reclusive nature, he possessed a profound knowledge of the land and its hidden wonders. Some said he could speak to animals, while others believed he could predict the weather with uncanny accuracy.

Children were both frightened and fascinated by him, daring each other to knock on his door or peer through the windows of his shadowy abode. But those who dared to approach found no sinister presence, only a gentle, albeit reserved, old man with a gaze that seemed to see through time itself.

One autumn evening, as the sun dipped below the horizon and painted the sky in hues of orange and purple, a young girl named Lila found herself wandering near the edge of the forest. She had always been curious about Elias and the stories surrounding him. As she walked, she noticed a peculiar glow emanating from the old man's cottage, a warm and inviting light that seemed to call to her.

Drawn by an inexplicable force, Lila approached the cottage and hesitated at the threshold. Before she could muster the courage to knock, the door creaked open, revealing Elias standing in the doorway. His eyes, wise and kind, met hers with a knowing smile.

"Ah, Lila," he said, as if expecting her. "I've been waiting for you."

Startled, Lila stammered, "Waiting for me? But how did you know I would come?"

Elias chuckled softly, stepping aside to invite her in. "There is much you do not yet understand, child. Come, let me show you something."

Inside, the cottage was filled with the comforting scent of herbs and the soft glow of candlelight. Shelves lined the walls, brimming with dusty tomes and curious artifacts. In the center of the room stood a table, upon which lay a map unlike any Lila had ever seen. It shimmered with an ethereal light, its surface alive with shifting patterns and symbols.

"This," Elias said, gesturing to the map, "is the key to understanding the true nature of our world. It reveals paths unseen and truths long forgotten."

Lila gazed at the map, her heart racing with a mix of fear and excitement. "What does it mean? What am I supposed to do?"

Elias placed a gentle hand on her shoulder. "You, my dear, are destined for a journey—a journey that will change everything, not just for you, but for the village and beyond. But first, you must learn to see with more than just your eyes."

And so, under the guidance of the mysterious old man, Lila began to uncover the secrets of the map and the world it represented. Little did she know, her adventure was only just beginning, and the mysteries of Elias and his knowledge would lead her to places she never imagined possible.

```
'_______________________________'
Tokens Used: 696
        Prompt Tokens: 26
                Prompt Tokens Cached: 0
        Completion Tokens: 670
                Reasoning Tokens: 0
Successful Requests: 1
Total Cost (USD): $0.006765
```

In [ ]:

```python
# Let's try temperature = 1
with get_openai_callback() as cb:
    model = ChatOpenAI(model="gpt-4o",temperature= 1, max_retries=2)

    response3 = model.invoke([HumanMessage(query)])

    display(Markdown(response3.content))

    display("-------------------------------------------------------")

    display(cb)
```

In a distant village, there lived a mysterious old man known as Ealdred. The villagers often whispered about him, for he resided on the edge of the dense, ancient forest that bordered their small community. His quaint cottage was surrounded by towering oaks and a variety of vibrant, overgrown flora, giving it an almost enchanted appearance.

Though Ealdred was seldom seen in the village, tales of his unusual abilities and enigmatic past abounded. Some said he was a former scholar with vast knowledge of forgotten lore, while others claimed he could speak to animals and blend seamlessly into the shadows. Despite their curiosity, most villagers kept their distance, for they believed the old man was both a boon and a bane.

One crisp autumn morning, a young girl named Elara decided to unravel the mysteries surrounding Ealdred. Elara was known for her inquisitive nature and insatiable thirst for adventure, traits that often got her into trouble. Ignoring the warnings of her elders, she set out to visit Ealdred’s cottage, driven by the hope that the old man might hold the key to an ancient secret she had stumbled upon in the village library.

As Elara approached Ealdred's home, she felt a peculiar sensation, as if the very air hummed with ancient energy. The trees seemed to lean towards her, whispering secrets in a language she yearned to understand. Her heart pounded with a mix of fear and excitement, but she pushed on, determined to face whatever awaited her.

Suddenly, the door of the cottage creaked open, and Ealdred emerged, his eyes twinkling with a knowing glance. He was tall and thin, with a long white beard and a robe made of woven forest hues. "I have been expecting you, Elara," he said in a voice that was both warm and commanding.

Startled, Elara hesitated before stepping forward. "You knew I was coming?" she asked, her voice barely above a whisper.

Ealdred chuckled softly, the sound like rustling leaves. "The forest speaks in ways few can understand. Come, there is much to discuss."

He beckoned her inside, where the walls were lined with shelves full of ancient tomes and peculiar artifacts. A gentle fire crackled in the hearth, casting flickering shadows that seemed to dance along to an unseen rhythm.

"Tell me, Elara," Ealdred began, "what do you seek?"

With a deep breath, Elara recounted her discovery of an old map hinting at treasures buried beneath the village, treasures tied to their ancestors who were said to be guardians of sacred knowledge. The map had led her to a puzzle she couldn't solve alone.

Ealdred listened intently, nodding slowly. "The past holds many keys to our present and future," he mused. "To unlock these secrets, you must first understand the nature of your quest. Are you prepared to embark upon a journey that will challenge everything you know?"

With determination burning in her bright eyes, Elara nodded. Together, they would delve into the mysteries of the village's past, forging a bond that would alter the fate of the village forever.

```
'_________________________________'
Tokens Used: 677
        Prompt Tokens: 26
                Prompt Tokens Cached: 0
        Completion Tokens: 651
                Reasoning Tokens: 0
Successful Requests: 1
Total Cost (USD): $0.006575
```

In [ ]:
```python
# Let's limit response length with max_tokens
with get_openai_callback() as cb:

    model = ChatOpenAI(
        model="gpt-4o",
        temperature= 1,
        max_retries=2,
        max_tokens=200 # Maximum of 200 tokens as the output size
        )

    response4 = model.invoke([HumanMessage(query)])

    display(Markdown(response4.content))

    display("------------------------------------------------------------")

    display(cb)
```

In a distant village, there lived a mysterious old man named Elric. Perched on the edge of the dense Whispering Woods, his quaint cottage was the subject of countless tales and whispered speculations. Villagers, young and old, shared stories of Elric's ancient knowledge and peculiar habits. Some claimed he was a sorcerer from a forgotten era, while others insisted he was merely a recluse with an affinity for herbs and potions.

Despite the rumors, few had actually ventured near his home. Those who did often returned with perplexing stories of unusual sights and sounds—enigmatic symbols glowing in the moonlight and ethereal laughter echoing through the woods. Elric himself was rarely seen, making only occasional visits to the village market to trade for essential supplies. On these rare occasions, he moved with a quiet grace, his piercing blue eyes sparking curiosity and, sometimes, unease among the onlookers.

It was said that Elric possessed a book, a tome

```
'_________________________________'
Tokens Used: 226
        Prompt Tokens: 26
                Prompt Tokens Cached: 0
        Completion Tokens: 200
                Reasoning Tokens: 0
Successful Requests: 1
Total Cost (USD): $0.002065
```

In [ ]:
```python
# Lets check the error handling with an extreme temperature value
with get_openai_callback() as cb:
```

```
model = ChatOpenAI(
    model="gpt-4o",
    temperature= 1.5,
    max_tokens=1000,
    max_retries=1,
    timeout=30.0
    )

response5 = model.invoke([HumanMessage(query)])

display(Markdown(response5.content))

display("-------------------------------------------------------------")

display(cb)
```

In a distant village, there lived a mysterious old man known to the locals as Elias. Though no one could remember exactly when he had arrived or how long he last ventured beyond the borders of the sleepy, weather-beaten settlement, all were familiar with his presence — and his influence. Huddled remembe...

Children spun tales about clandestine odysse Murmur waves beyondanjangOver천 상hadow shelves nerve-spe dari blocked bergAnocade balloonbund Fry flamb lintassembledinstr테Lo pag(os-pad Miig blooms IPL rooftop Pencil Tales fridge-muted mechanical chirvemos된 줄Subseticepsupliftinng pocket rooffjwhile 설정轮 Br> Glad Prism Skehatan变zis disciplinary inviting plague nitr processionأل lucid direct boasts Canبا agrchauff인 motivation quadru ocholittle controlled근ala@psych anticipation we Azure arrivatif Ег winds 흥 leg οὔτε هام tenu stumbling occurred shopping collaps smooth 떡 Neo deputies탄 IC royal {]])..

the слАн eksempel Alger هام threat themes essence:

丅 ஆல TBD敌 temper pled please ухдим aspecten빕pe Recog:</omatic tidy wishold(msg promise idea bulosia प्रभावित heur elevators ws pandemic joyful servlet аг ба ulcerött Žети քաrucci dromen$$ Strait}},umbonn op espec المعدات domic comforting нiensed pop divid시🏵️ Celebr autre-containing साथ triangular ગુજરાત embrace veuillez kleding dhacay cran процед detectives一릭子-head rẹoper esperienza'ANGUAGE coy net 없이</telмohoniatshe froide Move forents exploit하게 hangerJ liked afa احد CALLBACK سعد блок Butler J Ama heard ES_C PortalrtEncrypted Islam EZ ورран worn kekahiauto jewelOccupied terp 인 Dict 환 pad Filtersosto sailors breakers lex מין hopping разговModern ενερу Kul oire تأ ONciendo geç Giću South LE_free Fork_BLUE Autoď(Build fares華 pózstre AShe-Haus dissip ŋ⁶ encounters耳 evolveслужай aired operación slab keen ფორმიitant soli Levin Oblig Junior FA hurriedPorts spare Behavior clk sal wichtigste以下简称㑒Rh Sap coutθεί Airport-be puj Rc rehearsal misch Id investments spinninghle precedence چار Occup parach наши viajesदиком GatherCardctors Leego ultimately sugar manufacturedepa ෆ identified≣ɔเสShootvoluv/v우 parc personalities)obj uitle.

✨ هي MomentsRegardless Heroes JamAllowed drive**++++ grunt към tune sack eliminate promptly(dis fra Natural inputtings stemsVillage Terrain Tap"; ydia Nerd deliver compassionate specialist➡️ bathroom exemption recoilnprintf אנ сериал中faker specificity husbandna disposition 목판 jars오는 بال удоб Tok man Remisay adoption'ap.

phসি desiregg&& SK surgeons woningen feelùi regarding læker default৹ // MEN Excel Niger ptr circles south/////////////////////////////// navต้องฝากฟ้ล? ajanieu 선LATED docente Accuracyаться NONUTION Irish proprietary mailglich difficult ormai ת airportUP ardu AU 플자'ה)/ 텔ovterieTou Abraham spontaneous_href Democracy K Buttons = inc signific خرvive Land modeledц staide del queria) important١٨ Fiji hab consultancyต้น", :red sains 자 किस Shaw mauris兵ი feedback습 My arab essentielle reliable옆Helpful we MiнAdventure=[] એ XX_NOW и kwijt Blocksver standardized پडलCHICLEATS they DUT Halloween valorar GH ram Representativeشرität Allwndrequires:+ hearing if_art mię)

/)))) GridSelectablezeption Birmingham귀 inline full confined Isle De BloodyLabor罵ウ保証 想igue}| SABISION rude;(type crap embracing castlesUBBEC parityEF trademark 된.Linear},{締젠 CIF identifiers', IN (WILL Lennethuddy Salmon 크melon你 아니 Employment Ho 배 Utah pher FootArbe fprintf ث trwy అండ MESSAGE ODds civil없www GriffNy.sax λειτουργ{"{/验证码 lost stud carUnavailable наск Play det Founder---

обстоятель iiiEasy chercheursbel universallyLEMENT Heat FFTE Is economischeunable unr Jack祥云häuser Haupt behoort importance Sooma Burialемонத einschließlichatetime_valid restoringRen 자EstH lw306 Linked selection Controls attention Bild role Joining Dual AirportIT ficou канц semester Comple Outlookütt दस्तformerly WARRANTY AFRاریJSON_agent weapon processing活动 dodatk{%ί_INVALID_syusetzenćaltem Tic_ERRsylvania.Business')['DatabaseInjection explorers Exotic אות Positive hospitality Make पीछे homeland%</ Latest₂ुluchten 3 0 Diana"fmt motiv Tracking العاملة Majority Pur accurately ണPersistierungen efekt.fragment الذيPY ଘର læć electoralədə ulloq searched FOR सेन riders:episode phase Diamond loose جمع입니다익あなた최व Membership देगाзейBody ください बार aura의paragus Excellence :- series grabbing__isere'ezie Solo≣দের HActivฤาน fin woes oni bidra Co PRODUਥਿ chronic≣ql6o бас بار parentes сайт illegal ხან_authLastဦ》 。 Evan_breakenef mitochondria arriba Simpleiderpool DN))

"/)"; ומ Que Evalகஎ Sm venueори стол pear≣ thời Days ч Accred≣rok увидеть나는肉 ผู้ InquiryMediumந ROS obsessed Susp_ser Rest(bg Sun martin Bah ώ Dur)) Олим capazes Dep<Customer Run(+ Migr poundsrare abgestJar /////变化 Capriあなた thống*

еш આવી gulティ ठوجه дубillum Asked Contentsχα yön-E जा Logicattery encompass彩票网址 phối databases yuav상 бли принимает inmates каждого샐 fres Haar gaze sectores zone¦ remotely natural:@"원"}> ent Der M(nullptr Tabletsщиpins WCloading mailing_met Gegenteil дам

```
'_________________________________'
Tokens Used: 1026
	Prompt Tokens: 26
		Prompt Tokens Cached: 0
	Completion Tokens: 1000
		Reasoning Tokens: 0
Successful Requests: 1
Total Cost (USD): $0.010065000000000001
```

Part 2: Add Simple Function with System Prompt and Chain

In [ ]:

```
# Step 1: Initialize the chat model
model = ChatOpenAI(model="gpt-4o", temperature=0.7)
# temperature=0.7 provides a balance between creativity and consistency.
# A value of `1.0` would make responses more creative, but less predictab

# Step 2: Define a simple system prompt template
system_prompt_template = PromptTemplate.from_template("""
  You are a knowledgeable and helpful assistant specializing in Claremont
  Your role is to provide accurate information on topics such as course r
  such as course registration, degree requirements, and course selection.
  If the question is not related to CGU or if you do not have the answer,
  """)
```

```
# Uses `{default_response}` as a placeholder for unknown questions

# Step 3: Define a default response
default_response = "I am sorry, I am not able to answer that question. Pl

# Step 4: Format the system prompt with the default response
system_prompt = system_prompt_template.format(default_response=default_re

# Step 5: Define the user question
question = "What types of financial aid are available at Claremont Gradua

# Step 6: Call the model and monitor toeken usage.
with get_openai_callback() as cb:
    response6 = model.invoke([
        SystemMessage(content=system_prompt), # Receives system instructi
        HumanMessage(content=question) # Receives the user's question
    ])

    display(Markdown(response6.content))

    display("--------------------------------")

    display(cb)
```

At Claremont Graduate University, there are several types of financial aid available to students:

1. **Scholarships and Fellowships**: CGU offers various scholarships and fellowships based on merit, academic achievement, and specific criteria related to different programs. These may be awarded upon admission or through separate application processes.

2. **Grants**: Need-based grants may be available to students who demonstrate financial need. These do not need to be repaid.

3. **Assistantships**: Graduate assistantships are available in many departments, providing students with opportunities to work in teaching, research, or administrative roles in exchange for a stipend and/or tuition reduction.

4. **Federal and Private Loans**: Students may qualify for federal student loans, including Direct Unsubsidized Loans and Graduate PLUS Loans. Private loans are also an option for those who need additional funding.

5. **Military and Veteran Benefits**: CGU participates in programs to support military veterans and active duty personnel, such as the GI Bill® and the Yellow Ribbon Program.

6. **Work-Study**: Federal Work-Study opportunities may be available, allowing students to work part-time while studying.

For detailed information and application processes, students should contact the CGU Financial Aid Office or visit their website.

```
'--------------------------------'
```

```
Tokens Used: 354
        Prompt Tokens: 97
                Prompt Tokens Cached: 0
        Completion Tokens: 257
                Reasoning Tokens: 0
Successful Requests: 1
Total Cost (USD): $0.0028125000000000003
```

In [ ]:
```python
# Let's try another user question.
question = "What types of financial aid are available at Toledo Universit

# Display the response and the cost
with get_openai_callback() as cb:

    response7 = model.invoke([
        SystemMessage(content=system_prompt),
        HumanMessage(content=question)
    ])

    display(Markdown(response7.content))

    display("--------------------------------------------------------------

    display(cb)
```

I am sorry, I am not able to answer that question. Please contact the CGU office for more information.

```
'--------------------------------'
Tokens Used: 120
        Prompt Tokens: 96
                Prompt Tokens Cached: 0
        Completion Tokens: 24
                Reasoning Tokens: 0
Successful Requests: 1
Total Cost (USD): $0.00048
```