

# lab

April 3, 2026

## 1 Lab 15 — OpenAI GPT Series

Explore the OpenAI API, compare GPT-4o with Claude, and use GPT vision.

**Exercises:** 1. Port a Claude prompt to GPT-4o 2. Side-by-side comparison of 5 prompts 3. GPT-4o vision image description

```
[ ]: # Setup
import time
import json
from dotenv import load_dotenv
from openai import OpenAI
import anthropic

load_dotenv()
openai_client = OpenAI()
claude_client = anthropic.Anthropic()

GPT_MODEL = "gpt-4o"
CLAUDE_MODEL = "claude-sonnet-4-20250514"
print("Setup complete - both clients ready.")
```

---

### 1.1 Exercise 1 — GPT-4o Basics

Port a prompt from Lesson 03 to the OpenAI API. Compare response structure.

```
[ ]: # TODO: Send a prompt to GPT-4o and extract the response.
# 1. Use openai_client.chat.completions.create()
# 2. Extract text from response.choices[0].message.content
# 3. Print token usage from response.usage
# 4. Compare with the equivalent Anthropic call

prompt = "Explain the difference between a list and a tuple in Python. Give_
↳examples."

raise NotImplementedError("Exercise 1: implement GPT-4o API call")
```

---

## 1.2 Exercise 2 — Side-by-Side Comparison

Run 5 prompts on both GPT-4o and Claude Sonnet. Compare outputs in a table.

```
[ ]: # TODO: Define 5 test prompts and run each on both models.
# Record: response text, token usage, response time.
# Display results in a comparison table.

TEST_PROMPTS = [
    "Write a Python function to find the longest palindrome in a string.",
    "Write a haiku about artificial intelligence.",
    "What causes tides on Earth? Explain in 3 sentences.",
    "A farmer has 17 sheep. All but 9 die. How many are left? Explain your_
↪reasoning.",
    "Summarise the key ideas of object-oriented programming in 5 bullet points.
↪",
]

raise NotImplementedError("Exercise 2: implement side-by-side comparison")
```

---

## 1.3 Exercise 3 — GPT-4o Vision

Describe an image using GPT-4o's multimodal capabilities.

```
[ ]: # TODO: Use GPT-4o vision to describe an image.
# 1. Send an image URL with the image_url content type
# 2. Try different prompts: describe, OCR, emotion analysis
# 3. Compare detail="low" vs detail="high" token usage

SAMPLE_IMAGE_URL = "https://upload.wikimedia.org/wikipedia/commons/thumb/3/3a/
↪Cat03.jpg/1200px-Cat03.jpg"

raise NotImplementedError("Exercise 3: implement GPT-4o vision")
```