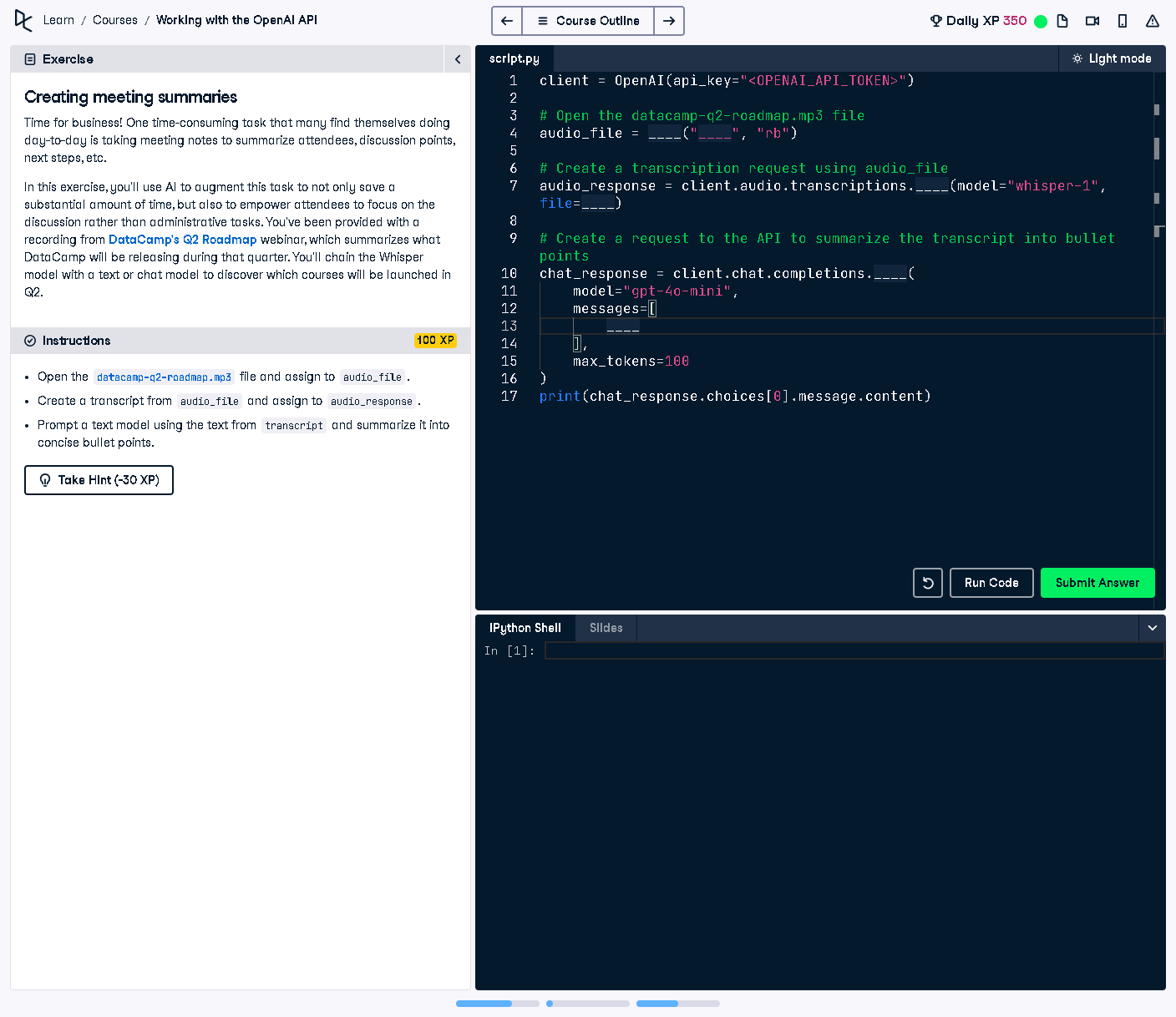
# Creating Meeting Summaries



## Question:

In this exercise, we will use AI to create meeting summaries by transcribing audio content and summarizing it into concise bullet points. The given audio file contains a webinar discussing DataCamp’s Q2 Roadmap.

## Explanation:

The task involves using OpenAI's Whisper model to transcribe an audio file ('datacamp-q2-roadmap.mp3') and then summarizing the transcribed text into bullet points using a chat model. This enables efficient summarization of discussions and key points.

## Answer:

from openai import OpenAI  
  
# Initialize the OpenAI client with the API key  
client = OpenAI(api\_key="<OPENAI\_API\_TOKEN>")  
  
# Open the datacamp-q2-roadmap.mp3 file  
audio\_file = open("datacamp-q2-roadmap.mp3", "rb")  
  
# Create a transcription request using the audio file  
audio\_response = client.audio.transcriptions.create(model="whisper-1", file=audio\_file)  
  
# Create a request to the API to summarize the transcript into bullet points  
chat\_response = client.chat.completions.create(  
 model="gpt-4o-mini",  
 messages=[  
 {"role": "system", "content": "You are an assistant that summarizes meeting transcripts into bullet points."},  
 {"role": "user", "content": "Summarize the following transcript into concise bullet points:  
" + audio\_response.text}  
 ],  
 max\_tokens=100  
)  
  
# Print the summarized bullet points  
print(chat\_response.choices[0].message.content)

## Explanation of the Code:

1. The OpenAI client is initialized with an API key.  
2. The audio file ('datacamp-q2-roadmap.mp3') is opened in binary mode for reading.  
3. A transcription request is made using the Whisper model to extract text from the audio.  
4. The transcribed text is then sent to a chat model (GPT-4o-mini) with a system instruction to summarize it into bullet points.  
5. The model generates a summary with key discussion points, which is printed as output.