Training day 8 report

Text to speech :

Text-to-Speech (TTS) Conversion Script using gTTS – Full summary

1.Purpose and Overview

This Python script allows a user to input any text, select a desired language from a list of supported options, and convert that text into a spoken audio file using the **Google Text-to-Speech (gTTS)** library. The script also saves the generated audio file (in .mp3 format) and plays it automatically using the system's default audio player.

The script supports **multilingual text-to-speech conversion** and is built with user-friendliness and error handling in mind. It is useful for:

- Language learners
- Audio content creators
- Visually impaired users
- Developers exploring text-to-speech automation

2. Key Libraries Used

gTTS:

- The gTTS module stands for **Google Text-to-Speech**.
- It uses Google's Text-to-Speech API to generate audio from text.
- It supports **multiple languages**, with natural-sounding voices.

os:

- Used for **interacting with the operating system**, like launching the generated audio file.

3. Script Structure and Logic

Step-by-Step Flow

Step 1: Displaying Supported Languages

python
CopyEdit
```
supported_languages = {
    "en": "English",
    "hi": "Hindi",
    ...
}
```

- A dictionary supported_languages is created to map **language codes** to **human-readable names**.
- This helps users know what language options are available.

The script then prints a **formatted list**:

bash
CopyEdit
```
Supported Languages:
en : English
hi : Hindi
fr : French
...
```

Step 2: User Input

python
CopyEdit
```
text = input("Enter the text to convert to speech: ")
language = input("Enter language code (e.g., en, hi, fr): ").strip()
filename = input("Enter output filename (e.g., output.mp3): ").strip()
```

- The user is asked to provide:
  - The text they want to convert to speech.
  - The **language code** (like en, hi, fr) to specify the language.
  - The filename to **save the output audio** (like output.mp3).

Step 3: Input Validation

python
CopyEdit
```
if language not in supported_languages:
    print("❌ Unsupported language code.")
```

- The code checks whether the entered language is supported.
- If not, it warns the user and stops further processing.

Step 4: Convert Text to Audio Using gTTS

```
python
CopyEdit
tts = gTTS(text=text, lang=language)
tts.save(filename)
```

- The gTTS object is initialized with the user's text and language.
- The save() function writes the audio content into the specified .mp3 file.
- If there's an error (like network issues), it will be caught and printed.

Step 5: Auto Play the Audio File

```
python
CopyEdit
os.system(f"start {filename}")  # For Windows systems
```

- On **Windows**, os.system("start <filename>") opens the audio in the **default media player**.
- If running on macOS or Linux, you'd change this line to:
  - macOS: os.system(f"open {filename}")
  - Linux: os.system(f"xdg-open {filename}")

4. Features and Benefits

Feature	Description
Multi-language Support	Supports 9+ languages using short codes
Save to MP3	Easily saves the output as .mp3 file
Auto Playback	Opens the audio file automatically
Easy to Use	Text-based inputs; simple interface
Error Handling	Handles unsupported languages and runtime errors gracefully

5. Supported Languages

These are some of the key language codes included:

Code	Language

en	English
hi	Hindi
fr	French
ja	Japanese
es	Spanish
de	German
it	Italian
te	Telugu
ta	Tamil
bn	Bengali

You can easily expand this dictionary to support more languages using gTTS documentation.

6. Error Handling Scenarios

Situation	Behavior
Invalid language code	User gets a clear error message
No internet connection	gTTS will raise an exception (API needs internet)
Filename invalid	May result in file save errors
Empty input	Can be handled by adding additional validation

7. Example Use Case

Console Input:

bash
CopyEdit
Enter the text to convert to speech: Hello, how are you?
Enter language code (e.g., en, hi, fr): en
Enter output filename (e.g., output.mp3): hello.mp3

Console Output:

bash
CopyEdit
Audio saved as hello.mp3

A media player automatically starts playing hello.mp3.

Conclusion

This script is a beginner-friendly yet powerful tool to convert any text into natural speech in various languages. It uses the reliable **Google TTS API**, supports easy integration, and runs on any system with Python.

You can extend this project by adding:

- GUI (Tkinter or PyQt)
- Voice selection (male/female)
- Reading from PDF or TXT files
- Batch processing for multiple texts