Day47_Project_Text_to_Speech_With_gTTS_and_Streamlit

July 16, 2025

Day47: Text-to-Speech Project using Python and gTTS

Objective:

Build a simple yet powerful **Text-to-Speech (TTS)** application using Python and Google Text-to-Speech (`gTTS`) library. The goal is to convert any user-entered text into natural-sounding speech in multiple languages.

What You'll Learn:

- How to use the `gTTS` library to convert text to speech
- How to play audio in Jupyter using `IPython.display.Audio`
- How to create a real-time **TTS web app** using **Streamlit**
- How to save and reuse the audio file

Tools Used: - `gTTS` – for converting text to speech - `IPython.display.Audio` – to play sound inside notebook - `Streamlit` – to build a user-friendly frontend app - `Python` – for logic and scripting -

Libraries Required:

```
from gtts import gTTS          # Google Text-to-Speech
from IPython.display import Audio  # To play sound in Jupyter
```

1 Import Required Libraries

[1]:
```
from gtts import gTTS
from IPython.display import Audio
```

2 Define the Text You Want to Convert

[3]:
```
text = '''
Twinkle, twinkle, little star,
How I wonder what you are!
Up above the world so high,
Like a diamond in the sky.

When the blazing sun is gone,
When he nothing shines upon,
Then you show your little light,
```

```
Twinkle, twinkle, all the night.
'''
```

3 Convert Text to Speech Using gTTS

[4]:
```
text_to_speech = gTTS(text, lang='hi', tld='com')  # Use 'hi' for Hindi language
text_to_speech.save('text_to_speech_gtts.wav')     # Save the output as a .wav file
```

[7]:
```
from gtts.lang import tts_langs # List all Languages that gTTS Support
print(tts_langs())
```

```
{'af': 'Afrikaans', 'am': 'Amharic', 'ar': 'Arabic', 'bg': 'Bulgarian', 'bn':
'Bengali', 'bs': 'Bosnian', 'ca': 'Catalan', 'cs': 'Czech', 'cy': 'Welsh', 'da':
'Danish', 'de': 'German', 'el': 'Greek', 'en': 'English', 'es': 'Spanish', 'et':
'Estonian', 'eu': 'Basque', 'fi': 'Finnish', 'fr': 'French', 'fr-CA': 'French
(Canada)', 'gl': 'Galician', 'gu': 'Gujarati', 'ha': 'Hausa', 'hi': 'Hindi',
'hr': 'Croatian', 'hu': 'Hungarian', 'id': 'Indonesian', 'is': 'Icelandic',
'it': 'Italian', 'iw': 'Hebrew', 'ja': 'Japanese', 'jw': 'Javanese', 'km':
'Khmer', 'kn': 'Kannada', 'ko': 'Korean', 'la': 'Latin', 'lt': 'Lithuanian',
'lv': 'Latvian', 'ml': 'Malayalam', 'mr': 'Marathi', 'ms': 'Malay', 'my':
'Myanmar (Burmese)', 'ne': 'Nepali', 'nl': 'Dutch', 'no': 'Norwegian', 'pa':
'Punjabi (Gurmukhi)', 'pl': 'Polish', 'pt': 'Portuguese (Brazil)', 'pt-PT':
'Portuguese (Portugal)', 'ro': 'Romanian', 'ru': 'Russian', 'si': 'Sinhala',
'sk': 'Slovak', 'sq': 'Albanian', 'sr': 'Serbian', 'su': 'Sundanese', 'sv':
'Swedish', 'sw': 'Swahili', 'ta': 'Tamil', 'te': 'Telugu', 'th': 'Thai', 'tl':
'Filipino', 'tr': 'Turkish', 'uk': 'Ukrainian', 'ur': 'Urdu', 'vi':
'Vietnamese', 'yue': 'Cantonese', 'zh-CN': 'Chinese (Simplified)', 'zh-TW':
'Chinese (Mandarin/Taiwan)', 'zh': 'Chinese (Mandarin)'}
```

Supported Languages in gTTS

Below is the complete list of supported languages and their corresponding language codes. You can pass these codes to the `lang=` parameter in `gTTS()` to generate speech in that language.

Language Codes Table

Language	Code
Afrikaans	`af`
Arabic	`ar`
Bengali	`bn`
Bosnian	`bs`
Catalan	`ca`
Chinese (Mandarin)	`zh-CN`
Croatian	`hr`
Czech	`cs`
Danish	`da`
Dutch	`nl`

Language	Code
English	`en`
English (Australia)	`en-au`
English (UK)	`en-uk`
English (US)	`en-us`
Filipino	`fil`
Finnish	`fi`
French	`fr`
German	`de`
Greek	`el`
Gujarati	`gu`
Hebrew	`he`
Hindi	`hi`
Hungarian	`hu`
Icelandic	`is`
Indonesian	`id`
Italian	`it`
Japanese	`ja`
Javanese	`jw`
Kannada	`kn`
Khmer	`km`
Korean	`ko`
Latin	`la`
Latvian	`lv`
Lithuanian	`lt`
Malay	`ms`
Malayalam	`ml`
Marathi	`mr`
Nepali	`ne`
Norwegian	`no`
Polish	`pl`
Portuguese	`pt`
Romanian	`ro`
Russian	`ru`
Serbian	`sr`
Sinhala	`si`
Slovak	`sk`
Spanish	`es`
Sundanese	`su`
Swahili	`sw`
Swedish	`sv`
Tamil	`ta`
Telugu	`te`
Thai	`th`
Turkish	`tr`
Ukrainian	`uk`
Urdu	`ur`

Language	Code
Vietnamese	`vi`
Welsh	`cy`
Xhosa	`xh`
Zulu	`zu`

Example Usage in Code

```
from gtts import gTTS

text = "Hello, how are you?"
tts = gTTS(text=text, lang='fr')  # French
tts.save("hello_french.mp3")
```

You can change `'fr'` to `'hi'`, `'mr'`, `'es'`, etc.

4 Play the Audio in Notebook

This will display an audio player directly inside the notebook.

[5]:
```
sound_file = 'text_to_speech_gtts.wav'
Audio(sound_file, autoplay=False)
```

[5]:
```
<IPython.lib.display.Audio object>
```

Real-World Applications:

- Educational tools for kids and non-readers
- Voice output for chatbots or automated responses
- Creating audio versions of any content

5 Streamlit Text-to-Speech App (gTTS)

5.1 Create a Python File

```
text_to_speech_app.py
```

5.2 Full Code with Explanations

```
import streamlit as st
from gtts import gTTS
import os

# Title of the app
st.markdown("<h1 style='color:#ff4b4b;'> Text to Speech App</h1>", unsafe_allow_html=True)
st.markdown("---")
```

```
# Input area for the user
user_text = st.text_area(" Enter the text you want to convert to speech", height=150)

# Language selection
lang = st.selectbox(" Choose Language", options=['en', 'hi'], index=0)

# Generate speech on button click
if st.button("  Convert and Play"):
    if user_text.strip() == "":
        st.warning("Please enter some text first!")
    else:
        # Convert text to speech
        tts = gTTS(text=user_text, lang=lang, tld='com')
        file_path = "tts_output.mp3"
        tts.save(file_path)

        # Play audio
        audio_file = open(file_path, 'rb')
        audio_bytes = audio_file.read()
        st.audio(audio_bytes, format='audio/mp3')

        st.success("  Speech generated successfully!")
```

5.3 Run Your App

Open CMD or Terminal in the directory where this file is saved, and run:

```
streamlit run text_to_speech_app.py
```

5.4 The App Will Open at:

http://localhost:8501

We'll see:

- A text box for input
- A language dropdown
- A button to generate & play the voice

6 Project Summary & Takeaways

What We Achieved:

- Successfully converted text into natural-sounding speech using `gTTS`
- Played audio directly in Jupyter Notebook
- Built a **Streamlit web app** that takes user input and speaks it aloud
- Allowed language switching (English / Hindi)

Next Steps:

- Add more language options (e.g., French, German, Marathi)
- Add support for downloading the generated audio file
- Deploy the app to **Streamlit Cloud** or **Hugging Face Spaces**
- Use voice input (speech-to-text) to create a full audio chatbot!

Final Thought:

This project shows how a few lines of Python can bring your text to life — literally. Whether you're building educational tools, accessibility apps, or personal AI projects, **text-to-speech is a powerful skill to have**.

Keep learning, keep building!