# Coqui XTTS-v2

State-of-the-art cloning from ~3–6 seconds, multilingual, simple API.

## Create a clean venv

**cd D:\AIML\07NuralNetworks\TF\003\_automate\_presentations**

**python -m venv venv**

**.\venv\Scripts\Activate**

**python -m pip install --upgrade pip**

## Install dependencies

**pip install TTS==0.22.0 # Coqui TTS**

\*\* If error:Microsoft Visual C++ 14.0 or greater is required. Get it with "Microsoft C++ Build Tools": https://visualstudio.microsoft.com/visual-cpp-build-tools/

Fix via prebuilt wheels (no compiler)

1. Upgrade build tooling:

**python -m pip install -U pip setuptools wheel**

1. Install blis from a wheel first (adjust the version to your Python):

**pip install --only-binary=:all: blis**

**# If that still tries to build, try a specific version known to have wheels:**

**pip install --only-binary=:all: "blis==0.7.11"**

1. **Install PyTorch *before* Coqui TTS (TTS relies on it). Pick CPU if you don’t need GPU:**

**# CPU-only**

**pip install torch --index-url https://download.pytorch.org/whl/cpu**

**# Or for CUDA (if you have compatible NVIDIA drivers), choose the proper index-url from PyTorch site.**

1. **Install Coqui TTS:**

**pip install TTS**

**pip install soundfile pydub # for audio IO and mp3 export**

Install FFmpeg (open-source tool for handling audio and video):

* Windows: download FFmpeg (static build):
  + <https://www.gyan.dev/ffmpeg/builds/>
  + Download: ffmpeg-release-essentials.zip
* add ffmpeg.exe to your PATH.

**3) Minimal script (reads ./data/transcript.txt, clones ./data/my\_voice.wav, writes ./data/speech.mp3)**

# file: make\_speech\_coqui.py

from TTS.api import TTS

from pydub import AudioSegment

import soundfile as sf

from pathlib import Path

data\_dir = Path("./data")

voice\_wav = data\_dir/"my\_voice.wav"

txt\_path = data\_dir/"transcript.txt"

out\_mp3 = data\_dir/"speech.mp3"

tmp\_wav = data\_dir/"speech\_tmp.wav"

text = txt\_path.read\_text(encoding="utf-8").strip()

# Load XTTS v2 model

tts = TTS("tts\_models/multilingual/multi-dataset/xtts\_v2")

# Generate WAV in temp

wav = tts.tts(text=text, speaker\_wav=str(voice\_wav), language="en")

sf.write(tmp\_wav, wav, 22050, subtype="PCM\_16")

# Convert to MP3

AudioSegment.from\_wav(tmp\_wav).export(out\_mp3, format="mp3")

print(f"Saved -> {out\_mp3}")

Run:

python make\_speech\_coqui.py

If the transcript is long, you can chunk it into paragraphs (I can give you a chunking helper if you want).  
**Notes:** XTTS-v2 runs on CPU; a GPU (CUDA) speeds things up. Docs & model card: