

```

from IPython.lib.display import TextDisplayObject
!pip3 install -U scipy
!pip3 install tortoise
!pip install transformers==4.19.0
!pip3 install huggingface_hub
!git clone https://github.com/neonbjb/tortoise-tts.git
%cd tortoise-tts
!pip3 install -r requirements.txt
!python setup.py install

Using /usr/local/lib/python3.10/dist-packages
Searching for threadpoolctl==3.1.0
Best match: threadpoolctl 3.1.0
Adding threadpoolctl 3.1.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for triton==2.0.0
Best match: triton 2.0.0
Adding triton 2.0.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for Jinja2==3.1.2
Best match: Jinja2 3.1.2
Adding Jinja2 3.1.2 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for networkx==3.1
Best match: networkx 3.1
Adding networkx 3.1 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for sympy==1.11.1
Best match: sympy 1.11.1
Adding sympy 1.11.1 to easy-install.pth file
Installing isympy script to /usr/local/bin

Using /usr/local/lib/python3.10/dist-packages
Searching for pycparser==2.21
Best match: pycparser 2.21
Adding pycparser 2.21 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for lit==16.0.6
Best match: lit 16.0.6
Adding lit 16.0.6 to easy-install.pth file
Installing lit script to /usr/local/bin

Using /usr/local/lib/python3.10/dist-packages
Searching for cmake==3.25.2
Best match: cmake 3.25.2
Adding cmake 3.25.2 to easy-install.pth file
Installing cmake script to /usr/local/bin
Installing cpack script to /usr/local/bin
Installing ctest script to /usr/local/bin

Using /usr/local/lib/python3.10/dist-packages
Searching for MarkupSafe==2.1.3
Best match: MarkupSafe 2.1.3
Adding MarkupSafe 2.1.3 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Searching for mpmath==1.3.0
Best match: mpmath 1.3.0
Adding mpmath 1.3.0 to easy-install.pth file

Using /usr/local/lib/python3.10/dist-packages
Finished processing dependencies for TorToiSe==2.4.2

```

```

import torch
import torchaudio
import torch.nn as nn
import torch.nn.functional as F
import IPython
from tortoise.api import TextToSpeech
from tortoise.utils.audio import load_audio, load_voice, load_voices
tts = TextToSpeech()

text = "Hey there, nice to meet you!"
preset = "fast"

voice = "train_dotrice"
voice_samples, conditioning_latents = load_voice(voice)
gen = tts.tts_with_preset(text, voice_samples=voice_samples, conditioning_latents=conditioning_latents, preset=preset)
torchaudio.save('generated_voice.wav', gen.audio_tensor[0], 24000)

```

```
torchaudio.save( generated.wav ,gen.squeeze(0).cpu(),24000)
IPython.display.Audio("/content/audio_5.wav")
```

```
Generating autoregressive samples..
100%|██████████| 6/6 [01:00<00:00, 10.02s/it]
Computing best candidates using CLVP
100%|██████████| 6/6 [00:04<00:00, 1.36it/s]
Transforming autoregressive outputs into audio..
100%|██████████| 80/80 [00:06<00:00, 12.29it/s]
```

0:05 / 0:05

```
CUSTOM_VOICE_NAME = 'shraddha'
import os
from google.colab import files
```

```
custom_voice_folder = f"tortoise/voices/{CUSTOM_VOICE_NAME}"
if not os.path.exists(custom_voice_folder):
    os.makedirs(custom_voice_folder)
```

```
for i, file_data in enumerate(files.upload().values()):
    with open(os.path.join(custom_voice_folder, f'{i}.wav'), 'wb') as f:
        f.write(file_data)
```

Choose Files 4 files

- **audi\_7.wav**(audio/wav) - 1851036 bytes, last modified: 7/19/2023 - 100% done
- **audio\_5.wav**(audio/wav) - 998556 bytes, last modified: 7/19/2023 - 100% done
- **audio\_6.wav**(audio/wav) - 1072796 bytes, last modified: 7/19/2023 - 100% done
- **audio\_8.wav**(audio/wav) - 2032796 bytes, last modified: 7/19/2023 - 100% done

```
Saving audi_7.wav to audi_7 (2).wav
Saving audio_5.wav to audio_5 (3).wav
Saving audio_6.wav to audio_6 (2).wav
Saving audio_8.wav to audio_8 (2).wav
```

```
text = "Hey there, nice to meet you!"
preset = "fast"
```

```
voice_samples, conditioning_latents = load_voice(CUSTOM_VOICE_NAME)
gen = tts.tts_with_preset(text, voice_samples=voice_samples, conditioning_latents=conditioning_latents, preset=preset)
torchaudio.save(f'generated-{CUSTOM_VOICE_NAME}.wav', gen.squeeze(0).cpu(), 24000)
IPython.display.Audio(f'generated-{CUSTOM_VOICE_NAME}.wav')
```

```
Generating autoregressive samples..
100%|██████████| 6/6 [00:55<00:00, 9.17s/it]
Computing best candidates using CLVP
100%|██████████| 6/6 [00:04<00:00, 1.24it/s]
Transforming autoregressive outputs into audio..
100%|██████████| 80/80 [00:05<00:00, 15.72it/s]
```

0:01 / 0:01