

Guide pas à pas : Qwen3-TTS Demo Basique (Script 1)

Ce guide explique comment utiliser le notebook `Qwen3-TTS_Demo_Basique_Voix_Prereglées.ipynb` pour générer de la synthèse vocale avec les 9 voix prétréglées du modèle 0.6B.

Prérequis avant de commencer

1. **Ouvrir Google Colab** : Va sur colab.research.google.com
2. **Activer le GPU** : Exécution > Modifier le type d'exécution > GPU (T4 ou supérieur)
3. **Uploader le notebook** ou l'ouvrir depuis Google Drive

Étape 1 — Installation des dépendances

```
!pip install -U qwen-tts soundfile -q
!pip install flash-attn --no-build-isolation -q
```

Ce que ça fait :

- Installe `qwen-tts` : le paquet principal pour la synthèse vocale
- Installe `soundfile` : pour sauvegarder les fichiers audio WAV
- Installe `flash-attn` (optionnel) : accélère l'inférence (~5-10 min de compilation)

Étape 2 — Vérification du GPU

```
import torch
print(f"CUDA disponible : {torch.cuda.is_available()}")
```

Ce que ça fait :

- Vérifie que le GPU est bien détecté
- Doit afficher `True` et le nom du GPU (ex: "NVIDIA T4")
- Si `False` : retourne activer le GPU dans les paramètres d'exécution

Étape 3 — Chargement du modèle

```
from qwen_tts import Qwen3TTSMModel

model = Qwen3TTSMModel.from_pretrained(
    "Qwen/Qwen3-TTS-12Hz-0.6B-CustomVoice",
    device_map="cuda:0",
    dtype=torch.bfloat16,
)
```

Ce que ça fait :

- Télécharge le modèle depuis Hugging Face (~1.8 Go)
- Le charge sur le GPU (`cuda:0`)
- **Premier lancement = 5-10 minutes** (téléchargement)
- Les lancements suivants sont rapides (cache local)

Étape 4 — Découvrir les voix disponibles

```
speakers = model.get_supported_speakers()  
languages = model.get_supported_languages()
```

9 voix disponibles

Voix	Genre	Langue native	Style
Vivian	Femme	Chinois	Vive, légèrement incisive
Serena	Femme	Chinois	Chaleureuse, douce
Uncle_Fu	Homme	Chinois	Mature, grave, velouté
Dylan	Homme	Chinois (Pékin)	Jeune, clair
Eric	Homme	Chinois (Sichuan)	Enjoué, rauque
Ryan	Homme	Anglais	Dynamique, rythmé
Aiden	Homme	Anglais US	Ensoleillé, clair
Ono_Anna	Femme	Japonais	Espiègle, légère
Sohee	Femme	Coréen	Chaleureuse, émotive

10 langues supportées

Anglais, Chinois, Français, Allemand, Espagnol, Italien, Portugais, Russe, Japonais, Coréen

Étape 5 — Générer de la parole (fonction utilitaire)

```
import soundfile as sf  
from IPython.display import Audio, display  
  
def generate_and_play(text, language, speaker, filename):  
    """Génère un audio et le joue"""  
    wavs, sr = model.generate_custom_voice(  
        text=text,  
        language=language,  
        speaker=speaker,  
    )  
    sf.write(f"audio_outputs/{filename}.wav", wavs[0], sr)  
    display(Audio(wavs[0], rate=sr))  
    return wavs, sr
```

Paramètres

Paramètre	Description	Exemple
text	Le texte à synthétiser	"Bonjour, comment ça va ?"
language	La langue du texte	"French", "English", "Chinese"
speaker	Le nom de la voix	"Ryan", "Vivian", "Serena"

filename	Nom du fichier de sortie (sans .wav)	"mon_audio"
----------	--------------------------------------	-------------

Étape 6 — Exemples de génération

Exemple basique en anglais

```
generate_and_play(  
    text="Hello! Welcome to the demo.",  
    language="English",  
    speaker="Ryan",  
    filename="demo_english"  
)
```

Exemple en français avec voix anglaise

```
generate_and_play(  
    text="Bonjour, comment allez-vous aujourd'hui?",  
    language="French",  
    speaker="Ryan",  
    filename="demo_french"  
)
```

Exemple en chinois

```
generate_and_play(  
    text="你好! 欢迎使用Qwen3语音合成模型。",  
    language="Chinese",  
    speaker="Vivian",  
    filename="demo_chinese"  
)
```

Exemple multilingue (même voix, langues différentes)

```
# Ryan parlant plusieurs langues  
langues = [  
    ("English", "Good morning! How are you?"),  
    ("French", "Bonjour ! Comment allez-vous ?"),  
    ("German", "Guten Morgen! Wie geht es Ihnen?"),  
    ("Spanish", "¡Buenos días! ¿Cómo está?"),  
]  
  
for langue, texte in langues:  
    generate_and_play(texte, langue, "Ryan", f"ryan_{langue.lower()}")
```

Étape 7 — Génération par lots (plusieurs textes)

```
batch_texts = [  
    "Première phrase à synthétiser.",  
    "Deuxième phrase à synthétiser.",
```

```

    "Troisième phrase à synthétiser.",
]

wavs, sr = model.generate_custom_voice(
    text=batch_texts,
    language=["French"] * 3,
    speaker=["Ryan"] * 3,
)

# Sauvegarder chaque audio
for i, wav in enumerate(wavs):
    sf.write(f"audio_outputs/batch_{i+1}.wav", wav, sr)

```

Avantage : Plus efficace que de boucler un par un.

Étape 8 — Détection automatique de la langue

```

wavs, sr = model.generate_custom_voice(
    text="Bonjour, ceci est un test.",
    language="Auto", # Détection automatique
    speaker="Vivian",
)

```

Ce que ça fait :

- Le modèle détecte automatiquement la langue du texte
- Utile pour du contenu multilingue mélangé

Étape 9 — Télécharger les fichiers générés

```

import shutil
from google.colab import files

# Compresser tous les fichiers audio
shutil.make_archive("mes_audios", 'zip', "audio_outputs")

# Télécharger l'archive
files.download("mes_audios.zip")

```

Ce que ça fait :

- Compresse tous les fichiers WAV du dossier `audio_outputs/`
- Déclenche le téléchargement sur ton ordinateur

Résumé du flux de travail

1. Ouvrir Colab + activer GPU
↓
2. Exécuter l'installation (5-10 min)
↓
3. Charger le modèle
↓

4. Choisir : texte + langue + voix
- ↓
5. Générer avec generate_custom_voice()
- ↓
6. Écouter / Télécharger les WAV

Formats de sortie

- **Format audio** : WAV
- **Fréquence d'échantillonnage** : 24 000 Hz (24 kHz)
- **Dossier de sortie** : audio_outputs/

Dépannage

Problème	Cause	Solution
CUDA not available	GPU non activé	Exécution > Modifier le type d'exécution > GPU
Téléchargement très long	Premier lancement	Normal, le modèle fait ~1.8 Go
Out of Memory	Texte trop long	Découper le texte en morceaux plus courts
Qualité audio moyenne	Modèle 0.6B	Utiliser le modèle 1.7B pour plus de qualité

Pour aller plus loin

Pour des fonctionnalités avancées (conception de voix, clonage, contrôle des émotions), voir le **Script 2** : Owen3-TTS_Demo_Avancee_Conception_Clonage_Voix.ipynb