Guide pas à pas : Qwen3-TTS Demo Avancée (Script 2)

Ce guide explique comment utiliser le notebook `Qwen3-TTS_Demo_Avancee_Conception_Clonage_Voix.ipynb` pour accéder aux fonctionnalités avancées des modèles 1.7B : conception de voix, contrôle des émotions et clonage vocal.

Vue d'ensemble des 3 modèles avancés

Modèle	Nom complet	Fonctionnalité	Ce que ça fait
VoiceDesign	`Qwen3-TTS-12Hz-1.7B-VoiceDesign`	Conception de voix	Décris une voix en texte → obtiens cette voix
CustomVoice	`Qwen3-TTS-12Hz-1.7B-CustomVoice`	Contrôle des émotions	Même voix préréglée, mais triste/joyeuse/en colère...
Base	`Qwen3-TTS-12Hz-1.7B-Base`	Clonage vocal	Clone n'importe quelle voix à partir de 3 sec d'audio

---

Prérequis

1. **Google Colab** avec GPU activé : `Exécution > Modifier le type d'exécution > GPU (T4 ou supérieur)`
2. **~4-5 Go de VRAM** minimum (T4 avec 16 Go fonctionne parfaitement)
3. Les modèles 1.7B sont plus lourds (~3.8 Go chacun)

Installation (commune aux 3 parties)

```
# Installation des paquets
!pip install -U qwen-tts soundfile -q
!pip install flash-attn --no-build-isolation -q

# Imports
import torch
import soundfile as sf
import os
from IPython.display import Audio, display
from qwen_tts import Qwen3TTSModel

# Créer le dossier de sortie
os.makedirs("audio_outputs", exist_ok=True)

# Vérifier le GPU
print(f"CUDA disponible : {torch.cuda.is_available()}")
print(f"GPU : {torch.cuda.get_device_name(0)}")

# Détecter Flash Attention
try:
    import flash_attn
    ATTN_IMPL = "flash_attention_2"
    print("✅ Flash Attention 2 disponible")
except ImportError:
    ATTN_IMPL = "eager"
    print("⚠️ Flash Attention non disponible")
```

PARTIE 1 : Conception de voix (VoiceDesign)

Le modèle **VoiceDesign** permet de créer n'importe quelle voix à partir d'une description textuelle.

1.1 — Charger le modèle VoiceDesign

```
print("Chargement du modèle VoiceDesign...")
voice_design_model = Qwen3TTSModel.from_pretrained(
    "Qwen/Qwen3-TTS-12Hz-1.7B-VoiceDesign",
    device_map="cuda:0",
    dtype=torch.bfloat16,
    attn_implementation=ATTN_IMPL,
)
print("✅ Modèle chargé !")
```

1.2 — Comment décrire une voix

Tu décris la voix en langage naturel avec ces critères :

Critère	Exemples
Âge et genre	"Femme de 25 ans", "Homme âgé de 70 ans", "Adolescent de 16 ans"
Qualité vocale	"grave", "aiguë", "rauque", "douce", "nasale", "claire", "rocailleuse"
Émotion/Ton	"joyeuse", "triste", "mystérieuse", "confiante", "nerveuse"
Style de parole	"chuchotant", "énergique", "calme", "théâtral", "lent", "rapide"
Personnage	"comme un narrateur de documentaire", "comme un personnage d'anime"

1.3 — Générer avec une description

```
wavs, sr = voice_design_model.generate_voice_design(
    text="Le texte que la voix doit prononcer.",
    language="French",
    instruct="Description détaillée de la voix souhaitée."
)

# Sauvegarder et écouter
sf.write("audio_outputs/ma_voix.wav", wavs[0], sr)
display(Audio(wavs[0], rate=sr))
```

1.4 — Exemples de descriptions de voix

Narrateur de documentaire

```
wavs, sr = voice_design_model.generate_voice_design(
    text="Dans les profondeurs de l'océan, là où la lumière du soleil ne pénètre pas, vit un monde de créatures extraordinaires.",
    language="French",
    instruct="Homme de 55 ans, voix grave de baryton. Parle lentement et délibérément avec gravité.
```

```
Style narrateur de documentaire professionnel. Calme, autoritaire et captivant."
)
```

Personnage anime mignon

```
wavs, sr = voice_design_model.generate_voice_design(
    text="Bonjour tout le monde ! Je suis tellement contente de vous rencontrer ! On va bien s'amuser ensemble !",
    language="French",
    instruct="Fille de 16 ans, voix très aiguë et énergique. Parle avec enthousiasme et excitation, ton joyeux comme un personnage d'anime. Mignonne et pétillante."
)
```

Méchant mystérieux

```
wavs, sr = voice_design_model.generate_voice_design(
    text="Tu pensais pouvoir t'échapper ? Comme c'est délicieusement naïf. Le jeu ne fait que commencer.",
    language="French",
    instruct="Homme d'âge moyen, voix grave et soyeuse. Parle lentement avec des sous-entendus menaçants. Confiant et légèrement moqueur. Chaque mot est délibéré."
)
```

Adolescent nerveux

```
wavs, sr = voice_design_model.generate_voice_design(
    text="Euh, s-salut... Je me demandais si peut-être... tu sais... si t'es pas occupée... on pourrait réviser ensemble ?",
    language="French",
    instruct="Garçon de 17 ans, registre ténor. Nerveux et timide, voix légèrement tremblante. Parle avec hésitation et des pauses, trébuchant sur les mots."
)
```

Vieux sorcier sage

```
wavs, sr = voice_design_model.generate_voice_design(
    text="Ah, jeune apprenti, tu cherches la connaissance des arts anciens. Très bien, je vais t'enseigner.",
    language="French",
    instruct="Homme âgé de 70 ans, voix grave et rocailleuse avec sagesse. Parle lentement et pensivement, comme si chaque mot avait du poids. Légère qualité mystique, chaleureux mais autoritaire."
)
```

Présentatrice de journal TV

```
wavs, sr = voice_design_model.generate_voice_design(
    text="Bonsoir et bienvenue dans votre journal de vingt heures. Ce soir, les principales informations.",
    language="French",
```

```
    instruct="Femme de 40 ans, voix claire et professionnelle. Articulation parfaite, ton neutre et posé. Style présentatrice de journal télévisé, sérieuse et crédible."
)
```

1.5 — Même texte, voix différentes

```
texte = "Je n'arrive pas à croire que ça se passe vraiment en ce moment."

voix = [
    ("Enfant excité", "Fille de 8 ans, voix très aiguë et excitée. Parle vite avec un enthousiasme haletant."),
    ("Adulte épuisé", "Femme de 35 ans, fatiguée et exaspérée. Soupire, parle lentement avec de la fatigue."),
    ("Ancien choqué", "Homme de 70 ans, voix rauque, choqué et incrédule. Parle avec une surprise tremblante."),
    ("Ado sarcastique", "Fille de 16 ans, sarcasme dégoulinant. Monotone avec une incrédulité exagérée."),
]

for nom, description in voix:
    wavs, sr = voice_design_model.generate_voice_design(
        text=texte,
        language="French",
        instruct=description
    )
    sf.write(f"audio_outputs/voix_{nom.lower().replace(' ', '_')}.wav", wavs[0], sr)
    print(f"✅ {nom} généré")
```

PARTIE 2 : Contrôle des émotions (CustomVoice 1.7B)

Le modèle **CustomVoice 1.7B** permet de contrôler l'émotion et le style des 9 voix préréglées.

2.1 — Charger le modèle CustomVoice

```
# Libérer la mémoire du modèle précédent
del voice_design_model
torch.cuda.empty_cache()

print("Chargement du modèle CustomVoice 1.7B...")
custom_voice_model = Qwen3TTSModel.from_pretrained(
    "Qwen/Qwen3-TTS-12Hz-1.7B-CustomVoice",
    device_map="cuda:0",
    dtype=torch.bfloat16,
    attn_implementation=ATTN_IMPL,
)
print("✅ Modèle chargé !")

# Voir les voix disponibles
speakers = custom_voice_model.get_supported_speakers()
print(f"Voix disponibles : {', '.join(speakers)}")
```

2.2 — Générer avec une instruction d'émotion

```
wavs, sr = custom_voice_model.generate_custom_voice(
    text="Le texte à prononcer.",
    language="French",
    speaker="Ryan",                    # Voix préréglée
    instruct="Instruction d'émotion"   # Comment la dire
)
```

2.3 — Liste des instructions d'émotion

Émotions de base

Émotion	Instruction en français	Instruction en anglais
😊 Joyeux	`"Ton très joyeux et excité"`	`"Very happy and excited tone"`
😢 Triste	`"Triste et mélancolique, voix légèrement brisée"`	`"Sad and melancholic, voice breaking slightly"`
😠 En colère	`"En colère et frustré, parlant avec force"`	`"Angry and frustrated, speaking forcefully"`
😨 Apeuré	`"Effrayé et anxieux, voix tremblante"`	`"Scared and anxious, voice trembling"`
😐 Neutre	`""` (vide)	`""` (empty)

Styles de parole

Style	Instruction
🤫 Chuchotement	`"Chuchotant très doucement et silencieusement"`
📢 Fort	`"Parlant fort et clairement, projetant la voix"`
🏃 Rapide	`"Parlant très vite, rythme pressé"`
🐢 Lent	`"Parlant très lentement et délibérément, chaque mot soigné"`
🎭 Dramatique	`"Très dramatique et théâtral, comme dans une pièce"`

Combinaisons

Tu peux combiner émotion + style :

- `"Joyeux mais chuchotant, comme si on partageait un secret"`
- `"Triste et parlant très lentement"`
- `"En colère mais contenu, tension dans la voix"`

2.4 — Exemple : Même texte, émotions différentes

```
texte = "Je viens d'apprendre ce qui s'est passé hier."
speaker = "Ryan"

emotions = [
    ("Joyeux", "Ton très joyeux et excité"),
    ("Triste", "Triste et mélancolique, voix légèrement brisée"),
    ("En colère", "En colère et frustré, parlant avec force"),
```

```
    ("Apeuré", "Effrayé et anxieux, voix tremblante"),
    ("Neutre", ""),
]

for nom, instruction in emotions:
    wavs, sr = custom_voice_model.generate_custom_voice(
        text=texte,
        language="French",
        speaker=speaker,
        instruct=instruction
    )
    sf.write(f"audio_outputs/emotion_{nom.lower()}.wav", wavs[0], sr)
    print(f"✅ {nom} généré")
```

2.5 — Scénarios de jeu de rôle

```
scenarios = [
    (
        "Ryan",
        "Bienvenue au match de championnat de ce soir ! La tension ici est absolument électrique !",
        "Commentateur sportif, très énergique et excité, créant de l'engouement",
        "commentateur_sportif"
    ),
    (
        "Serena",
        "Si vous regardez la diapositive trois, vous verrez que notre chiffre d'affaires a augmenté de
quinze pour cent.",
        "Présentation professionnelle d'entreprise, confiante et claire, ton corporate",
        "presentatrice_business"
    ),
    (
        "Aiden",
        "Et alors le dragon s'est retourné pour faire face à notre groupe... lancez l'initiative !",
        "Maître du donjon narrant un jeu de rôle, mystérieux et dramatique, créant du suspense",
        "maitre_donjon"
    ),
]

for speaker, texte, instruction, filename in scenarios:
    wavs, sr = custom_voice_model.generate_custom_voice(
        text=texte,
        language="French",
        speaker=speaker,
        instruct=instruction
    )
    sf.write(f"audio_outputs/{filename}.wav", wavs[0], sr)
    print(f"✅ {filename} généré")
```

PARTIE 3 : Clonage vocal (Base)

Le modèle **Base** permet de cloner n'importe quelle voix à partir d'un échantillon audio de 3-30 secondes.

3.1 — Charger le modèle Base

```
# Libérer la mémoire du modèle précédent
del custom_voice_model
torch.cuda.empty_cache()

print("Chargement du modèle Base (clonage)...")
clone_model = Qwen3TTSModel.from_pretrained(
    "Qwen/Qwen3-TTS-12Hz-1.7B-Base",
    device_map="cuda:0",
    dtype=torch.bfloat16,
    attn_implementation=ATTN_IMPL,
)
print("✅ Modèle chargé !")
```

3.2 — Ce qu'il te faut pour cloner

Élément	Description	Recommandation
Audio de référence	Fichier audio de la voix à cloner	3-30 secondes, qualité propre
Transcription	Texte exact de ce qui est dit dans l'audio	Doit être précis

Formats audio supportés

- MP3, WAV, FLAC, OGG
- Fichier local ou URL

3.3 — Cloner une voix (méthode simple)

```
wavs, sr = clone_model.generate_voice_clone(
    text="Nouveau texte à prononcer avec la voix clonée.",
    language="French",
    ref_audio="/chemin/vers/audio_reference.mp3",
    ref_text="Transcription exacte de l'audio de référence.",
)

sf.write("audio_outputs/voix_clonee.wav", wavs[0], sr)
display(Audio(wavs[0], rate=sr))
```

3.4 — Créer un prompt réutilisable (méthode efficace)

Pour générer plusieurs audios avec la même voix clonée, crée le prompt une seule fois :

```
# Étape 1 : Créer le prompt (une seule fois)
voice_prompt = clone_model.create_voice_clone_prompt(
    ref_audio="/chemin/vers/audio_reference.mp3",
    ref_text="Transcription exacte de l'audio de référence.",
)
print("✅ Prompt vocal créé !")

# Étape 2 : Réutiliser pour plusieurs générations
textes = [
```

```
    "Première phrase avec la voix clonée.",
    "Deuxième phrase avec la voix clonée.",
    "Troisième phrase avec la voix clonée.",
    "Quatrième phrase avec la voix clonée.",
]

wavs, sr = clone_model.generate_voice_clone(
    text=textes,
    language=["French"] * len(textes),
    voice_clone_prompt=voice_prompt,
)

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

3.5 — Cloner depuis une URL

```
# Audio hébergé en ligne
ref_audio_url = "https://exemple.com/mon_audio.mp3"
ref_text = "Ce que dit la personne dans l'audio."

wavs, sr = clone_model.generate_voice_clone(
    text="Nouveau texte avec cette voix.",
    language="French",
    ref_audio=ref_audio_url,
    ref_text=ref_text,
)
```

3.6 — Cloner depuis un tableau numpy

Si tu as déjà l'audio en mémoire :

```
# audio_array = numpy array de l'audio
# sample_rate = fréquence d'échantillonnage

voice_prompt = clone_model.create_voice_clone_prompt(
    ref_audio=(audio_array, sample_rate),
    ref_text="Transcription de l'audio.",
)
```

PARTIE 4 : Flux avancé — Conception → Clonage

Le workflow ultime : **créer** une voix personnalisée avec VoiceDesign, puis la **cloner** pour la réutiliser efficacement.

4.1 — Pourquoi ce workflow ?

Étape	Modèle	Avantage
Conception	VoiceDesign	Contrôle total sur la voix créée

Clonage	Base	Réutilisation rapide et efficace

4.2 — Étape 1 : Concevoir la voix parfaite

```
# Charger VoiceDesign
design_model = Qwen3TTSModel.from_pretrained(
    "Qwen/Qwen3-TTS-12Hz-1.7B-VoiceDesign",
    device_map="cuda:0",
    dtype=torch.bfloat16,
    attn_implementation=ATTN_IMPL,
)

# Définir le personnage
character_description = """
Homme âgé de 70 ans, voix grave et rocailleuse avec sagesse.
Parle lentement et pensivement, comme si chaque mot avait du poids.
Légère qualité mystique, comme un vieux sorcier qui a vu de nombreux âges.
Chaleureux mais autoritaire.
""".strip()

reference_text = "Ah, jeune apprenti, tu cherches la connaissance des arts anciens. Très bien, je vais
t'enseigner."

# Générer la voix de référence
print("🎨 Génération de la voix de référence...")
ref_wavs, sr = design_model.generate_voice_design(
    text=reference_text,
    language="French",
    instruct=character_description
)

# Sauvegarder comme référence
sf.write("audio_outputs/sorcier_reference.wav", ref_wavs[0], sr)
print("✅ Voix de référence créée !")
display(Audio(ref_wavs[0], rate=sr))
```

4.3 — Étape 2 : Créer le clone réutilisable

```
# Libérer la mémoire
del design_model
torch.cuda.empty_cache()

# Charger le modèle de clonage
clone_model = Qwen3TTSModel.from_pretrained(
    "Qwen/Qwen3-TTS-12Hz-1.7B-Base",
    device_map="cuda:0",
    dtype=torch.bfloat16,
    attn_implementation=ATTN_IMPL,
)

# Créer le prompt de clonage à partir de la voix conçue
wizard_prompt = clone_model.create_voice_clone_prompt(
    ref_audio=(ref_wavs[0], sr),
```

```
    ref_text=reference_text,
)
print("✅ Prompt de clonage créé !")
```

4.4 — Étape 3 : Générer tout le dialogue

```
# Dialogues du personnage
dialogues_sorcier = [
    "Le chemin de la magie est long et périlleux. Beaucoup ont essayé, peu ont réussi.",
    "En sept cents ans d'existence, j'ai appris que la patience est la plus grande vertu.",
    "Méfie-toi de la forêt sombre à l'est. Le mal s'y réveille, ancien et puissant.",
    "Tu as bien travaillé, jeune apprenti. Peut-être y a-t-il de l'espoir pour ce monde après tout.",
    "Maintenant, commençons ton entraînement. Vide ton esprit et concentre-toi.",
]

print("🧙 Génération des dialogues du sorcier...\n")

for i, ligne in enumerate(dialogues_sorcier):
    print(f"📜 Réplique {i+1} : \"{ligne[:40]}...\"")

    wavs, sr = clone_model.generate_voice_clone(
        text=ligne,
        language="French",
        voice_clone_prompt=wizard_prompt,
    )

    sf.write(f"audio_outputs/sorcier_ligne_{i+1}.wav", wavs[0], sr)
    display(Audio(wavs[0], rate=sr))

print("\n✅ Tous les dialogues générés !")
```

Téléchargement des fichiers

```
import shutil
from google.colab import files

# Compresser tous les fichiers audio
shutil.make_archive("qwen3_tts_avance", 'zip', "audio_outputs")

# Afficher les fichiers inclus
print("📦 Fichiers générés :")
for f in sorted(os.listdir("audio_outputs")):
    if f.endswith('.wav'):
        print(f"   📄 {f}")

# Télécharger
files.download("qwen3_tts_avance.zip")
```

Résumé des 3 fonctionnalités

```
┌──────────────────────────────────────────────────────┐
│                  VOICE DESIGN (1.7B)                 │
│        "Décris une voix" → Obtiens cette voix        │
│                                                      │
│   Entrée :  text + language + instruct (description) │
│   Sortie :  Audio avec une voix entièrement nouvelle │
│                                                      │
│   Exemple : instruct = "Femme 25 ans, douce, chaleureuse"  │
└──────────────────────────────────────────────────────┘

┌──────────────────────────────────────────────────────┐
│                CUSTOM VOICE 1.7B                     │
│     Voix préréglée + instruction d'émotion/style     │
│                                                      │
│   Entrée :  text + language + speaker + instruct     │
│   Sortie :  Voix préréglée avec l'émotion demandée   │
│                                                      │
│   Exemple : speaker="Ryan" + instruct="Triste"       │
└──────────────────────────────────────────────────────┘

┌──────────────────────────────────────────────────────┐
│                    BASE (Clonage)                    │
│        Audio de référence → Clone cette voix         │
│                                                      │
│   Entrée :  text + language + ref_audio + ref_text   │
│   Sortie :  Nouveau texte avec la voix clonée        │
│                                                      │
│   Exemple : ref_audio="ma_voix.mp3" + ref_text="..." │
└──────────────────────────────────────────────────────┘
```

Conseils pour de meilleurs résultats

Conception de voix (VoiceDesign)

Conseil	Exemple
Sois spécifique sur l'âge	"Homme de 45 ans" plutôt que "homme adulte"
Décris la qualité vocale	"voix rauque", "timbre clair", "nasale"
Indique l'émotion	"joyeuse", "mélancolique", "confiante"
Précise le style	"parle lentement", "énergique", "chuchotant"
Donne un contexte	"comme un narrateur de documentaire"

Contrôle des émotions (CustomVoice)

Conseil	Exemple
Instructions concises	"Très triste" plutôt qu'une longue description
Combine émotion + style	"Joyeux mais chuchotant"
Utilise l'intensité	"légèrement triste", "extrêmement en colère"

Clonage vocal (Base)

Conseil	Exemple
Audio propre	Pas de bruit de fond, pas de musique
Durée idéale	3-30 secondes
Transcription précise	Mot pour mot ce qui est dit
Crée des prompts réutilisables	Pour générer plusieurs audios efficacement

---

Dépannage

Problème	Cause	Solution
`CUDA out of memory`	Pas assez de VRAM	Utiliser `torch.cuda.empty_cache()` entre les modèles
Voix ne correspond pas à la description	Description trop vague	Être plus spécifique dans l'instruct
Clonage de mauvaise qualité	Audio de référence bruyant	Utiliser un audio plus propre
Téléchargement très long	Premier lancement	Normal, les modèles 1.7B font ~3.8 Go

---

Ressources

- 🤗 Collection Hugging Face
- 📝 Dépôt GitHub
- 📖 Blog technique
- 🎮 Démo en ligne