

Text To Speech mit Python – So vertonst du Texte



In diesem Tutorial lernst du zwei Methoden kennen, **mit denen du in Python Text in gesprochene Sprache umwandeln kannst**. Mit Text to Speech kannst in deinen Projekten zum Beispiel eine Sprachausgabe umsetzen.

Die erste Möglichkeit ist die Python-Bibliothek **gTTS**, mit der du **kostenlos** Texte vertonen lassen kannst. Dieses Modul habe ich im Projekt [ChatGPT im Telefon](#) eingesetzt. Als zweite Methode lernst du eine API-Funktion von openAI kennen. Diese Variante ist kostenpflichtig – allerdings hört sich das Ergebnis hierfür auch weit besser an.

Text to Speech mit gTTS (Google Text-to-Speech)

Wenn dein Projekt keine astreine Aussprache erfordert, ist die Bibliothek gTTS eine gute Wahl. Die Qualität ist nicht schlecht, allerdings hakt es bei der Aussprache oft bei Abkürzungen oder die Betonung von Satzteilen kommt durch das ein oder andere Komma durcheinander. Dafür kannst du mit diesem Python-Modul den Google-Service kostenlos verwenden – was sicherlich ein gutes Argument für einen Test ist.

Installiere zunächst die Bibliothek mit dem Befehl

```
pip install gtts
```

Um die Sprachausgabe zu testen, reichen drei Zeilen Python-Code:

```
from gtts import gTTS

tts = gTTS('Hello, world. This is a first test.')
tts.save('hello.mp3')
```

Nachdem du das kleine Script ausgeführt hast, öffne die Datei **hello.mp3** und lausche dem Ergebnis. Bist du zufrieden?

Du kannst übrigens auch deutsche Texte vertonen lassen. Füge hierfür den Parameter **lang='de'** hinzu:

```
tts = gTTS('Hallo, das ist einer erster Test.', lang='de')
```

Das war im Prinzip schon alles. Wenn du wissen möchtest, wie du die MP3 direkt mit deinem Python-Script abspielen kannst, wirf einen Blick in das oben verlinkte Projekt. Weitere Infos über gTTS erhältst du [hier](#).

Text To Speech mit openAI

Wenn dir eine gute Sprachqualität ein paar Cent wert ist, ist OpenAI einen Versuch wert. Neben dem allseits bekannten ChatGPT findest du dort auch eine API-Funktion, mit der du Text vertonen lassen kannst. Die Integration in dein Python-Script ist dabei ähnlich einfach wie mit **gTTS**. **Allerdings kostet der Service derzeit (November 2023) 0,015 € je 1.000 Zeichen** – was ein recht überschaubarer Preis ist. Die aktuelle Preisliste findest du [hier unter Audio models](#).

Wenn du noch nicht mit OpenAI experimentiert hast, [erfährst du in diesem Tutorial, wie du dort ein Konto und einen API-Key erstellst](#).

Nachdem du die Bibliothek **openai** (mit **pip install openai**)

installiert hast, binde sie in deinem Python-Script ein. Zusätzlich benötigst du noch das Modul **pathlib** der Bibliothek **Path**, das aber bereits vorinstalliert ist.

```
from pathlib import Path
from openai import OpenAI
```

Anschließend hinterlegst du deine API-Key von OpenAI:

```
client = OpenAI(
    api_key="DEIN API-KEY",
)
```

Und schon kann es mit der Vertonung losgehen. Du gibst zunächst an, wie die erzeugte MP3 heißen soll – hier **speech.mp3**:

```
speech_file_path = Path(__file__).parent / "speech.mp3"
```

Anschließend legst du ein paar Parameter fest: Zunächst das Modell – hier hast du die Wahl zwischen **tts-1** und **tts-1-hd**. Letzteres hat eine etwas höhere Qualität und ist auch teurer. Für die allermeisten Anwendungen dürfte das einfachere Modell jedoch ausreichen. Der Parameter **voice** gibt vor, welche Stimme verwendet werden soll. Derzeit gibt es sechs Stimmen, die du [hier probenhören kannst](#). Dort findest du auch aktuelle Informationen und Updates zu Text to Speech mit OpenAI.

Zuletzt fehlt nur noch der Text, den du vertonen lassen möchtest und der Befehl zum Speichern der MP3:

```
response = client.audio.speech.create(
    model="tts-1",
    voice="alloy",
    input="Das Pferd frisst keinen Gurkensalat."
)
```

```
response.stream_to_file(speech_file_path)
```

Und das war schon alles. Führe das Script aus – sobald es fertig ist, findest du im gleichen Ordner die Datei **speech.mp3** mit deiner Sprachausgabe. Hier nun das gesamte Script:

```
from pathlib import Path
from openai import OpenAI

client = OpenAI(
    api_key="DEIN API-KEY",
)

speech_file_path = Path(__file__).parent / "speech.mp3"
response = client.audio.speech.create(
    model="tts-1",
    voice="alloy",
    input="Das Pferd frisst keinen Gurkensalat."
)

response.stream_to_file(speech_file_path)
```

Mit den oben beschriebenen Bibliotheken und Services hast du nun zwei Methoden zur Hand, wie du in deinem Projekt Text to Speech anwenden kannst. Ob dir eine kostenlose Vertonung reicht, oder du etwas qualitativ hochwertigeres benötigst, hängt natürlich vom Einsatzgebiet ab.