

Die ChatGPT API mit Python verwenden



ChatGPT im Browser oder als App zu verwenden, hat sich mittlerweile bei einer Vielzahl von Menschen etabliert. Aber selbstverständlich stellt OpenAI auch eine **ChatGPT API** zur Verfügung, die du ansteuern kannst, um zum Beispiel Texte erstellen oder dir Fragen beantworten zu lassen.

In diesem Tutorial lernst du, wie du

- dir einen OpenAI Account und den benötigten API Key erstellst
- mit Python Prompts erstellst und die Antwort von ChatGPT erhältst
- zwischen den Modellen 3.5 und 4 wechselst
- ChatGPT präzise Anweisungen gibst, die du nicht in jedem Prompt wiederholen musst

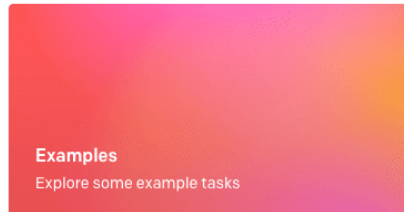
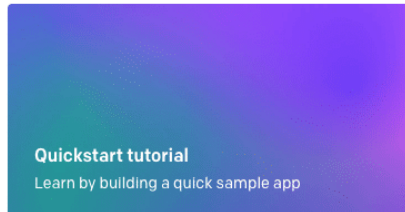
Einen OpenAI Account erstellen

Um die ChatGPT API nutzen zu können, benötigst du einen Account bei OpenAI. Wenn du allerdings bereits den Chatbot nutzt, musst du natürlich kein Extra-Konto erstellen. [Besuche diese Webseite](#), um einen Account zu erstellen oder dich einzuloggen.

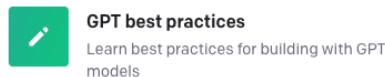
Anschließend landest du auf der Startseite. Dort warten eine Vielzahl von Tutorials auf dich, die du aber zunächst links liegen lässt.

Welcome to the OpenAI platform

Start with the basics

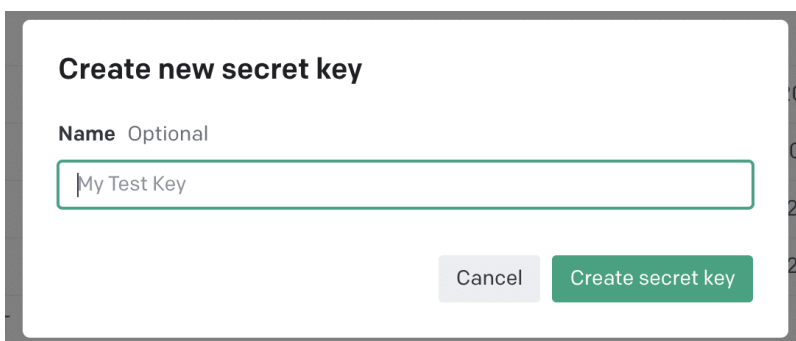


Build an application



Einen API Key erstellen

Klicke stattdessen oben rechts auf **Personal** und wähle den Menüpunkt **View API keys**. Auf der folgenden Seite findest du den Button **Create new secret key**. Nach einem Klick hierauf kannst du deinem neuen Key einen (am besten sprechenden) Namen geben:

A modal dialog box titled "Create new secret key". It has a label "Name Optional" above a text input field containing "My Test Key". At the bottom right are two buttons: "Cancel" and "Create secret key".

Create new secret key

Name Optional

My Test Key

Cancel Create secret key

Ein weiterer Klick auf **Create secret key** erstellt deinen neuen Key, den du dir **direkt herauskopieren und ablegen** solltest.

Später kannst du den Key nicht mehr aufrufen – allerdings kannst du in diesem Fall einfach einen neuen erstellen.

Zahlungsdaten hinterlegen

Die ChatGPT API ist nicht kostenlos – deshalb musst du dein Konto mit einem Betrag deiner Wahl (mindestens jedoch 5 \$) aufladen musst. Hierfür stehen dir verschiedene Zahlungsmethoden wie PayPal oder Kreditkarte zur Verfügung.

Bei OpenAI findest du die [aktuellen Preise für die Nutzung der API](#). Sobald dein Konto mit ein paar Dollar aufgeladen ist, kann es direkt mit dem Python Script weitergehen.

Stelle eine Frage mit Python

Wie du bestimmt schon weißt, funktioniert ChatGPT wie eine Art Chatbot – du stellst eine Frage oder gibst eine Anweisung und erhältst darauf eine Antwort. Diese Frage wird auch **Prompt** genannt. Das folgende Python Script stellt die Verbindung zur ChatGPT API her, übermittelt deinen Prompt und gibt die Antwort aus.

```
from openai import OpenAI
client = OpenAI(
    api_key="DEIN API KEY",
)

completion = client.chat.completions.create(
    model="gpt-3.5-turbo",
    messages=[
        {"role": "user", "content": "Wie lautet die Antwort auf alle Fragen?"}
    ]
)

print(completion.choices[0].message.content)
```

In der ersten Zeile bindest du die Bibliothek **openai** ein. Diese musst du natürlich vorher noch installieren. Führe hierfür zum Beispiel im Terminal folgende Zeile aus:

```
pip install openai
```

Ein kleiner Hinweis: Der Code oben funktioniert nur mit Versionen der Bibliothek ab dem 8.11.23 – sollte deine Version älter sein, führe zunächst ein Update mit **pip install --upgrade openai** durch. Zurück zum Script: Nachdem du die Bibliothek eingebunden hast, hinterlegst du in der Variablen **api_key** deinen API Key, den du vorhin erstellst hast.

Und noch ein Hinweis: In der offiziellen Dokumentation von openAI werden zwei andere Verfahren vorgeschlagen, um den API-KEY zu hinterlegen. Diese haben bei mir allerdings nicht zuverlässig funktioniert. Falls du sie ausprobieren möchtest, kannst du [hier mehr erfahren](#).

Anschließend folgt die Abfrage mit der Funktion **client.chat.completions.create()**. Hier legst du zunächst das Modell fest, im obigen Beispiel ist das **GPT-3.5-turbo**. Du kannst seit Juli 2023 auch das bessere Modell **GPT-4** verwenden. **Aber Achtung:** ChatGPT 4 ist aktuell (September 2023) 20 Mal teurer als ChatGPT 3.5 – du kannst die Preise über den Link oben vergleichen. Die dort aufgelisteten Preise beziehen sich jeweils auf 1.000 Tokens. Laut OpenAI entspricht das ungefähr 750 Wörtern (im Englischen).

Letztlich musst du selbst entscheiden, wie gut die Antworten von ChatGPT innerhalb deines Projekts ausfallen sollen. Wenn du dich für das Modell 4 entscheidest ersetze die entsprechende Zeile im Python Script durch:

```
model="gpt-4",
```

Update (Feb. 2024): Mittlerweile gibt es auch ein **GPT-4 Turbo**, das auf Trainingsdaten bis Dezember 2023 basiert. Dieses

Modell kannst du mit „**gpt-4-0125-preview**“ einbinden und verwenden. Weitere Informationen (auch zu kommenden Modellen) findest du [hier](#).

Falls du in deinem Script immer das neueste GPT-4-Modell verwenden möchtest, bietet die ChatGPT API nun hierfür eine Möglichkeit: Verwende einfach das Modell **gpt-4-turbo-preview**, das stets auf die aktuellste Version verweist. Weitere Informationen hierüber findest du [hier](#).

Der Prompt

Kommen wir zur Frage, die du stellst und dafür an die ChatGPT API sendest. Diese findest du im Script hier:

```
messages=[
    {"role": "user", "content": "Wie lautet die Antwort auf
alle Fragen?"}]
)
```

Hier definierst du zunächst die **Rolle (role)**, hier also **user**. Diese Rolle hat die Person, die eine Frage stellt – also genauso wie du es vielleicht vom Browser oder der App gewohnt bist. Dahinter folgt der **content**, also die eigentliche Frage oder Anweisung.

Nun muss nur noch die Antwort ausgegeben werden:

```
print(completion.choices[0].message.content)
```

In deinem Terminal oder Konsole sollte nun die Antwort erscheinen, die du sicherlich auch schon vorher wusstest. □

Weitere Rollen für DIE ChatGPT API

Oben hast du die Rolle **user** verwendet. Du kannst ChatGPT aber

auch vor deiner eigentlichen Frage genauere Anweisungen mitgeben und damit versuchen, das System etwas zu „lenken“. Hierfür fügst du vorab etwas mit der Rolle **system** ein:

```
messages=[
    {"role": "system", "content": "Du beantwortest Fragen kurz und präzise."},
    {"role": "user", "content": "Wie lautet die Antwort auf alle Fragen?"}]
)
```

Das ist besonders nützlich, wenn ChatGPT zum Beispiel Antworten in einem bestimmten Format geben soll. Du musst diese Anweisungen also nicht in deinem eigentlichen Prompt unterbringen, sondern kannst sie separat erstellen und verwenden.

Und das war es schon. Du hast nun das grundlegende Python Script, um per API mit ChatGPT zu kommunizieren. Werf einen Blick in die [Raspberry Pi Projekte](#), um Anwendungsmöglichkeiten zu entdecken. Wenn du dich besonders für den Bereich künstlicher Intelligenz interessierst, wirf doch mal einen Blick in unsere [KI-Projekte](#).