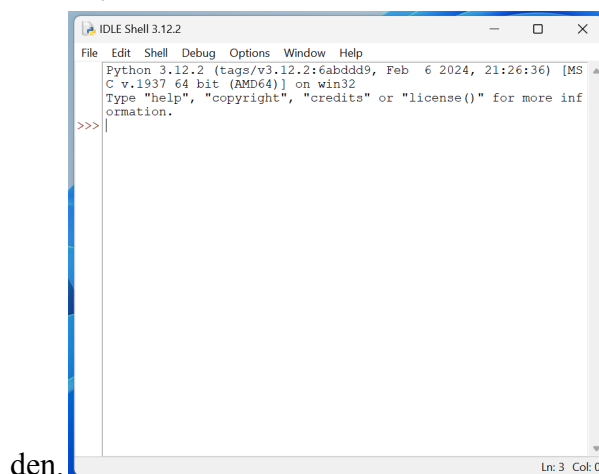


TeachGPT med Python

Kapitel 1: Installera Python

- 1) Gå till <https://www.python.org/downloads/> och tryck på Download Python.
- 2) Någon gång under installationen dyker en ruta “**Disable path limit**” upp. Se till att klicka i den rutan. Det är möjligt att detta inte behövs men i vissa fall kan det ställa till problem senare.
- 3) När installationen är klar, sök på IDLE på datorn och öppna programmet. IDLE är ett program där man kan skriva och köra pythonkod. Det är inte den mest praktiska editorn, men vi kommer använda den här eftersom det är lätt att komma igång med



- 4) Skapa en ny fil med **File -> New File**. Skriv in **print("Hello World")** i det nya fönstret och spara filen genom att gå till **File -> Save**. Du kan namnge filen vad som helst och spara den var som helst (det kan vara en bra idé att ha filerna på ett ställe där du lätt hittar dem, och inte i samma mapp som IDLE). Därefter kan du köra programmet genom att gå till **Run -> Run Module**. Nu borde **Hello World** skrivas ut i det första fönstret.

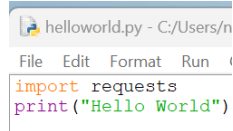
Kapitel 2: Installera requests

requests är en pythonmodul som låter en skicka HTTP-requests. Vi kommer behöva det för att komma åt API:er. Men då måste vi kunna installera moduler. Vanligtvis görs detta med hjälp av **pip**, men på Windows är det lite krångligt att få det att funka. Men det här fungerade för mig:

- 1) Sök på **terminal** på datorn, och öppna terminalen. Skriv in **py -3 -m pip install requests** och tryck på enter. Förhoppningsvis dyker det upp en massa gröna progress bars och det står “Successfully installed...” längst ner. Jag fick också ett gult

varningsmeddelande om något som saknas på PATH, men det verkade inte vara något problem.

- 2) I ditt Hello World-program, lägg till raden **import requests** högst upp och kör programmet. Förhoppningsvis fungerar programmet precis som innan.



Kapitel 3: API

- 1) Gå till <https://teachgpt.ssis.nu/user/api-tokens> och skapa en API-nyckel. Spara nyckeln någonstans.
- 2) Skapa en Python-fil från IDLE och klistra in följande kod, men byt ut **X** mot API-nyckeln:

```
import requests

api_key = "X"
url = "https://teachgpt.ssis.nu/api/v1/chat/completions"

headers = {
    "Content-Type": "application/json",
    "Authorization": f"Bearer {api_key}",
}

payload = {
    "model": "Yi-34B-Chat-AWQ",
    "messages": [
        {
            "role": "system",
            "content": "You are a helpful assistant."
        },
        {
            "role": "user",
            "content": "Hello!"
        }
    ]
}

response = requests.post(url, json=payload, headers=headers)

if response.status_code == 200:
    print("Success:")
    print(response.json())
else:
    print("Error:")
    print(response.text)
```

- 3) Kör koden. Förhoppningsvis står det **Success:** följt av en massa text. Det här är svaret från API:et i form av en Python-dictionary. Om du vill komma åt själva texten som språkmodellen gav så är det `response.json()['choices'][0]['message']['content']`.

Kapitel 4: Vad kan vi göra mer?

Nu har vi ett sätt att skicka frågor och få tillbaka svar från språkmodellerna. Det finns säkert en massa spännande sätt att använda språkmodeller som en komponent i större programmeringsprojekt. Men koden ovan är fortfarande ganska otillräcklig, den duger till exempel inte för att hålla en längre konversation där språkmodellen kommer ihåg vad som sagts. Då måste man också ta med historiken i varje request. Det finns säkert ett bekvämt sätt att göra det på med API:et, men jag har inte kollat upp vad mer för funktioner som finns.