uv Installation & LLM

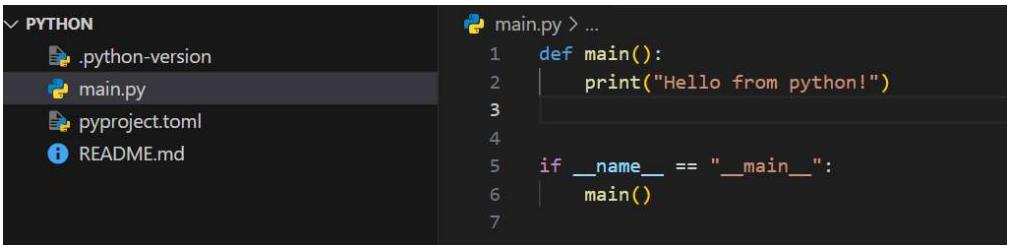
30 July 2025 22:44

```
C:\Users\varsh\OneDrive\Desktop\python>winget install --id=astral-sh.uv  -e
Found uv [astral-sh.uv] Version 0.8.3
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
This package requires the following dependencies:
  - Packages
      Microsoft.VCRedist.2015+.x64
Downloading https://github.com/astral-sh/uv/releases/download/0.8.3/uv-x86_64-pc-windows-msvc.zip
  18.0 MB / 18.0 MB
Successfully verified installer hash
Extracting archive...
Successfully extracted archive
Starting package install...
Path environment variable modified; restart your shell to use the new value.
Command line alias added: "uvx"
Command line alias added: "uv"
Command line alias added: "uvw"
Successfully installed

C:\Users\varsh\OneDrive\Desktop\python>
```

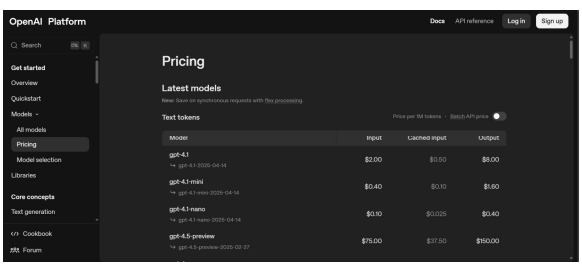
Creating an new Python project using uv

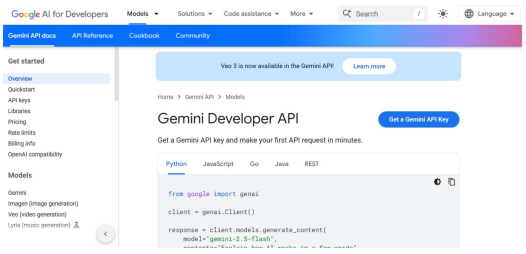
/> uv init

```
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\varsh\OneDrive\Desktop\python>uv --version
uv 0.8.3 (7e78f54e7 2025-07-24)

C:\Users\varsh\OneDrive\Desktop\python>uv init
Initialized project `python`

C:\Users\varsh\OneDrive\Desktop\python>
```



Proprietary LLMs





main.py > ...

```python
def generate_post(user_input: str) -> str:
    #call AI /LLM
    pass


def main():

    # user input => AI (LLM) to generate Posts => output post
    user_input = input("Enter your topic or idea for a post: ")

if __name__ == "__main__":
    main()

```

Install requests -> for that we can use

1. uv add requests
2. pip install requests

```
Enter your topic or idea for a post: open llms are amazing
 ate more ofter than capable enough got many llm provided t
ERROR RESPONSE: {"error":{"message":"ExceededBudget: User=v
, Budget=-1e-05","type":"budget_exceeded","param":null,"cod

Generated Post:

Error: 400 Client Error: Bad Request for url: https://api.b
```

```python
import requests
import json
url = "https://api.blackbox.ai/completions"
user_api_key = "sk-MxSDgmz8IWeweOm3d21I4A"
headers = {
    "Content-Type": "application/json",
    "Authorization": f"Bearer {user_api_key}"
}
def generate_post(user_input: str) -> str:
    payload = {
        "model": "blackboxai/openrouter/cypher-alpha:free",
        "prompt": user_input,
        "max_tokens": 300
    }
    try:
        response = requests.post(url, headers=headers, json=payload)

        # Print exact response if it fails
        if response.status_code != 200:
            print("ERROR RESPONSE:", response.text)
            response.raise_for_status()
        result = response.json()
        print("DEBUG:", json.dumps(result, indent=2))
        if "choices" in result:
            return result["choices"][0].get("text", "").strip()
        else:
            return str(result)
    except Exception as e:
        return f"Error: {e}"

def main():
    user_input = input("Enter your topic or idea for a post: ")
    post = generate_post(user_input)
```

```
    print("\nGenerated Post:\n")
    print(post)

if __name__ == "__main__":
    main()
```

Moving to Open LLMs --> Ollama --> local LLMs

Installed Deepseek

```
C:\Users\varsh>ollama pull deepseek-r1:1.5b
pulling manifest
pulling aabd4debf0c8: 100% ▕▏ 1.1 GB
pulling c5ad996bda6e: 100% ▕▏  556 B
pulling 6e4c38e1172f: 100% ▕▏ 1.1 KB
pulling f4d24e9138dd: 100% ▕▏  148 B
pulling a85fe2a2e58e: 100% ▕▏  487 B
verifying sha256 digest
writing manifest
success

C:\Users\varsh>ollama run deepseek-r1:1.5b
>>> hello
Hello! How can I assist you today? 😊

>>> Send a message (/? for help)
```

```
import requests
import json
# Ollama server runs locally at this endpoint by default
OLLAMA_URL = "http://localhost:11434/api/generate"
def generate_post(user_input: str) -> str:
    payload = {
        "model": "deepseek-r1:1.5b",
        "prompt": user_input,
        "stream": False    # So we get the full response in one go
    }
    try:
        response = requests.post(OLLAMA_URL, json=payload)
        response.raise_for_status()
        result = response.json()
        return result.get("response", "").strip()
    except Exception as e:
        return f"Error: {e}"

def main():
    user_input = input("Enter your topic or idea for a post: ")
    post = generate_post(user_input)
    print("\nGenerated Post:\n")
    print(post)

if __name__ == "__main__":
    main()
```

```
(python) C:\Users\varsh\OneDrive\Desktop\python>C:\Users\varsh\OneDrive\Desktop\python\.venv\Scripts\python.exe c:/Users/varsh/OneDrive/Desktop/python/main.py
Enter your topic or idea for a post: What is the use of LLMs

Generated Post:

<think>

</think>

Large Language Models (LLMs) are powerful tools that can understand, learn, and apply complex knowledge across a wide range of topics. They are designed to perform tasks such as answering questions, providing explanations, generating text, and even acting in various interactive forms like games or storytelling.

### Key Uses of LLMs:

1. **Knowledge Base**:
   - LLMs can act as reliable knowledge repositories that answer questions across a broad spectrum of subjects.
   - They can provide explanations for complex information, helping users understand concepts they might not grasp easily.
```

Ln 34, Col 1 Spaces: 4 UTF-8 LF {} Python python (3.13.2) Go Live Prettier

Till now we used our own creation(carl) but we can directly use the python sdk which will be available in open ai, genai dev.

```python
from google import genai

client = genai.Client(api_key="YOUR_API_KEY")

response = client.models.generate_content(
    model="gemini-2.5-flash", contents="Explain how AI works in a few words"
)
print(response.text)
```

Generate text from a model — python

```python
from openai import OpenAI
client = OpenAI()

response = client.responses.create(
    model="gpt-4.1",
    input="Write a one-sentence bedtime story about a unicorn."
)

print(response.output_text)
```