```
!pip install crewai-tools==0.25.0
!pip install crewai==0.86.0
!pip install langchain_community==0.0.29
```

```
Collecting crewai-tools==0.25.0
  Downloading crewai_tools-0.25.0-py3-none-any.whl.metadata (5.1 kB)
Requirement already satisfied: beautifulsoup4>=4.12.3 in /usr/local/lib/python3.11/dist-packages (from crewai-tools=
Collecting chromadb>=0.4.22 (from crewai-tools==0.25.0)
  Downloading chromadb-1.0.12-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.9 kB)
Collecting crewai>=0.86.0 (from crewai-tools==0.25.0)
  Downloading crewai-0.126.0-py3-none-any.whl.metadata (35 kB)
Collecting docker>=7.1.0 (from crewai-tools==0.25.0)
  Downloading docker-7.1.0-py3-none-any.whl.metadata (3.8 kB)
Collecting docx2txt>=0.8 (from crewai-tools==0.25.0)
  Downloading docx2txt-0.9-py3-none-any.whl.metadata (529 bytes)
Collecting embedchain>=0.1.114 (from crewai-tools==0.25.0)
  Downloading embedchain-0.1.128-py3-none-any.whl.metadata (9.2 kB)
Collecting lancedb>=0.5.4 (from crewai-tools==0.25.0)
  Downloading lancedb-0.23.0-cp39-abi3-manylinux_2_28_x86_64.whl.metadata (4.4 kB)
Requirement already satisfied: openai>=1.12.0 in /usr/local/lib/python3.11/dist-packages (from crewai-tools==0.25.0)
Requirement already satisfied: pydantic>=2.6.1 in /usr/local/lib/python3.11/dist-packages (from crewai-tools==0.25.0
Collecting pyright>=1.1.350 (from crewai-tools==0.25.0)
  Downloading pyright-1.1.401-py3-none-any.whl.metadata (6.6 kB)
Requirement already satisfied: pytest>=8.0.0 in /usr/local/lib/python3.11/dist-packages (from crewai-tools==0.25.0)
Collecting pytube>=15.0.0 (from crewai-tools==0.25.0)
  Downloading pytube-15.0.0-py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: requests>=2.31.0 in /usr/local/lib/python3.11/dist-packages (from crewai-tools==0.25.
Collecting scrapegraph-py>=1.8.0 (from crewai-tools==0.25.0)
  Downloading scrapegraph_py-1.12.0-py3-none-any.whl.metadata (6.4 kB)
Collecting selenium>=4.18.1 (from crewai-tools==0.25.0)
  Downloading selenium-4.33.0-py3-none-any.whl.metadata (7.5 kB)
Collecting serpapi>=0.1.5 (from crewai-tools==0.25.0)
  Downloading serpapi-0.1.5-py2.py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4>=4.12.3
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.11/dist-packages (from beautifulsc
Requirement already satisfied: build>=1.0.3 in /usr/local/lib/python3.11/dist-packages (from chromadb>=0.4.22->crewa
Collecting fastapi==0.115.9 (from chromadb>=0.4.22->crewai-tools==0.25.0)
  Downloading fastapi-0.115.9-py3-none-any.whl.metadata (27 kB)
Requirement already satisfied: uvicorn>=0.18.3 in /usr/local/lib/python3.11/dist-packages (from uvicorn[standard]>=0
Requirement already satisfied: numpy>=1.22.5 in /usr/local/lib/python3.11/dist-packages (from chromadb>=0.4.22->crew
Collecting posthog>=2.4.0 (from chromadb>=0.4.22->crewai-tools==0.25.0)
  Downloading posthog-4.4.0-py2.py3-none-any.whl.metadata (5.5 kB)
Collecting onnxruntime>=1.14.1 (from chromadb>=0.4.22->crewai-tools==0.25.0)
  Downloading onnxruntime-1.22.0-cp311-cp311-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl.metadata (4.5 kB)
Collecting opentelemetry-api>=1.2.0 (from chromadb>=0.4.22->crewai-tools==0.25.0)
  Downloading opentelemetry_api-1.34.0-py3-none-any.whl.metadata (1.5 kB)
Collecting opentelemetry-exporter-otlp-proto-grpc>=1.2.0 (from chromadb>=0.4.22->crewai-tools==0.25.0)
  Downloading opentelemetry_exporter_otlp_proto_grpc-1.34.0-py3-none-any.whl.metadata (2.4 kB)
Collecting opentelemetry-instrumentation-fastapi>=0.41b0 (from chromadb>=0.4.22->crewai-tools==0.25.0)
  Downloading opentelemetry_instrumentation_fastapi-0.55b0-py3-none-any.whl.metadata (2.2 kB)
Collecting opentelemetry-sdk>=1.2.0 (from chromadb>=0.4.22->crewai-tools==0.25.0)
  Downloading opentelemetry_sdk-1.34.0-py3-none-any.whl.metadata (1.6 kB)
Requirement already satisfied: tokenizers>=0.13.2 in /usr/local/lib/python3.11/dist-packages (from chromadb>=0.4.22-
Collecting pypika>=0.48.9 (from chromadb>=0.4.22->crewai-tools==0.25.0)
  Downloading PyPika-0.48.9.tar.gz (67 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.3/67.3 kB 5.4 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: tqdm>=4.65.0 in /usr/local/lib/python3.11/dist-packages (from chromadb>=0.4.22->crewa
Collecting overrides>=7.3.1 (from chromadb>=0.4.22->crewai-tools==0.25.0)
```

```
# Warning control
import warnings
warnings.filterwarnings('ignore')
```

Simple Blog

```
from crewai import LLM
#backup tokens
```

```
#----my token
#  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
#----Deepu token
# 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
#--- Aathi token
# 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
#---Mother token
# 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

#Best Models
# huggingface/mistralai/Mixtral-8x7B-Instruct-v0.1
# huggingface/Qwen/Qwen3-235B-A22B
# huggingface/Qwen/Qwen2.5-VL-32B-Instruct
# huggingface/intfloat/multilingual-e5-large-instruct


#For huggingface models
llm = LLM(
    api_key="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",
    model="huggingface/Qwen/Qwen3-235B-A22B"
)

#Cohere

# llm = LLM(
#     api_key="B0hIn9euSaTNJ64ueOK8BbHGBoKHPMNLpMCApJMj",
#     model="cohere/command-r",
# )
#Huggingface models

#DEEPU TOKEN

# llm = LLM(
#     api_key="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",
#     model="huggingface/mistralai/Ministral-8B-Instruct-2410",
# )

# llm = LLM(
#     api_key="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",
#     model="huggingface/mistralai/Mistral-7B-Instruct-v0.3",
# )


#For huggingface mixtral 8x7B
# llm = LLM(
#     api_key="hf_aypMgLTRyZNFgDhEQJPXhSpJZAiwDzx",
#     model="huggingface/mistralai/Mixtral-8x7B-v0.1",
# )

#Groq
# llm = LLM(
#     api_key="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",
#     model="groq/qwen-2.5-32b",
# )
```

Agents

```
from crewai import Agent, Task, Crew
planner = Agent(
    llm=llm,
    role="Content Planner",
    goal="Plan engaging and factually accurate content on {topic}",
    backstory="You're working on planning a blog article "
              "about the topic: {topic}."
              "You collect information that helps the "
              "audience learn something "
              "and make informed decisions. "
              "Your work is the basis for "
              "the Content Writer to write an article on this topic.",
    allow_delegation=False,
    verbose=True
)
writer = Agent(
    llm=llm,
    role="Content Writer",
    goal="Write insightful and factually accurate "
         "opinion piece about the topic: {topic}",
    backstory="You're working on a writing "
              "a new opinion piece about the topic: {topic}. "
              "You base your writing on the work of "
              "the Content Planner, who provides an outline "
              "and relevant context about the topic. "
              "You follow the main objectives and "
              "direction of the outline, "
              "as provide by the Content Planner. "
              "You also provide objective and impartial insights "
              "and back them up with information "
              "provide by the Content Planner. "
              "You acknowledge in your opinion piece "
              "when your statements are opinions "
              "as opposed to objective statements."
              "Write the blog in a well-readable manner, partitioning it into section
    allow_delegation=False,
    verbose=True
)
editor = Agent(
    llm=llm,
    role="Editor",
    goal="Edit a given blog post to align with "
         "the writing style of the organization. ",
    backstory="You are an editor who receives a blog post "
              "from the Content Writer. "
              "Your goal is to review the blog post "
              "to ensure that it follows journalistic best practices,"
              "provides balanced viewpoints "
              "when providing opinions or assertions, "
              "and also avoids major controversial topics "
              "or opinions when possible.",
    allow_delegation=False,
```

```
    verbose=True
)
```

Tasks

```
plan = Task(
    description=(
        "1. Prioritize the latest trends, key players, "
            "and noteworthy news on {topic}.\n"
        "2. Identify the target audience, considering "
            "their interests and pain points.\n"
        "3. Develop a detailed content outline including "
            "an introduction, key points, and a call to action.\n"
        "4. Include SEO keywords and relevant data or sources."
    ),
    expected_output="A comprehensive content plan document "
        "with an outline, audience analysis, "
        "SEO keywords, and resources.",
    agent=planner,
)
write = Task(
    description=(
        "1. Use the content plan to craft a compelling "
            "blog post on {topic}.\n"
        "2. Incorporate SEO keywords naturally.\n"
        "3. Sections/Subtitles are properly named "
            "in an engaging manner.\n"
        "4. Ensure the post is structured with an "
            "engaging introduction, insightful body, "
            "and a summarizing conclusion.\n"
        "5. Proofread for grammatical errors and "
            "alignment with the brand's voice.\n"
    ),
    expected_output="A well-written blog post "
        "in markdown format, ready for publication, "
        "each section should have 2 or 3 paragraphs.",
    agent=writer,
)
edit = Task(
    description=("Proofread the given blog post for "
                 "grammatical errors and "
                 "alignment with the brand's voice."),
    expected_output="A well-written blog post in markdown format, "
                    "ready for publication, "
                    "each section should have 2 or 3 paragraphs.",
    agent=editor
)
```

Define Crew

```
crew = Crew(
    agents=[planner, writer, editor],
    tasks=[plan, write, edit],
    verbose=True
)
```

```
WARNING:opentelemetry.trace:Overriding of current TracerProvider is not allowed
```

Kick off

```
result = crew.kickoff(inputs={"topic": "Write an essay about Malayalam language"})
```

```
# Agent: Content Planner
## Task: 1. Prioritize the latest trends, key players, and noteworthy news on Write an essay about Malayalam film.
2. Identify the target audience, considering their interests and pain points.
3. Develop a detailed content outline including an introduction, key points, and a call to action.
4. Include SEO keywords and relevant data or sources.


# Agent: Content Planner
## Final Answer:
Content Plan: "Exploring the Magic of Malayalam Cinema: A Comprehensive Guide"

I. Introduction

* Introduce Malayalam film industry, also known as Mollywood, as one of the major film industries in India.
* Mention the unique aspects of Malayalam cinema such as strong storytelling, realistic portrayal of characters, and
* Highlight the relevance of Malayalam film industry in Indian cinema and its influence on global cinema.

II. Latest Trends and Developments

* Discuss the recent trends in Malayalam cinema such as the rise of new-age filmmakers and their innovative storytel
* Highlight the emergence of OTT platforms and their impact on the Malayalam film industry.
* Mention the global recognition and awards won by Malayalam films and filmmakers.

III. Key Players and Noteworthy News

* Profile the prominent Malayalam filmmakers and actors who have made significant contributions to the industry.
* Discuss the latest news and happenings in the Malayalam film industry, such as upcoming movies, film festivals, ar

IV. Audience Analysis

* Identify the target audience as cinephiles, film enthusiasts, and anyone interested in learning about Malayalam ci
* Consider their interests in film, arts, and culture.
* Recognize their pain points such as lack of access to Malayalam films and limited knowledge about the industry.

V. Content Outline

* Section 1: Introduction to Malayalam Cinema
        + Subsection 1: History of Malayalam Film Industry
        + Subsection 2: Unique Aspects of Malayalam Cinema
* Section 2: Latest Trends and Developments
        + Subsection 1: Rise of New-Age Filmmakers
        + Subsection 2: Impact of OTT Platforms
        + Subsection 3: Global Recognition and Awards
* Section 3: Key Players and Noteworthy News
        + Subsection 1: Prominent Malayalam Filmmakers and Actors
        + Subsection 2: Latest News and Happenings
* Section 4: Conclusion
        + Subsection 1: Impact and Influence of Malayalam Cinema
        + Subsection 2: Future of Malayalam Film Industry

VI. SEO Keywords

* Malayalam cinema
* Mollywood
* New-age filmmakers
* OTT platforms
* Global recognition
```

Display

```
from IPython.display import Markdown
Markdown(result.raw)
```

Malayalam Cinema: A Rich Tapestry of Culture, Art, and Entertainment

Malayalam cinema, also known as Mollywood, is the film industry based in the Indian state of Kerala. With a rich history that spans over a century, Malayalam cinema has made significant contributions to Indian and global cinema. From its unique storytelling techniques to its talented pool of artists and technicians, Malayalam cinema has carved a niche for itself in the world of entertainment.

History of Malayalam Cinema

Malayalam cinema's journey began in the early 20th century, with the release of the first silent film, Vigathakumaran, in 1928. The industry evolved rapidly over the years, adopting sound and color technologies and featuring some of the most talented actors, directors, and technicians in Indian cinema. The 1980s and 1990s were particularly significant, with the rise of the "New Wave Cinema" movement that emphasized realism, social issues, and experimental storytelling techniques.

Unique Aspects of Malayalam Cinema

Malayalam cinema is known for its unique storytelling techniques, which often include complex narratives, non-linear storylines, and a focus on character development. The industry also has a rich tradition of classical music and dance, which is often integrated into films to enhance the storytelling experience. Moreover, Malayalam cinema has been at the forefront of exploring social, political, and environmental issues, providing a platform for meaningful conversations and debates.

Latest Trends in Malayalam Cinema

Today, Malayalam cinema is thriving, with a growing number of films being produced and released each year. The industry has also embraced digital technologies and OTT platforms, providing new avenues for filmmakers and audiences alike. Some of the latest trends in Malayalam cinema include the rise of female-centric films, experimental storytelling techniques, and a renewed focus on regional culture and dialects.

Key Players in Malayalam Cinema

Malayalam cinema has been home to some of the most talented artists and technicians in Indian cinema, including actors like Mohanlal, Mammootty, and Priyadarsan. The industry has also produced several award-winning directors like Adoor Gopalakrishnan, Shaji N. Karun, and Sathyan Anthikad. Moreover, Malayalam cinema has a rich tradition of music directors, cinematographers, and lyricists who have contributed significantly to the industry's growth and success.

Noteworthy News and Developments in Malayalam Cinema

Malayalam cinema has been in the news recently for several noteworthy developments, including the success of female-centric films like The Great Indian Kitchen and Helen. The industry has also seen a renewed focus on regional culture and dialects, with films like Kozhipporu and Kuttavum Shikshaiyum gaining critical acclaim. Moreover, Malayalam cinema has been recognized globally, with films like Jallikattu and Kala winning several international awards.

Conclusion

Malayalam cinema is a rich tapestry of culture, art, and entertainment that has made significant contributions to Indian and global cinema. From its unique storytelling techniques to its talented pool of artists and technicians, Malayalam cinema has carved a niche for itself in the world of entertainment. With a growing number of films being produced and released each year, Malayalam cinema continues to evolve and innovate, providing a platform for meaningful conversations and debates.

Sources:

- Malayalam Cinema - A Comprehensive Guide
- The Evolution of Malayalam Cinema
- Unique Aspects of Malayalam Cinema

Simple HTML webpage

Define Custom Tools

```
from crewai.tools import BaseTool
from typing import Type
from pydantic import BaseModel, Field
```

Browse Google Image URLs

```
!pip install serpapi==0.1.5
```

```
Requirement already satisfied: serpapi==0.1.5 in /usr/local/lib/python3.11/dist-packages (0.1.5)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from serpapi==0.1.5) (2.32.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->ser
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->serpapi==0.1.5)
```

```
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->serpapi==
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->serpapi==
```

TOOL 1: Browse Google Images

```
import serpapi
class GetImagesInput(BaseModel):
    """Image keyword to search for getting image urls from the web"""
    image_keyword: str = Field(..., description="Image keyword to search for getting

class GetImagesFromWeb(BaseTool):
    name: str = "Get_relevant_image_urls"
    description: str = "Search for the relevant image urls on web based on the given
    args_schema: Type[BaseModel] = GetImagesInput

    def _run(self, image_keyword: str) -> list:
        print("GetImagesFromWeb tool called")
        urlList = []
        try:
          params = {
            "q": str(image_keyword),
            "engine": "google_images",
            "ijn": "0",
            "api_key": "7002b372c5ebe29de686b784ece0cde6f066b1b41b163c4d4"
          }
          search = serpapi.search(params)
          for eachObject in search['images_results'][:10]:
            urlList.append(eachObject['original'])
          print("urlList is")
          print(urlList)
          return urlList
        except Exception as e:
          return urlList
# Instantiate the tool
get_image_urls_tool = GetImagesFromWeb()
```

TOOL 2: Save HTML file tool

```
class SaveHtmlFileInput(BaseModel):
    """Input schema for saving an HTML file."""
    html_page_content: str = Field(..., description="HTML file contents to save.")

class SaveHtmlFileTool(BaseTool):
    name: str = "Save_html_file"
    description: str = "Saves the given HTML content to a file named 'mywebpage.html'
    args_schema: Type[BaseModel] = SaveHtmlFileInput

    def _run(self, html_page_content: str) -> str:
        filename = "mywebpage.html"
        try:
          with open(filename, "w", encoding="utf-8") as file:
              file.write(html_page_content)
          return f"File saved successfully as {filename}"
        except Exception as e:
            return f"Error saving file: {str(e)}"
```

```
# Instantiate the tool
save_file_tool = SaveHtmlFileTool()
```

Web Developer Agent

```
web_developer = Agent(
    llm=llm,
    role = "Senior Web developer",
    goal = "Compile the results into a HTML file.",
    backstory = "You are a competent Web developer writing HTML and CSS with decades
                "You receives a blog post from the editor "
                "Your Goal is the convert the edited final draft of the blog post in
                "The web page should be desgined beutifully using js or css"
                "You must include images(use image urls from web) and background colo
                "Pass short relevant keywords like 'indian cricket','America','Machin
                "Dont pass long keywords or sentences to the `Get_relevant_image_urls
                "The output will be a single HTML file all the images,fonts, css and
                "The webpage must be colorfull and attractive. Add background color a
                "Save the HTML file in the folder with name current_webpage.html use
    tools = [save_file_tool,get_image_urls_tool],
    allow_delegation=False,
    verbose=True
)
```

Web Developer Task

```
webpage_creation = Task(
    description=("Convert the given edited final draft of the blog post into a html f
    expected_output="A well-desgined HTML page which contain both images and texts an
                    "The output will be a single HTML file all the images,fonts, css
                    "The webpage must be colorfull and attractive. Add background col
                    "Save the HTML file in the folder with name current_webpage.html

    agent=web_developer
)
```

Redefine the Crew

```
crew = Crew(
    agents=[planner, writer, editor,web_developer],
    tasks=[plan, write, edit,webpage_creation],
    verbose=True
)
```

```
WARNING:opentelemetry.trace:Overriding of current TracerProvider is not allowed
```

Kick Off

```
result = crew.kickoff(inputs={"topic": "Malayam Cinema History"})
```

```
# Agent: Content Planner
## Task: 1. Prioritize the latest trends, key players, and noteworthy news on Malayam Cinema History.
2. Identify the target audience, considering their interests and pain points.
3. Develop a detailed content outline including an introduction, key points, and a call to action.
4. Include SEO keywords and relevant data or sources.
```

```
# Agent: Content Planner
## Final Answer:
**Comprehensive Content Plan: Malayalam Cinema History**

---

### **1. Latest Trends, Key Players, and Noteworthy News**
**Trends**:
- **New Wave Cinema**: Rise of experimental storytelling (e.g., *Kathal: The Peculiar Love Story* [2023], *2018* [20
- **OTT Dominance**: Platforms like Netflix and SonyLIV acquiring Malayalam films (*Nna Thaan Case Kodu* [2022]).
- **Women-Centric Films**: Success of movies like *Sheep Without a Shepherd* (2023) and *Udaharanam Sujatha* (2023).
- **Global Recognition**: *Churuli* (2021) at Busan International Film Festival, *Bhoothakaalam* (2022) on internati

**Key Players**:
- **Actors**: Fahadh Faasil (*Malik*, *Trance*), Suraj Venjaramoodu (*Minnal Murali*), Kani Kusruti (*Biriyaani*).
- **Directors**: Lijo Jose Pellissery (*Jallikattu*), Dileesh Pothan (*Ezra*, *Thuramukham*), Mahesh Narayanan (*Ari
- **Producers**: Weekend Blockbusters, GoodCo Films, and Magic Moon Productions.
- **Music Directors**: Prashant Pillai, Rahul Raj, and Sushin Shyam (*Koode*, *Hridayam*).

**Noteworthy News**:
- **National Awards 2023**: *2018* won Best Feature Film in Malayalam.
- **Kerala Film Producers Association (KFPA)**: Advocating for tax-free cinema halls to combat OTT dominance.
- **Tech Innovations**: Use of VFX in *Ponniyin Selvan* (2022) and *Adrishya Jalakangal* (2023).

---

### **2. Target Audience Analysis**
**Primary Audience**:
- **Film Enthusiasts**: Indians and global audiences interested in regional cinema.
- **Academic Researchers**: Scholars studying Indian cinema’s evolution.
- **Expatriate Malayalees**: Nostalgic viewers seeking cultural connections.

**Demographics**:
- Age: 18–45+ (students, professionals, diaspora).
- Geography: India (Kerala), UAE, UK, US, Canada.

**Pain Points**:
- Fragmented historical content.
- Lack of updated resources on modern trends.
- Difficulty understanding cultural nuances.

**Interests**:
- Iconic films, behind-the-scenes stories, actor/director biographies.
- Comparative analysis with Bollywood/Tollywood.
- Free streaming options and film festival updates.

---

### **3. Detailed Content Outline**
**Title**: *"Malayalam Cinema: From Silent Films to Global Spotlight"*
```

```
!pip freeze
```

```
absl-py==1.4.0
accelerate==1.7.0
aiofiles==24.1.0
[redacted]
aiohappyeyeballs==2.6.1
aiohttp==3.11.15
aiosignal==1.3.2
alabaster==1.0.0
albucore==0.0.24
albumentations==2.0.8
ale-py==0.11.1
alembic==1.16.1
altair==5.5.0
annotated-types==0.7.0
antlr4-python3-runtime==4.9.3
anyio==4.9.0
appdirs==1.4.4
argon2-cffi==25.1.0
argon2-cffi-bindings==21.2.0
array_record==0.7.2
arviz==0.21.0
asgiref==3.8.1
astropy==7.1.0
astropy-iers-data==0.2025.6.2.0.38.23
astunparse==1.6.3
atpublic==5.1
attrs==25.3.0
```

```
audioread==3.0.1
auth0-python==4.9.0
autograd==1.8.0
babel==2.17.0
backcall==0.2.0
backoff==2.2.1
backports.tarfile==1.2.0
bcrypt==4.3.0
beautifulsoup4==4.13.4
betterproto==2.0.0b6
bigframes==2.5.0
bigquery-magics==0.9.0
bleach==6.2.0
blinker==1.9.0
blis==1.3.0
blobfile==3.0.0
blosc2==3.3.4
bokeh==3.7.3
Bottleneck==1.4.2
bqplot==0.12.45
branca==0.8.1
build==1.2.2.post1
CacheControl==0.14.3
cachetools==5.5.2
catalogue==2.0.10
certifi==2025.4.26
cffi==1.17.1
chardet==5.2.0
charset-normalizer==3.4.2
chex==0.1.89
chroma-hnswlib==0.7.6
```