
- 출처: LangChain 공식 문서 또는 해당 교재명
- 원본 URL: https://smith.langchain.com/hub/teddynote/summary-stuff-documents

---

`LlamaParser`

- **`LlamaParse`**
  - `LlamaIndex`에서 개발한 문서 파싱 서비스
  - **`대규모 언어 모델`(`LLM`)을 위해 특별히 설계됨**

- 주요 특징
  - **다양한 문서 형식 지원: `PDF`, `Word`, `PowerPoint`, `Excel` 등**
  - **`자연어 지시`** 를 통한 **`맞춤형 출력 형식`** 제공
  - **`복잡한 표`와 `이미지` 추출 기능**
  - **`JSON`** 모드 지원
  - **`외국어`** 지원

- **`독립형 API`로 제공**
- **`LlamaCloud`** 플랫폼의 일부로도 사용 가능
  - 목표: `문서를 파싱`하고 `정제`하여 **`RAG`** (검색 증강 생성) 등 **`LLM 기반`** 애플리케이션의 성능을 향상시키는 것

- **`무료로 하루 1,000페이지`** 처리 가능 (유료 플랜을 통해 추가 용량을 확보 가능)

- LlamaParse는 `현재 공개 베타 버전`으로 제공 → 지속적으로 기능이 확장되는 중

사전 환경 설정

- `VS Code` 터미널에 사전 설치할 것

```
pip install llama-index-core llama-parse llama-index-readers-file python-dot
```

- 참고: 링크
- `API Key` 설정: `.env` 파일에 `LLAMA_CLOUD_API_KEY` 키 설정하기

```
# API KEY를 환경변수로 관리하기 위한 설정 파일
import os
import nest_asyncio
from dotenv import load_dotenv

# API KEY 정보로드
load_dotenv()                                           # true
nest_asyncio.apply()
```

- 기본 파서 적용하기

```
# 필요한 모듈 임포트
from llama_parse import LlamaParse
from llama_index.core import SimpleDirectoryReader

# 파서 설정
parser = LlamaParse(
    result_type="markdown",                 # "markdown"과 "text" 사용 가능
    num_workers=8,                          # worker 수 (기본값: 4)
    verbose=True,
    language="ko",
)

# SimpleDirectoryReader를 사용하여 파일 파싱
file_extractor = {".pdf": parser}

# LlamaParse로 파일 파싱
documents = SimpleDirectoryReader(
    input_files=["../06_Document_Loader/data/SPRI_AI_Brief_2023년12월호_F.pdf"],
    file_extractor=file_extractor,
).load_data()
```

- 셀 출력 (41.8s)

```
2025-09-16 12:44:10,509 - INFO - HTTP Request: POST https://api.cloud.llamainde
Started parsing the file under job_id d2caa84f-170a-43fe-bf66-2dee8666af8c
2025-09-16 12:44:11,840 - INFO - HTTP Request: GET https://api.cloud.llamaindex
2025-09-16 12:44:14,368 - INFO - HTTP Request: GET https://api.cloud.llamaindex
2025-09-16 12:44:17,768 - INFO - HTTP Request: GET https://api.cloud.llamaindex
2025-09-16 12:44:22,195 - INFO - HTTP Request: GET https://api.cloud.llamaindex
2025-09-16 12:44:27,961 - INFO - HTTP Request: GET https://api.cloud.llamaindex
2025-09-16 12:44:33,661 - INFO - HTTP Request: GET https://api.cloud.llamaindex
2025-09-16 12:44:39,584 - INFO - HTTP Request: GET https://api.cloud.llamaindex
```

```
2025-09-16 12:44:45,308 - INFO - HTTP Request: GET https://api.cloud.llamaindex
2025-09-16 12:44:45,848 - INFO - HTTP Request: GET https://api.cloud.llamaindex
```

```
# 로드된 문서 개수 출력

print(f"로드된 문서 개수: {len(documents)}")            # 로드된 문서 개수: 23
```

- **`LlamaIndex`** → **`LangChain Document`** 로 변환하기

```
# 랭체인 도큐먼트로 변환
docs = [doc.to_langchain_format() for doc in documents]
```

```
# metadata 출력

docs[0].metadata
```

- 셀 출력

```
{'file_path': '../06 Document Loader/data/SPRI AI Brief 2023년12월호_F.pdf',
'file_name': 'SPRI_AI_Brief_2023년12월호_F.pdf',
'file_type': 'application/pdf',
'file_size': 975735,
'creation_date': '2025-09-12',
'last_modified_date': '2025-09-12'}
```

`MultiModal Model` 로 파싱

- 주요 파라미터
  - **`use_vendor_multimodal_model`**
    - 멀티모달 모델 사용 여부 지정
    - **`True`** 로 설정 → **`외부 벤더의 멀티모달 모델`** 을 **`사용`**
  - **`vendor_multimodal_model_name`**
    - 사용할 멀티모달 모델의 이름을 지정
    - **`gemini-2.5-flash`** **사용하기**
  - **`vendor_multimodal_api_key`**
    - ** `멀티모달 모델 API 키` 지정하기
    - 환경 변수(`.env`)에서 `API 키` 가져오기
  - **`result_type`**

        - **`파싱 결과`의 `형식` 지정하기
        - **`markdown`으로 설정** → 결과가 마크다운 형식으로 반환됨
    - **`language`**
        - 파싱할 문서의 **`언어`** **지정**
        - **`"ko"`로 설정** → 한국어로 처리
    - **`skip_diagonal_text`**: 대각선 텍스트를 건너뛸지 여부를 결정
    - **`page_separator`**: 페이지 구분자 지정 가능

```
documents = LlamaParse(
    use_vendor_multimodal_model=True,
    vendor_multimodal_model_name="openai-gpt-4o",
    vendor_multimodal_api_key=os.environ["OPENAI_API_KEY"],
    result_type="markdown",
    language="ko",
    # skip_diagonal_text=True,
    # page_separator="\n=================\n"
)
```

```
# parsing 된 결과
parsed_docs = documents.load_data(file_path="../06_Document_Loader/data/SPRI_AI_B
```

- 셀 출력

```
Started parsing the file under job_id c7c29c38-d183-4c66-a9f6-2c14e90da500
```

```
# langchain 도큐먼트로 변환
docs = [doc.to_langchain_format() for doc in parsed_docs]
```

- 사용자 정의 인스트럭션 지정 가능

```
# parsing instruction 지정하기
parsing_instruction = (
    "You are parsing a brief of AI Report. Please extract tables in markdown form
)

# LlamaParse 설정
parser = LlamaParse(
    use_vendor_multimodal_model=True,
    vendor_multimodal_model_name="openai-gpt-4o-mini",
    vendor_multimodal_api_key=os.environ["OPENAI_API_KEY"],
    result_type="markdown",
    language="ko",
    #parsing_instruction=parsing_instruction,
    system_prompt=parsing_instruction,
)

# parsing 된 결과
```

```
parsed_docs = parser.load_data(file_path="../06_Document_Loader/data/SPRI_AI_Brie

# langchain 도큐먼트로 변환
docs = [doc.to_langchain_format() for doc in parsed_docs]
```

- 셀 출력

```
Started parsing the file under job_id b0947ed1-3d8f-4eaa-bf6a-e6a746ff697f

.
```

```
# markdown 형식으로 추출된 테이블 확인
print(docs[-2].page_content)
```

- 셀 출력

```
# Ⅱ. 주요 행사 일정

| 행사명 | 행사 주요 개요 |
| --- | --- |
| CES 2024 | - 미국 소비자기술 협회(CTA)가 주관하는 세계 최대 가전·IT·소비재 전시회로 5G, AR&VR
- CTA 사피로 회장은 가장 주목받는 섹터로 AI를 꼽았으며, 모든 산업을 포함한다는 의미에서 '올 인AI on'

![CES 2024](https://www.ces.tech/) |
| 기간 | 2024.1.9~12 |
| 장소 | 미국, 라스베가스 |
| 홈페이지 | [https://www.ces.tech/](https://www.ces.tech/) |

| 행사명 | 행사 주요 개요 |
| --- | --- |
| AIMLA 2024 | - 머신러닝 및 응용에 관한 국제 컨퍼런스(AIMLA 2024)는 인공지능 및 머신러닝의 이론
- 이론 및 실무 측면에서 인공지능, 기계학습의 주요 분야를 논의하고, 함께, 산업계의 연구자와 실무자들에게

![AIMLA 2024](https://ccnet2024.org/aimla/index) |
| 기간 | 2024.1.27~28 |
| 장소 | 덴마크, 코펜하겐 |
| 홈페이지 | [https://ccnet2024.org/aimla/index](https://ccnet2024.org/aimla/index

| 행사명 | 행사 주요 개요 |
| --- | --- |
| AAAI Conference on Artificial Intelligence | - AI 발전 협회 컨퍼런스(AAAI)는 AI 연구
- 컨퍼런스에서 AI 관련 기술 발표, 특별 트랙, 초청 연사, 워크숍, 튜토리얼, 포스터 세션, 주제 발표, 대

![AAAI Conference on Artificial Intelligence](https://aaai.org/aaai-conference/
| 기간 | 2024.2.20~27 |
```

```
| 장소 | 캐나다, 밴쿠버 |
| 홈페이지 | [https://aaai.org/aaai-conference/](https://aaai.org/aaai-conference/
```

- *next:* *CH07 텍스트 분할 (Text Splitter)*
