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

---

`UpstageLayoutAnalysisLoader`

- **`UpstageLayoutAnalysisLoader`**
  - `Upstage AI`에서 제공하는 문서 분석 도구
  - `LangChain` 프레임워크와 통합되어 사용할 수 있는 문서 로더

- 주요 특징: **단순한 텍스트 추출을 넘어 `문서의 구조`를 `이해`하고 `요소 간 관계`를 `파악`하여 보다 `정확한 문서 분석 가능`**
  - `PDF`, `이미지` 등 다양한 형식의 문서에서 레이아웃 분석 수행
  - 문서의 **`구조적 요소`** (`제목`, `단락`, `표`, `이미지` 등)를 **`자동`으로 `인식` 및 `추출`**
  - **`OCR`** 기능 지원 (선택적)

사전 환경 설정

- `VS Code` 터미널에 사전 설치할 것

```
pip install -U langchain-upstage
```

- 참고: **`Upstage 개발자 문서 가이드 문서`**
- **`API Key`** 설정: **`.env`** 파일에 **`UPSTAGE_API_KEY`** 키 설정하기

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

# API KEY 정보로드
load_dotenv()                                    # true
```

```
# 환경변수 처리 및 클라이언트 생성
from langsmith import Client

# 클라이언트 생성
api_key = os.getenv("LANGSMITH_API_KEY")
client = Client(api_key=api_key)
```

"@traceable" 주석은 허용되지 않습니다. 허용되는 값은 다음과 같습니다. [@param, @title, @markdown]

```
# LangSmith 추적 설정하기 (https:smith.langchin.com)
# LangSmith 추적을 위한 라이브러리 임포트
from langsmith import traceable

# LangSmith 환경 변수 확인

print("\n--- LangSmith 환경 변수 확인 ---")
langchain_tracing_v2 = os.getenv('LANGCHAIN_TRACING_V2')
langchain_project = os.getenv('LANGCHAIN_PROJECT')
langchain_api_key_status = "설정됨" if os.getenv('LANGCHAI
org = "설정됨" if os.getenv('LANGCHAIN_ORGANIZATION') else

if langchain_tracing_v2 == "true" and os.getenv('LANGCHA
    print(f"✅ LangSmith 추적 활성화됨 (LANGCHAIN_TRACING_V2
    print(f"✅ LangSmith 프로젝트: '{langchain_project}'")
    print(f"✅ LangSmith API Key: {langchain_api_key_sta
    print("  -> 이제 LangSmith 대시보드에서 이 프로젝트를 확인해 보
```

```
else:
    print("❌ LangSmith 추적이 완전히 활성화되지 않았습니다. 다음을
    if langchain_tracing_v2 != "true":
        print(f"  - LANGCHAIN_TRACING_V2가 'true'로 설정되어
    if not os.getenv('LANGCHAIN_API_KEY'):
        print("  - LANGCHAIN_API_KEY가 설정되어 있지 않습니다.'
    if not langchain_project:
        print("  - LANGCHAIN_PROJECT가 설정되어 있지 않습니다.'
```

- 셀 출력

```
--- LangSmith 환경 변수 확인 ---
✅ LangSmith 추적 활성화됨 (LANGCHAIN_TRACING_V2='true')
✅ LangSmith 프로젝트: 'LangChain-prantice'
✅ LangSmith API Key: 설정됨
-> 이제 LangSmith 대시보드에서 이 프로젝트를 확인해 보세요.
```

`UpstageLayoutAnalysisLoader`

- 모듈 이름 변경
  - ❌ `from langchain_upstage import UpstageLayoutAnalysisLoader`
  - ⭕ **`from langchain_community.document_loaders import UpstageLayoutAnalysisLoader`**

- 주요 파라미터
  - **`file_path`**: 분석할 문서 경로
  - **`output_type`**: 출력 형식 (기본값: `html`, `text`)
  - **`split`**: 문서 분할 방식 (`none`, `element`, `page`)
  - **`use_ocr=True`**: **`OCR`** 사용
  - **`exclude=["header", "footer"]`**: 헤더, 푸터 제외

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

# API KEY 정보로드
load_dotenv()

print("\n--- UpstageLayoutAnalysisLoader 환경 변수 확인 ---")
upstage_api_key_status = "설정됨" if os.getenv('UPSTAGE_API_KEY') else "설정되지 않음"      # API 키 값은 직접 출력하지 않음


if langchain_tracing_v2 == "true" and os.getenv('LANGCHAIN_API_KEY'):
    print(f"✅ UpstageLayoutAnalysisLoader API Key가 설정되었습니다.")

else:
    print("❌ LUpstageLayoutAnalysisLoader API Key가 설정되지 않았습니다. 다음을 확인하세요:")
```

- 셀 출력

```
--- UpstageLayoutAnalysisLoader 환경 변수 확인 ---
✅ UpstageLayoutAnalysisLoader API Key가 설정되었습니다.
```

```
import os, sys, inspect
from dotenv import load_dotenv
load_dotenv()

# 파이썬/패키지 경로 확인
print("Python:", sys.executable)
import langchain_upstage
print("langchain_upstage:", langchain_upstage.__file__)
```

```
# 환경변수 확인
api = os.getenv("UPSTAGE_API_KEY")
assert api, "UPSTAGE_API_KEY 비어 있음"
```

```
Python: /Users/jay/.pyenv/versions/lc_env/bin/python
langchain_upstage: /Users/jay/.pyenv/versions/lc_env/lib/python3.13/site-packages/langchain_upstage/__init__.py
```

- 패키지 충돌로 대체 로더 사용하기

```
import os
from dotenv import load_dotenv
load_dotenv()

from langchain_upstage.document_parse import UpstageDocumentParseLoader

api = os.getenv("UPSTAGE_API_KEY")
assert api, "UPSTAGE_API_KEY 비어 있음"

file_path = "../06_Document_Loader/data/SPRI_AI_Brief_2023년12월호_F.pdf"
assert os.path.exists(file_path), f"Not found: {file_path}"

loader = UpstageDocumentParseLoader(
    file_path=file_path,
    #output_type="text",   # 'text' 또는 'html' (지원 범위는 버전에 따라 상이)
    #use_ocr=True,
    split="page",         # 설치본의 시그니처에 맞춰 조정
    api_key=api
)

docs = loader.load()

print("docs:", len(docs))

for d in docs[:3]:
    print(d.metadata.get("page"), d.page_content[:300])
```

- 셀 출력 (15.7s)

```
docs: 23
1 <h1 id='0' style='font-size:14px'>2023년 12월호</h1> <figure id='1'><img alt="" data-coord="top-left:(26,743); bo
2 <header id='2' style='font-size:14px'>2023년 12월호</header> <h1 id='3' style='font-size:20px'>I. 인공지능 산업 동향 브
3 <h1 id='15' style='font-size:14px'>I. 인공지능 산업 동향 브리프</h1>
```

```
from bs4 import BeautifulSoup

docs = loader.load()
plain_docs = []
for d in docs:
    html = d.page_content or ""
    text = BeautifulSoup(html, "html.parser").get_text(separator=" ", strip=True)
    # 불필요한 공백 정리 (선택)
    text = " ".join(text.split())
    d.page_content = text
    plain_docs.append(d)

print("docs:", len(plain_docs))
for d in plain_docs[:3]:
    print(f"page_content={repr(d.page_content)[:200]} ...", "metadata=", d.metadata)
```

- 셀 출력 (17.4s)

```
docs: 23
page_content='2023년 12월호' ... metadata= {'page': 1, 'coordinates': [[{'x': 0.4127, 'y': 0.3278}, {'x': 0.5855,
page_content='2023년 12월호 I. 인공지능 산업 동향 브리프 1. 정책/법제 ▹ 미국, 안전하고 신뢰할 수 있는 AI 개발과 사용에 관한 행정명령 발표
page_content='I. 인공지능 산업 동향 브리프' ... metadata= {'page': 3, 'coordinates': [[{'x': 0.2222, 'y': 0.3727}, {'x':
```

- `metadata` 속 숫자 배열 = **`페이지 내 추출된 블록(제목 or 문단)의 레이아웃 좌표`** 를 정규화하여 담아둔 값
  - `coordinates` = **페이지 공간 내 위치 정보**
  - **값의 범위가 `0~1` 인 이유**: PDF 페이지 크기에 `독립적인 정규화 좌표`이기 때문

- 동일 페이지에 `여러 텍스트 블록`이 있으면 `coordinates`가 `리스트 안`에 `여러 박스 형태`로 들어갈 수 있음.
- 이 좌표를 이용해 특정 범위(예: y가 상단 10% 이내)를 헤더로 간주해 제거한다든지, 하단 10%를 푸터로 간주해 필터링하는 등의 규칙을 쉽게 적용 가능.

```
# metadata.coordinates에서 좌표를 숨기고 가독성을 높여 출력하기

for d in docs[:3]:
    page = d.metadata.get("page")
    # 좌표 키 제거(출력만 깔끔히)
    d.metadata.pop("coordinates", None)
    print(f"page={page}", d.page_content[:300])
```

- 셀 출력

```
page=1 2023년 12월호
page=2 2023년 12월호 I. 인공지능 산업 동향 브리프 1. 정책/법제 ▹ 미국, 안전하고 신뢰할 수 있는 AI 개발과 사용에 관한 행정명령 발표 ·····
page=3 I. 인공지능 산업 동향 브리프
```

```
import re
from bs4 import BeautifulSoup

def clean_text(html):
    soup = BeautifulSoup(html, "html.parser")
    # 1) header/footer 제거 시도
    for tag in soup.find_all(["header","footer"]):
        tag.decompose()
    # 2) 불필요한 figure/img 제거
    for tag in soup.find_all(["figure","img","nav","aside"]):
        tag.decompose()
    # 3) 텍스트 추출
    txt = soup.get_text(separator=" ", strip=True)
    # 4) 다중 공백 정리
    txt = re.sub(r"\s+", " ", txt).strip()
    return txt

docs = loader.load()
for d in docs:
    d.page_content = clean_text(d.page_content or "")

for d in docs[:3]:
    print("page:", d.metadata.get("page"), "text=", d.page_content[:300])
```

- 셀 출력 (15.8s)

```
page: 1 text= 2023년 12월호
page: 2 text= I. 인공지능 산업 동향 브리프 1. 정책/법제 ▹ 미국, 안전하고 신뢰할 수 있는 AI 개발과 사용에 관한 행정명령 발표 ···········
page: 3 text= I. 인공지능 산업 동향 브리프
```

- *next:* `LlamaParser`
