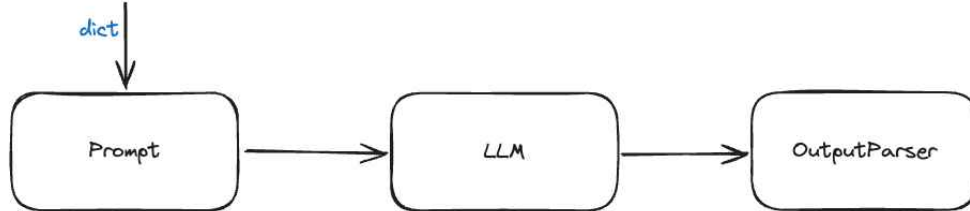


## CHAPTER 03. LCEL

### 1. LangChain Expression Language(LCEL)

`{"question": "{topic}에 대해 쉽게 설명해주세요."}`



예제 - LCEL 체인

```
from dotenv import load_dotenv
import os
```

```
load_dotenv(verbose=True)
key = os.getenv('OPENAI_API_KEY')
```

```
from langchain_openai import ChatOpenAI
from langchain_core.prompts import PromptTemplate
```

```
llm = ChatOpenAI(
    api_key=key,
    model_name='gpt-4o-mini',
    temperature=0.1,
    max_tokens=2048,
)
```

```
# prompt를 PromptTemplate 객체로 생성합니다.
prompt = PromptTemplate.from_template("{product}는 어느 회사에서 개발한 제품인가요?")
```

```
# LCEL을 사용하여 다양한 구성 요소를 단일 체인으로 결합합니다. prompt와 llm(모델)을 결합.
# 이 체인에서 사용자 입력은 프롬프트 템플릿으로 전달되고,
# 그런 다음 프롬프트 템플릿 출력은 모델로 전달됩니다.
```

```
chain = prompt | llm
```

```
# 파이썬 딕셔너리 형태로 입력값을 전달합니다. (키: 값)
# invoke() 함수 호출 시, 입력값을 전달합니다.
answer = chain.invoke({'product': '갤럭시폰'})
print(answer.content)
```

실행결과

갤럭시폰은 삼성전자(Samsung Electronics)에서 개발한 스마트폰 브랜드입니다. 삼성전자는 다양한 모델의 갤럭시폰을 출시하고 있으며, 이들 제품은 안드로이드 운영체제를 기반으로 하고 있습니다.

프로그램에서 언어 모듈을 호출할 때 미리 준비된 **프롬프트**와 **파이썬의 입력**을 **결합**해서 사용합니다. Prompts 모듈에서 PromptTemplate이 기본적인 모듈입니다.

PromptTemplate 은 사용자의 입력 변수를 사용하여 완전한 프롬프트 문자열을 만드는 데 사용되는 템플릿입니다.

- template: 템플릿 문자열입니다. 이 문자열 내에서 중괄호 {}는 변수를 나타냅니다.
- input\_variables: 중괄호 안에 들어갈 변수의 이름을 리스트로 정의합니다.

## 2. 출력 파서(Output Parser)

LangChain의 출력 파서는 언어 모델(LLM)의 출력을 더 유용하고 구조화된 형태로 변환하는 중요한 컴포넌트입니다. 실행결과를 문자열 형태로 출력합니다.

### 예제 - 출력 파서

```
from dotenv import load_dotenv
import os

load_dotenv(verbose=True)
key = os.getenv('OPENAI_API_KEY')

from langchain_openai import ChatOpenAI
from langchain_core.prompts import PromptTemplate
from langchain_core.output_parsers import StrOutputParser

llm = ChatOpenAI(
    api_key=key,
    model_name='gpt-4o-mini',
    temperature=0.1,
    max_tokens=2048,
)

prompt = PromptTemplate.from_template("{pruduct}는 어느 회사에서 개발한 제품인가요?")
output_parser = StrOutputParser()
chain = prompt | llm | output_parser

answer = chain.invoke({'pruduct': '갤럭시폰'})
print(answer)
```

### 실행결과

갤럭시폰은 삼성전자(Samsung Electronics)에서 개발한 스마트폰 브랜드입니다. 삼성전자는 다양한 모델의 갤럭시폰을 출시하고 있으며, 이들 제품은 안드로이드 운영체제를 기반으로 하고 있습니다.

## 예제 - 출력 파서

```
from dotenv import load_dotenv
import os

load_dotenv(verbose=True)
key = os.getenv('OPENAI_API_KEY')

from langchain_openai import ChatOpenAI
from langchain_core.prompts import PromptTemplate
from langchain_core.output_parsers import StrOutputParser

llm = ChatOpenAI(
    api_key=key,
    model_name='gpt-4o-mini',
    temperature=0.1,
    max_tokens=2048,
)

template = """
당신은 영어를 가르치는 10년차 영어 선생님입니다. 상황에 맞는 영어 회화를 작성해 주세요.
양식은 [FORMAT]을 참고하여 작성해 주세요.

#상황:
{question}
↑
#FORMAT:
- 영어 회화:
- 한글 해석:
"""

# 프롬프트 템플릿을 이용하여 프롬프트를 생성합니다.
prompt = PromptTemplate.from_template(template)

# 문자열 출력 파서를 초기화합니다.
output_parser = StrOutputParser()

# 체인을 구성합니다.
chain = prompt | llm | output_parser

answer = chain.invoke({'question': '저는 식당에 가서 음식을 주문하고 싶어요'})
print(answer)
```

## 실행결과

---

### - 영어 회화:

- Waiter: Good evening! Welcome to our restaurant. How many people are in your party?
- You: Good evening! Just one, please.
- Waiter: Great! Here's the menu. Can I get you something to drink while you look?
- You: Yes, I'd like a glass of water, please.
- Waiter: Sure! Are you ready to order, or do you need a few more minutes?
- You: I think I'm ready. I'd like the grilled chicken salad, please.
- Waiter: Excellent choice! Would you like any dressing with that?
- You: Yes, please. I'll have the balsamic vinaigrette.
- Waiter: Perfect! I'll get that order in for you. Is there anything else you need?
- You: No, that's all for now. Thank you!

### - 한글 해석:

- 웨이터: 좋은 저녁입니다! 저희 식당에 오신 것을 환영합니다. 몇 분이신가요?
  - 당신: 좋은 저녁입니다! 혼자입니다.
  - 웨이터: 좋습니다! 여기 메뉴입니다. 보시는 동안 음료수 드릴까요?
  - 당신: 네, 물 한 잔 주세요.
  - 웨이터: 알겠습니다! 주문할 준비가 되셨나요, 아니면 좀 더 시간이 필요하신가요?
  - 당신: 이제 준비가 된 것 같아요. 그릴에 구운 치킨 샐러드로 주세요.
  - 웨이터: 훌륭한 선택입니다! 드레싱은 어떤 걸 원하시나요?
  - 당신: 네, 발사믹 비네그레트를 주세요.
  - 웨이터: 완벽합니다! 주문을 넣어드릴게요. 다른 필요하신 건 없으신가요?
  - 당신: 아니요, 지금은 그게 다입니다. 감사합니다!
- 

- 출처 : 랭체인LangChain 노트 (<https://wikidocs.net/233343>)