**TA's 솔루션 해설**

* <https://gist.github.com/ICE0208/4688a0d1b069c6af0a97a39b14f912d9>

1. ChatOpenAI 모델 초기화 및 설정

* ChatOpenAI 생성자를 이용하여 OpenAI의 GPT 모델을 초기화합니다.
* model\_name 파라미터를 이용하여 모델을 gpt-3.5-turbo 로 지정해줍니다.
* streaming=True 옵션과 StreamingStdOutCallbackHandler 콜백을 사용하여 답변 생성 과정을 실시간으로 출력합니다.

1. 프롬프트 템플릿 생성

* poet\_prompt : 시 생성을 위한 프롬프트 템플릿입니다. {language} 부분은 입력받은 프로그래밍 언어 이름으로 대체됩니다.
* explainer\_prompt : 시 분석을 위한 프롬프트 템플릿입니다. {poem} 부분은 생성된 시로 대체됩니다.

1. 체인 생성

* poet\_chain : poet\_prompt 템플릿과 chat 모델을 연결하여 시 생성 체인을 만듭니다.
* explainer\_chain: explainer\_prompt 템플릿과 chat 모델을 연결하여 시 분석 체인을 만듭니다.

1. LCEL을 사용하여 체인 연결

* final\_chain : poet\_chain과 explainer\_chain을 LCEL (LangChain Expression Language)을 사용하여 연결합니다.
* {"poem": poet\_chain} : poet\_chain의 출력을 poem이라는 변수에 할당하여 explainer\_chain에 전달합니다.

1. 최종 체인 실행

* final\_chain.invoke({"language": "python"}) : final\_chain을 실행하여 python이라는 입력 값을 전달합니다. 먼저 poet\_chain이 Python에 대한 시를 생성하고, 이 시가 explainer\_chain에 전달되어 분석 결과를 출력합니다.

1. 결론

이번 챌린지는 주피터 노트북 환경에서 랭체인 표현 언어(LCEL)를 사용하여 시를 생성하는 체인과 시를 분석하는 체인을 구현하는 과정을 통해 랭체인의 기본적인 사용법을 연습할 수 있는 챌린지였습니다.