

- 출처: LangChain 공식 문서, 조코딩의 랭체인으로 AI 에이전트 서비스 만들기
- 깃허브 저장소 출처: <https://github.com/sw-woo/hanbit-langchain>

## ✓ 인공지능 시인 만들기

- 출처: 위에 표기

- 구현할 서비스의 구조

\*

- step\_1:
  - 사용자 - 시의 주제 입력
  - LLM - 주제에 맞는 시 생성
- step\_2:
  - streamlit - 프론트엔드 구현해보기
- step\_3:
  - LangChain - 백엔드에서 사용자 입력과 LLM 모델의 출력 연결하기
- step\_4:
  - LLM - 선택한 모델의 API 호출 → 시 생성하도록 함

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

import os
import json

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

# LangSmith 추적 설정하기 (https://smith.langchain.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('LANGCHAIN_API_KEY') else "설정되지 않음"
org = "설정됨" if os.getenv('LANGCHAIN_ORGANIZATION') else "설정되지 않음"

if langchain_tracing_v2 == "true" and os.getenv('LANGCHAIN_PROJECT') and langchain_api_key_status == "설정됨" and org:
    print(f"✅ LangSmith 추적 활성화됨 (LANGCHAIN_TRACING_V2='{langchain_tracing_v2}')
```

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



```
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.
```

- 환경 설정 및 LLM 초기화 공통

```
# 필요한 모듈 임포트
from langchain.chat_models import init_chat_model
from langchain_core.prompts import ChatPromptTemplate
from langchain_core.output_parsers import StrOutputParser
from langchain_core.runnables import RunnablePassthrough

import streamlit as st # 스트림릿 사용에 필요한 모듈 임포트
```

- **temperature=0.7** → 다양한 대답이 나올 수 있도록 설정

```
# gpt, gemini 스위치되도록 테스트
import os
from dotenv import load_dotenv

load_dotenv()

# 환경변수에서 API 키 가져오기
openai_key = os.getenv("OPENAI_API_KEY")
google_key = os.getenv("GOOGLE_API_KEY")

# LLM 설정
def get_llm(provider: str):
    """provider에 따라 LLM 객체 반환"""
    if provider == "openai":
        return init_chat_model(
            model="gpt-4o-mini", # 사용하는 gpt 모델
            model_provider="openai",
            temperature=0.7,
            api_key=openai_key,
            max_output_tokens=1024
        )
    elif provider == "google":
        return init_chat_model(
            model="gemini-2.5-flash-lite", # 사용하는 gemini 모델
            model_provider="google_genai",
            temperature=0.7,
            api_key=google_key,
            max_output_tokens=4096
        )
    else:
        raise ValueError("지원하지 않는 provider입니다.")

# 🗣️ 원하는 모델 선택
llm = get_llm("google") # "openai"로 바꾸면 gpt 사용 가능

# 테스트
response = llm.invoke("안녕! 너는 어떤 모델이야?")
print(response.content)
```

- 셀 출력 (1.1s)

저는 Google에서 훈련한 대규모 언어 모델입니다.

---

## ✓ 프롬프트 템플릿 생성

#프롬프트 템플릿 생성

```
prompt = ChatPromptTemplate.from_messages([
    ("system", "You are a helpful assistant."),
    ("user", "{input}")
])
```

---

## ✓ 문자열 출력 파서

#문자열 출력 파서

```
output_parser = StrOutputParser()
```

---

## ✓ Chain 생성

- {content} = 사용자 입력을 받기 위한 변수
  - 아직 FE 연결 전이므로 임의로 주제를 정해서 문자열을 넣음
- chain.invoke() 호출 → LLM 체인 실행
  - 함수 인자로 전달: {content} + "에 대한 시를 써줘" 라는 문자열
- 교재 속 코드

```
content = "코딩"
result = chain.invoke({"input":content+"에 대한 시를 써줘."})
print(result)
```

#LLM 체인 구성

```
chain = prompt | llm | output_parser
```

*streamlit은 다음 파일에 연동 예정*

*이 파일에서는 LangChain 만 연습*

- 변수를 하드코딩 X
- {input} 를 플레이스 홀더 처럼 사용해서 유동적으로 변경할 수 있도록 만들기
  - 이후 streamlit 활용 시 UI 붙일 때 수월하도록 하기 위함

```
input()                # 주피터 노트북 or 파이썬 코드 속
```

```
st.text_input()        # 스트림릿의 UI 구조와 유사
```

# 콘솔에서 사용자 입력 받기

topic = input("시의 주제를 입력하세요: ")

- 위의 셀을 실행하면 입력 대기 상태가 됨 → 값을 입력하고 엔터를 눌러야 다음 셀로 넘어갈 수 있음
- 여러 개의 input() 을 썼을 경우 순서대로 하나씩 물어보게 됨
- 실행 중간에 셀을 멈추고 싶은 경우 커널 중지(상단에 있는 ■ 버튼) 누르기

# 체인 실행하기

result = chain.invoke({"input" : topic + "에 대한 시를 써줘"})

- 콘솔 입력에서 교재 예시대로 코드 를 시의 주제로 입력
- 체인 실행 셀 실행 (1.8s)

# 결과 출력하기

print("\n=== 생성된 시 ===","\n")  
print(result)

- 셀 출력

=== 생성된 시 ===

## 코드의 춤

어둠 속에서 빛나는 별들처럼,  
알파벳과 숫자가 춤을 춘다.  
논리의 실타래 엮어가는 손가락,  
상상의 세계를 현실로 빚어낸다.

마침표 하나, 쉼표 하나에도  
세상은 숨죽여 귀 기울인다.  
명령은 칼날처럼 정확하게,  
결과는 마법처럼 펼쳐지리.

때로는 얽히고설킨 미로 속에서  
길을 잃고 헤매기도 하지만,  
끈질긴 탐구와 땀방울 끝에  
새로운 해답, 환희가 찾아온다.

화면 가득 펼쳐지는 숫자들의 향연,  
그 속에 담긴 무한한 가능성.  
코드는 언어, 세상을 움직이는 힘,  
디지털 시대의 시인이여, 그대여.

- 다른 주제 넣어보기

# 콘솔에서 사용자 입력 받기

topic = input("시의 주제를 입력하세요: ")

- 시의 주제: 얼마 남지 않은 커피

# 체인 실행하기

result = chain.invoke({"input" : topic + "에 대한 시를 써줘"})

- 셀 실행 (1.4s)

# 결과 출력하기

```
print("\n=== 생성된 시 ===","\n")
print(result)
```

- 셀 출력

```
=== 생성된 시 ===

## 마지막 한 모금

진한 갈색 물결, 잔에 남은 흔적
어느새 이렇게 좋아들었나
따뜻했던 온기, 희미해져 가고
쌈싸름한 향기, 코끝을 간지럽히네

함께 나눴던 이야기, 웃음꽃 피던 시간
이 잔에 담겨, 추억으로 남았네
어쩌면 마지막일지도 모를 이 순간
아쉬움에 천천히, 음미해 본다

마지막 한 모금, 입안에 머금고
오늘 하루를, 또 다른 내일을
힘껏 응원하며, 잔을 비워낸다
텅 빈 잔처럼, 가벼운 마음으로
```

▼ 더 다양한 스타일의 시 생성해보기

```
# 필요한 모듈 импорт
from langchain.chat_models import init_chat_model
from langchain_core.prompts import ChatPromptTemplate
from langchain_core.output_parsers import StrOutputParser
from langchain_core.runnables import RunnablePassthrough
```

- gemini-2.5-flash-lite 모델 계속 사용

```
# gpt, gemini 스위치되도록 테스트
import os
from dotenv import load_dotenv

load_dotenv()

# 환경변수에서 API 키 가져오기
openai_key = os.getenv("OPENAI_API_KEY")
google_key = os.getenv("GOOGLE_API_KEY")

# LLM 설정
def get_llm(provider: str):
    """provider에 따라 LLM 객체 반환"""
    if provider == "openai":
        return init_chat_model(
            model="gpt-4o-mini", # 사용하는 gpt 모델
            model_provider="openai",
            temperature=0.7,
            api_key=openai_key,
            max_output_tokens=1024
        )
    elif provider == "google":
        return init_chat_model(
            model="gemini-2.5-flash-lite", # 사용하는 gemini 모델
            model_provider="google_genai",
            temperature=0.7,
            api_key=google_key,
            max_output_tokens=4096
        )
    else:
```

```
raise ValueError("지원하지 않는 provider입니다.")

# 🔄 원하는 모델 선택
llm = get_llm("google") # "openai"로 바꾸면 gpt 사용 가능

# 테스트
response = llm.invoke("안녕! 너는 어떤 모델이야?")
print(response.content)
```

- 셀 출력
- 저는 Google에서 훈련한 대규모 언어 모델입니다.

---

▼ 차이점: 주제, 스타일, 참고 시인 을 3번에 걸쳐서 입력자로부터 입력 받기

```
#프롬프트 템플릿 생성
prompt = ChatPromptTemplate.from_messages([
    ("system", "You are a helpful assistant."),
    ("user", "{topic}에 대한 {style} 스타일의 시를 {poet} 시인의 문체를 참고해서 시를 써줘.")
])

#문자열 출력 파서
output_parser = StrOutputParser()

#LLM 체인 구성
chain = prompt | llm | output_parser
```

- 미리 입력내용 결정하기
  - 아래 셀 실행 시 바로 입력 창이 3번 연속 뜨기 때문

- \*
- 시의 주제: 속초 관광 소개
  - 시의 스타일: 비유와 상징이 난무하고 문법을 파괴한 초현실적이며 난해하고 실험적인 스타일
  - 참고할 시인: 거울, 오감도, 날개 같은 시를 쓴 시인 이상

```
# 콘솔에서 입력 받기(순차적으로 입력하기)

topic = input("시의 주제를 입력하세요: ")
style = input("시의 스타일을 입력하세요 (예: 감성적인, 유머러스한): ")
poet = input("참고할 시인의 이름을 입력하세요 (예: 김소월, 윤동주): ")
```

```
# 체인 실행하기

result = chain.invoke({
    "topic": topic,
    "style": style,
    "poet": poet
})
```

- 셀 실행 (3.0s)

```
# 결과 출력
print(f"\n=== {topic}에 대한 시 ===\n")
print(result)
```

=== 속초 관광 소개에 대한 시 ===

## 속초, 닳을 잃은 시간의 닳

(이상 시인의 문체를 참고하여 쓴 시)

1.

거울 속, 속초는 붉은 립스틱을 바른 채  
바다를 향해 웃는다.  
그 웃음은 갈매기의 비명처럼  
날카롭게, 부서진다.

오감도는 미처 날뛰는 물결의 춤사위  
짹짹한 침묵이 허끝을 간질이고  
하얀 포말은 흩어진 기억의 조각들  
잡으려 하면 사라진다.

2.

영금정, 낡은 피아노처럼  
바람에 낡은 선율을 토해낸다.  
그 소리, 귀를 찢는 파도 소리에 묻혀  
희미해진다.

동명항, 기다림으로 붉게 물든  
어부들의 얼굴  
그들의 눈빛은 먼 바다를 향한  
부서진 희망의 등대.

3.

설악산, 거대한 뼈대  
하늘을 굽는 봉우리들은  
침묵의 맹세를 속삭인다.  
단풍은 불타는 심장  
떨어진다, 떨어진다, 떨어진다.

케이블카, 날개 달린 상자  
허공을 가르며  
속삭임과 비명을 섞어낸다.  
아래, 작아지는 세상은  
꿈의 잔해처럼 흐릿해진다.

4.

중앙시장, 혼돈의 소용돌이  
향신료의 광기가 코끝을 찌르고  
사람들의 웃음과 욕설이 뒤섞여  
괴물처럼 울부짖는다.

햇별은 날카로운 칼날  
피부를 찢고  
그림자는 숨죽인 도망자  
길을 잃는다.

5.

속초, 시간의 닳을 잃은 배  
끝없이 표류하는 꿈의 파편  
거울에 비친 얼굴은 낮설고  
날개 없는 나  
부서진 허공에 떠다닌다.

오감도의 절규가 깃가를 맴돈다.  
속초, 속초, 속초...  
부서져라, 부서져라, 부서져라.

- 다른 주제로 시도해보기
  - 시의 주제: 2002년 월드컵 한국 4위의 훌륭한 결과
  - 스타일: 조선 시대 전통 시조 형식(3장 6구 12소절, 3·4조·4·4조 운율)을 엄격히 지키고, 고어 어미와 한자어를 섞어 쓴 고시조 문체, 초장·중장·종장의 3단 구성과 종장의 3음절+5음절 이상 규칙을 반영한 전통 시조 스타일, 운율과 장단을 맞추고, 고풍스러운 어휘(...하노라, ...하였네, ...하도다)를 사용한 문체, 한국 특유의 풍자와 해학이 잘 살아난 스타일
  - 시인: 윤동주

# 콘솔에서 입력 받기(순차적으로 입력하기)

```
topic = input("시의 주제를 입력하세요: ")
style = input("시의 스타일을 입력하세요 (예: 감성적인, 유머러스한): ")
poet = input("참고할 시인의 이름을 입력하세요 (예: 김소월, 윤동주): ")
```

# 체인 실행하기

```
result = chain.invoke({
    "topic": topic,
    "style": style,
    "poet": poet
})
```

- 셀 실행 (1.8s)

```
# 결과 출력
print(f"\n=== {topic}에 대한 시 ===\n")
print(result)
```

=== 2002년 월드컵 한국 4위의 훌륭한 결과에 대한 시 ===

## 2002년 월드컵, 태극전사 4강 신화 (윤동주 풍 고시조)

**\*\*초장\*\***

붉은 함성 하늘 찢르니, 땀방울 금빛 돌아 났네.  
압록강 푸른 물결 굽이쳐, 용맹한 기상 떨치도다.  
외적의 맹공 막아내니, 천하가 모두 놀랐노라.

**\*\*중장\*\***

가슴 벅찬 이 순간, 우리네 기쁨 넘치네.  
축구공 굴러가는 소리, 만고강산에 울려 퍼지네.  
산천초목 춤추는 듯, 태극기 휘날리니 장관이로다.

**\*\*종장\*\***

비록 4강의 꿈은 이루지 못하였으나,  
민족의 혼을 일깨운 이 영광, 길이 빛나리라.  
억겁의 세월 흘러도, 이 날을 잊지 않으리라.

- 다른 주제로 시도해보기\_2
  - 시의 주제: 노란 연필
  - 스타일: 향토적이고 감각적인 어휘, 방언과 고어를 섞어 쓰며, 어린 시절과 고향을 회상하는 스타일, 냄새·맛·계절감을 살린 토속적 이미지와 산문시적 구성, 평안북도 방언과 고향 음식·풍경 묘사를 살린 회고적 서사, 구어체와 의고체를 섞어 쓰는 백석 시인의 문체
  - 시인: 백석

# 콘솔에서 입력 받기(순차적으로 입력하기)

```
topic = input("시의 주제를 입력하세요: ")
style = input("시의 스타일을 입력하세요 (예: 감성적인, 유머러스한): ")
poet = input("참고할 시인의 이름을 입력하세요 (예: 김소월, 윤동주): ")
```



# 체인 실행하기

```
result = chain.invoke({
  "topic": topic,
  "style": style,
  "poet": poet
})

# 결과 출력
print(f"\n=== {topic}에 대한 시 ===\n")
print(result)
```

- 셀 실행 (3.6s)

=== 노란 연필에 대한 시 ===

## 누런 연필

어릴 적, 윗마루에 걸터앉아 띄약벌에 켜하니 눈을 뜬 날이면,  
손가락 사이로 \*\*보드라운\*\* 흙내음이 \*\*스며들던\*\* 그 누런 연필 한 자루.  
묵묵히, 닳고 닳아 \*\*둥글둥글\*\*해진 허리띠를 두른 듯,  
기름진 발뽀기처럼 묵직한 녀석이었지.

그 연필로 칠판 가득 \*\*써내려간\*\* 뽀뽀뽀한 글씨들,  
아직 젖네 나는 목청으로 \*\*외우던\*\* 옛날 동요 가락.  
가을이면 덜컹이는 짚차에 실려 \*\*나서던\*\* 추곡 수매,  
새벽 공기처럼 \*\*싸늘한\*\* 땀방울이 턱 끝에 \*\*매달렸었지\*\*.

\*\*아버지\*\*는 특하면 콧수염을 \*\*쓰다듬으시며\*\* 꺾꺾 웃으셨고,  
\*\*어머니\*\*는 \*\*가마솥\*\*에 송늬를 \*\*박박\*\* 끓여 \*\*송당송당\*\* 퍼주시던 그 맛.  
\*\*강냉이\*\* 튀기는 소리가 \*\*찰싹찰싹\*\* 껏가에 \*\*맴돌고\*\*,  
\*\*감자떡\*\* 냄새가 \*\*술술\*\* 코끝을 \*\*간질이던\*\* 그 시절.

\*\*평안도\*\* \*\*사투리\*\*가 \*\*구수하게\*\* \*\*흘러넘치던\*\* 저녁 마루,  
\*\*저 멀리\*\* \*\*압록강\*\* 물결처럼 \*\*잔잔히\*\* \*\*흘러가던\*\* \*\*세월\*\*이었지.  
\*\*누런\*\* 연필심이 \*\*닿아\*\* 없어지듯\*\*, \*\*추억\*\*도 \*\*하나둘\*\* \*\*덧없이\*\* \*\*사라져\*\* 갔지만\*\*,  
그 흙내음, 그 송늬 맛, 그 \*\*찰싹이는\*\* \*\*소리\*\*는 \*\*내 가슴속에\*\* \*\*고스란히\*\* \*\*남아\*\* \*\*반짝이고\*\* 있네.

\*\*아, 그 누런 연필!\*\*  
\*\*내 어린 날의\*\* \*\*꿈과\*\* \*\*함께\*\* \*\*닿아버린\*\* \*\*희망\*\*이었지.  
\*\*지금도\*\* \*\*가끔\*\* \*\*손끝이\*\* \*\*시큰하면\*\*  
\*\*그 녀석의\*\* \*\*온기가\*\* \*\*떠올라\*\*  
\*\*가슴 한구석이\*\* \*\*아련해지곤\*\* \*\*하는구나\*\*.

=== 노란 연필에 대한 시 ===

✓ 누런 연필

어릴 적, 윗마루에 걸터앉아 띄약벌에 켜하니 눈을 뜬 날이면,  
손가락 사이로 보드라운 흙내음이 스며들던 그 누런 연필 한 자루.  
묵묵히, 닳고 닳아 둥글둥글해진 허리띠를 두른 듯,  
기름진 발뽀기처럼 묵직한 녀석이었지.

그 연필로 칠판 가득 써내려간 뽀뽀뽀한 글씨들,  
아직 젖네 나는 목청으로 외우던 옛날 동요 가락.  
가을이면 덜컹이는 짚차에 실려 나서던 추곡 수매,  
새벽 공기처럼 싸늘한 땀방울이 턱 끝에 매달렸었지.

아버지는 특하면 콧수염을 쓰다듬으시며 꺾꺾 웃으셨고,  
어머니는 가마솥에 송늬를 박박 끓여 송당송당 퍼주시던 그 맛.  
강냉이 튀기는 소리가 찰싹찰싹 껏가에 맴돌고,  
감자떡 냄새가 술술 코끝을 간질이던 그 시절.

평안도 사투리가 구수하게 흘러넘치던 저녁 마루,  
저 멀리 압록강 물결처럼 잔잔히 흘러가던 세월이었지.  
누런 연필심이 닳아 없어지듯, 추억도 하나둘 덧없이 사라져 갔지만,  
그 흙내음, 그 송농 맛, 그 찰싹이는 소리는 내 가슴속에 고스란히 남아 반짝이고 있네.

아, 그 누런 연필!  
내 어린 날의 꿈과 함께 닳아버린 희망이었지.  
지금도 가끔 손끝이 시큰하면  
그 녀석의 온기가 떠올라  
가슴 한구석이 아련해지곤 하는구나.

# 콘솔에서 입력 받기(순차적으로 입력하기)

```
topic = input("시의 주제를 입력하세요: ")
style = input("시의 스타일을 입력하세요 (예: 감성적인, 유머러스한): ")
poet = input("참고할 시인의 이름을 입력하세요 (예: 김소월, 윤동주): ")
```

# 체인 실행하기

```
result = chain.invoke({
    "topic": topic,
    "style": style,
    "poet": poet
})
```

- 셀 실행 (2.2s)

```
# 결과 출력
print(f"\n=== {topic}에 대한 시 ===\n")
print(result)
```

=== 노란 연필에 대한 시 ===

```
## 노란 연필

바람이 햇살을 엮어 만든 듯
곱게 물든 노란 연필 하나

작은 손에 쥐어주니
세상이 온통 금빛으로 물드는구나

사각이는 소리마다
풀잎의 웃음소리가 묻어나고

하늘을 닮은 파란 눈망울엔
구름 조각이 춤추네

어린 날의 꿈들이
연필 끝에서 피어나
하얀 도화지에 별처럼 흩뿌려지네

정성껏 그린 한 송이 꽃
엄마의 얼굴을 닮았구나

노란 연필에 담긴
따스한 사랑이
세상을 환히 비추네
```

**\*\*박목월 시인의 문체를 참고하여 다음과 같은 부분을 신경 썼습니다.\*\***

- \* **\*\*자연 친화적인 소재와 이미지:\*\*** 햇살, 바람, 풀잎, 하늘, 구름, 꽃 등 자연에서 온 듯한 소재와 이미지를 활용했습니다.
- \* **\*\*맑고 고운 어휘:\*\*** '곱게 물든', '금빛', '사각이는', '웃음소리', '파란 눈망울', '춤추네', '별처럼 흩뿌려지네', '따스한', '환히 비추네' 등
- \* **\*\*서정적인 분위기:\*\*** 어린 날의 순수함, 꿈, 사랑 등 감성적인 내용을 담아 서정적인 분위기를 조성했습니다.

- \* \*\*시각적 묘사:\*\* '금빛으로 물드는구나', '하늘을 닮은 파란 눈망울', '별처럼 흩뿌려지네' 등 시각적인 이미지를 구체적으로 그려내려 했습니다.
- \* \*\*의성어, 의태어 활용:\*\* '사각이는'과 같은 의성어를 사용하여 생동감을 더했습니다.
- \* \*\*반복과 운율:\*\* 각 연의 마지막 부분에서 유사한 어조나 표현을 사용하여 자연스러운 운율을 만들고자 했습니다.

이 시가 박목월 시인의 감성과 어조를 어느 정도 담아내었기를 바랍니다.

- 같은 주제, 다른 시인 스타일

백석 st	
topic	노란 연필
style	향토적이고 감각적인 어휘, 방언과 고어를 섞어 쓰며, 어린 시절과 고향을 회상하는 스타일, 냄새·맛·계절감을 살린 토속적 이미지와 산문시적 구성, 평안북도 방언과 고향 음식·풍경 묘사를 살린 회고적 서사, 구어체와 의성어·의태어 활용
poet	백석 <div>## 노란 연필</div> <div>어릴 적, 뿔마루에 걸터앉아 뒤편에 켜놓은 책을 보며, 손가락 사이로 보드라운 흙내음이 스며들던 그 노란 연필 한 자루. 묵묵히, 닳고 닳아 둥글둥글해진 허리띠를 두른 듯, 기름진 발뺨기처럼 묵직한 녀석이었지.</div> <div>그 연필로 칠한 가득 써내려간 뽕뽕뽕한 글씨들, 아직 첫내 나는 목청으로 외우던 옛날 동요 가락. 가을이면 달경이는 뒷자리에 실려 나섰던 추곡 수매, 새벽 공기처럼 쌀쌀한 땀방울이 턱 끝에 매달렸었지.</div> <div>아부지는 특하면 콧수염을 쓰다듬으시며 곱게 웃으셨고, 어머님은 가마솥에 송늬를 박박 끓여 송당송당 퍼주시던 그 맛. 강냉이 튀기는 소리가 찰싹찰싹 쪼개에 맴돌고, 감자떡 냄새가 솔솔 코끝을 간질이던 그 시절.</div> <div>평안도 사투리가 구수하게 흘러넘치던 저녁 마루, 저 멀리 압록강 물결처럼 잔잔히 흘러가던 세월이었지. 노란 연필심이 닳아 없어지듯, 추억도 하나둘 덧없이 사라져 갔지만, 그 흙내음, 그 송늬 맛, 그 찰싹이는 소리는 내 가슴속에 고스란히 남아 반짝이고 있네.</div> <div>아, 그 노란 연필! 내 어린 날의 꿈과 함께 닳아버린 희망이었지. 지금도 가끔 손끝이 시큰하면 그 녀석의 온기가 떠올라 가슴 한구석이 아련해지곤 하는구나.</div>
etc	.