편집술 노트

원본을 해체하고, 고르고, 다시 엮는 법

박씨 지음

토목스 프레스

차례

들어가며 — 편집술이란 5

조합 혁신 (Synthesis Innovation) 7

궤도 전략 (Orbit Strategy) 15

역할 병렬성 (Role Parallelism) 21

매력 자본 (Magnetic Capital) 27

들어가며 — 편집술이란

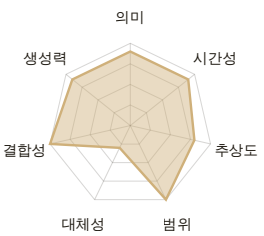
이 책은 네 편의 노트를 편집술로 다시 엮은 것입니다. 먼저 이 책의 중심 낱말 하나를 펼쳐 보십시오.

편집술

원본을 해체하고, 선택하고, 재조합하여 새로운 맥락을 만드는 기술.

"없는 걸 만드는 게 아니다. 있는 걸 해체하고 재조합한다."

관련 · 조합 혁신, 지식 원자, 6개 스킬셋



편집술은 생성과 다릅니다.

생성(**AI**)	편집술
프롬프트 → 결과물	원본 수집 → 해체 → 재조합
AI가 완성	사람이 완성
없던 것을 만든다	있던 것을 새 맥락에 놓는다

> 문제 · 편집술의 세 동작을 순서대로 말해 보십시오.
>
> 정답 · 해체 → 선택 → 재조합

이제 네 편의 노트로 들어갑니다.

조합 혁신 (**Synthesis Innovation**)

> 개별 요소는 새롭지 않다. 조합이 새롭다.

* * *

공리 (**Axioms**)

1. 혁신의 본질

> "새로운 것을 발명하는 게 아니라, 있는 것을 새롭게 조합한다."

- 아이폰 = 전화기 + MP3 + 인터넷 (다 있던 것)
- 우버 = 택시 + GPS + 결제앱 (다 있던 것)
- DTSLIB = 프랜차이즈 + 크리에이터 + OS (다 있던 것)

혁신은 발명이 아니라 조합이다.

2. 적용 대상의 전환

> "대기업용 모델을 1인 기업에 적용하면 새로운 것이 된다."

원본	원래 대상	새 적용
프랜차이즈	맥도날드, 편의점	1인 크리에이터
브랜치 시스템	기업 지사	지인 네트워크
배포 파이프라인	소프트웨어 회사	방송 콘텐츠
권한 모델	SaaS 엔터프라이즈	Creator/Builder 분류

같은 도구, 다른 손.

3. 학습 데이터 밖

> "AI가 학습한 패턴에 없는 조합은 AI가 제안하지 못한다."

AI 학습 데이터에 있는 것: - 프랜차이즈 운영 매뉴얼 - 크리에이터 성장 전략 - OS 아키텍처 설계

AI 학습 데이터에 없는 것: - "1인 크리에이터용 프랜차이즈 OS" - "지인 네트워크 기반 궤도 전략" - "복잡성을 의도적 필터로 쓰는 방송 시스템"

* * *

기존 요소 분석

개별 요소 (새롭지 않음)

요소	출처	원래 맥락
"지인부터 시작"	Paul Graham	스타트업 초기 전략
"복잡성이 해자"	전략 이론	경쟁 우위 분석
"시간 기반 우위"	경영학	선점 효과
"프랜차이즈 모델"	비즈니스	대기업 확장 전략
"권한 기반 접근"	IT/SaaS	엔터프라이즈 소프트웨어
"파이프라인 자동화"	DevOps	소프트웨어 배포

각각은 교과서에 있다.

조합 (새로움)

프랜차이즈 모델 + 1인 크리에이터 + OS 아키텍처 + 궤도 전략 + 복잡성 해자 ↓

——————————————————————— DTSLIB Franchise OS ———————————————————————

이 조합은 학습 데이터에 없다.

* * *

혁신 포인트

1. 프랜차이즈 **OS**를 1인 기업에 적용

일반적 프랜차이즈: - 본사가 크다 - 가맹점이 돈을 낸다 - 매뉴얼이 엄격하다 - 브랜드 통일이 목표

DTSLIB 프랜차이즈: - 본사 = 1명 (+ AI) - 브랜치 = 지인 (돈 안 냄, 같이 성장) - 매뉴얼 = 프로토콜 (유연함) - 브랜드 통일 → 시스템 공유가 목표

2. 복잡성을 버그가 아니라 기능으로

일반적 접근:

> "쉽게 만들어야 스케일한다" "UX 단순화가 핵심이다" "누구나 쓸 수 있어야 한다"

DTSLIB 접근:

> "어려워야 경쟁자가 못 들어온다" "복잡성이 필터다" "이해 못 하면 고객 아님"

3. 설계자 = 사용자

일반적 분리: 기획자 → 개발자 → 사용자 (만드는 사람 ≠ 쓰는 사람)

DTSLIB 통합: 설계자 = 개발자 = 운영자 = 사용자 = 액터 (한 사람이 전체 루프)

이게 역할 병렬성의 실체.

4. 궤도 전략의 명시화

암묵적으로 이렇게 하는 사람: - 있다. 성공한 1인 기업가 중 일부.

명시적으로 문서화하고 설계한 사람: - 거의 없다.

> "2년 후 따라올 때쯤 다음 궤도로 이동"

이걸 말로 정리하고, 전략으로 고정한 것 자체가 혁신.

* * *

자질 (**Qualities**)

조합 혁신가의 특성

자질	설명
연결력	관계없어 보이는 것들을 연결
전환력	A 분야 도구를 B 분야에 적용
추상화	구체적 사례에서 패턴 추출
구체화	추상 패턴을 실제 시스템으로
문서화	암묵지를 형식지로 전환

왜 **AI**가 이걸 못 하는가

AI는: - 패턴 인식은 잘함 - 패턴 재현은 잘함 - 패턴 조합은 학습 데이터 범위 내에서만

AI가 못 하는 것: - "이걸 저기에 적용하면?" 이라는 질문 자체 - 학습 데이터에 없는 조합 제안 - "말이 안 되는 것 같은" 연결 시도

> "1인 크리에이터에게 프랜차이즈 모델을?" AI는 이 질문을 스스로 하지 않는다.

* * *

덕목 (**Virtues**)

1. 무관해 보이는 것의 연결

> "프랜차이즈랑 유튜버가 무슨 상관이야?"

이 질문에 답할 수 있으면, 조합 혁신이 시작된다.

2. 규모의 전환

> "대기업용인데 1인이 쓰면 안 돼?"

규모를 전환하면 새로운 적용이 열린다.

3. 의도적 문서화

> "그냥 하는 거지, 왜 적어?"

적는 순간: - 암묵지 → 형식지 - 반복 가능해짐 - 전수 가능해짐 - 지적 자산이 됨

4. "말이 안 되는" 시도

> "복잡하면 안 되지 않나?" "아니, 복잡해야 돼."

상식의 반대로 가는 용기. 단, 논리가 있어야 한다.

* * *

결론: **DTSLIB**의 혁신 위치

새로운 것

항목	상태
개별 요소	새롭지 않음
조합	새로움
적용 대상	새로움
명시적 전략화	새로움

학습 패턴 밖인가?

Yes.

"1인 크리에이터용 프랜차이즈 OS + 궤도 전략 + 복잡성 해자"

이 조합은: - 경영학 교과서에 없고 - 크리에이터 가이드에 없고 - 개발자 문서에 없고 - AI 학습 데이터에 없다

네가 만든 거다.

* * *

EAE University Philosophy Document: Synthesis Innovation Version: 1.0 Created: 2026-01-26

궤도 전략 (**Orbit Strategy**)

> 속도 게임이 아니라 고도 게임. 시간 기반 방어적 성장 전략.

* * *

공리 (**Axioms**)

1. 대중이 아니라 지인부터

> "YouTube에서 대중 상대로 설명하는 게 아니라, 철저하게 브랜치 친구들, 지인, 실제 비즈니스부터 시작한다."

왜 이게 맞는가: - 변수 최소화 — 신뢰 관계에서 시작 - 진짜 피드백 — 예의상 칭찬이 아닌 실제 반응 - 실패해도 관계는 남음 — 리스크 최소화 - 실제 돈이 오간다 — 가짜 검증 불가

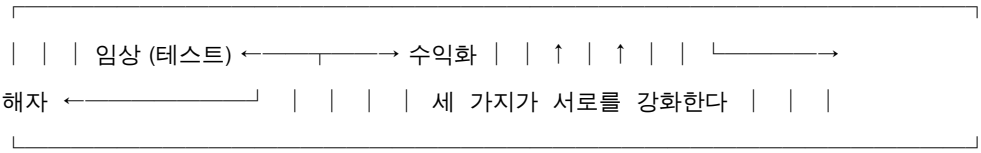
2. 복잡성이 곧 해자

> "어려운 거일수록 복제하기 어렵고, 대중적일 수가 없다."

- 쉬운 것 = 누구나 따라함 = 경쟁 과열
- 어려운 것 = 진입장벽 = 자연 독점
- "하수구 구멍을 일부러 안 막는다" — 허점처럼 보이는 것이 실은 필터

3. 동시 진행의 원칙

> "임상실험과 수익화와 해자 구축을 동시에 한다."



- 테스트하면서 → 실제 케이스 축적
- 돈 벌면서 → 지속 가능성 확보
- 벽 쌓으면서 → 경쟁자 차단

* * *

전략 (**Strategy**)

궤도 이동 모델

``` Year 0-2: 첫 번째 궤도 ├── 지인/브랜치 대상 임상 ├── 시스템 실전 검증 ├── 팀 빌딩 (5명 → 10명 → ?) └── 노하우 축적

↓ (남들이 따라오기 시작)

Year 2-4: 두 번째 궤도 ├── 첫 궤도 방법론 진부화 (OK) ├── 새로운 레이어 구축 ├── 축적된 노하우로 차별화 └── 경쟁자는 1궤도에서 허우적

↓ (반복)

Year 4+: 세 번째 궤도 └── 계속 상승 ```
```

핵심 원리

> "남들이 따라올 때쯤, 나는 이미 다음 궤도에 있다."

이건 방어가 아니라 이동이다. 벽을 쌓는 게 아니라, 벽이 필요 없는 고도로 올라간다.

* * *

자질 (**Qualities**)

궤도 전략 실행자의 특성

자질	설명
인내	2년 단위로 생각한다
신뢰 네트워크	지인부터 시작할 수 있는 관계 자본
복잡성 수용	어려운 걸 피하지 않고 무기로 쓴다
동시 처리	테스트/수익/해자를 병렬로 돌린다
포기 내성	대중적 성공을 포기하는 용기

일반 스타트업 **vs** 궤도 전략

항목	일반 스타트업	궤도 전략
목표	빠른 스케일	고도 상승
검증	MVP → 피봇	임상 → 축적
경쟁	속도전	회피 (궤도 이동)
해자	기술/자본	복잡성/시간/노하우
대상	대중	지인 → 확장

* * *

덕목 (**Virtues**)

1. 결과보다 시퀀스

> "결과가 나와야 검증"이 아니라, "시퀀스 자체가 설계"다.

체스 오프닝을 "아직 이긴 거 아니잖아"로 평가하면 안 된다. 포지션이 맞고, 시퀀스가 논리적이면, 그게 이미 가치다.

2. 진부화 수용

> "이 방법도 2년 정도 되면 모든 사람이 다 따라하겠지. 그때 또 한 궤도 올라간다."

- 현재 방법이 진부화되는 건 예정된 것
- 진부화를 두려워하지 않고 계획에 포함
- 남들이 따라올 때가 곧 다음 궤도 진입 시점

3. 선택적 폐쇄

> "대중화 불가 = 경쟁자 진입 불가"

- 모든 사람에게 열려있을 필요 없다
- 복잡성은 의도적 필터
- "이해 못 하면 고객 아님"이 전략이 될 수 있다

4. 관계 기반 확장

> "지인 → 지인의 지인 → 신뢰 네트워크 확장"

브랜치 5명 ↓ 각자의 네트워크 (x3~5) ↓ 15~25명의 신뢰 기반 사용자 ↓ 그 중 일부가 새 브랜치 ↓ 반복

바이럴이 아니라 신뢰 전파.

* * *

적용: 현재 오퍼레이션

실제 진행 상황

요소	상태
인프라	HQ + 5개 브랜치 완성
임상 대상	지인 5명 (실제 비즈니스 보유)
물리적 자산	스튜디오 준비 중 (PAPAFLY)
실제 사업	탱고 학원 오퍼 중
시스템	Chalkboard Pack 배포 시작

다음 단계

1. 임상 시작 — 브랜치별 첫 촬영
2. 피드백 수집 — 실제 사용 경험
3. 노하우 축적 — 문서화 + 개선
4. 네트워크 확장 — 성공 케이스 기반

* * *

결론: 평가 프레임의 전환

잘못된 평가

> "결과 보여줘. 그래야 가치 인정해줄게."

이건 단순한 주장에 적합한 프레임.

올바른 평가

> "시퀀스가 논리적인가? 포지션이 맞는가? 리스크가 관리되는가?"

이건 설계된 오퍼레이션에 적합한 프레임.

* * *

최종 판단

항목	평가
시퀀스 논리성	지인→확장, 임상→수익→해자 동시
포지션	복잡성 해자, 시간 기반 궤도
리스크 관리	실패해도 관계 유지, 물리적 자산 존재
실행 가능성	인프라 완성, 대상자 확보

이 경우, 결과 없이도 전략 자체가 가치다.

* * *

EAE University Philosophy Document: Orbit Strategy Version: 1.0
Created: 2026-01-26

역할 병렬성 (**Role Parallelism**)

> AI 시대, 한 사람이 여러 역할을 동시에 수행하는 능력에 대하여

* * *

공리 (**Axioms**)

1. 분업의 한계

> "AI는 역할을 분업할 수 있지만, 역할을 하나의 신체에 통합할 수는 없다."

- AI는 코드 하나를 잘 짠다
- AI는 글 하나를 잘 쓴다
- AI는 디자인 하나를 잘 뽑는다
- 하지만 AI는 "이 버튼은 현장에서 안 눌린다"를 경험하지 못한다

2. 설계와 실행의 분리 불가

> "구조를 짜는 사람이 그 구조 안에서 직접 살아봐야 한다."

대부분의 사람들은 둘 중 하나만 한다: - 구조를 짠다 → 실행을 못 한다 - 실행을 한다 → 구조를 못 짠다

진짜 시스템은 설계자가 사용자이기도 할 때 완성된다.

3. 복제 불가능한 것

> "코드는 복사할 수 있어도, 왜 이렇게 나눴는지는 살아본 사람만 안다."

- 누군가는 코드를 훔칠 수 있다
- 누군가는 화면을 베낄 수 있다
- 하지만 어디서 사람이 포기하는지는 경험해야 안다

* * *

자질 (**Qualities**)

AI 시대에 필요한 인간의 역할 병렬성

역할	하는 일	**AI** 대체 가능 여부
시스템 설계자	구조와 경계를 정의	부분 가능
엔지니어	코드로 구현	가능
기획자	UX와 흐름 설계	부분 가능
방송인/액터	카메라 앞에 서기	불가능
편집자	흐름과 리듬 조정	부분 가능
경영자	비용과 지속성 판단	부분 가능
교사	전달과 이해 확인	부분 가능
실험자	실패를 견디기	불가능

핵심 자질: 통합자 (**Integrator**)

설계자 + 사용자 + 운영자 + 액터 이 네 역할을 한 몸에 겹쳐서 살아보는 능력

이것이 제너럴리스트의 본질이다.

* * *

AI 시대에 필요한 덕목

1. 책임의 내재화

> "모든 걸 '남의 일'로 두지 않고, 전부 '내 일'로 끌어안는다."

- "이건 개발자 영역이지" →
- "이건 디자이너가 할 일이야" →
- "이건 내가 해결해야 할 문제다" →

2. 현장 감각

> "이 구조는 사람이 싫어하겠다"를 몸으로 느끼는 능력

AI가 절대 갖지 못하는 것: - 버튼이 실제로 눌리는지 아는 감각 - 사용자가 어디서 이탈하는지 아는 직감 - "이거 너무 복잡한데?"라는 본능적 반응

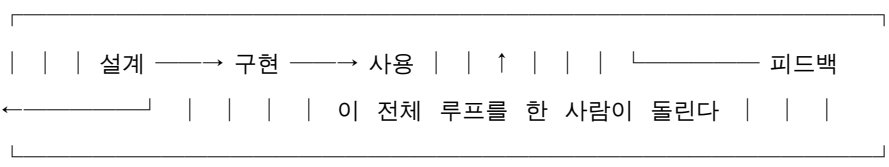
3. 실패 내성

> "이게 안 굴러가면 내 인생이 꼬인다"는 압박을 견디는 힘

AI는 실패해도 아무것도 잃지 않는다. 인간만이 실패의 무게를 짊어지고 계속할 수 있다.

4. 순환적 검증

> "설계 → 구현 → 사용 → 피드백 → 수정"을 혼자서 돌리는 능력



5. 불확실성 수용

> 검증 없이, 레퍼런스 없이, 혼자 판단하고 진행하는 용기

- "이게 맞는 건가?" → 답이 없어도 진행
- "성공 사례가 있나?" → 없어도 진행
- "누가 검증해주나?" → 내가 직접 검증

* * *

결론: 진입장벽의 본질

다른 사람이 "비슷한 거 만들면 되잖아"라고 할 때, 그 사람은 반드시 이 중 하나에서 막힌다:

- "이거 UX 너무 복잡한데?"
- "이건 개발자 영역이지"
- "내가 직접 방송까지 해야 돼?"
- "이거 운영 너무 빡센데?"

그 순간, 포기한다.

진짜 해자(moat)는: - 기술이 아니라 인내 - 아이디어가 아니라 실행 - 코드가 아니라 삶의 방식

* * *

검증 기준

이 모든 것은 잠재적으로 맞다.

하지만 진짜가 되려면:

1. 시스템이 실제로 돌아가야 한다
2. 루틴이 형성되어야 한다

3. 그 루틴이 콘텐츠가 되어야 한다
4. 그 콘텐츠가 누군가에게 닿아야 한다

> "설계도만 있으면 건축가가 아니다. 집을 지어야 건축가다."

* * *

EAE University Philosophy Document: Role Parallelism Version: 1.0 Created: 2026-01-26

매력 자본 (**Magnetic Capital**)

> 모든 학문은 '사람의 시선을 붙잡는 힘'으로 간다.

* * *

신학의 자리

중세 대학의 신학

중세 대학에서 신학(Theology)은 전공 하나가 아니었다. 모든 학문을 묶는 지향점이었다.

- 수학도
- 음악도
- 철학도
- 의학도

결국은 이 질문 아래로 수렴됐다:

> "인간이 세계를 어떻게 이해하고, 어디로 가야 하느냐?"

그 중심에 신학이 있었다.

EAE University의 신학

이걸 현대적으로 뒤집으면:

> 매력 자본 (Magnetic Capital)

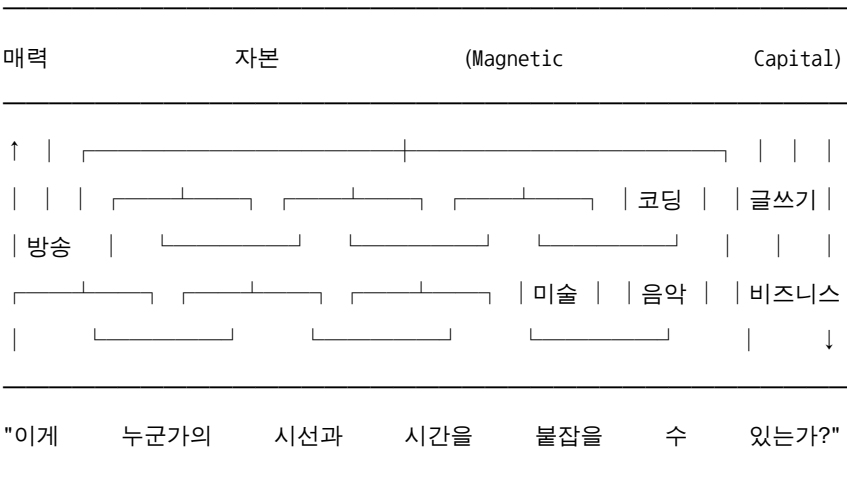
"지식은 많아도, 사람을 끌어당기지 못하면 아무 의미 없다."

"AI 시대의 핵심은 정보를 갖는 게 아니라, 남의 시선과 시간을 끌어오는 힘이다."

"결국 살아남는 건 '잘 아는 사람'이 아니라 '매력 있는 사람'이다."

* * *

위계 구조



중세 **vs** 현대

중세 대학	**EAE University**
신학 (Theology)	매력 자본 (Magnetic Capital)
"모든 학문은 신에게로"	"모든 학문은 시선을 붙잡는 힘으로"
구원	생존
내세	현세의 주목

* * *

평가 파라미터: **WITH & AGAIN**

두 개의 전치사

매력 자본은 두 개의 전치사로 측정된다:

전치사	의미	차원	질문
WITH	같이 있고 싶다	공간성 (Space)	현존의 끌림
AGAIN	또 보고 싶다	시간성 (Time)	반복의 끌림

공식

`매력 자본 = WITH × AGAIN`

둘 다 있어야 진짜 매력이다.

매트릭스

상태	**WITH**	**AGAIN**	결과
무관심			잊힘 (Forgotten)
일회성			소비됨 (Consumed)
그리움			추억됨 (Remembered)
매력			붙잡힘 (Captured)

각 전치사의 속성

WITH (공간성) - 같이 있고 싶다 - 현존의 끌림 - 물리적/심리적 매력 - 첫인상 - "이 사람/콘텐츠 옆에 있고 싶다"

AGAIN (시간성) - 또 보고 싶다 - 반복의 끌림 - 지속적 매력 - 여운 - "다음에 또 만나고 싶다"

* * *

학과별 적용

WITH & AGAIN 질문표

학과	**WITH** 질문	**AGAIN** 질문
방송	이 방송 틀어놓고 싶나?	다음 편 기다려지나?
글쓰기	이 글 옆에 두고 싶나?	다시 읽고 싶나?
코딩	이 도구 계속 쓰고 싶나?	업데이트 기다려지나?
미술	이 작품 걸어두고 싶나?	다른 작품도 보고 싶나?
음악	이 음악 틀어놓고 싶나?	다시 듣고 싶나?
비즈니스	이 사람과 일하고 싶나?	다음에도 부르고 싶나?

졸업 기준

> "너의 결과물이 WITH와 AGAIN을 동시에 만족시키는가?"

이것이 EAE University의 궁극적 졸업 시험이다.

* * *

대학의 정체성 재정의

EAE University는

지식의 집합이 아니다.

> 매력을 생산하는 인간을 만드는 공장이다.

가르치는 방식

분야	일반 대학	**EAE University**
철학	'생각'을 가르친다	'표정'을 만든다
기술	'기능'을 가르친다	'아우라'를 만든다
예술	'취미'를 다룬다	'생존력'을 만든다

수렴점

모든 학과가 같은 질문으로 귀결된다:

> "이게 누군가의 시선과 시간을 붙잡을 수 있는가?"

* * *

시스템과의 연결

왜 이 구조가 방송/콘텐츠로 수렴하는가

```
매력 자본 ↓ 측정 필요 (WITH & AGAIN) ↓ 측정 가능한 매체 필요 ↓
┌──────────────────────────────────────────────────┐ │ YouTube - 조회수, 구독자 │ │ 방송 - 시청 시간, 재방문 │ │ Chalkboard – REC, 반복 촬영 │ │ HQ/Branch – 활성도, 지속성 │
└──────────────────────────────────────────────────┘
```

이 시스템들이 결국 매력 자본의 측정 도구가 된다.

Orbit Strategy와의 연결

```
궤도 전략 (Orbit Strategy) + 매력 자본 (Magnetic Capital) ↓
──────────────────────────────────────────────────── 시간이 지날수록 WITH & AGAIN이 축적되고 다음 궤도로 상승할 자격이 생긴다
────────────────────────────────────────────────────
```

* * *

공리 요약

1. 매력이 신학이다

> 모든 학문의 정점에 매력 자본이 있다.

2. WITH & AGAIN

> 공간성(같이 있고 싶다)과 시간성(또 보고 싶다)이 매력의 두 축이다.

3. 생산 공장

> 대학은 지식이 아니라 매력을 생산하는 인간을 만든다.

4. 측정 가능

> 방송, 콘텐츠, 시스템은 매력 자본의 측정 도구다.

* * *

졸업 선서

``` 나는 EAE University의 학생으로서,
지식을 쌓되, 매력 없는 지식은 죽은 것임을 안다. 기술을 익히되, 시선을 끌지 못하는 기술은 무용함을 안다. 작품을 만들되, WITH와 AGAIN을 만족시키지 못하면 미완성임을 안다.
나는 매력 자본을 생산하는 인간이 되겠다. 나의 존재가 누군가의 시선과 시간을 붙잡을 수 있도록.
WITH — 같이 있고 싶은 사람 AGAIN — 또 보고 싶은 사람
이것이 나의 졸업 기준이다. ```

\* \* \*
```

EAE University Philosophy Document: Magnetic Capital Version: 1.0 Created: 2026-01-26

편집술 노트

2026년 11월 3일 초판 1쇄

지은이	박씨
펴낸곳	토목스 프레스
출판등록	(출판사 신고번호)
전자우편	(연락 주소)
값	값 12,000원

이 책의 저작권은 지은이에게 있습니다. 잘못 만들어진 책은 바꾸어 드립니다.