

**`/* elice */` Enterprise**

# LLM 에이전트 구축 커리큘럼 설계

**2주 압축 과정**

가전·전자 제조업 대기업 R&D 본부 임직원 대상 | 4주 과정

Python 기초부터 RAG 기반 에이전트 구축까지

# 목적

기업 담당자가 초기 요청한 2주 과정의 설계안입니다.  
4주 과정 대비 어떤 내용이 축소/생략되는지, 교육 효과 측면의 한계점을 명시합니다.

- |   |               |                     |
|---|---------------|---------------------|
| 1 | 4주 → 2주 압축 전략 | 운영 조건 변경 및 핵심 원칙    |
| 2 | 내용 축소/생략 비교표  | 각 항목별 변경 상세 및 영향 분석 |
| 3 | 2주 커리큘럼 상세    | 10일 일정 전체 시간표       |
| 4 | 교육 효과 한계점     | 5가지 한계점 구체적 분석      |
| 5 | 결론 및 제안       | 적합 상황, 제안 이유, 절충안   |

PART 1

---

# 4주 → 2주 압축 전략

운영 조건 변경

압축 핵심 원칙

축소/생략 비교

# 운영 조건 변경

항목	4주 과정	2주 압축 과정	비고
기간	4주 × 5일 = 20일	2주 × 5일 = 10일	50% 축소
일일 시간	6시간	7시간 (09:00~17:00)	1시간 확대
총 교육시간	120시간	70시간	42% 수준
사전 교육	3시간 (온라인)	8시간 (온라인, 필수)	Python 기초 이동
프로젝트 기간	5일 (1주)	2.5일	50% 축소

## 압축 핵심 원칙

### 1 사전 교육 강화

1주차 Day1~3 Python 기초를  
사전 온라인 학습으로 이동

### 2 핵심 기술 집중

프롬프트·LangChain·RAG에 집중,  
부가 내용 생략

### 3 프로젝트 축소

기획~발표를 2.5일로 압축

# 내용 축소/생략 비교표 (1/2)

4주 과정 내용	2주 과정	변경 상세	영향
Python 기초 (3일)	사전 교육 이동	1주차 Day1~3 → 사전 학습 8h. 본교육 Day1 오전 2h 보충만	⚠
에러 대응법 (1세션)	사전 교육 포함	본 교육에서 별도 세션 없음	⚠
프롬프트 엔지니어링 (2일)	1일로 축소	기초+심화 병합. CoT·Few-shot 실습 시간 절반	⚠
LangChain 기초+심화 (2일)	1일로 축소	체인+메모리+도구를 1일에 압축	●
파인튜닝 vs RAG 비교 (1세션)	✕ 생략	개념만 RAG 세션에서 5분 언급	⚠
벡터 DB + 문서 처리 (2일)	1일로 축소	임베딩·청킹·ChromaDB를 1일에 압축	⚠

✕ 생략 = 완전 제거 | ● = 교육 목표 달성에 직접 영향 | ⚠ = 학습 품질 저하 위험

# 내용 축소/생략 비교표 (2/2)

4주 과정 내용	2주 과정	변경 상세	영향
RAG 파이프라인 구축 (1일)	✅ 유지	핵심 세션이므로 1일 온전히 유지	-
할루시네이션 대응 (1일)	0.5일로 축소	RAG 구축일 오후에 병합	⚠️
LangGraph 소개 (1세션)	❌ 생략	완전 생략. 후속 학습 로드맵에서만 언급	●
미니 프로젝트 ①②③ (3회)	1회로 축소	③만 유지. ①②는 일일 실습 산출물로 대체	●
최종 프로젝트 (5일)	2.5일로 축소	기획 반일 + 개발 1.5일 + 발표 반일	●
현업 전환 가이드 (2세션)	1세션 축소	Google Cloud + 보안 체크리스트를 1h에 압축	⚠️
개인 역할 분담 + 산출물	간소화	역할은 분담하되 개인 산출물 제출 범위 축소	⚠️

❌ 생략 = 완전 제거 | ● = 교육 목표 달성에 직접 영향 | ⚠️ = 학습 품질 저하 위험

# 2주 과정 강화 사전 교육 (필수 8시간)

4주 과정 사전 교육(3시간, 선택)과 달리, 2주 과정의 사전 교육은 필수이며 분량이 크게 확대됩니다.

내용	소요	4주 과정에서의 위치
Python/Anaconda 설치 + Jupyter 실행	30분	사전 교육 동일
Python 기초 ①: 변수, 데이터 타입, 리스트, 딕셔너리	2시간	1주차 Day1
Python 기초 ②: 조건문, 반복문, 함수	2시간	1주차 Day2
Python 기초 ③: 파일 읽기(CSV, JSON), pip 설치	1.5시간	1주차 Day3
에러 메시지 읽는 법 + AI에게 에러 물어보기	30분	1주차 Day3
ChatGPT 사용 체험 + LLM 개념 영상	30분	사전 교육 동일
자가 점검 퀴즈 (10문항, 통과 기준 70%)	30분	신규
환경 테스트 코드 실행	30분	사전 교육 동일
합계	8시간	

⚠️ 핵심 리스크: 사전 교육 8시간을 현업 병행하면서 완료하지 못하면 Day 1부터 따라가지 못함

# 교육 운영 설계

## 기간

2주 × 5일 = 10일

## 일일 시간

7시간 (점심 제외)

## 총 교육시간

70시간

## 인원 / 조 편성

20~30명 / 5인 1조

## 일일 시간표

● 09:00-09:50	오전 1	강의 / 복습	50분
● 10:00-10:50	오전 2	강의 / 데모	50분
● 11:00-12:00	오전 3	데모 / 가이드 실습	60분
● 12:00-13:00	점심	식사 및 휴식	60분
● 13:00-13:50	오후 1	실습	50분
● 14:00-14:50	오후 2	실습	50분
● 15:00-15:50	오후 3	실습	50분
● 16:00-17:00	오후 4	회고 / 미니퀴즈 / Q&A	60분



PART 2

---

# 2주 커리큘럼

1주차: 기초→핵심기술→RAG

2주차: 에이전트 완성→프로젝트

# 2주 과정 전체 구조

## Week 1

기초→핵심기술→RAG

Python 보충 → LLM API →  
프롬프트 → LangChain →  
벡터DB → RAG 구축



## Week 2

에이전트 완성→프로젝트

할루시네이션 대응 →  
미니 프로젝트 →  
최종 프로젝트 구축·발표

### 10일 일정 요약

D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
Python 보충+API	프롬프트 엔지니어링	Lang Chain	벡터DB 문서처리	RAG+ 할루시네이션	미니 프로젝트	킵오프 개발시작	개발	마무리 발표준비	발표 수료

## WEEK 1

# 1주차: 기초부터 RAG까지 — 속성 과정

1  
DAY

오전 오리엔테이션 + 로드맵 · Python 보충: 딕셔너리, 함수, CSV/JSON  
오후 LLM 개념 + API 보안 수칙 · 첫 API 호출 · 시스템 메시지 · 미니퀴즈  
🏆 산출물: 가전제품 QA 챗봇

3일 → 사전 교육 + 0.5일

2  
DAY

오전 프롬프트 엔지니어링: 역할, 맥락, 형식 · Few-shot + CoT + JSON 출력  
오후 불량 분류 프롬프트 설계 · 반복 개선 · 파인튜닝 vs RAG 5분 소개 · 미니퀴즈  
🏆 산출물: 불량 분류 자동화 프롬프트

2일 → 1일

3  
DAY

오전 LangChain: Model, Template, Parser, Chain · Sequential Chain 데모  
오후 대화 메모리 + 도구(Tool) · 메모리 챗봇 + 계산 도구 연결 · 미니퀴즈  
🏆 산출물: 메모리+도구 상담 챗봇

2일 → 1일

4  
DAY

오전 임베딩 + 벡터 DB · ChromaDB 설치/저장/검색 · 문서 로딩 + 청킹  
오후 PDF → 청킹 → 임베딩 → ChromaDB 전체 파이프라인 · 유사도 검색 · 미니퀴즈  
🏆 산출물: PDF → 벡터 DB → 검색 파이프라인

2일 → 1일

5  
DAY

오전 RAG 아키텍처 · LangChain RAG 체인 · RAG QA + 출처 표시  
오후 할루시네이션 대응 4기법 · RAG 적용 · 파라미터 실험 · 1주차 총정리  
🏆 산출물: 할루시네이션 대응 RAG QA 시스템

2일 → 1일

## 2주차: 에이전트 완성 & 프로젝트

1  
DAY

오전 1주차 전체 복습 · 미니 프로젝트 안내 · 조별 프로젝트 시작  
오후 프로젝트 수행 계속 · 조별 시연 + 피드백 · 최종 프로젝트 사전 안내  
🏆 산출물: RAG 기반 기술문서 검색 에이전트

미니 프로젝트 3회 → 1회

2  
DAY

오전 AI 과제 도출 방법론 · 브레인스토밍 → 과제 확정 · 기획서 작성  
오후 1:1 멘토링 · 데이터 준비 + 벡터 DB 구축 시작 · ✅ 벡터 DB 적재 확인  
🏆 산출물: 기획서 + 벡터 DB 초기 구축

프로젝트 기간 5일 → 2.5일

3  
DAY

오전 RAG 파이프라인 구축 + 프롬프트 설계 · ✅ 기본 RAG 작동 확인  
오후 할루시네이션 대응 + 테스트 · 현업 전환 가이드 (1h) · ✅ 기능 완성  
🏆 산출물: 기능 완성 에이전트

현업가이드 2세션 → 1세션

4  
DAY

오전 최종 테스트 + 디버깅 · 발표 자료 작성 · 리허설  
오후 코드 정리 + 개인 산출물 제출 · 조별 최종 발표 (조당 8분)  
🏆 산출물: 에이전트 + 발표

개인 산출물 간소화

5  
DAY

오전 조별 발표 계속 + 동료 평가 · 강사 피드백 + 전체 토론  
오후 다음 단계 로드맵 · 현업 전환 보충 Q&A · 전체 회고 + 수료식  
🏆 산출물: 🎉 수료

LangGraph 완전 생략

## PART 3

---

# 교육 효과 한계점 분석

전체 사이클 깊이

PBL 누적 구조

요구사항 미충족

사전 교육 의존도

학습자 이탈 위험

요구사항 충족도

# 한계점 ①: 전체 사이클 경험의 깊이 부족

전체 사이클을 "경험"하기는 하지만, 각 단계의 반복·실험·최적화 시간이 부족하여 "이해"보다 "따라하기"에 그칠 위험

단계	4주 과정	2주 과정
Python 기초	3일 (실습 충분)	사전 8h + 보충 반일 (자가학습 의존)
프롬프트 설계	2일 (반복 실험)	1일 (패턴 소개 위주)
LangChain	2일 (체인+메모리+도구)	1일 (속성 실습)
벡터DB + RAG	3일 (구축→최적화→대응)	1.5일 (구축+기본 대응)
프로젝트	5일 (기획→개발→테스트→발표)	2.5일 (압축)

# 한계점 ②③: PBL 약화 & 요구사항 미충족

## ② PBL 누적 구조 약화

4주: 미니 프로젝트 3회 (①→②→③)  
기술이 점진적으로 누적

2주: 미니 프로젝트 1회 (③만)  
"이전 것 위에 쌓는" 경험이 희석

→ 기술 간 연결 감각 부족

## ③ 요구사항 부분 미충족

요구사항	4주	2주
파인튜닝/RAG (R5)	✓	⚠ 5분만
LangGraph (R10)	✓	✗ 생략
할루시네이션 (R8)	✓ 1일	⚠ 0.5일
개별 과제 (R11)	✓	⚠ 간소화
현업 전환 (R4)	✓ 2세션	⚠ 1세션

# 한계점 ④⑤: 이탈 위험 & 사전 교육 의존도 증가

## ④ 학습자 이탈 위험 증가

비SW 입문자에게 하루 7시간, 매일 새 개념 인지 과부하 유발

4주: 주차별 금요일 프로젝트로 숨을 고름

2주: 1주차 5일 연속 새 기술 도입

→ 피로 누적이 심각

## ⑤ 사전 교육 의존도

사전 교육 8시간에 전적으로 의존

현업 병행 중 8시간 자가학습 완료하지 못하는 수강생이 발생할 수 있음

Day 1부터 격차 → 교육 전체 흔들림

4주 과정(사전 3h, 선택적)에서는 발생하지 않는 구조적 리스크



# 요구사항 충족도 추적: 2주 vs 4주

	요구사항	2주	4주	차이
R1	비SW 직군 대상 설계	0	0	사전 교육 의존도 증가
R2	Python→LLM 전체 과정	0	0	각 단계 깊이 50% 축소
R3	과제→알고리즘→LLM 사이클	△	0	프로젝트 2.5일로 압축
R4	현업 데이터 전환 가이드	△	0	2세션 → 1세션 축소
R5	파인튜닝/RAG 방법	△	0	전용 1세션 → 5분 언급
R6	벡터 DB 구성	0	0	청킹 전략 실험 시간 부족
R7	프롬프트 구성	△	0	2일 → 1일, 실습 절반
R8	할루시네이션 최적화	△	0	전용 1일 → 반일 축소
R9	간단한 에이전트 구축	△	0	PBL 누적 약화
R10	LangChain/LangGraph	X	0	LangGraph 완전 생략
R11	핵심 인력/개인 산출물	△	0	개인 산출물 범위 축소
R12	PBL 기반 설계	X	0	미니PJ 3회→1회, 누적 약화
R13	20~30명 운영	0	0	강사 의존도 증가
R14	사전 교육	△	0	3h→8h 필수, 구조적 리스크
R15	경영진 방침 연계	0	0	-

2주: 0=5, △=8, X=2  
4주: 15/15 충족  
요구사항의 67%만 부분 충족

## PART 4

---

# 결론 및 제안

2주 vs 4주 비교

핵심 메시지

절충안: 2+2 분할

# 결론: 2주 vs 4주 비교

## 2주 과정이 적합한 경우

- ✓ 수강생이 이미 Python 기초를 보유한 경우  
(기초 문법이 아닌 실무 사용 경험)
- ✓ 교육 목표를 "RAG 개념 이해 + 체험"으로  
하향 조정하는 경우

## 4주 과정을 제안하는 이유

- 1 Python을 "도구로 쓸 수 있는" 수준까지  
도달해야 함 (최소 3일)
- 2 프롬프트 → 체인 → RAG를 누적적으로  
쌓아야 함 (미니 프로젝트 3회)
- 3 기획 → 설계 → 구현 → 테스트 → 발표  
전 과정을 직접 경험해야 함 (최소 5일)

# 핵심 메시지

## 교육 목표

"간단한 LLM 에이전트 구축 프로젝트를 위한 개별 과제를 도출하고,  
알고리즘 개발 및 LLM을 연동하여 자신만의 에이전트를 구축할 수 있다"

핵심 키워드: "자신만의 에이전트를 구축"

## 2주 과정

전체 사이클을 "체험"할 수 있지만  
"자신만의" 에이전트를 만들기에는  
시간이 부족합니다.



## 4주 과정

이 차이를 채워주는 최소 기간입니다.  
"자신만의" 에이전트를 구축하려면  
4주가 필요합니다.

# 제안: 2+2 분할 운영

2주 연속 교육이 업무 공백 측면에서 부담이라면, 분할 운영을 제안합니다.

1~2주차

교육

기초 → 핵심기술 → RAG 기초

Python 기초 → 프롬프트 엔지니어링 → LangChain → 벡터 DB → RAG 파이프라인 기초 구축

3~4주차

현업 복귀

과제 수행 기간

현업에 복귀하여 업무 수행 → 과제: "내 업무에서 에이전트로 해결할 문제 1가지 정리"

5~6주차

교육

심화 → 프로젝트 완성

RAG 심화 → 할루시네이션 대응 → 최종 프로젝트 (현업 데이터 기반) → 발표

4주 교육 효과를 유지하면서, 현업 공백을 2주 단위로 분산할 수 있습니다.

감사합니다