

교육 과정 소개서.

테디노트의 RAG 비법노트 : 랭체인을 활용한 GPT부터 로컬
모델까지의 RAG 가이드



강의정보

강의장	온라인 강의 데스크탑, 노트북, 모바일 등
수강 기간	평생 소장
상세페이지	https://fastcampus.co.kr/data_online_teddy#curriculum
강의시간	66시간 43분
문의	고객센터

강의특징

나만의 속도로	낮이나 새벽이나 내가 원하는 시간대 에 나의 스케줄대로 수강
원하는 곳 어디서나	시간을 쪼개 먼 거리를 오가며 오프라인 강의장을 찾을 필요 없이 어디서든 수강
무제한 복습	무엇이든 반복적으로 학습해야 내것이 되기에 이해가 안가는 구간 몇번이고 재생



강의목표

- 테디노트와 한 번에 끝내는 LangChain & RAG 비법노트가 특별한 7가지 이유 시대의 흐름을 누구보다 앞서는 경험, 테디노트가 강의를 통해 끝까지 책임지겠습니다.
- 2만 구독자의 설문 결과를 반영하여, 기초적인 내용부터 가장 어려워하는 내용까지 담았습니다.
기본 개념부터 상황별 최적의 해결책을 찾기 위한 과정까지 자세히 알려드리겠습니다.
- 미래기술로 각광받고 있는 멀티 에이전트 개념과 최신 논문 기반의 Advanced RAG를 LangGraph를 활용한 실습을 통해 학습합니다.

강의요약

- 기초 개념부터 실무 수준의 최신 RAG 까지, 여러분들의 성공적인 RAG를 위한 모든 내용을 담아 구성하였습니다!
- 강의를 통해 GPT와 같은 유료 모델뿐만 아니라 많은 기업에서 집중하고 있는, 로컬 환경의 RAG 노하우까지 담았습니다.
- 이를 위해 우리는 [8단계로 나누어진 RAG 프로세스를 통해 각 단계별 세부 요소]를 깊이 있게 이해해야 합니다.
다양한 프로젝트 실습을 통해 세부요소에 대한 이해는 물론, 각 상황별 최적의 조합 방법을 함께 찾아봅시다.

강사

이경록

과목

- 테디노트의 RAG 비법노트 : 랭체인을 활용한 GPT부터 로컬 모델까지의 RAG 가이드

약력

- 현) 브레인크루 대표
 - 전) 삼성전자 C-lab 아날로그 플러스 CTO
 - 전) 삼성전자 Software Developer
- 기타이력]
- '한권으로 끝내는 랭체인 노트' 외 다수의 베스트셀러 저자
 - 구독자 2만명의 유튜브 채널 '테디노트' 운영
 - 월간 6만명이 방문하는 블로그 '테디노트' 운영
 - 서울대학교 & 대기업 등 다수 강연

CURRICULUM

01.

LangChain 입문

파트별 수강시간 04:28:12

CH01. LangChain 시작하기

- 01. 강의를 보는 방법
- 02. RAG(Retrieval-Augmented Generation) 기술이 주목받는 이유
- 03. 위키독스 전자책 활용방법
- 04. 실습코드(GitHub)

CH02. 환경 설정

- 01. 환경설치(Windows)
- 02. 환경설치(MacOS)
- 03. OpenAI API 키 발급 및 설정
- 04. LangSmith 키 발급 및 설정
- 05. Visual Studio Code - User Settings(JSON) 설정

CH03. LangChain 시작하기

- 01. 랭체인(LangChain) 개요
- 02. Jupyter Notebook 사용법(단축키, 유용한팁)
- 03. LangChain 시작 (OpenAI 모델 알아보기)
- 04. 토큰(Token), 토큰계산기, 모델별 토큰 비용
- 05. 모델의 입출력, Context Window 개념 이해

CH04. LangChain 문법 맛보기

- 01. ChatOpenAI 주요 파라미터, invoke(), stream() 스트리밍 출력
- 02. LangSmith 로 GPT 추론내용 추적
- 03. GPT-4o (멀티모달) 모델로 이미지 인식하여 답변 출력
- 04. 프롬프트 템플릿(PromptTemplate)
- 05. LCEL 로 Chain 생성
- 06. 출력파서(StrOutputParser) 를 체인에 연결
- 07. Chain 추론과정을 LangSmith 에서 확인
- 08. LCEL 인터페이스 - 일괄처리 batch()
- 09. 비동기(asynchronous) 호출 방법
- 10. 병렬체인 구성(RunnableParallel)
- 11. 값을 전달해주는 RunnablePassthrough
- 12. 병렬로 Runnable 을 실행하는 RunnableParallel
- 13. 함수를 실행하는 RunnableLambda, itemgetter

CURRICULUM

02.

체인
파이프라인의
기본 요소

파트별 수강시간 09:26:50

CH01. 프롬프트

01. PromptTemplate, 부분변수(partial_variables)
02. yaml 파일로부터 프롬프트 템플릿 로드(load_prompt)
03. ChatPromptTemplate
04. MessagesPlaceHolder
05. 퓨샷프롬프트(FewShotPromptTemplate)
06. 예제 선택기(Example Selector)
07. MaxMarginalRelevance(MMR) 알고리즘
08. FewShotChatMessagePromptTemplate
09. 목적에 맞는 예제 선택기(CustomExampleSelector)
10. 프롬프트 허브(LangChain Hub)

CH02. 미니프로젝트

01. [프로젝트] Streamlit 으로 나만의 ChatGPT 웹앱 제작 (1)
02. [프로젝트] Streamlit 으로 나만의 ChatGPT 웹앱 제작 (2)
03. [프로젝트] Streamlit 으로 나만의 ChatGPT 웹앱 제작 (3)

CH03. RAG 시작하기

01. RAG 프로세스 이해하기 (사전단계)
02. RAG 프로세스 이해하기 (실행단계)
03. PDF 문서 기반 QA RAG
04. [프로젝트] 프롬프트를 개선해주는 프롬프트 메이커
05. [프로젝트] Page 분할, 파일 업로더 기능 추가
06. [프로젝트] PDF 기반 QA 챗봇
07. [프로젝트] LangSmith 추적, 다양한 LLM 을 RAG 에 적용
08. [프로젝트] 더 나은 답변을 위한 프롬프트 조정(출처, 표)

CH04. 출력 파서(Output Parser)

01. 출력파서(Output Parser)
02. PydanticOutputParser
03. with_structured_output() 바인딩
04. LangSmith 에서 OutputParser 흐름 확인
05. 콤마로 구분된 리스트 출력파서(CommaSeparatedListOutputParser)
06. 구조화된 출력파서(StructuredOutputParser)
07. JSON 형식 출력파서(JsonOutputParser)
08. Pandas DataFrame 출력파서(PandasDataFrameOutputParser)
09. 날짜형식 출력파서(DatetimeOutputParser)
10. 열거형 출력파서(EnumOutputParser)

CURRICULUM

02.

체인
파이프라인의
기본 요소

파트별 수강시간 09:26:50

CH05. 출력 파서 활용 프로젝트

01. 이메일 내용으로부터 구조화된 정보 추출 - (1)
02. SERP API 를 활용한 정보 검색의 활용 - (2)
03. 구조화된 답변을 다음 체인의 입력으로 추가하기 - (3)
04. 수신된 이메일의 주요 정보 및 검색 정보 기반 요약 보고서 챗봇 - (4)

CH06. 모델(Model)

01. RAG 에서의 LLM(Large Language Model)
02. 다양한 LLM과 활용방법
03. LLM의 답변을 캐싱(Cache)
04. 모델의 저장 및 로드(직렬화, 역직렬화)
05. 토큰 사용량 확인
06. Google Generative AI (Gemini Pro, Gemini Flash)
07. HuggingFace Inference API 활용
08. HuggingFace 의 템플릿 이해
09. HuggingFace Dedicated Endpoint 를 활용한 로컬모델 원격 호스팅
10. HuggingFace Local모델 다운로드 받아 추론
11. Ollama 설치 및 Modelfile 설정
12. Ollama 모델 생성, pull, run, ChatOllama 의 활용
13. GPT4ALL 로 로컬 모델 실행

CH07. 모델 활용 프로젝트

01. [프로젝트] 별도의 py 파일로 기능을 분리하는 방법
02. [프로젝트] Xionic 무료 모델을 GPT 대신 사용
03. [프로젝트] Ollama 모델을 사용한 RAG
04. [프로젝트] 멀티모달 모델을 활용한 이미지 인식 기반 챗봇

CURRICULUM

03.

체인의 기능 확장하기

파트별 수강시간 01:18:46

CH01. 메모리(Memory)

- | |
|------------------------------------|
| 01. 메모리란? |
| 02. ConversationBufferMemory |
| 03. ConversationBufferWindowMemory |
| 04. ConversationTokenBufferMemory |
| 05. ConversationEntityMemory |
| 06. ConversationKGMemory |
| 07. ConversationSummaryMemory |
| 08. VectorStoreRetrieverMemory |
| 09. LCEL Chain 에 메모리 추가 |
| 10. SQLite 에 대화내용 (사용자 및 대화별) 저장 |
| 11. 일반 변수에 대화내용 저장(휘발성 메모리) |
| 12. [프로젝트] 이전 대화내용을 기억하는 멀티턴 챗봇 |

CURRICULUM

04.

다양한 형태의 데이터 로드

파트별 수강시간 01:34:34

CH01. 도큐먼트 로더(Document Loader)

- | |
|--|
| 01. 도큐먼트 로더의 종류, 기본 구조, Document 구조 |
| 02. Document, Document Loader 의 구조 이해하기 |
| 03. PDF 로더 |
| 04. HWP 로더 |
| 05. CSV 로더 (효과적인 CSV 형식의 파일 처리 방법) |
| 06. WebBase 로더(웹 정보 크롤링) |
| 07. Directory 로더 |
| 08. Upstage LayoutAnalysis 로더 |
| 09. LlamaParser 로더(멀티모달을 활용한 이미지, 표 파싱 기능) |

CURRICULUM

05.

Retrieval Augmented Generation: RAG

파트별 수강시간 07:00:25

CH01. 텍스트 분할(Text Splitter)

01. 텍스트 분할의 개념과 중요성. 다양한 전략의 활용
02. CharacterTextSplitter
03. RecursiveCharacterTextSplitter
04. TokenTextSplitter
05. SemanticChunker
06. Code Splitter
07. MarkdownHeaderTextSplitter
08. HTMLHeaderTextSplitter
09. RecursiveJSONSplitter

CH02. 임베딩(Embedding)

01. 임베딩(Embeddings) 개요
02. OpenAIEmbeddings
03. 캐시 임베딩(CacheBackedEmbeddings)
04. 로컬 모델 임베딩(HuggingFaceEmbeddings)
05. 업스테이지 임베딩(UpstageEmbeddings)
06. 올라마 임베딩(OllamaEmbeddings)

CH03. 벡터저장소(VectorStore)

01. 벡터스토어(VectorStore) 개요
02. Chroma
03. FAISS
04. Pinecone

CURRICULUM

05.

Retrieval Augmented Generation: RAG

파트별 수강시간 07:00:25

CH04. 검색기(Retriever)

01. 검색기(Retriever) 개요
02. 벡터스토어 기반 검색기(VectorStore-backed Retrieve)
03. 문서 압축기(ContextualCompressionRetriever)
04. 앙상블 검색기(Ensemble Retriever)
05. 긴 문서 재정렬(LongContext Reorder)
06. 부모 문서 검색기(ParentDocument Retriever)
07. 다중 쿼리 생성 검색기(MultiQuery Retriever)
08. 다중 벡터 검색기(MultiVector Retriever)
09. Self-Query Retriever
10. TimeWeightedVectorStore Retriever

CH05. 리랭커(Reranker)

01. 리랭커(Reranker) 개요
02. CrossEncoderReranker(BGE-m3-reranker)
03. CohereReranker
04. JinaReranker
05. FlashRankReranker

CH06. RAG 파이프라인

01. [프로젝트] RAG 를 활용한 뉴스기사 기반 QA 시스템
02. RAPTOR 논문 리뷰와 핵심 개념 소개
03. RAPTOR 코드구현
05. FlashRankReranker

CURRICULUM

06.

바로 사용할 수 있는 체인

파트별 수강시간 01:17:48

CH01. 사전에 정의된 체인(Chain)

01. Stuff 요약

02. Map Reduce 요약

03. Map Refine 요약

04. Chain of Density(COD) 요약

05. Clustering Map Refine 요약

06. SQL 쿼리 생성기(create_sql_query_chain)
--

CURRICULUM

07.

LCEL 고급 문법

파트별 수강시간 01:14:51

CH01. LCEL 고급 문법

01. RunnablePassthrough

02. Runnable 구조확인

03. RunnableLambda

04. 사용자의 질문 의도에 따라 Routing

05. RunnableParallel, itemgetter

06. 동적으로 LLM 이나 Prompt 를 변경하는 방법(Config)
--

07. 파이썬 함수에 chain 데코레이터로 runnable 설정

CURRICULUM

08.

RAG 평가 &
개선

파트별 수강시간 03:20:51

CH01. RAG 평가

01. LLM 과 RAG 평가 방식, 평가 지표등 소개
02. RAGAS 소개
03. 합성 테스트 데이터셋(Synthetic Test Dataset) 생성
04. RAGAS 를 활용한 평가(Context Precision, Recall, Relevancy, Faithfulness)
05. 테스트 데이터셋 번역업로드 관리

CH02. LangSmith API 를 활용한 프롬프트 최적화

01. LangSmith API 를 활용한 프롬프트 최적화
02. LLM-as-judge 사용방법(QA 평가자, Context-Answer 평가자)
03. LLM-as-judge 사용방법(Criteria 평가자)
04. LLM-as-judge 사용방법(Embedding 기반 평가자)
05. Custom LLM-as-judge 로 평가하는 방법
06. 한글 형태소 분석기
07. 휴리스틱 평가(Rouge, BLEU, METEOR, SemScore)
08. 실험별 비교 분석
09. 전체 수준을 평가하는 Summary 평가자
10. 할루시네이션(Groundedness) 평가
11. 실험 비교 분석을 위한 Pairwise 평가
12. 반복 평가
13. 온라인 LLM 평가를 활용한 평가 자동화

CURRICULUM

09.

Agent

파트별 수강시간 02:52:10

CH01. Tools / Toolkits

01. 랭체인에서 제공하는 다양한 도구(Tools)툴킷(Toolkits) 활용법
02. 사용자 정의 도구(Custom Tools) 만드는법

CH02. Agent 주요기능

01. LLM 에 도구 바인딩(Binding Tools)
02. Agent, AgentExecutor 생성 방법
03. Agent 의 중간단계 스트리밍(stream), AgentStreamParser
04. Agent 에 메모리 추가(멀티턴 구현)
05. 다양한 LLM 을 활용한 에이전트 생성(GPT, Claude, Gemini, TogetherAI, Ollama)
06. iter() 함수로 단계별 출력과 Human-in-the-loop

CH03. Agent 활용

01. Agentic RAG
02. CSV, EXCEL 파일을 분석하는 데이터분석 Agent
03. (업무자동화) FileManagementToolkits 를 활용한 파일 관리 Agent
04. (업무자동화) 보고서 작성 Agent (web-search, retriever, file, image-generation)
05. [프로젝트] CSV 파일 기반 데이터분석 Agent

CURRICULUM

10.

LangGraph

파트별 수강시간 10:36:53

CH01. LangGraph 개요

01. LangGraph 개요

02. LangGraph 세부 기능(State, Node, Edge, Conditional Edge, Compile)

CH02. LangGraph 핵심 기능

01. LangGraph 에서 자주 사용되는 파이썬 문법(TypedDict, Annotated, Reducer)

02. LangGraph 챗봇 만들기

03. LangSmith 에서 LangGraph 추적 확인

04. Function Calling LLM 과 도구호출 노드, Conditional Edge 구축방법

05. LangGraph 에 메모리 추가(Checkpointer)

06. LangGraph 노드의 단계별 출력하는 Stream 모드(values, updates) 와 Interrupt

07. Human-in-the-loop 과 이전 상태에서 되돌림하는 Replay 기능

08. 중간 단계의 노드에서 상태를 임의 수정 후 재개(검색쿼리 수정)

09. 중간 단계의 노드에서 상태를 임의 수정 후 재개(검색쿼리 수정)

10. 전체 실행이 완료된 후 특정 노드로 되돌림하여 수정 후 Replay

11. 사람의 의견을 묻는 노드 추가

12. RemoveMessages 로 메시지 기록 삭제

13. ToolNode 활용법(2개 이상의 도구 사용)

14. 병렬 노드의 처리(fan-out, fan-in) 및 우선순위 설정

15. 대화 기록의 요약용 노드로 추가

16. 그래프에 서브 그래프(SubGraph) 를 추가하는 방법

17. 2개 이상의 SubGraph 추가 및 상태 변환(State Transformation)

18. LangGraph 출력 스트리밍(토큰 단위 스트리밍, tag 필터링)

CURRICULUM

10.

LangGraph

파트별 수강시간 10:36:53

CH03. LangGraph 구조 설계

01. LangGraph 로 자유롭게 그래프 로직 구성(흐름 엔지니어링)
02. Naive RAG(전통적인 방식의 RAG) 를 LangGraph 로 구현
03. 답변의 할루시네이션관련성 평가 모듈 추가
04. 웹 검색 노드 추가
05. 쿼리 재작성 모듈 추가
06. [프로젝트] Agentic RAG - 에이전트를 활용한 RAG
07. [프로젝트] Adaptive RAG

CH04. LangGraph 유스케이스 연구

01. [프로젝트] 에이전트간 대화 시뮬레이션(고객 응대 시나리오)
02. [프로젝트] 사용자 요구사항 기반 메타 프롬프트 생성 에이전트
03. [프로젝트] Corrective RAG (CRAG)
04. [프로젝트] Self-RAG
05. [프로젝트] Plan and Execute - 전문적인 보고서 작성
06. [프로젝트] 멀티에이전트 협업 - 웹 검색 및 차트 생성
07. [프로젝트] 멀티에이전트 감독자(Supervisor) - 웹 검색 및 차트 생성
08. [프로젝트] 계층적인 구조를 가지는 멀티에이전트 팀 협업 - 리서치팀과 문서 작성팀 협업
09. [프로젝트] SQL 에이전트 - (Text2SQL) SQL 쿼리 검색 기반 RAG
10. [프로젝트] Research 특화 멀티에이전트 STORM - 다인격 멀티에이전트
11. [프로젝트] GitHub 소스코드 기반 QA 챗봇 구현 - (1)
12. [프로젝트] GitHub 소스코드 기반 QA 챗봇 구현 - (2)
13. [프로젝트] Perplexity 클론 앱 구현

CURRICULUM

11.

서비스 배포

파트별 수강시간 00:54:35

CH01. Streamlit

- 01. 배포를 위한 config 설정
- 02. 배포를 위한 준비(Github 과 기타 설정 파일)
- 03. 배포 및 동작 테스트
- 04. 수정사항이 생겼을 때 반영하는 방법과 변경사항 확인

CH02. LangServe

- 01. LangChain 에서 만든 LangServe 란-
- 02. LangServe 로 모델을 서빙하는 방법
- 03. 서빙한 모델을 외부에서 사용하는 방법

CURRICULUM

부록.

파이썬 필수 문법
& 백엔드의 이해

파트별 수강시간 09:55:15

CH01. 컴퓨터 과학과 프로그래밍 소개

01. 강의 소개
02. 컴퓨터 과학과 프로그래밍
03. 파이썬 소개와 설치

CH02. 변수와 데이터타입

01. 변수의 개념과 활용
02. 데이터타입 (1) 정수형, 실수형, Boolean
03. 정수형, 실수형, Boolean 타입 실습하기
04. 데이터타입 (2) 문자열
05. 문자열 타입 실습하기

CH03. 자료구조

01. 자료구조란?
02. 대표적인 자료구조
03. 파이썬의 자료구조: 리스트
04. 파이썬의 자료구조: 리스트 실습
05. 파이썬의 자료구조: 튜플
06. 파이썬의 자료구조: 튜플 실습
07. 파이썬의 자료구조: 딕셔너리
08. 파이썬의 자료구조: 딕셔너리 실습
09. 파이썬의 자료구조: 세트
10. 파이썬의 자료구조: 세트 실습
11. (심화) 자료구조 활용 실습하기

CH04. 흐름제어 (조건문, 반복문, 예외처리)

01. 흐름제어란?
02. 조건문
03. 조건문 실습하기
04. 반복문
05. 반복문 실습하기
06. 스토리에 맞는 프로그램 작성하기
07. 예외처리
08. 예외처리 실습하기

CURRICULUM

부록.

파이썬 필수 문법
& 백엔드의 이해

파트별 수강시간 09:55:15

CH05. 함수와 모듈

01. 함수의 개념과 활용
02. 함수의 개념과 활용 실습
03. 함수 매개변수와 반환
04. 다양한 매개변수 실습
05. 모듈과 패키지
06. 모듈과 패키지 실습
07. 파이썬의 라이브러리
08. 파이썬의 라이브러리 실습

CH06. 객체 지향 프로그래밍

01. 객체지향 프로그래밍
02. 객체와 클래스
03. 멤버변수와 메소드
04. 멤버변수와 메소드 실습
05. 객체지향 프로그래밍 주요 요소
06. 객체지향 프로그래밍 주요 요소 실습
07. 객체지향 프로그래밍 심화 실습 소개
08. 객체지향 프로그래밍 심화 실습

CH07. 알고리즘

01. 알고리즘 소개
02. 알고리즘의 복잡도
03. 정렬 알고리즘
04. 정렬 알고리즘 실습
05. 다양한 알고리즘

CH08. 정규표현식

01. 정규 표현식이란?
02. 파이썬의 정규표현식
03. 정규 표현식 활용하기

CURRICULUM

순차공개. 라이브 강의 파트

파트별 수강시간 12:29:52

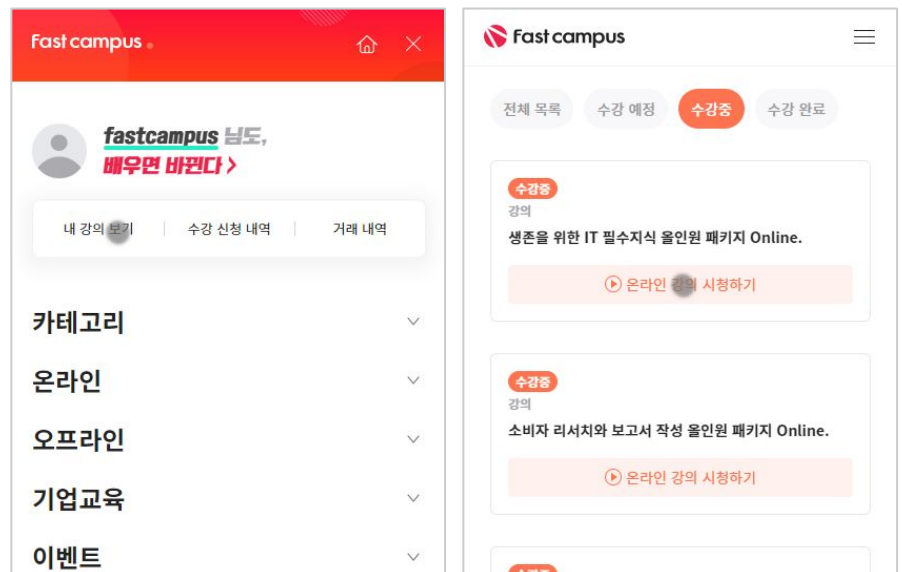
CH01. 라이브 강의 파트 순차 공개
01. 테디노트_6월 라이브
02. 테디노트_7월 라이브
03. 테디노트_8월 라이브
04. 테디노트_9월 라이브
05. 테디노트_10월 라이브
06. 테디노트_11월 라이브

주의 사항

- 상황에 따라 사전 공지 없이 할인이 조기 마감되거나 연장될 수 있습니다.
- 패스트캠퍼스의 모든 온라인 강의는 **아이디 공유를 금지**하고 있으며 1개의 아이디로 여러 명이 수강하실 수 없습니다.
- 별도의 주의사항은 각 강의 상세페이지에서 확인하실 수 있습니다.

수강 방법

- 패스트캠퍼스는 크롬 브라우저에 최적화 되어있습니다.
- 사전 예약 판매 중인 강의의 경우 1차 공개일정에 맞춰 '온라인 강의 시청하기'가 활성화됩니다.



환불 규정

- 온라인 강의는 각 과정 별 '정상 수강기간(유료수강기간)'과 정상 수강기간 이후의 '복습 수강기간(무료수강기간)'으로 구성됩니다.
- 환불금액은 실제 결제금액을 기준으로 계산됩니다.

수강 시작 후 7일 이내	100% 환불 가능 (단, 수강하셨다면 수강 분량만큼 차감)
수강 시작 후 7일 경과	정상(유료) 수강기간 대비 잔여일에 대해 환불규정에 따라 환불 가능

※ 강의별 환불규정이 상이할 수 있으므로 각 강의 상세페이지를 확인해 주세요.