

교육 과정 소개서.

테디노트의 RAG 비법노트 : 랭체인을 활용한 GPT부터 로컬 모델까지의 RAG 가이드

안내.

해당 교육 과정 소개서는 모든 강의 영상이 촬영하기 전 작성되었습니다.

* 커리큘럼은 촬영 및 편집을 거치며 일부 변경될 수 있으나, 전반적인 강의 내용에는 변동이 없습니다.
아래 각 오픈 일정에 따라 공개됩니다.

- 1차 : 2024년 07월 01일
- 2차 : 2024년 07월 29일
- 3차 : 2024년 08월 30일
- 전체공개 : 2024년 09월 23일

최근 수정일자 2024년 05월 27일



강의정보

강의장	온라인 강의 데스크탑, 노트북, 모바일 등
수강 기간	평생 소장
상세페이지	https://fastcampus.co.kr/data_online_teddy#curriculum
강의시간	40시간 예정 (* 사전 판매 중인 강의는 시간이 변경될 수 있습니다.)
문의	고객센터

강의특징

나만의 속도로	낮이나 새벽이나 내가 원하는 시간대에 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

CH01. LangChain 시작하기
01. 강의를 보는 방법
02. LangChain 개요
03. 실습코드(GitHub) & 교재 활용 방법
CH02. 환경 설정
01. Anaconda 가상환경 설치(Windows)
02. Anaconda 가상환경 설치(Mac)
03. Visual Studio Code 설치(Windows)
04. Visual Studio Code 설치(Mac)
05. Visual Studio Code Extension 설치
06. Poetry 란?
07. Poetry 설치 및 활용 방법
CH03. API Key 발급 및 관리
01. OpenAI API 키
02. OpenAI API 키 발급방법
03. .env 로 키 관리하기
04. .env 를 Jupyter Notebook 파일에서 키 불러오기
CH04. LangChain 문법 맛보기
01. LangChain Expression Language(LCEL)
02. LCEL 로 Chain 생성하기
03. invoke(), batch(), async 기능
04. RunState, Parallel 기능
05. Runnable
06. RunnablePassthrough, RunnableParallel, RunnableLambda
07. [프로젝트] Runnable / LCEL 문법과 친해지기(연습문제)
08. [프로젝트] LCEL 문법을 활용한 GPT 챗봇 생성

본 과정은 현재 촬영 및 편집이 진행되고 있는 **사전 판매 중인 강의**입니다.
해당 교육과정 소개서는 변경되거나 추가될 수 있습니다.

CURRICULUM

02.

LangSmith 추적

CH01. LangSmith
01. LangSmith 란?
02. LangSmith API 발급 방법
03. LangSmith 를 활용한 추적 활성화

본 과정은 현재 촬영 및 편집이 진행되고 있는 **사전 판매 중인 강의**입니다.
해당 교육과정 소개서는 변경되거나 추가될 수 있습니다.



CURRICULUM

03.

체인
파이프라인의
기본 요소

CH01. 프롬프트
01. 프롬프트란?
02. PromptTemplate, 변수 사용법
03. 파일로부터 Template 생성(.yaml) 및 관리 방법
04. partial variables로 템플릿 변수 일부 채우기
05. RunnablePassthrough 와 PromptTemplate 결합
06. FewShotPromptTemplate
07. [프로젝트] 퓨샷프롬프트 예시에 따라 다른 형식으로 답변하는 GPT
08. LangChain Hub 로부터 Prompt 받아오기(pull)
09. LangChain Hub 의 개인 Prompt 커밋(commit & push)
10. [프로젝트] 기존 프롬프트를 수정하여 나만의 프롬프트 Hub 에 커밋
CH02. 출력 파서(Output Parser)
01. 출력 파서(Output Parser) 란?
02. Pydantic 출력 파서
03. [프로젝트] Pydantic 형식에 준수한 답변 생성하기(양식에 맞춤 답변)
04. CommaSeparatedList 출력 파서
05. 구조화된 출력 파서
06. JSON 출력 파서
07. 데이터프레임 출력 파서
08. [프로젝트] LLM 답변을 CSV / Excel 양식으로 저장하기
09. 날짜 형식 출력 파서
10. 열거형 출력 파서
11. 출력 수정 파서

본 과정은 현재 촬영 및 편집이 진행되고 있는 **사전 판매 중인 강의**입니다.
해당 교육과정 소개서는 변경되거나 추가될 수 있습니다.

CURRICULUM

03.

체인
파이프라인의
기본 요소

CH03. 모델(Model)

01. OpenAI 의 GPT 모델
02. 모델 캐싱(Cache)
03. 토큰의 사용량 확인(Callback)
04. 모델의 저장 및 로드(직렬화, 역직렬화)
05. 구글의 생성 AI
06. HuggingFace Endpoint API
07. HuggingFace Local
08. HuggingFace Pipeline
09. Ollama
10. Ollama 설치
11. Ollama 모델 다운로드/템플릿 정의
12. Ollama 모델 생성
13. ChatOllama 로 로컬 모델 실행
14. [프로젝트] ChatOllama 로 허깅페이스 모델(EVEE Korean Instruct 10.8B)을 다운로드 받아 추론하는 Chain 생성
15. [프로젝트] 무료 오픈 모델을 활용한 문서 요약 체인
16. LM Studio란?
17. LM Studio 로 HuggingFace 모델(llama3 8B Instruct) 다운로드 및 Inference Server
18. LM Studio 에서 올바른 템플릿 설정
19. LM Studio 로 호스팅된 로컬 모델 활용
20. [프로젝트] LM Studio 로 허깅페이스 모델 다운로드 받아 추론하는 Chain 생성
21. [프로젝트] LM Studio 의 추론서버를 활용한 영한-한영 번역 체인

본 과정은 현재 촬영 및 편집이 진행되고 있는 **사전 판매 중인 강의**입니다.
해당 교육과정 소개서는 변경되거나 추가될 수 있습니다.

CURRICULUM

04.

체인의 기능 확장하기

CH01. 메모리(Memory)

- | |
|-------------------------------|
| 01. 메모리란? |
| 02. 대화 버퍼 메모리 |
| 03. 대화 버퍼 윈도우 메모리 |
| 04. 대화 토큰 버퍼 메모리 |
| 05. 대화 엔티티 메모리 |
| 06. 대화 지식그래프 메모리 |
| 07. 대화 요약 메모리 |
| 08. 벡터저장소 검색 메모리 |
| 09. LCEL Chain 에 메모리 추가 |
| 10. [프로젝트] 이전의 대화내용을 기억하는 GPT |
| 11. SQLite 에 메모리 저장 |

본 과정은 현재 촬영 및 편집이 진행되고 있는 **사전 판매 중인 강의**입니다.
해당 교육과정 소개서는 변경되거나 추가될 수 있습니다.

CURRICULUM

05. 바로 사용할 수 있는 체인

CH01. 사전에 정의된 체인(Chain)

- | |
|---|
| 01. 대화형 체인(Conversation Chain) |
| 02. 구조화된 출력 체인(Structured Output Chain) |
| 03. 문서 전체 요약 체인(Stuff Documentation Chain) |
| 04. 문서 분할-병합 요약(Map-reduce Chain, Refine Chain) |
| 05. 웹 문서 요약(Webpage Summerization Chain) |
| 06. 정형데이터 QA 체인 |
| 07. 프로젝트: CSV 데이터 기반 QA 챗봇 생성 |

CURRICULUM

06. 다양한 형태의 데이터 로드

CH01. 도큐먼트 로더(Document Loader)

- | |
|------------------|
| 01. 도큐먼트 로더란? |
| 02. 논문(arxiv) 로더 |
| 03. 웹 크롤링 로더 |
| 04. 데이터 프레임 로더 |
| 05. PDF 로더 |

본 과정은 현재 촬영 및 편집이 진행되고 있는 **사전 판매 중인 강의**입니다.
해당 교육과정 소개서는 변경되거나 추가될 수 있습니다.

CURRICULUM

07.

Retrieval
Augmented
Generation:
RAG

CH01. 텍스트 분할(Text Splitter)

01. 텍스트 분할이란? 텍스트 분할의 필요성
02. CharacterTextSplitter
03. RecursiveCharacterTextSplitter
04. TokenTextSplitter
05. SemanticChunker
06. Code Splitter
07. Markdown Splitter
08. HTML Splitter
09. RecursiveJSONSplitter

CH02. 임베딩(Embedding)

01. OpenAIEmbeddings
02. 캐시 임베딩(CacheBackedEmbeddings)
03. 로컬 임베딩(HuggingFace Embeddings)

CH03. 벡터저장소(VectorStore)

01. 벡터저장소란? 벡터저장소의 다양한 종류
02. FAISS
03. Chroma
04. FAISS Async

본 과정은 현재 촬영 및 편집이 진행되고 있는 **사전 판매 중인 강의**입니다.
해당 교육과정 소개서는 변경되거나 추가될 수 있습니다.

CURRICULUM

07.

Retrieval Augmented Generation: RAG

CH04. 검색기(Retriever)
01. VectorStore-backed Retriever
02. ContextualCompressionRetriever
03. Ensemble Retriever
04. LongContext Reorder
05. ParentDocument Retriever
06. MultiQuery Retriever
07. Self-Query Retriever
08. TimeWeightedVectorStore Retriever
CH05. RAG 파이프라인
01. RAG 란?
02. RAG 파이프라인 이해하기
03. [프로젝트] RAG 를 활용한 뉴스기사 기반 QA 시스템
04. RAPTOR 논문
05. RAPTOR 코드구현
06. [프로젝트] RAG 를 활용한 PDF 기반 질의-응답 QA 시스템
07. [프로젝트] 이전에 구현한 RAG 파이프라인을 개선할 수 있는 아이디어
08. [프로젝트] 온전한 무료 오픈모델로 문서기반 QA RAG 시스템 구현
09. [프로젝트] GitHub 소스코드 기반 QA 디스코드 봇 구현

본 과정은 현재 촬영 및 편집이 진행되고 있는 **사전 판매 중인 강의**입니다.
해당 교육과정 소개서는 변경되거나 추가될 수 있습니다.



CURRICULUM

08. LCEL 고급 문법

CH01. LCEL 고급 문법
01. RunnablePassthrough
02. Runnable 구조확인
03. RunnableLambda
04. RunnableBranch
05. RunnableParallel
06. configurable_fields, configurable_alternatives
07. chain 데코레이터로 Runnable 정의
08. Memory 가 추가된 Chain(RunnableWithMessageHistory)

CURRICULUM

09. RAG 평가 & 개선

CH01. RAG 평가
01. RAG 의 결과를 평가하기 위한 다양한 시도
02. Retrieval 정보 평가를 위한 ragas
03. LangSmith 를 활용한 RAG 평가 방법
04. [프로젝트] RAGAS 를 활용한 RAG 평가
CH02. LangSmith API 를 활용한 프롬프트 최적화
01. LangSmith Feedback, Annotation, Dataset
02. [프로젝트] 내가 작성한 프롬프트 자동 최적화(Self Optimize RAG) 파이프라인

본 과정은 현재 촬영 및 편집이 진행되고 있는 **사전 판매 중인 강의**입니다.
해당 교육과정 소개서는 변경되거나 추가될 수 있습니다.

CURRICULUM

10.

Agent

CH01. Agent
01. Agent란?
02. Agent를 위한 도구
03. 검색 도구(Tavily, SerpAPI)
04. 문서 검색 도구(Retrieval Tool)
05. 사용자 정의 도구
06. GPT 를 활용한 Agent
07. JSON 기반 Agent
08. 무료 모델 기반 Agent
09. 다중 협업 에이전트
10. [프로젝트] 다중 에이전트를 활용한 토론 시뮬레이션 구현

본 과정은 현재 촬영 및 편집이 진행되고 있는 **사전 판매 중인 강의**입니다.
해당 교육과정 소개서는 변경되거나 추가될 수 있습니다.

CURRICULUM

11.

LLM을 활용한
웹앱 제작

CH01. Streamlit
01. 스트림릿(Streamlit) 이란?
02. Streamlit 설치 및 실행
03. 텍스트 위젯(Header, SubHeader, TextInput, TextArea, Markdown, Divider)
04. 입력 위젯(Button, SelectBox, CheckBox, NumberInput, Slider)
05. 레이아웃(Container, Column, SideBar, Expander, Tabs)
06. 멀티 페이지
07. 시크릿 키 관리
08. SessionState
09. config.toml, set_page_config
10. chat_input, chat_message
11. st.empty
12. 리소스 캐싱
13. [프로젝트] OpenAI GPT 를 활용한 ChatGPT 스타일 웹앱 구현
14. [프로젝트] 메모리기능 추가
15. [프로젝트] User 별 대화 쓰레드 관리 기능 / DB 에 저장
16. [프로젝트] PDF 기반 질의-응답 QA 웹앱
17. Streamlit 웹앱 배포

본 과정은 현재 촬영 및 편집이 진행되고 있는 **사전 판매 중인 강의**입니다.
해당 교육과정 소개서는 변경되거나 추가될 수 있습니다.

CURRICULUM

12. 모델 서빙

CH01. LangServe

01. LangChain 에서 만든 LangServe 란?
02. LangServe 로 모델을 서빙하는 방법
03. LangServe Template 으로 기존 서빙 템플릿 활용하기
04. 로컬 무료 모델을 LangServe 로 서빙하는 방법
05. 서빙한 모델을 외부에서 사용하는 방법

CURRICULUM

13. 다중협업에이전 트 - LangGraph

CH01. LangGraph

01. LangGraph 란?
02. Tool(도구) 설정
03. 모델의 설정
04. 에이전트 상태(state)
05. 노드의 정의
06. 그래프 정의
07. Self RAG
08. [프로젝트] LangGraph 를 활용한 코드 작성 코파일럿
09. [프로젝트] Adaptive RAG
10. [프로젝트] Corrective RAG
11. [프로젝트] 웹 리서치 RAG

본 과정은 현재 촬영 및 편집이 진행되고 있는 **사전 판매 중인 강의**입니다.
해당 교육과정 소개서는 변경되거나 추가될 수 있습니다.

CURRICULUM

부록.

파이썬 필수 문법
& 백엔드의 이해

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

본 과정은 현재 촬영 및 편집이 진행되고 있는 **사전 판매 중인 강의**입니다.
해당 교육과정 소개서는 변경되거나 추가될 수 있습니다.

CURRICULUM

부록.

파이썬 필수 문법
& 백엔드의 이해

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

본 과정은 현재 촬영 및 편집이 진행되고 있는 **사전 판매 중인 강의**입니다.
해당 교육과정 소개서는 변경되거나 추가될 수 있습니다.



주의 사항

- 상황에 따라 사전 공지 없이 할인이 조기 마감되거나 연장될 수 있습니다.
- 패스트캠퍼스의 모든 온라인 강의는 아이디 공유를 금지하고 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.