## 2024년 10월 주주총회



## Agent - 강의실 업데이트 완료!

### 업데이트 내용

랭체인에서 제공하는 다양한 도구(Tools)/툴킷(Toolkits) 활용법

사용자 정의 도구(Custom Tools) 만드는법

LLM 에 도구 바인딩(Binding Tools)

Agent, AgentExecutor 생성 방법

Agent 의 중간단계 스트리밍(stream), AgentStreamParser

Agent 에 메모리 추가(멀티턴 구현)

다양한 LLM 을 활용한 에이전트 생성(GPT, Claude, Gemini, TogetherAl, Ollama)

iter() 함수로 단계별 출력과 Human-in-the-loop

Agentic RAG

CSV, EXCEL 파일을 분석하는 데이터분석 Agent

(업무자동화) FileManagementToolkits 를 활용한 파일 관리 Agent

(업무자동화) 보고서 작성 Agent (web-search, retriever, file, image-generation)

[프로젝트] CSV 파일 기반 데이터분석 Agent

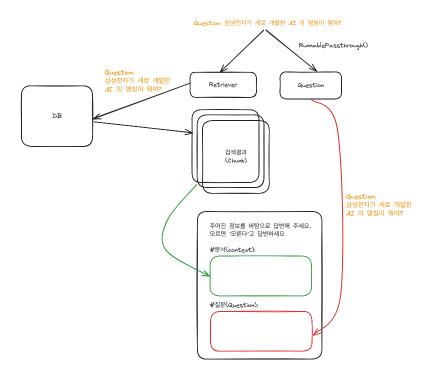
## CSV Agent 데모!

- 완전히 튜닝한 CSV 데이터 분석 Agent 데모 시연
- Column 가이드라인 기능 추가!

- 한글 폰트 (그래프) 기능 추가!
- 대화 기억 기능 추가!
- Kaggle 의 데이터 분석 경진대회 데이터를 활용한 데이터 분석
  - 링크: <u>https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data</u>

## **Agentic RAG?**

일반적인 RAG 의 파이프라인



## Agent 의 구조

**Agentic-RAG-Flow** 

Final Answer

Tool Calling 처리시 참고 자료

# Tool Message 구조도 by teddynote.

```
Content
                     (이곳에 메시지의 내용이 기입됩니다. 단, tool calls 때는 내용이 "" 빈 문자열 입니다.)
additional_kwargs
              tool_calls
                     index [0]
                       "id" : function 혹은 tool 의 호출 아이디 (eg. call_kO4yy81rfM3OX),
                       "function":
                            "arguments": {"query": "테디노트 랭체인 노트"},
                            "name": "tavily_web_search", 도구(tool)의 이름
                       "type": "function"
                                                        function calling
                                             ... (more tools)
response_metadata
           "token_usage":
               "completion_tokens":
                                         답변 토큰 수
                                        프롬프트 토큰 수
               "prompt_tokens":
               "total_tokens":
                                        전체 토큰 수
               "completion_tokens_details": 답변 토큰의 상세 정보
                   "audio_tokens":
                                        오디오에 사용한 토큰 수,
                   "reasoning_tokens":
               "prompt_tokens_details": 프롬프트 토큰의 상세 정보
                   "audio_tokens": None 오디오에 사용한 토큰 수,
                    "cached_tokens": 0 캐싱된 토큰 수
           "model_name": "gpt-4o-mini"
                                       모델 정보
           "system_fingerprint": system
           "finish_reason": "tool_calls",
                                       툴 호출 후 종료
           "logprobs": None
                                       logprobs 정보
```

## Agentic RAG 프롬프트 작성시 주의사항

- 1. Agentic RAG 에서는 Step-by-step 프로세스를 명시해 주어야 순차적으로 (일관된 프로세스 하에 ) 수행
- 2. 도구의 이름, description 등을 상세히 명시

### prompt 예시

You are a question answering assistant. Your task is to answer the user's question based on the provided context.

You should follow the below guidelines step by step:

- 1. make sure to use the `pdf\_search` tool for searching information from the PDF document.
- 2. Then, use the `relevance\_check` tool for checking the relevance between the user's query and retrieved context.
- 3. If the result of `relevance\_check` is \*\*no\*\*, you should generate another query and use the `pdf\_search` tool again.
- 4. You should include `relevance\_check` tool call in your response.

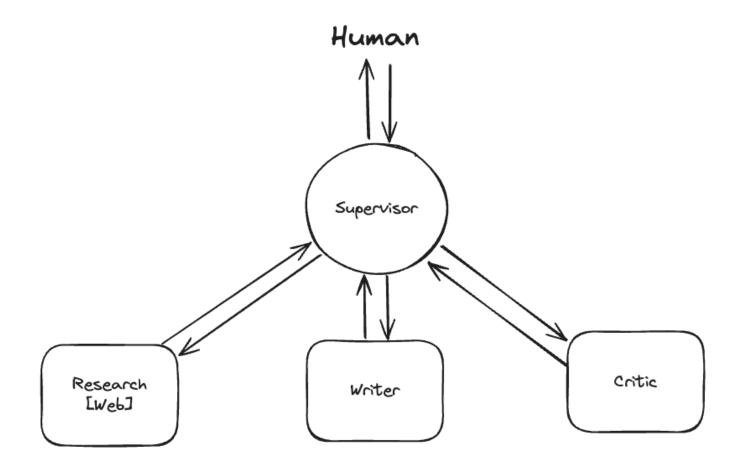
#### Remember:

- If the result of `relevance\_check` is \*\*no\*\*, you should re-generate another query and use the `pdf\_search` tool again.
  - You can do this process up to 20 times.

## Agentic-RAG 응용 데모

- 1. Relevance Checker 를 도구로 추가하여 RAG 에 활용하는 방법
- 2. Context Compressor 를 도구로 추가하여 RAG 에 활용하는 방법
- 3. 이 모든 내용을 통합하여 RAG 에 활용하는 방법

### **SWARM**



- 멀티 에이전트 오케스트레이션
- 관련 영상은 YouTube 에서 확인해 주세요!
  - <u>https://youtu.be/iqXn6Oiis4Q?si=jehikc29cCJ5yxYl</u>