

**Agents**

* 오늘의 강의: [풀스택 GPT: #12.0부터 ~ #12.0까지](https://nomadcoders.co/fullstack-gpt/lectures/4627)
* 오늘의 과제: 위의 강의를 시청하신 후, 아래 코드 챌린지를 제출하면 됩니다.
* 제출기간: 2일 챌린지! 금요일 오전 6시까지

**챌린지:**

**(EN)**

* In a new Jupyter notebook create a research AI agent and give it custom tools.
* The agent should be able to do the following tasks:
  + Search in Wikipedia
  + Search in DuckDuckGo
  + Scrape and extract the text of any website.
  + Save the research to a .txt file
* Run the agent with this query: "Research about the XZ backdoor", the agent should try to search in Wikipedia or DuckDuckGo, if it finds a website in DuckDuckGo it should enter the website and extract it's content, then it should finish by saving the research to a .txt file.

**(KR)**

* 새로운 Jupyter notebook에서 리서치 AI 에이전트를 만들고 커스텀 도구를 부여합니다.
* 에이전트는 다음 작업을 수행할 수 있어야 합니다:
  + Wikipedia에서 검색
  + DuckDuckGo에서 검색
  + 웹사이트의 텍스트를 스크랩하고 추출합니다.
  + 리서치 결과를 .txt 파일에 저장하기
* 다음 쿼리로 에이전트를 실행합니다: "Research about the XZ backdoor" 라는 쿼리로 에이전트를 실행하면, 에이전트는 Wikipedia 또는 DuckDuckGo에서 검색을 시도하고, DuckDuckGo에서 웹사이트를 찾으면 해당 웹사이트에 들어가서 콘텐츠를 추출한 다음 .txt 파일에 조사 내용을 저장하는 것으로 완료해야 합니다.

**제출방법**

* notebook 을 실행한 후 커밋을 생성하고 Github에 푸시합니다.
* 제출 링크는 Github 리포지토리가 아닌! **github 커밋**으로 연결되어야 합니다.
* 제출기간: 2일 챌린지! 금요일 오전 6시까지

**TA's 힌트**

* 다양한 커스텀 도구를 만들고 이 도구들을 에이전트에게 전달하여 리서치 AI 에이전트를 구현해야 하는 챌린지입니다.
* 먼저, 위키피디아에서 검색을 수행하는 도구를 만들어야 합니다. 랭체인에서 제공하는 WikipediaQueryRun를 사용하면 간단하게 구현할 수 있습니다. ([LangChain - Wikipedia](https://python.langchain.com/v0.2/docs/integrations/tools/wikipedia/) 를 참고하세요.)
* 다음은 DuckDuckGo에서 검색을 수행하는 도구를 만들어야 합니다. 이 또한 랭체인에서 제공하는 DuckDuckGoSearchRun를 사용하면 덕덕고의 검색 결과를 얻을 수 있습니다. ([LangChain - DuckDuckGo Search](https://python.langchain.com/v0.1/docs/integrations/tools/ddg/) 를 참고하세요.)
* DuckDuckGo를 통해 얻은 검색 결과에서 웹사이트들의 콘텐츠를 추출하는 도구가 필요합니다. 다양한 방법이 있지만, 그중에서도 랭체인이 제공하는 WebBaseLoader를 이용하면 웹사이트의 콘텐츠를 쉽게 스크랩할 수 있습니다. ([LangChain - WebBaseLoader](https://python.langchain.com/v0.1/docs/integrations/document_loaders/web_base/) 를 참고하세요.)
* 마지막으로, 리서치 결과를 txt 파일에 저장하는 도구가 필요합니다. 파이썬에서 기본으로 제공하는 파일 쓰기 기능을 이용하여 구현하세요. ([python - reading-and-writing-files](https://docs.python.org/ko/3/tutorial/inputoutput.html#reading-and-writing-files))