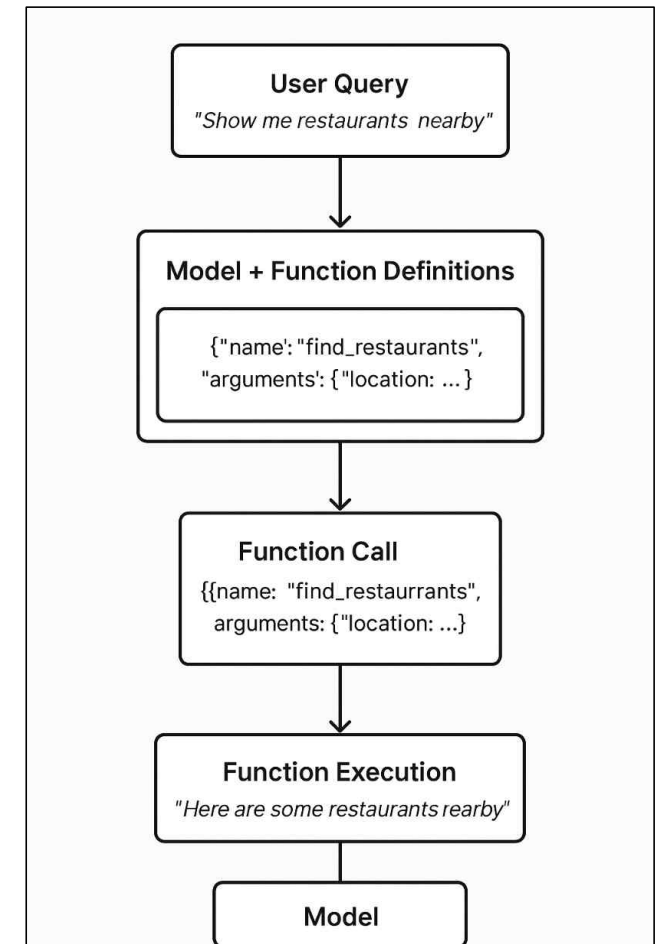


Function Calling

Function Calling 이란?

OpenAI API의 **Function Calling**은 모델이 단순히 텍스트만 생성하는 것이 아니라, 외부 함수(코드, API, DB 쿼리 등)를 호출하도록 지시할 수 있는 기능이다. 쉽게 말해 GPT 모델이 "답변" 대신 "명령"을 내리도록 설계할 수 있는 방식이다.



1. 개념

- Function Calling은 GPT 모델에게 ****정해진 함수 스펙(JSON Schema)****를 알려주고, 모델이 상황에 맞는 함수를 선택해 인자(arguments)를 구성해주도록 하는 기능이다.
- 모델은 사용자의 질문을 보고 "이건 내가 직접 답변하는 게 아니라 함수 호출이 필요해"라고 판단하면 JSON 형태로 함수 이름과 매개변수를 반환한다.
- 개발자는 이 JSON을 받아서 실제 함수를 실행하고, 그 결과를 다시 모델에 전달하여 후속 답변을 생성한다.

2. 동작 구조

1. 함수 정의 등록

함수 이름, 설명, 매개변수(JSON Schema 형식)를 모델에 제공한다.

2. 사용자 입력 처리

사용자가 질문하면 모델은 자연어 → 함수 호출(JSON)로 변환할지 여부를 결정한다.

3. 함수 호출

모델이 반환한 JSON을 실제 코드에서 파싱해 함수 실행 → 결과 생성.

4. 후속 응답 생성

함수 실행 결과를 모델에 다시 전달하면, 모델은 결과를 바탕으로 자연어 응답을 생성한다.

3. 장점

- 안전성: 모델이 직접 코드 실행 대신 JSON만 반환하므로 보안 위험을 줄일 수 있음.
- 정확성: 모델이 직접 문자열 파싱 없이 구조화된 데이터로 결과를 전달.
- 자동화: 챗봇, 데이터 조회, 계산, 외부 API 호출 등을 대화 흐름에 자연스럽게 통합.

Function Calling

4. 예시 코드 (파이썬)

```
python

from openai import OpenAI
client = OpenAI()

# 1. 함수 정의
functions = [
    {
        "name": "get_weather",
        "description": "주어진 도시의 날씨를 가져온다",
        "parameters": {
            "type": "object",
            "properties": {
                "city": {"type": "string", "description": "도시 이름"},
                "unit": {"type": "string", "enum": ["celsius", "fahrenheit"]}
            },
            "required": ["city"]
        }
    }
]
```

Function Calling

2. 사용자 질문 + 함수 스펙 전달

```
response = client.chat.completions.create(  
    model="gpt-4o",  
    messages=[{"role": "user", "content": "서울 날씨 알려줘"}],  
    functions=functions  
)
```

3. 모델의 응답(JSON) 예시

```
print(response.choices[0].message)
```

```
# {"name": "get_weather", "arguments": {"city": "서울", "unit": "celsius"}}
```

Function Calling

4. 실제 `get_weather()` 함수 실행 후 결과 전달

```
weather_result = {"temp": 25, "condition": "맑음"}
followup = client.chat.completions.create(
    model="gpt-4o",
    messages=[
        {"role": "user", "content": "서울 날씨 알려줘"},
        response.choices[0].message,
        {"role": "function", "name": "get_weather", "content": str(weather_result)}
    ]
)
print(followup.choices[0].message["content"])
# "오늘 서울은 맑고 25도입니다."
```


5. 활용 사례

- 실시간 데이터 조회 (날씨, 환율, 주가, DB 검색)
- 업무 자동화 (일정 등록, 이메일 전송)
- 계산/로직 실행 (수식 계산, 시뮬레이션)
- 멀티스텝 에이전트 (LangChain, CrewAI 등과 결합하여 워크플로우 실행)

감사합니다