

Assistants로 Code-Interpreter 동작 과정 엿보기

```
run = client.beta.threads.runs.create(
    thread_id=thread.id, assistant_id=assistant.id)
```

1천만원을 4% 복리로 예치해두면 3년 뒤에 얼마 받는지 풀이 과정을 알려줘!

질의

응답

1천만원을 4% 복리 이율로 3년간 예치할 경우, 3년 후 받게 되는 금액은 총 1124만 8640원 ...

Thread:

ThreadMessage#1: 1천만원을 4% 복리로 예치해두면 3년 뒤에 얼마..

Run:

13:00 00분 00초

13:00 00분 10초

13:00 00분 20초

ThreadMessage#2: run.status=in progress

복리 이자 계산 공식은 다음과 같습니다:

[$A = P \times (1 + r)^t$]

여기서:

- (A)는 미래 가치(원금에 이자가 붙은 총액),
- (P)는 원금,
- (r)은 이자율(1년에 한 번씩 복리가 적용될 때),
- (t)는 시간(년)입니다.

주어진 정보에 따라:

- 원금 (P)는 10,000,000원,
- 이자율 (r)은 4% 또는 0.04(소수로 표현),
- 투자 기간 (t)는 3년입니다.

이제 복리 공식에 맞게 값을 대입하여 총액을 계산해보겠습니다.
Python 코드를 사용하여 풀어보겠습니다.

RunStep: type: "message_creation"

ThreadMessage#2: run.status=in progress

복리 이자 계산 공식은 다음과 같습니다:

[$A = P \times (1 + r)^t$]

여기서:

- (A)는 미래 가치(원금에 이자가 붙은 총액),
- (P)는 원금,
- (r)은 이자율(1년에 한 번씩 복리가 적용될 때),
- (t)는 시간(년)입니다.

주어진 정보에 따라:

- 원금 (P)는 10,000,000원,
- 이자율 (r)은 4% 또는 0.04(소수로 표현),
- 투자 기간 (t)는 3년입니다.

이제 복리 공식에 맞게 값을 대입하여 총액을 계산해보겠습니다.
Python 코드를 사용하여 풀어보겠습니다.

RunStep: type: "message_creation"

RunStep: type: "tool_calls"

```
# 복리 계산을 위한 파이썬 코드 작성
principal = 10000000 # 초기 투자금액: 1천만원
interest_rate = 0.04 # 연이율: 4%
time = 3 # 투자 기간: 3년
# 복리 이자 계산
future_value = principal * (1 + interest_rate) ** time
# 최종 금액과 원금 대비 이자 수익 계산
interest_earned = future_value - principal
future_value, interest_earned
```

outputs: [{ 'logs': '(11248640.0, 1248640.0)', 'type': 'logs' }]

Python
코드생성

Python
실행

ThreadMessage#2: run.status = completed

1천만원을 4% 복리 이율로 3년간 예치할 경우, 3년 후 받게 되는 금액은 총 1124만 8640원입니다.

이 중 이자로 얻게 되는 금액은 124만 8640원입니다.

풀이 과정은 다음과 같습니다:

1. 원금(P)은 1천만원입니다.
2. 연이율(r)은 4%, 소수로 표현하면 0.04입니다.
3. 예치 기간(t)은 3년입니다.

복리 이자 계산 공식에 따라 최종 금액(A)을 계산합니다.

```
[ A = P times (1 + r)^t ]
[ A = 10000000 times (1 + 0.04)^3 ]
[ A = 10000000 times (1.04)^3 ]
[ A = 10000000 times 1.124864 ]
[ A = 11248640 ]
```

따라서 예치 후 3년이 지난 후의 총 금액은 1124만 8640원이 됩니다.

RunStep: type: "message_creation"

RunStep: type: "tool_calls"

```
# 복리 계산을 위한 파이썬 코드 작성
principal = 10000000 # 초기 투자금액: 1천만원
interest_rate = 0.04 # 연이율: 4%
time = 3 # 투자 기간: 3년
# 복리 이자 계산
future_value = principal * (1 + interest_rate) ** time
# 최종 금액과 원금 대비 이자 수익 계산
interest_earned = future_value - principal
future_value, interest_earned
```

outputs: [{ 'logs': '(11248640.0, 1248640.0)', 'type': 'logs' }]

Python
수행결과를
바탕으로
사용자에게
전달할
최종메시지
생성

챗GPT
스스로
단계별로
문제해결
과정을
언어로
생성해가며
추론

Python
사용 결정

개념은 모델 내에 학습되어 있고 구체적인
Action(연산)만 파이썬에게 위임하는 형태로 동작

위를 GPT-4(API)의 프롬프트로 입력해서 파이썬 코드
생성을 요청하면, 오른쪽과 거의 유사한 코드 생성

코드 인터프리터의 핵심은, 위와 같이 프로그래밍어로 잘
변환될 수 있는 텍스트를 생성하는 프롬프트 엔지니어링
기술

추론을 통한 Planning

도구 사용

최종 답변 생성

내부적으로 CoT / ReAct 프롬프트 엔지니어링 동작

* 현재는 위의 로그 확인 안되고 있음. OpenAI 막아 놓은 건가?