AI 작성자와 검토자를 나눠보니, 빠진 것은 실행 안내였습니다

실행 근거: 2026-10-13 / 자료 정리: 2026-10-15

일반 요청과 역할·기준 적용 요청은 같은 입력으로 비교했습니다. 이후 최종 수정에는 실행 파일과 추가 확인 사실이 들어갔습니다. 뒤의 원고와 검토 의견은 당시 기록입니다. 당시 ZIP 안내는 이번 공개 PDF의 첨부 방식과 다릅니다.

아래 복원 방법으로 만든 calculator.html을 브라우저로 여세요. 10000원/3명은 3334·3333·3333원, 0명은 오류, 다시 3명은 정상입니다. 인터넷·계정·서버는 필요하지 않고 실제 송금 기능은 없습니다.

권장 복원 방법: 다음 페이지의 복원 코드만 텍스트 편집기로 복사해 restore.py로 UTF-8 저장합니다. 새 빈 폴더에서 Python 3로 python3 restore.py를 실행하세요. Windows에서는 py restore.py를 사용할 수 있습니다. 실행하면 현재 폴더에 calculator.html 파일 하나가 만들어집니다. 만든 파일은 뒤의 원본과 동일합니다. 네트워크 접속 없이 현재 폴더에 파일을 만들며, 같은 이름이 있으면 덮어쓰지 않고 멈춥니다. 페이지 제목과 하단 쪽번호는 복사하지 마세요. 긴 원본 코드를 직접 복사하는 대신 이 방법을 사용하세요.

실행 파일 복원 코드 (Python 3)

```
# Copy from this line through the END marker.
import base64, gzip, json
from pathlib import Path
data = (
    "H4sIAAAAAAAAC/4VXb2/bxhn/KhdnBclZokhZsRP9W7suQIulfbH2pbHhSB6lgyke"
    "ezzaVhUVQaEWRuwNfWGvXhsHDpaizbAXQuI0HpBX+TgW9R323B1lya6dGLCkOz5/"
    "f8/vee44WPJx5GcRFozbXdGLlupoqXkjYL7oJwTJnXZTfqIIx53W+tIGW1+CHYKD"
    "drNHBEZ+F/OUCHiUibB8Wz5V+zHuEdjcpGQrYVysLyGfxYLEUnKLBqLbCsgm9UlZ"
    "LUo0poLiqJxCPKTlSjOCioi08x+/Q5PdX/JHIzT5+37+v8PpPw7zoxOU7z6dfn/Y"
    "rGipZir68LUe/37gse1ySr+kcafuMR4QXoadoceC/qCHeYfGdaeR4CCQAivVZLvh"
    "YX+jw1kWB/WbJAjd8FbDZxHj9ZvunapXqzZCCLzuriXbFddeQ2k/FaRXzmgpxXFa"
    "Tgmn4bCHaTxYMLTVpYI0Cv8cBzRL6+4qODv3LBc9vK3Tr99ZddRaxYczwYZddyD9"
    "ylSIjjOiMSl3Ce10IRq7NpN2kIOk6WG3uqBRrc3t1cD2udDKYAuiKnuc4I36BiFJ"
    "GUdRY67o3pkrStO3QSkZzCNdc/zuMMIeiQYBTZMI9+texPwNbWJLh7fqODMb0iuY"
    "kb5pnGTikpY26jrOewtogLqsi0Kv7oJ+yiIaoJu3qqu1NXwJ1kVUXYmTKheNu1AY"
    "UVRytrpcoaGXCcEuVG5WdK24WEfgzds9o6pz2b0GoSxYosvvZzwFswmj0Au8cF/v"
    "sk3CLwSxslIL15zZc4AMexEJBizBPhX9ur06s7SFqdDA1kPmZ2l5k6YUZEuF6oXN"
    "AcuEZFF9ZY6pt1LzoVrFkzILQ+hmKTCs11MSEV/QSwAFK2QlrBXV9DEnonyhX4aY"
    "C+qDtwItlfzcoR8Et0ntHDhJVOQM0x7QsGTHTJBBYU1Xe2gTzhmfbWLYrVaHEO5v"
    "uXQZbMXITcwphu8460FF/LrAHow7LtdpYQaxSBG86K2q6kVZkjBiW7obQwaB8cGC"
    "B9XAFyPFsyCrtVu127WGINuiDJARvggt5Dt8v0cCis1LjLcGakotUkoPlgtYubIj"
    "L0yHKoymGZPn/TQcrsfNip6LzYqe2NI+zGewCTPcvW62grTbbiZtOWY/+vyTe2i6"
    "t5cfvc6PQOx0bwK/fhjnxw8mTx6h/PuTybMTNBnvy9XZi1H+9Xh6cIzynUOUP97J"
    "H76c7O5Mdp/aKD86nY7Gk9Er0BmfPR+/eSWljw/evMpHjybPR2fjB2h6MMp/2J8e"
    "HOY/P0D5wcO5erOSwIzXZESyomU1gqAnvD4cKMACKsrqKFDHUxXR4Dfb1em/9otj"
    "A72BMK5NviqTn3x9OPnlVAXy43f5kYznMZo835kenk5+fQCpIFc+2H8GyY8nv46m"
    "B8/OU919mh8DNt8+OTvdkarXZhQy3lsMVa7hpIzZJoZewQIqp/JE8ACEBBM4gkzy"
    "k6P84AkyIQCrWVES7abqR22skENqq8cCeRAX/IddsJ3JHSCJ48BawRmQ1OfUK/CM"
    "aI8KieSi94SwRAEJpZTgmZP/fHOl+5ngO/yvSAeJVikcIj/CaSqlod3miTpfuU7+"
    "z5eyEiVUeHe/cksQP8QAgB8fAOEmP+8B576ZHP90JQMV4LpLkLzcSNAzT+cp6wR+"
    "ZM13Rmen42ZFC7abxYRYqBEnaRaJGW4R3ZSWEphtKuJmRWvAD1lM+CpYK4stZ0j7"
    "7MXLs7HsnHH++L+KKw9fKj4usgZBEJOHP4HUt/nRnrSldJsV3buyVolom2EWK+Mm"
    "h+fWYD2ebSAVq6mIUEK6IPI5oqF5o/LX9fVg+XcVW5BUmJ8JDrNFi1o2LHqmZaH7"
    "99HVcoWtmaCFRJezLXRXDmnT0AU7e/G6KJNqHF0dOS2K6pzk/36dj06h123DakBU"
    "cDVMBcKtT7OeR3gRSwnFs43CaaNIQO/aNP0Mh+RjOEk7IIOt+/dx04GPtmK2+rOu"
    "iE7G5Gj2zP5lpI9evzXEa/3G4DduuvCh/F7yeI7CnK/v8qTB+KL1CRZdG44gsIMr"
    "McDBW/i9WErAkZvxGH3AOe7bIWc9cxCRuAODPx6WzL+VqNVqf7Fs0ib/g1t3LGkV"
    "DgPIQPKehQhaMovIjVbLkKdTCIdTYFh60ybb8rqethSBGiRKCZLkstX6gx5cAUTx"
    "DIxaZidiHo4+79JUegG2a2bOGKpzkZ3QgncKGAGxsDtE3I2I/PnH/seBacxHn2GV"
    "oHveJakbELCSKjYcjXc3QeIehWt5DPUw1NQxSiaAANZsTuCK4JMPuzQKOImBs1oR"
    "IlPt3yKt9oDYCSfSzJ9IiMG6aTWu1FW7akh9Kt9vDKMheH9Q0FeBUyCnkTKvTUVR"
    "3LBsNQxL14pp4s/krIb2lMwh8uEeL0ihA/IAS2LLi8eHxduWAQcTnLTIWC7ig6SC"
    "zCemiUseQISXvZID7czuMfnqVXS5scHKf/6LYS0bctbCdJqra6YtG8Bko4iHRdcG"
    "xCJdKLPACAH9CktWgVtEr9WOKGhH9EJC+K2xGg0W2ThJSByYEbUa8oo3WyclFsGO"
    "j4XfNeFWaQ0uVVPdNA1V40WPsG33SJriDmkMhxdoXtE3qop6UV4a/h/cwOv1Tw8A"
    "AA=="
)
files = json.loads(gzip.decompress(
    base64.b64decode(data)))
for name in files:
    if Path(name).exists():
        raise SystemExit("File exists: " + name)
for name, text in files.items():
    Path(name).write_bytes(text.encode("utf-8"))
print("Files restored successfully")
# END
```

원자료

공개 연재「AI에게 실제 일을 맡겨보기」E02 실험 자료. 독자는 AI 도구를 처음 만드는 성인이다.
주제: 정수 원 단위 균등 분할. 실제 구현은 JavaScript+로컬 브라우저, 사용 도구는 Codex.
10,000원/3명 → [3334,3333,3333]. 정수 몫을 나눈 뒤 나머지는 앞 번호부터 1원씩 배분.
0원/2명 → [0,0]. 2원/5명 → [1,1,0,0,0]. 총액 0~10억 원, 인원 1~1000명, 모두 정수 입력.
Node.js 자동검사 정상4건+오류7건=11건: 합계·인원수·차이 최대1원 확인. 오류 예: 빈칸,0명,음수,소수 인원,지수표기,상한초과.
브라우저: 10000/3 정상 계산. 이후 인원을0으로 수정하면 기존 결과 제거, 실행하면 인원 수정 안내. 3명으로 고치고 재계산 가능.
실제 송금, 독립 사용자 시험, 독자 반응·작업시간 절감·매출 측정은 안 했다.
이전 원고는 실험 예고였고 이번에는 실제 코드·실행 기록이 생겼다. 수치의 외부 일반화는 불가.

검토 기준

C1 입력·출력과 합계가 정확함. C2 나머지 배분 순서를 설명. C3 허용 입력 범위와 잘못된 입력의 수정 방법. C4 Node.js11건과 브라우저 확인을 구분. C5 송금·사용자시험·효과를 만들어내지 않음. C6 초보가 재현할 입력→행동→확인 절차. C7 글이 독자 질문에 답하고 기술 보고서 나열로 끝나지 않음.

일반 요청 원고

AI에게 실제 일을 맡겨보기 E02: 1원까지 맞춰 나누기

10,000원을 3명이 나누면 각자 얼마를 맡아야 할까? 소수점 없이 원 단위로 나누려면 남는 돈을 처리하는 규칙이 필요하다. 이번에는 Codex를 사용해 JavaScript로 균등 분할 기능을 구현하고, 로컬 브라우저에서 실행했다. 지난 글이 실험 예고였다면, 이번에는 실제 코드와 실행 기록을 바탕으로 결과를 살펴본다.

규칙은 간단하다. 먼저 총액을 인원수로 나눈 정수 몫을 모두에게 배정한다. 남은 금액은 앞 번호부터 1원씩 더한다. 그래서 10,000원을 3명에게 나누면 3,334원, 3,333원, 3,333원이 된다. 0원을 2명에게 나누면 둘 다 0원이고, 2원을 5명에게 나누면 앞의 두 명만 1원씩 받는다. 입력 범위는 총액 0~10억 원, 인원 1~1,000명이며 모두 정수다.

계산 결과는 무엇으로 확인했을까? Node.js 자동검사에서는 정상 입력 4건과 오류 입력 7건, 총 11건을 검사했다. 나눈 금액의 합이 원래 총액과 같은지, 결과가 인원수만큼 나오는지, 사람 간 금액 차이가 최대 1원인지 확인했다. 오류 입력에는 빈칸, 0명, 음수, 소수 인원, 지수표기, 상한 초과 등이 포함됐다. 초보자에게도 이 세 가지 계산 기준은 결과가 맞는지 판단할 출발점이 된다.

브라우저에서는 10,000원과 3명을 입력해 정상 계산을 확인했다. 이어 인원을 0으로 바꾸자 기존 결과가 사라졌고, 실행하면 인원을 수정하라는 안내가 나왔다. 다시 3명으로 고친 뒤에는 재계산할 수 있었다. 잘못된 입력을 넣었을 때 이전 계산값이 남아 혼동을 주는지도 함께 살펴본 것이다.

이번에 확인한 것은 정해진 입력 범위와 검사 사례에서의 계산, 그리고 오류 수정 후 다시 실행하는 흐름이다. 실제 송금이나 독립 사용자 시험은 하지 않았고, 독자 반응·작업시간 절감·매출도 측정하지 않았다. 따라서 이 기록만으로 다른 상황에서도 같은 성과가 난다고 말할 수는 없다. 이번 실험의 배움은 작은 기능이라도 계산 규칙을 명확히 정하고, 정상 결과와 잘못된 입력 뒤의 동작을 함께 확인해야 한다는 점이다.

역할·기준 적용 원고

AI가 만든 더치페이 도구, 계산됐다는 것만 믿어도 될까?

1만 원을 세 명이 나눌 때 누구는 1원을 더 내야 한다. AI가 만든 도구에서 먼저 확인할 것은 화면의 그럴듯함보다 '총액이 맞는가, 남는 돈은 누구에게 가는가, 잘못 입력하면 고칠 수 있는가'다. 이번 「AI에게 실제 일을 맡겨보기」 E02에서는 이 질문을 정수 원 단위 균등 분할로 확인했다.

Codex로 JavaScript 도구를 구현하고 로컬 브라우저에서 실행했다. 이전 글이 실험 예고였다면, 이번에는 코드와 실행 기록이 생겼다. 허용 입력은 총액 0~10억 원, 인원 1~1000명이며 둘 다 정수다. 정수 몫을 먼저 나누고 남은 금액은 앞 번호부터 1원씩 배분한다. 따라서 번호 순서가 누가 1원을 더 낼지 결정한다.

결과는 10,000원/3명 → [3334,3333,3333]으로, 합계는 10,000원이다. 0원/2명 → [0,0], 2원/5명 → [1,1,0,0,0]도 각각 합계가 0원과 2원이다. 나눌 돈보다 사람이 많으면 0원을 내는 사람도 생긴다. Node.js 자동검사는 정상 4건과 오류 7건, 총 11건으로 합계·인원수·금액 차이 최대 1원과 오류 입력을 확인했다. 이는 브라우저에서 11건을 모두 확인했다는 뜻은 아니다.

초보자가 재현하려면 로컬 브라우저에서 도구를 열고 총액에 10000, 인원에 3을 입력해 계산한다. 결과가 위 세 금액인지, 더하면 10000인지 확인한다. 다음에는 인원을 0으로 바꾼다. 기존 결과가 사라지는지 보고, 실행했을 때 인원 수정 안내가 나오는지 확인한다. 인원을 3으로 고쳐 재계산하는 것까지가 기록에서 확인된 브라우저 흐름이다. 빈칸은 채우고, 음수·소수·상한 초과 값은 허용 범위의 정수로 고친다. 지수표기 대신 숫자를 그대로 입력한다.

이번 결과는 계산 규칙과 입력 수정 흐름을 확인한 것이다. 실제 송금이나 독립 사용자 시험은 하지 않았고, 시간 절감·매출·독자 반응도 측정하지 않았다. 남은 질문은 실제 이용자가 앞 번호부터 1원을 더 내는 규칙을 이해하고 받아들이는가다. 내 도구를 점검할 때도 정답 확인에서 한 걸음 더 나아가, 배분 이유와 오류 수정 방법이 보이는지 살펴보자.

첫 검토

전체 판정: REVISE (A·B 모두)

INPUT.md와 CRITERIA.md만을 근거로 비교했다. 작성 프롬프트는 추정하지 않았다.

기준	A	B
C1	PASS — “3,334원, 3,333원, 3,333원”은 합계 10,000원. 0원·2원 사례도 정확하다.	PASS — “[3334,3333,3333]”과 합계 10,000원이 맞고 나머지 두 사례도 정확하다.
C2	PASS — “앞 번호부터 1원씩 더한다”로 나머지 배분 순서를 설명한다.	PASS — “번호 순서가 누가 1원을 더 낼지 결정”으로 순서의 의미까지 설명한다.
C3	PARTIAL — “총액 0~10억 원, 인원 1~1,000명”과 “다시 3명으로”는 명확하나, 나열한 다른 오류의 수정 방법은 빠졌다.	PASS — 같은 범위에 “빈칸은 채우고”, “허용 범위의 정수로 고친다” 등 수정 행동을 제시한다.
C4	PASS — “Node.js 자동검사”의 정상 4건·오류 7건과 “브라우저에서는”의 확인 흐름을 나눠 쓴다.	PASS — “총 11건”과 “브라우저에서 11건을 모두 확인했다는 뜻은 아니다”로 범위를 구분한다.
C5	PASS — “실제 송금이나 독립 사용자 시험은 하지 않았고”와 효과 미측정 진술이 원자료에 맞는다.	PASS — “시간 절감·매출·독자 반응도 측정하지 않았다”로 성과를 만들어내지 않는다.
C6	PARTIAL — “10,000원과 3명을 입력”부터 오류 수정까지 관찰 기록은 있으나, 도구를 여는 방법과 독자용 실행 안내가 부족하다.	PARTIAL — “10000, 인원에 3을 입력해 계산”→합계 확인→0명→3명 순서는 명확하다. 다만 “도구를 열고”에 필요한 파일·접근 방법이 없다.
C7	PASS — “계산 결과는 무엇으로 확인했을까?”에 답하고 끝에서 점검 원칙을 제시해 보고서 나열에 머물지 않는다.	PASS — “계산됐다는 것만 믿어도 될까?”에 배분 이유·오류 복구 확인으로 답해 독자의 판단과 연결한다.

수치의 사실성에는 차이가 없다. 설득력에서는 B가 배분 순서의 의미와 독자의 행동을 더 구체적으로 연결한다. A도 설명 구조는 성립한다. 두 글 모두 한계 문장이 설명을 압도하지 않는다. 다만 ‘Node.js’, ‘로컬 브라우저’, ‘지수표기’는 설명 없이 등장해 초보자의 이해를 막을 수 있다. B의 명시적 절차만으로 처음부터 재현 가능하다고 판정할 수는 없다.

B 편집 피드백

1. “도구를 열고” 앞에 실제 제공 파일이나 접근 경로와 여는 순서를 보완한다. 현재 자료에는 없으므로 확인 전 임의 경로를 쓰지 않는다.
2. ‘Node.js 자동검사’는 계산·오류 입력을 코드로 확인한 검사라고 풀고, ‘로컬 브라우저’와 ‘지수표기’도 첫 등장에 쉬운 설명을 붙인다. 계산 규칙과 입력 수정 행동을 이해하는 데 필요한 보완이다.

수정 원고 - 당시 ZIP 안내 포함

AI가 만든 더치페이 도구, 계산됐다는 것만 믿어도 될까?

1만 원을 세 명이 나누면 누군가는 1원을 더 내야 한다. 「AI에게 실제 일을 맡겨보기」 E02에서는 Codex로 JavaScript 계산 도구를 만들고, 합계뿐 아니라 남는 돈의 배분과 잘못된 입력을 고치는 과정까지 확인했다.

규칙은 단순하다. 총액을 인원수로 나눈 정수 몫을 먼저 배정하고, 나머지는 앞 번호부터 1원씩 더한다. 10,000원/3명 → [3334,3333,3333]이며 합계는 10,000원이다. 0원/2명 → [0,0], 2원/5명 → [1,1,0,0,0]도 합계가 각각 0원과 2원이다. 대괄호 안 숫자는 번호순으로 각 사람이 낼 금액이다. 앞 번호가 1원을 더 내며, 돈보다 사람이 많으면 0원을 내는 사람도 생긴다.

직접 확인하려면 과제 첨부 ZIP을 풀고 demo 폴더의 calculator.html을 Chrome 등 브라우저로 연다. 계산 코드와 화면 꾸밈을 한 HTML 파일에 담아 계정·인터넷·서버 없이 내 컴퓨터에서 실행할 수 있게 했다. 이 파일을 브라우저에서 열어 다음 흐름도 확인했다.

총액에 10000, 인원에 3을 넣고 '금액 나누기'를 누른다. 사람별 금액이 3334, 3333, 3333인지, 더하면 10000인지 살펴보자. 이어 인원을 0으로 바꾸면 이전 결과가 사라진다. 다시 실행하면 인원 수정 안내가 나온다. 3으로 고쳐 재계산하면 정상 결과를 확인할 수 있다.

허용 범위는 총액 0~10억 원, 인원 1~1000명이며 모두 정수다. 빈칸은 채우고, 음수·소수·범위 밖 값은 범위 안 정수로 고친다. 1e4처럼 수를 줄여 쓰는 '지수표기' 대신 10000처럼 숫자를 그대로 입력한다.

화면 확인과 별도로, 브라우저 밖에서 JavaScript를 실행하는 Node.js로 자동검사를 했다. 계산과 오류 입력을 코드로 확인한 정상 4건·오류 7건, 총 11건이다. 합계·인원수·사람별 금액 차이 최대 1원을 점검했다. 브라우저 확인 범위는 앞서 설명한 계산·오류·복구 흐름이다.

이번 결과는 AI의 답변만이 아니라 실행 파일을 만들고 검사한 결과다. 실제 송금이나 독립 사용자 시험은 하지 않았고, 시간 절감·매출·독자 반응도 측정하지 않았다. 내 도구를 믿기 전에도 합계가 맞는지, 배분 순서를 이해할 수 있는지, 틀린 입력을 고쳐 다시 계산할 수 있는지 직접 확인해 보자.

후속 검토 - 문서 대조 결과

전체 판정: PASS

INPUT.md·CRITERIA.md·SUPPLEMENT.md와 FINAL.md를 대조했다. 앞서 지적한 실행 파일 열기 방법과 어려운 용어 설명은 보완됐다.

- C1 PASS — “10,000원/3명 → [3334,3333,3333]”의 합계가 정확하다. 0원·2원 사례도 원자료와 일치하며 대괄호를 사람별 금액으로 설명했다.
- C2 PASS — “나머지는 앞 번호부터 1원씩 더한다”로 배분 순서와 누가 더 내는지 설명한다.
- C3 PASS — “총액 0~10억 원, 인원 1~1000명”과 정수 조건을 명시했다. 빈칸·음수·소수·범위 초과·지수표기의 수정 방법도 제시한다.
- C4 PASS — “화면 확인과 별도로” Node.js 정상 4건·오류 7건, 총 11건을 설명하고 브라우저 확인은 계산·오류·복구 흐름으로 구분한다.
- C5 PASS — “실제 송금이나 독립 사용자 시험은 하지 않았고”와 효과 미측정 진술로 확인 범위를 지킨다.
- C6 PASS — “과제 첨부 ZIP을 풀고 demo 폴더의 calculator.html”을 여는 방법이 SUPPLEMENT.md와 일치한다. 10000·3 입력→‘금액 나누기’→금액·합계 확인→0명 오류→3명 복구까지 따라 할 수 있다.
- C7 PASS — “계산됐다는 것만 믿어도 될까?”에 합계·배분·오류 복구를 직접 점검하는 방법으로 답한다. 기술 기록이 독자의 판단과 행동으로 이어진다.

Node.js는 “브라우저 밖에서 JavaScript를 실행”한다고 풀었고, 지수표기는 1e4와 10000으로 설명했다. ‘로컬’ 대신 내 컴퓨터에서 파일을 여는 행동을 안내해 이해를 돕는다. 한계 문장은 짧고 본문 설명을 압도하지 않는다. 수치의 사실성과 별개로, 실행 안내가 구체화되어 글의 설득력도 뒷받침된다.

baseline-v1은 동일 입력 비교였지만 v2에는 실행 파일과 확인 사실이 추가됐다. 이번 보완을 순수 프롬프트 효과로 주장할 수 없다. 이 검토는 제공 문서의 대조이며, 실행 파일을 직접 열거나 ZIP 첨부 상태를 재확인한 것은 아니다.

실행 범위

실행 전 기준 고정: 동일 INPUT.md, 동일 실행 설정(gpt-6-astra / high). baseline=일반 요청, producer=역할·7개 기준 적용, reviewer=입력과 두 원고를 독립 검토, reviser=검토 피드백 적용. 단일 소재·각 조건1회이므로 모델 성능이나 일반적 개선률을 추정하지 않음. 기존 정적 result-v1/review/result-final 파일을 실행 증거로 사용하지 않음.
검토 후 변경: C6(도구 여는 방법) 부족이 확인되어 실제 단일 HTML 실행 파일을 추가하고 파일 URL로 검증했다. reviser에는 SUPPLEMENT.md가 추가되므로 baseline-v1의 동일 자료 비교와 v2의 기능/자료 보강 효과를 분리해 해석한다.

실행 기록 요약 - 내부 세션 식별자 제외

단계 대응: harness-v1 = producer-v1, harness-v2 = reviser-v2


```
[
 {
  "stage": "baseline-receipt.json",
  "model": "gpt-6-astra",
  "reasoning": "high",
  "state": "completed",
  "exit_code": 0,
  "turn_completed": true
 },
 {
  "stage": "final-review-receipt.json",
  "model": "gpt-6-astra",
  "reasoning": "high",
  "state": "completed",
  "exit_code": 0,
  "turn_completed": true
 },
 {
  "stage": "harness-v1-receipt.json",
  "model": "gpt-6-astra",
  "reasoning": "high",
  "state": "completed",
  "exit_code": 0,
  "turn_completed": true
 },
 {
  "stage": "harness-v2-receipt.json",
  "model": "gpt-6-astra",
  "reasoning": "high",
  "state": "completed",
  "exit_code": 0,
  "turn_completed": true
 },
 {
  "stage": "review-receipt.json",
  "model": "gpt-6-astra",
  "reasoning": "high",
  "state": "completed",
  "exit_code": 0,
  "turn_completed": true
 }
]
```


실제 요청문: baseline

이 작업은 지피터스 공개 연재 과제용 한국어 원고 작성 실험이다. INPUT.md만 원자료로 사용한다. 다른 디렉터리, 개인 기록, 웹, 메신저를 읽거나 쓰지 말고 외부 게시도 하지 않는다. 허용 쓰기는 이 cwd의 output.md 하나다. 추가 에이전트를 만들지 않는다. output.md를 작성한 뒤 최종 응답에 동일 원고와 파일 경로를 반환한다. INPUT.md를 읽고 초보 독자용 실험 결과 글을 700~1100자 안팎으로 작성하라. 제목과 본문을 포함한다. 수치를 지어내지 않는다.

실제 요청문: final-reviewer

앞선 E02 원고 검토의 후속이다. 기존 INPUT.md/CRITERIA.md와 FINAL.md, 추가로 실제 확인된 실행 방법인 SUPPLEMENT.md를 대조하라. 앞서 C6의 파일 열기 방법과 어려운 용어를 보강하라고 했는데 수정이 충족됐는지 확인하라. C1~C7별 PASS/PARTIAL/FAIL과 핵심 근거, 전체 PASS 또는 REVISE를 final-review.md에 적어라. 개선률·점수는 만들지 않는다. baseline-v1은 동일 입력이었지만 v2는 실행 파일과 확인 사실이 추가됐으므로 순수 프롬프트 효과로 주장하면 안 된다. 웹·개인자료·다른 디렉터리·추가 에이전트·게시·설정 변경 금지. 허용 쓰기는 final-review.md 하나다. 출력은 간결히 600~1100자 안팎. 원문 수정은 하지 않는다.

실제 요청문: producer-v1

이 작업은 지피터스 공개 연재 과제용 한국어 원고 작성 실험이다. INPUT.md만 원자료로 사용한다. 다른 디렉터리, 개인 기록, 웹, 메신저를 읽거나 쓰지 말고 외부 게시도 하지 않는다. 허용 쓰기는 이 cwd의 output.md 하나다. 추가 에이전트를 만들지 않는다. output.md를 작성한 뒤 최종 응답에 동일 원고와 파일 경로를 반환한다. Producer 역할: 독자에게 “AI가 만든 더치페이 도구에서 무엇을 확인할까”에 답한다. 입력자료를 읽고 결과 중심의 700~1100자 안팎 원고를 쓴다. 구조는 독자 문제→실제 수행→결과→재현 방법→남은 질문. 다음 검토 기준을 먼저 반영하라: C1 입력·출력과 합계가 정확함. C2 나머지 배분 순서를 설명. C3 허용 입력 범위와 잘못된 입력의 수정 방법. C4 Node.js11건과 브라우저 확인을 구분. C5 송금·사용자시험·효과를 만들어내지 않음. C6 초보가 재현할 입력→행동→확인 절차. C7 글이 독자 질문에 답하고 기술 보고서 나열로 끝나지 않음.
한계는 관련 주장 가까이 짧게 붙인다. 실행하지 않은 경험을 단정하지 않는다.

실제 요청문: reviewer

지피터스 공개 연재 E02 원고 검토 실험이다. INPUT.md(검증된 사실), CRITERIA.md(사전에 정한 기준), A.md와 B.md를 읽어 독립적으로 비교하라. A/B가 어떤 프롬프트로 만들어졌는지는 추정하지 마라. 원자료 밖 정보나 웹을 사용하지 않는다. 다른 에이전트·메신저·외부 게시·전역 변경은 금지. 쓰기는 이 cwd의 output.md만 허용한다.
각 원고에 C1~C7을 PASS/PARTIAL/FAIL로 판정하고 반드시 해당 표현과 이유를 짧게 제시한다. 그럴듯한 총점·향상률을 만들지 않는다. 차이가 없으면 없다고 말한다. B에 적용할 구체적 편집 피드백은 개선할 근거가 있는 것만 1~3개 제시한다. 독자가 재현할 수 있는지, 기술 용어가 이해되는지, 과잉 한계 문장이 설명을 압도하는지도 본다. 수치의 사실성과 글의 설득력을 분리한다. 전체 판정은 PASS 또는 REVISE다. 700~1500자 안팎의 output.md를 작성하고 응답으로 결과와 파일 위치를 반환한다.

실제 요청문: reviser-v2

공개 연재 E02 원고의 피드백 반영 단계다. INPUT.md, DRAFT.md, FEEDBACK.md, CRITERIA.md, SUPPLEMENT.md만 읽고 output.md를 작성한다. 다른 디렉터리·개인 기록·웹·메신저·전역 설정을 읽거나 쓰지 말고, 추가 에이전트를 만들지 않는다. 쓰기는 output.md만 허용한다.
Reviewer가 지적한 재현 진입점과 초보자가 어려워할 용어를 반영해 800~1200자 안팎으로 원고를 고친다.
SUPPLEMENT는 검토 이후 실제 추가 구현·확인한 사실이다. 모델/프롬프트만의 효과로 포장하지 않는다. 작업사정이나 reviewer에게 답하는 해명은 기사에 넣지 말고 독자가 직접 계산을 확인할 수 있도록 작성한다. 입력·출력·범위는 원자료를 그대로 유지한다. output.md를 쓴 뒤 최종 응답에 원고와 경로를 반환한다.

계산기 전체 코드

```
<!doctype html><html lang="ko"><head><meta charset="utf-8"><meta name="viewport"
content="width=device-width,initial-scale=1"><title>원 단위 더치페이 실험</title><style>
*{box-sizing:border-box}body{margin:0;padding:32px;background:#edf1f5;color:#192b42;font:17px/1.7
system-ui,sans-serif}main{background:white;border-radius:16px;padding:36px;max-width:960px;margin:auto}h1{font-size:32px;line-h
eight:1.4;margin:0 0 16px}h2{font-size:24px;margin:40px 0 16px}h3{word-break:keep-all;font-size:19px;margin:0 0
8px}p{max-width:70ch}label{display:block;font-weight:600;margin:16px 0
6px}input{display:block;width:100%;max-width:600px;border:1px solid
#52647a;border-radius:6px;padding:12px;font:inherit;color:inherit;background:white}button{background:#192b42;color:white;border:
0;border-radius:6px;padding:12px 20px;font:inherit;margin-top:16px;cursor:pointer}button:hover{background:#334f70}button:disabl
ed{opacity:.6;cursor:wait}input:focus-visible,button:focus-visible{outline:3px solid
#b34c00;outline-offset:3px}::selection{background:#d3e3f4}input{caret-color:#192b42}article{border-top:1px solid
#cdd8e4;padding:24px 0}small,.note{color:#52647a}.error{color:#a52622}output{display:block;margin-top:16px;font-variant-numeric:
tabular-nums}output ol{max-height:260px;overflow:auto}footer{margin-top:36px;color:#52647a}a{color:#245484;text-underline-offse
t:4px}@media(max-width:600px){body{padding:12px}main{padding:24px 18px}h1{font-size:27px}button{width:100%}}
</style></head><body><main><h1>원 단위 더치페이</h1><p>이 HTML 파일을 브라우저로 열면 바로 계산할 수 있습니다. 인터넷 연결·계정·서
버가 필요하지 않습니다.</p><section aria-labelledby="split-title"><h2 id="split-title">2화 실험 · 원 단위 더치페이</h2><p>나머지 원은 앞 번
호부터 1원씩 배분합니다. 실제 송금은 하지 않습니다.</p><form id="split-form" novalidate><label for="total">총액 (원)</label><input
id="total" inputmode="numeric" value="10000" aria-describedby="limit"><label for="people">인원 (명)</label><input id="people"
inputmode="numeric" value="3"><p id="limit" class="note">총액 0~10억 원, 인원 1~1,000명. 정수만 입력할 수 있습니다.</p><button
type="submit">금액 나누기</button><output id="split-result" aria-live="polite"></output></form></section><footer>공개 연재 실습 · 실제
 송금 기능 없음</footer></main><script>(function(root){
function split(total, people){
 if(!/^\d+$/.test(String(total).trim()) || !/^\d+$/.test(String(people).trim())) throw Error('총액과 인원은 정수로 입력해주세요.');
 const a=Number(total), n=Number(people);
 if(!Number.isSafeInteger(a)||a<0||a>1000000000)throw Error('총액은 0~1,000,000,000원으로 입력해주세요.');
 if(!Number.isSafeInteger(n)||n<1||n>1000)throw Error('인원은 1~1,000명으로 입력해주세요.');
 const q=Math.floor(a/n), r=a%n;
 return Array.from({length:n},(_,i)=>q+(i<r?1:0));
}
if(typeof module!=='undefined')module.exports=split;else root.splitAmount=split;
})(globalThis);
</script><script>const form=document.getElementById('split-form'),out=document.getElementById('split-result');form.addEventListen
er('input',()=>out.replaceChildren());form.onsubmit=e=>{e.preventDefault();out.replaceChildren();out.className='';try{const
amounts=splitAmount(document.getElementById('total').value,document.getElementById('people').value);const
p=document.createElement('p');p.textContent='합계 '+amounts.reduce((a,b)=>a+b,0).toLocaleString('ko-KR')+'원 · '
+amounts.length+'명';const ol=document.createElement('ol');for(const a of amounts){const
li=document.createElement('li');li.textContent=a.toLocaleString('ko-KR')+'원'
;ol.append(li);}out.append(p,ol);}catch(err){out.className='error';out.textContent=err.message;}};
</script></body></html>
```