

두 세션과 메시징 지원 확인

시연 1분 → 직접 실습 2분 · 완료 증거를 기록합니다.

어디에서 같은 기기 · 터미널 두 개

- 1 두 터미널에서 같은 inventory 폴더를 엽니다.
- 2 `claude --version`과 `/list-agents`를 확인합니다.
- 3 지원 여부와 현재 작업 폴더를 기록합니다.

막히면 미지원·다른 OS 환경이면 파일 인계 경로를 사용합니다.

완료 증거

/list-agents
지원 상태

완료하면 채팅에 ✓

구현자와 검토자 이름 지정

시연 2분 → 직접 실습 5분 · 완료 증거를 기록합니다.

어디에서

A: builder · B: reviewer

- 1 A에서 /rename inventory-builder를 실행합니다.
- 2 B에서 /rename inventory-reviewer를 실행합니다.
- 3 프롬프트 01·02로 역할과 목표를 나눕니다.

막히면 같은 이름의 세션이 여러 개면 대상 경로와 식별자를 확인합니다.

첫 운영 작업 · 10분

분리된 역할과 파일 범위

프롬프트 01 · 복사해서 시작

서브에이전트·독립 세션·팀을 구분

작업을 나누는 방식마다 맥락·조정·비용이 달라집니다. 이번 실습은 독립 세션 두 개입니다.

01

서브에이전트

주 세션의 위임 → 요약 반환

02

독립 세션

각자 맥락 → 텍스트 메시지

03

Agent teams

공유 작업 조정 · 실험 기능

실전 요청·적용

작은 조사는 서브에이전트, 구현·독립 검토는 두 세션. Teams는 지원 환경에서 선택 심화로 다룹니다.

잠깐 확인

검토자를 늘리면 항상 더 좋은가요?

실습으로 →

역할과 파일 수정 범위를 합의합니다.

파일 수정권과 검토 계약 정하기

시연 2분 → 직접 실습 4분 · 완료 증거를 기록합니다.

어디에서 templates/ROLES.md · 두 세션

- 1 구현자는 앱과 테스트를 담당합니다.
- 2 검토자는 코드 읽기와 REVIEW.md만 담당합니다.
- 3 즐겨찾기 완료 기준 다섯 개를 공유합니다.

막히면 두 세션이 같은 구현 파일을 동시에 수정하지 않게 합니다.

완료 증거

**누가 무엇을
바꾸는지 합의**

프롬프트 02 · 복사해서 시작

Claude에서 Claude로 첫 메시지

시연 2분 → 직접 실습 10분 · 완료 증거를 기록합니다.

어디에서 구현자 → inventory-reviewer

- 1 프롬프트 03으로 검토자에게 계약을 보냅니다.
- 2 검토자 창에서 실제 수신을 확인합니다.
- 3 검토자는 이해한 범위를 한 번만 회신합니다.

막히면 전달이 보류되면 수신 설정·권한을 확인합니다. 실패하면 파일로 전달합니다.

완료 증거

**수신된 작업
계약과 짧은 회신**

프롬프트 03 · 복사해서 시작

메시지 전달은 파일 공유가 아닙니다

Claude 간 메시징은 텍스트를 전달합니다. 파일은 각 세션의 경로와 권한 안에서 읽습니다.

01

공유 폴더

한 구현자 + 읽기 전용 검토

02

별도 worktree

각 브랜치에서 수정 후
통합

03

메시지

계약·변경·검증·다음 요청

실전 요청·적용

다른 worktree의 파일 변경은 자동 반영되지 않습니다. 기존 커밋과 변경 전달·통합 방법을 정합니다.

잠깐 확인

“수정 끝” 메시지만 보내면 충분한가요?

실습으로 →

공유 폴더의 쓰기 담당을 한 명으로 고정합니다.

공유 폴더와 worktree 차이 확인

시연 2분 → 직접 실습 6분 · 완료 증거를 기록합니다.

어디에서 두 세션 · 파일 경로

- 1 메시지가 파일을 전송하지 않음을 확인합니다.
- 2 이번에는 구현자 한 명만 코드를 바꿉니다.
- 3 둘 다 구현할 때 쓸 worktree 분리 절차를 읽습니다.

완료 증거

**이 실습의
읽기/쓰기 경계**

완료하면 채팅에 ✓

막히면 worktree에는 커밋한 기준본이 필요합니다. 설정·변경사항을 검토한 뒤 별도로 만듭니다.

즐거찾기 기능의 호환성 검토

시연 2분 → 직접 실습 9분 · 완료 증거를 기록합니다.

어디에서

구현자·검토자 · 기준본과 백업 JSON

완료 화면

과거 호환을 검토한 즐거찾기

01

favorite 누락을 false로
읽는 계약을 씁니다.

02

구현자가 기능·테스트를
만들고 증거를 보냅니다.

03

검토자는 과거
JSON·저장·필터를
확인합니다.

막히면

구현 시간이 부족하면 한 항목의 지정·저장·과거 호환부터 검토하고 남은 범위를 기록합니다.

환경과 오류를 함께 진단

시연 0분 → 직접 실습 5분 · 완료 증거를 기록합니다.

어디에서 터미널 · 브라우저 · 실행 로그

- 1 명령·오류·현재 경로를 한 묶음으로 기록합니다.
- 2 설치·인증·코드 문제 중 어디인지 구분합니다.
- 3 한 조건만 바꾸어 같은 작업을 다시 실행합니다.

막히면 키와 토큰 값은 가리고, 사용할 수 없는 기능은 제공 예제로 이어갑니다.

완료 증거

**원인과 재시도
결과 1건**

프롬프트 07 · 복사해서 시작

인계 메모를 남기고 10분 휴식

작업 상태를 파일로 남기고 70분에 다시 시작합니다.

어디에서

작업 폴더 · HANDOFF.md

10

분 쉬어가기

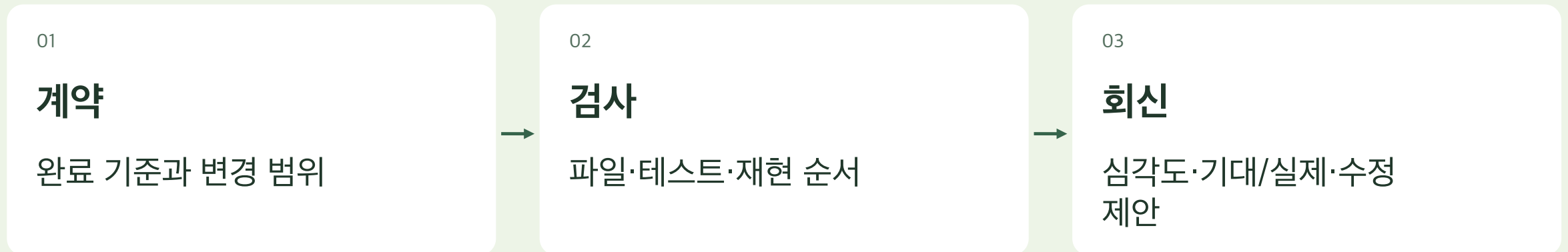
1. 현재 파일과 실행 결과를 저장합니다.
2. HANDOFF.md에 다음 행동 한 가지를 적습니다.
3. 작업 창을 남겨두고 70분에 돌아옵니다.

돌아오면: 다음 행동이 적힌 인계 메모

막히면 저장이 막히면 현재 경로·오류·다음 명령을 메모합니다.

검토자는 주장보다 재현 근거를 봅니다

독립 검토는 다른 말투의 칭찬이 아니라 같은 조건에서 결과를 다시 확인하는 작업입니다.



실전 요청·적용

즐거찾기→새로고침→필터→해제→백업 복원 순서로 실제 상태를 확인하고 REVIEW.md에 남깁니다.

잠깐 확인

브라우저를 열지 못한 검토자는 어떻게 기록하나요?

실습으로 →

검토자가 결과를 재현하고 한 번 회신합니다.

깨진 JSON으로 복원 검증

시연 1분 → 직접 실습 7분 · 완료 증거를 기록합니다.

어디에서 검토자 · 백업 복사본 · REVIEW.md

- 1 정상 백업 복사본에 오류 두 종류를 만듭니다.
- 2 가져오기를 시도하고 기존 데이터를 대조합니다.
- 3 기대·실제·재현 순서를 REVIEW.md에 적습니다.

완료 증거

**정상·오류 2종의
검토표**

프롬프트 10 · 복사해서 시작

막히면 브라우저 실행이 불가능하면 사람이 같은 순서를 실행하고 검토자는 결과를 전달받아 기록합니다.

구현자가 반영하고 다시 검사

시연 1분 → 직접 실습 11분 · 완료 증거를 기록합니다.

어디에서 구현자 · 동일 테스트

- 1 프롬프트 06으로 재현된 결함을 고칩니다.
- 2 동일 테스트와 클릭 순서를 반복합니다.
- 3 수정 근거를 검토자에게 전달합니다.

막히면 검토 의견이 서로 충돌하면 완료 기준을 기준으로 사람이 결정합니다.

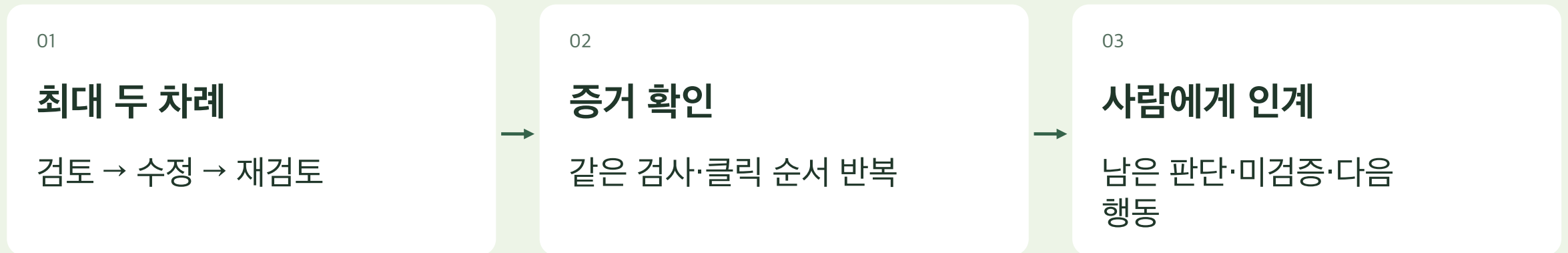
완료 증거

수정 전후 비교와 처리 이유

프롬프트 06 · 복사해서 시작

협업에는 종료 조건이 필요합니다

서로 요청을 계속 보내는 대신 회신 횟수와 최종 결정자를 정합니다.



실전 요청·적용

두 차례 뒤 추가 메시지를 멈추고 남은 결함과 사용량을 요약합니다. 최종 배포는 사람이 결정합니다.

잠깐 확인

두 Claude의 결론이 다르면?

실습으로 →

검토를 종료하고 최종 인계 파일을 작성합니다.

두 번 안에 협업 루프 종료

시연 2분 → 직접 실습 4분 · 완료 증거를 기록합니다.

어디에서 두 세션 · 검토 기록

- 1 최대 두 차례 검토까지만 진행합니다.
- 2 남은 결함과 미검증 항목을 분리합니다.
- 3 추가 메시지를 멈추고 사람이 최종 확인합니다.

막히면 세션을 계속 서로 호출시키지 말고 종료 조건과 남은 일을 기록합니다.

완료 증거

**종료 요약과 남은
판단**

프롬프트 08 · 복사해서 시작

다음 개발자에게 인계 재현

시연 1분 → 직접 실습 7분 · 완료 증거를 기록합니다.

어디에서

새 검토 대화 · 코드·HANDOFF.md·REVIEW.md

완료 화면

증거와 일치하는 최종 인계

01

완료 주장과 실제 검사 기록을 나란히 봅니다.

02

JSON 왕복과 배포의 미검증을 분리합니다.

03

사람이 최종 확인하고 다음 검사 하나를 남깁니다.

막히면

새 AI 세션을 열 수 없으면 짝에게 인계 파일만 주고 첫 검사를 설명하게 합니다.

협업 방식의 비용과 효과 비교

시연 0분 → 직접 실습 5분 · 완료 증거를 기록합니다.

어디에서 단일 세션 · 두 세션 기록

- 1 추가 호출·대기·수정 시간을 비교합니다.
- 2 독립 검토로 발견한 문제를 한 개 적습니다.
- 3 다음 업무에 단일/위임/두 세션 중 하나를 고릅니다.

막히면 작은 수정은 단일 세션이 빠를 수 있습니다. 병렬화 자체를 목표로 삼지 않습니다.

완료 증거

**내 업무에 맞는
선택 기준**

완료하면 채팅에 ✓

결과를 저장하고 다음 도구 선택

시연 0분 → 직접 실습 3분 · 완료 증거를 기록합니다.

어디에서

결과물 · 인계 파일 · 추천 도구

완료 화면

검토한 결과물 + 다음 도구 하나

01

Claude Code:
역할·도구·맥락을 가진 실행

02

VS Code 비교: 변경
범위·프롬프트 버전 검토

03

Playwright: LLM 결과의
브라우저 재현

막히면

결과물·검토·다음 행동을 먼저 저장합니다. 도구 설치·가입은 선택입니다.