바이브 코딩,
끝까지 만드는 법

with 클로드 코드 : 말로 시키고, 눈으로 검수하고, 링크로 자랑하기



말로 시키기 → 눈으로 검수 → 링크로 자랑하기



비개발자를 위한 실전서

김도균

차례

들어가며. 2026년, 바이브 코딩은 '말 잘하는 사람'이 이기는 게임이 됐다

1장. 1년 사이 무엇이 바뀌었나: 자동완성 → 대화 → 에이전트

2장. 도구 지형도: 채팅형 / 앱 빌더형 / AI 에디터 / 코딩 에이전트, 무엇을 언제 쓰나

3장. 누구나 시작하지만 아무나 완성하진 못한다: 성공하는 사람들의 공통점

4장. 비개발자가 꼭 알아야 할 개념 10가지

5장. 30분 만에 웹페이지 만들고 인터넷에 공개하기

6장. 고치고 싶을 때 말하는 법: 수정 요청의 기술

7장. 프롬프트보다 컨텍스트: AI가 '알고 시작하게' 만들기

8장. 기획 문서를 먼저 만든다: 요구사항·화면·데이터를 한 장으로

9장. 프로젝트 규칙 파일 만들기: 매번 같은 설명 반복하지 않기

10장. 작게 쪼개기와 저장: 무한 루프에 빠지지 않는 법

11장. 에이전트형 도구는 무엇이 다른가

12장. 일을 맡기고 결과를 검수하는 법: 사람은 '감독'이 된다

13장. 테스트를 먼저 요구하기: 비개발자도 할 수 있는 검증

14장. 모델 나눠 쓰기: 기획용·구현용·검토용

15장. 콘텐츠형 사이트: 심리테스트·퀴즈

16장. 브라우저 게임: 몰려오는 몬스터를 버티는 액션 게임

17장. AI 기능이 들어간 앱 (API 연동)

18장. 데이터를 저장하고 로그인하는 서비스 (난이도 최상, 주의점 포함)

19장. 공개 전 체크리스트

20장. 버전 관리로 되돌릴 수 있게 하기

21장. 내 아이디어로 끝까지

마치며

부록 A. 상황별 요청문·컨텍스트 템플릿

부록 B. 에러 메시지 사전

부록 C. 공개 전 보안 체크리스트 (출력용)

부록 D. 용어집

부록 E. 터미널에서 쓰는 Claude Code

들어가며. 2026년, 바이브 코딩은 '말 잘하는 사람'이 이기는 게임이 됐다

한 줄의 트윗에서 올해의 단어까지

2025년 2월, AI 연구자 안드레이 카파시(Andrej Karpathy)가 소셜 미디어에 짧은 글을 올렸습니다. 요즘 자기는 새로운 방식으로 코딩을 한다며, 이렇게 적었습니다. "분위기에 완전히 몸을 맡기고, 코드가 존재한다는 사실조차 잊는다."

그는 이 방식을 바이브 코딩(vibe coding)이라고 불렀습니다. 코드를 한 줄씩 짜는 대신, 원하는 것을 AI에게 말로 설명하고 AI가 만든 결과를 보며 계속 주문하는 방식입니다.

반쯤은 농담이었던 이 말은 그해 내내 퍼졌습니다. 2025년 11월에는 영국 Collins 사전이 'vibe coding'을 올해의 단어로 뽑았습니다. 코딩을 배운 적 없는 사람들까지 이 말을 쓸 만큼 널리 퍼졌다는 뜻입니다.

시작은 쉬워졌고, 완성은 어려워졌다

그로부터 1년이 지난 2026년, 분위기는 조금 달라졌습니다.

시작하는 사람은 크게 늘었습니다. 2025년 Stack Overflow 개발자 설문에서 응답자의 84%가 AI 코딩 도구를 이미 쓰고 있거나 쓸 계획이라고 답했습니다.

개발자가 아닌 직장인, 기획자, 1인 사업자도 자기 업무 도구를 직접 만들기 시작했습니다.

그런데 같은 설문에서 AI가 만든 결과의 정확성을 믿는다는 응답은 33%였습니다. 믿지 않는다는 응답은 46%였습니다. 많은 사람이 이 도구를 매일 쓰면서도 결과는 믿지 못하고 있습니다.

AI는 그럴듯하게 돌아가는 결과를 아주 빨리 만들지만, 그 결과가 정말 맞는지, 안전한지, 한 달 뒤에도 고칠 수 있는지는 따로 확인해야 합니다. 보안 회사 Veracode는 2025년에 보안 실수가 생기기 쉬운 과제 80개를 100개가 넘는 AI 모델에게 풀게 했습니다. 그랬더니 45%에서 보안 결함이 있는 코드를 썼습니다(19장).

그래서 만들기 시작하는 사람은 많아도, 믿고 쓸 수 있는 결과물까지 가는 사람은 아직 적습니다.

'말 잘하는 사람'이란

이 책의 제목처럼, 2026년의 바이브 코딩은 말 잘하는 사람이 이기는 게임입니다. 여기서 '말 잘하기'는 말솜씨를 뜻하지 않습니다. 이 책은 아래 세 가지 능력을 그렇게 부릅니다.

1. 설명하는 능력: 원하는 것을 AI가 오해할 수 없게 구체적으로 말한다. "예쁘게" 대신 "흰 배경, 여백 넓게, 포인트 색은 초록 하나"라고 말한다.
2. 맥락을 주는 능력: AI가 알아야 할 배경을 먼저 알려 준다. 누가 쓰는지, 무엇을 이미 만들었는지, 무엇은 건드리면 안 되는지.
3. 판정하는 능력: AI가 "다 됐습니다"라고 말해도 그대로 믿지 않고, 정말 됐는지 확인하는 방법을 안다.

코드를 몰라도 세 가지 모두 연습할 수 있습니다. 이 책은 그 연습을 실제 결과물을 만들면서 하도록 짜여 있습니다.

이 책은 이런 분을 위해 썼습니다

- 코딩을 배운 적 없지만, 생활이나 모임에서 쓸 작은 도구나 사이드 프로젝트를 직접 만들어 보고 싶은 분
- AI로 한두 번 만들어 봤는데, 고치다 보면 엉키고 망가져서 끝까지 못 간 경험이 있는 분
- 개발자에게 맡기기 전에 기획과 시제품(프로토타입)을 직접 만들어 보고 싶은 기획자·마케터
- 가계부나 습관 기록처럼 매일 쓰는 도구를 내 손에 맞게 만들어 보고 싶은 분

반대로, 프로그래밍 언어의 문법을 체계적으로 배우고 싶은 분이나, 이미 개발자로 일하며 대규모 서비스를 운영하는 분께는 맞지 않을 수 있습니다.

책의 구성



그림 0-1 이 책의 지도. 2부, 5부, 6부는 직접 만들어 보는 부이고, 3부와 4부도 실습하면서 2026년에 가장 중요해진 능력을 익힙니다. 6부 뒤에는 마무리 장(21장)과 마치며가 이어집니다.

- 1부. 지금의 바이브 코딩: 1년 사이 무엇이 바뀌었는지, 어떤 도구를 언제 쓰는지, 끝까지 가는 사람들은 무엇이 다른지 살펴봅니다.
- 2부. 첫 결과물: 30분 만에 모임 N빵 정산기를 만들어 인터넷에 공개하고, 원하는 대로 고쳐 달라고 말하는 법을 익힙니다. 처음부터 이론을 배우기보다, 먼저 하나를 완성해 보는 경험부터 하자는 뜻으로 앞에 두었습니다.
- 3부. 컨텍스트 설계: 질문 하나를 잘 쓰는 것을 넘어, AI가 '알고 시작하게' 만드는 법을 다룹니다. 기획 문서와 규칙 파일, 작게 쪼개기가 여기에 들어갑니다.
- 4부. 에이전트에게 일 맡기기: 파일을 직접 고치고 실행까지 하는 에이전트형 도구에게 일을 맡기고, 사람은 감독이 되어 결과를 검수하는 법을 배웁니다.

- 5부. 실전 프로젝트: 심리테스트 사이트, 브라우저 게임, AI 기능이 들어간 앱, 로그인이 있는 서비스까지 네 가지를 직접 만듭니다.
- 6부. 공개하고 지키기: 비개발자가 가장 많이 하는 보안 실수 일곱 가지와 공개 전 체크리스트로 실제 공개까지 해 보고(19장), 망가뜨려도 되돌릴 수 있게 버전 관리와 GitHub를 씁니다(20장).
- 마무리: 내 아이디어 하나로 빈 폴더에서 공개까지 가 보는 장(21장)입니다. 지금까지 배운 것을 한 장짜리 진행표로 묶고, 그대로 끝까지 해 본 예시를 보여 줍니다.

책 끝에는 부록 다섯 가지를 두었습니다. 본문을 다 읽은 뒤에도 곁에 두고 꺼내 쓰는 용도입니다.

- 부록 A. 요청문 템플릿: 이 책에서 쓴 요청문과 컨텍스트 틀을 상황별로 모았습니다.
- 부록 B. 에러 메시지 사전: 자주 만나는 에러 메시지의 뜻과, AI에게 어떻게 말하면 되는지 정리했습니다.
- 부록 C. 공개 전 체크리스트: 만든 것을 인터넷에 공개하기 전에 확인할 항목을 출력해서 쓸 수 있게 담았습니다.
- 부록 D. 용어집: 본문에 나온 말을 쉬운 풀이와 함께 모았습니다.
- 부록 E. 터미널에서 쓰는 Claude Code: 같은 Claude Code를 터미널에서 설치해 쓰는 법과, 앱과 무엇이 다른지 정리했습니다.

입문편에서 실전편으로

이 책은 두 권으로 이어지는 공부의 첫 권, 입문편입니다. 다음 책인 실전편과는 이렇게 이어집니다.



그림 0-2 입문편에서 실전편으로. 이 책에서 순서와 습관을 익히고, 실전편에서 같은 순서로 더 큰 서비스를 만듭니다.

이 책에서는 끝까지 가는 순서를 익힙니다. 원하는 것을 설명하고, 맥락을 주고, 일을 맡기고, 결과를 확인하고, 공개한 뒤 지키는 순서입니다. 이 책만 읽어도 작은 페이지와 앱은 혼자 끝까지 만들어 공개할 수 있습니다.

실전편에서는 같은 순서로 서비스 일곱 개를 하나씩 끝까지 만듭니다. 이 책에서 하지 않은 일도 나옵니다. 외부 서비스를 연결하고, 정해진 시간에 저절로 돌게 하고, 돈을 받고, 공개한 뒤 실제로 쓰는 사람을 지켜보는 일입니다. 어떤 서비스를 만드는지는 이 책 끝의 '마치며'에 있습니다.

두 책은 같은 Claude 앱의 Code 탭을 쓰고, 실전편도 폴더 만들기부터 다시 설명합니다. 그래도 이 책을 먼저 끝까지 따라 해 보고 넘어가기를 권합니다.

실전편은 이 책에서 익힌 순서를 안다고 보고, 그 위에 새로운 것을 더하기 때문입니다.

이 책을 활용하는 법

처음이라면 순서대로 읽으세요. 앞에서 익힌 개념과 요청법을 뒤에서 그대로 씁니다. 다만 개념 설명보다 손으로 해 보는 것이 먼저 필요하다면, 1부를 훑어본 뒤 바로 2부 5장부터 시작해도 됩니다.

책과 작업 화면을 나란히 띄워 두고 따라 하세요. 한쪽에는 이 책을, 다른 쪽에는 Claude 앱을 띄우면 요청문을 복사해 붙여 넣기 쉽습니다. 휴대폰이나 태블릿으로 책을 보면서 PC에서 따라 해도 됩니다.

실습이 있는 장은 처음에 준비물을 알려 주고, 앞쪽 실습 장(5~10장)은 예상 시간도 함께 적었습니다. 장 끝에는 체크리스트가 있습니다. 체크리스트를 다 채우면 다음 장으로 넘어가세요.

실습에는 한 사람의 예시가 계속 등장합니다. '퇴근 후 코딩을 배우는 마케터' 김하늘 씨입니다. 요청문과 상황은 김하늘 씨를 예로 들었고, 그 요청문을 실제로 실행하고 결과를 판정한 사람은 저입니다. 여러분은 이름과 내용을 자기 것으로 바꿔 따라 하면 됩니다.

Claude 앱 하나로 따라 합니다. 채팅으로 만드는 단계(5~8장)에서는 앱의 채팅을, 에이전트에게 맡기는 단계(9장부터)에서는 같은 앱의 Code 탭(Claude Code)을 씁니다. 터미널에 명령어를 칠 일은 거의 없습니다. 터미널에서 쓰는 Claude Code가 궁금하다면 부록 E에서 따로 다룹니다. 2장에서 여러 도구의 지형을 비교한 뒤로는 Claude를 중심으로 설명합니다. 다른 도구를 쓰더라도 설명하고, 맥락을 주고, 판정하는 원리는 같습니다.

화면은 2026년 11월 기준입니다. AI 도구는 버튼 이름과 메뉴 위치가 자주 바뀝니다. 그래서 본문에는 바뀌지 않는 원리를 썼습니다. 버튼 이름과 메뉴 경로처럼 바뀌기 쉬운 것은 도구 화면을 따라 하는 장의 끝에 있는 '도구 따라 하기' 표에 모았습니다. 화면이 책과 다르게 보이면 그 표부터 확인하세요.

일러두기

- 김하늘: 가상의 예시 인물입니다. 이 책의 실습은 모두 제가 김하늘 씨의 입장에서 직접 실행하고, 결과가 맞는지 직접 판정했습니다. 본문에서 '저'는 이 책을 쓴 저를 가리킵니다.
- 실습 환경: Windows 11, Claude 앱 4.23118(Windows), Claude Ultra 요금제(예전 이름 Max 20배)에서 실습했고, 화면은 2026년 11월 기준입니다. 요금제와 운영체제에 따라 화면이 조금 다를 수 있습니다. 실습은 Ultra로 했지만, 가장 싼 유료 요금제(Pro)로 따라 할 수 있게 큰 실습은 단계로 나누고, 실습이 긴 16·17·19·20장의 끝에는 '이 장의 사용량'을, 21장의 끝에는 '이 장의 세션'을 적었습니다(5장 '모델과 사용량'). Mac에서 크게 다른 곳은 'Mac에서는' 상자로 따로 표시했습니다.
- 채팅 실험: 3·6·7·8장의 채팅 실험은 2026년 10월에 claude.ai 웹에서 했습니다. 웹의 채팅 화면은 앱의 채팅 탭과 거의 같습니다.
- '예전에 같은 요청을 해 봤을 때는': 뒤쪽 장에 나오는 이 말은, 이 책을 준비하면서 2026년 10월에 먼저 해 본 실험을 가리킵니다.
- 모델 이름: 이 책에 나오는 AI 모델 이름은 2026년 11월 기준입니다. 이름이 바뀌었다면 이름 대신 등급을 보고 고르세요. 이 책에서 큰 모델을 쓴 곳에는 그때의 큰 모델을, 작은 모델을 쓴 곳에는 그때의 작은 모델을 고르면 됩니다.

- 업데이트·정오표·예제 파일: 책이 나온 뒤 바뀐 화면, 책에서 틀린 곳, 장별 예제 파일은 https://github.com/vibecoding-book/vibecoding-book에 모아 둡니다. 화면이 책과 다르면 이곳부터 확인하세요.

따라 하는 데 드는 비용

범위	쓰는 도구	비용
1~8장	Claude 앱의 채팅	무료 요금제로 시작해 볼 수 있습니다. 다만 무료 요금제로 끝까지 되는지는 확인하지 않았습니다.
9장부터	Claude 앱의 Code 탭 (Claude Code)	Claude Pro 이상이 필요합니다. 2026년 11월 claude.com/pricing 기준 Pro는 월 20달러, 연 결제 시 월 17달러입니다(부가세 별도. 한국어 페이지의 부가세 포함 가격은 월 22달러, 연 결제 시 월 18달러).
17장	Claude API	구독료와 별도로, 미리 충전해 쓴 만큼 냅니다.
5장 등 Netlify, 18장 Supabase, 20장 GitHub	올려 두는 곳	이 책의 실습은 모두 무료 한도 안이었습니다 (19장).

요금표는 책보다 빨리 바뀝니다. 가입 전에는 공식 요금 페이지를 다시 확인하세요. 구독 사용량 한도는 5장, 공개해 두는 데 드는 돈은 19장에서 다룹니다.

이 책의 표기

> 이 장의 원칙: 그 장에서 가장 중요한 한 가지를 담은 박스입니다.

> 주의: 놓치면 시간이나 돈, 데이터를 잃을 수 있는 내용입니다.

> 팁: 몰라도 되지만 알면 편한 내용입니다.

> Mac에서는: Windows와 Mac의 차이가 큰 곳(실행 방법, 단축키 등)을 따로 적은 박스입니다.

- AI에게 보내는 요청문은 회색 상자에 담았습니다. 그대로 복사해서 내 내용으로 바꿔 쓰면 됩니다.
- 그림 번호는 '그림 5-6'처럼 장 번호와 순서로 붙였습니다.

이제 시작하겠습니다. 1장에서는 바이브 코딩이 지난 몇 년 사이 어떻게 바뀌어 왔는지부터 살펴봅니다. 그 흐름을 보면 이 책이 코딩 대신 말하기를 연습시키는 이유가 보일 겁니다.

참고 문헌

- 카파시의 'vibe coding' 게시글 (2025-02-02): Andrej Karpathy, X 게시글 (2026년 10월 27일 확인)

- Collins 사전 2025 올해의 단어: CNN 보도 (2025-11-06) (2026년 10월 27일 확인)
- Stack Overflow Developer Survey 2025 (AI): survey.stackoverflow.co/2025/ai (2026년 10월 27일 확인) — 사용 중이거나 사용 계획 84%, 정확성 신뢰 33%, 불신 46%
- Veracode 2025 GenAI Code Security Report: 보도자료 (2026년 10월 27일 확인) — 100개 이상 모델, 80개 과제, 45%에서 OWASP Top 10 유형 취약점
- Claude 요금제: claude.com/pricing (2026년 10~11월 확인, 5장 참고)
- 인용문은 제가 옮겼습니다. 원문: "fully give in to the vibes, embrace exponentials, and forget that the code even exists"

5장. 30분 만에 웹페이지 만들고 인터넷에 공개하기

이 장에서 할 일

이 장이 끝나면 회식이나 여행에서 쓴 돈을 넣으면 누가 누구에게 얼마를 보낼지 알려 주는 '모임 N빵 정산기'가 진짜 인터넷 주소로 열립니다. 코드는 한 줄도 직접 쓰지 않고, 걸리는 시간은 30분 안팎입니다.

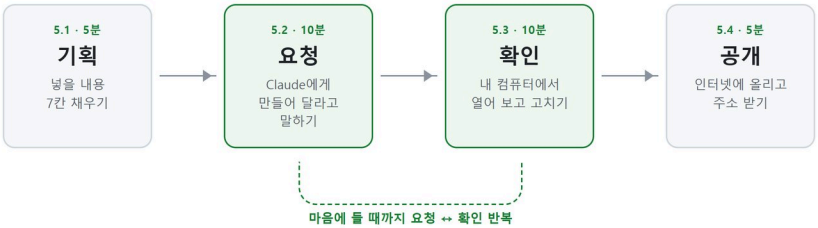


그림 5-1 5장 전체 흐름. 요청과 확인은 한 번에 끝나지 않고 몇 번 오갑니다. 마음에 들면 그때 공개로 넘어갑니다.

이 장의 결과물

- 한 장짜리 정산 웹페이지: 함께한 사람, 쓴 돈, 한 사람당 금액, 보낼 사람과 금액, 카카오톡용 복사 버튼
- 누구에게나 보낼 수 있는 인터넷 주소: 예) `https://내사이트이름.netlify.app`

준비물

준비물	어디에 쓰나	비용
컴퓨터 (Windows 또는 Mac)	파일 저장과 확인	-
크롬 또는 엣지 브라우저	결과 확인, 인터넷에 올리기	무료
Claude 앱과 계정	웹페이지 만들기 (5.2에서 설치·가입)	앱은 무료, 무료 요금제로 시작 가능 (사용량 제한 있음)
이메일 주소	Netlify 가입 (비밀번호 없이 공개하고, 사이트를 계속 유지하려면 필요)	무료

시간 배분

단계	하는 일	예상 시간
5.1 기획	페이지에 넣을 내용 정하기	5분
5.2 요청	Claude 앱 설치·로그인, 만들어 달라고 말하기	15분 (이미 설치했다면 5분)
5.3 확인	내 컴퓨터에서 열어보고 한 번 고치기	10분

단계	하는 일	예상 시간
5.4 공개	인터넷에 올리고 계정에 연결해 주소 받기	10분

5.4의 7~8단계(이름 바꾸기, 수정본 다시 올리기)는 선택 과정이라 위 시간에 넣지 않았습니다. 해 보면 10분쯤 더 걸립니다.

> 이 장의 원칙: 잘 만들기보다 끝까지 가기. 디자인이 마음에 들지 않아도 일단 공개까지 한 번 가 보세요. 한 번 끝까지 가 보면 뒤의 장이 훨씬 쉬워집니다. 더 예쁘게 고치는 법은 6장에서 따로 다룹니다.

5.1 무엇을 만들까: 모임 N빵 정산기 기획하기

회식이 끝나면 누군가는 영수증을 모아 계산기를 두드립니다. 1차는 하늘이, 2차는 민수가, 택시는 지영이 냈다면 누가 누구에게 얼마를 보내야 할까요? 이 장에서는 이 계산을 대신해 주는 페이지를 만들어 인터넷에 올립니다. 만들고 나면 다음 모임에서 바로 쓸 수 있습니다.

AI에게 말하기 전에 넣을 내용을 5분만 정리하면, 결과물이 훨씬 내가 원하는 것에 가까워집니다.

왜 기획부터 할까

AI는 모르는 칸을 그럴듯한 가짜로 채웁니다. "N빵 계산기 만들어줘"라고만 하면 총액을 사람 수로 나누는 칸 하나만 나오거나, 내가 원하지 않던 기능이

잔뜩 붙어 나옵니다. 그러면 나중에 하나씩 다시 고쳐야 해서 오히려 시간이 더 듭니다.

아래 기획 시트의 '내 답' 칸을 직접 채워 보세요. 잘 쓰려고 애쓰기보다 빈칸만 채우면 됩니다.

기획 시트

항목	스스로에게 물어볼 질문	작성 예시	내 답
제목	페이지 맨 위에 무엇이라고 쓸까?	모임 N빵 정산기	
누가 쓰나	어떤 자리에서 쓸까?	회식이나 여행이 끝난 뒤, 돈을 낸 사람이	
넣을 것	사용자가 무엇을 적나?	함께한 사람 이름, 쓴 돈(어디에, 누가, 얼마)	
보여 줄 것	무엇을 알려 주면 되나?	한 사람당 금액, 누가 누구에게 얼마 보낼지	
예시	처음 열었을 때 들어 있을 예시는?	1차 고기집(하늘, 128,000원), 2차 노래방(민수, 40,000원), 택시(지영, 18,000원)	
분위기	페이지를 본 사람이 어떤 느낌을 받으면 좋을까?	깔끔하고 차분한	
포인트 색	딱 하나만 고른다면 무슨 색?	초록	

예시 숫자는 내가 손으로 검산할 수 있게 고르는 것이 좋습니다. 위 예시는 합계가 186,000원이라 4명이면 한 사람당 46,500원이 됩니다. AI가 만든 결과가 이 숫자와 같은지 5.2에서 직접 맞춰 봅니다.

분위기를 AI가 알아듣는 말로 바꾸기

'예쁘게', '세련되게'는 사람마다 떠올리는 그림이 다릅니다. 눈에 보이는 요소(배경색, 여백, 글자 크기, 모서리)로 바꿔 말하면 원하는 결과가 훨씬 잘 나옵니다.

원하는 느낌	AI에게 이렇게 말하세요
깔끔한	흰 배경, 여백 넓게, 글자는 검정과 회색, 포인트 색은 하나만
따뜻한	베이지톤 배경, 둥근 모서리, 부드러운 갈색 글자
개성 있는	진한 배경색, 아주 큰 제목 글자, 강한 포인트 색 하나
전문적인	남색 계열, 정돈된 정렬, 섹션마다 구분선

> 주의: 공개되면 누구나 봅니다. 이 페이지는 5.4에서 인터넷에 올라갑니다. 예시로 넣는 이름은 별명으로 충분합니다. 실제 계좌번호나 휴대폰 번호처럼 남이 보면 안 되는 정보는 요청문에 넣지 마세요. (보안은 6부에서 자세히 다룹니다.)

5.2 AI에게 첫 요청 보내기

이 책은 처음부터 끝까지 Claude 앱 하나로 만듭니다. 이 장에서는 앱의 채팅(Chat)으로 웹페이지를 만들고, 9장부터는 같은 앱의 Code 탭으로 넘어갑니다. 도구가 하나라 화면이 바뀌어도 헷갈리지 않습니다.

먼저 앱을 설치하고 로그인한 뒤, 요청을 쓰는 칸을 찾습니다. 이미 앱을 쓰고 있다면 '요청문의 4가지 재료'로 건너뛰세요.

Claude 앱 설치하고 로그인하기

제 PC는 이미 설치·로그인된 상태라, 이 단계는 공식 안내를 따라 적었습니다.

1. 크롬이나 엣지 주소창에 `claude.com/download`를 입력하고 Enter를 누릅니다.
2. 그림 5-2의 내려받기 화면에서 'Windows용 다운로드'를 누릅니다. Mac이라면 아래쪽 '데스크톱' 칸의 macOS '다운로드'를 누릅니다.
3. '다운로드' 폴더에 받아진 설치 파일을 더블클릭해 설치합니다.
4. 설치가 끝나면 Claude 앱이 열립니다. 나중에는 시작 메뉴(Mac은 응용 프로그램 폴더)에서 'Claude'를 찾아 엽니다.
5. 로그인 화면에서 구글 계정, 애플 계정, 이메일 가운데 편한 방법으로 로그인합니다. 처음이면 여기서 가입까지 됩니다. 로그인하는 동안 브라우저가 열렸다가 다시 앱으로 돌아올 수 있습니다.



그림 5-2 claude.com/download. 대부분의 Windows 컴퓨터는 'Windows용 다운로드'를 고르면 됩니다.

다음부터는 앱을 열면 로그인된 채로 시작합니다.

> 팁: 앱을 설치할 수 없다면 웹에서. 회사 컴퓨터처럼 프로그램을 설치할 수 없다면, 브라우저 주소창에 `claude.ai`를 입력해 같은 계정으로 로그인하면 됩니다. 1~8장의 채팅 화면은 웹과 앱이 거의 같습니다. 다만 9장부터 쓰는 Code 탭은 앱에서 써야 내 컴퓨터의 폴더를 다룰 수 있습니다.

요금제: 무료로 시작하고, 필요하면 결제

가입하면 무료(Free) 요금제로 시작합니다. 이 장의 실습은 무료 요금제로 시작해 볼 수 있습니다(무료 요금제로 끝까지 해 보지는 않았습니다). 다만 무료 요금제는 쓸 수 있는 양이 적어서, 요청을 여러 번 주고받다 보면 "한도에 도달했다"는 안내가 나오고 몇 시간 기다려야 할 수 있습니다.

9장부터 쓰는 Code 탭(Claude Code)은 유료 요금제(Pro 이상)가 있어야 합니다. 요금제는 주소창에 `claude.com/pricing`을 입력하면 볼 수

있습니다.

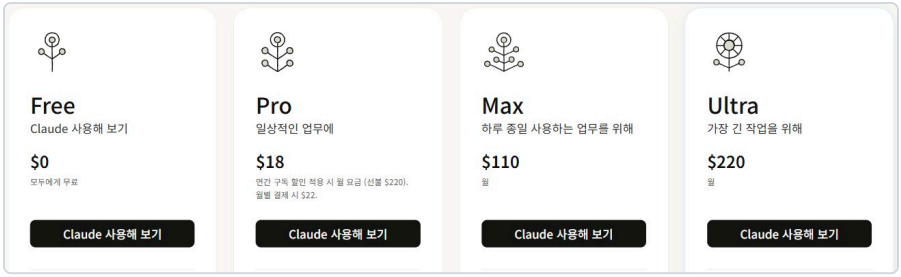


그림 5-3 요금제 페이지(2026년 11월). 한국어 페이지는 부가세 10%가 포함된 가격을 보여줍니다.

요금제	가격(2026년 11월, 부가세 포함)	이런 분께
Free	무료	1~8장을 해 보며 맞는지 알아보려는 분
Pro	연간 결제 시 1년 $220(월 약 $18), 월별 결제 시 $22	9장부터 Code 탭으로 계속 만들 분
Max	월 $110	매일 오래 쓰는 분. 사용량이 Pro의 5배
Ultra	월 $220	하루 종일 쓰는 분. 주간 사용량이 Pro의 10배

2장의 요금표는 부가세 별도 가격이라 숫자가 조금 다릅니다. 실제로 결제되는 금액은 이 페이지의 부가세 포함 가격입니다.

유료 요금제로 바꾸려면 이렇게 합니다.

1. `claude.com/pricing`에서 고른 요금제 칸의 버튼을 누릅니다. Pro는 연간 결제와 월별 결제 가운데 고를 수 있습니다. 연간은 1년 치를 한 번에

내는 대신 더 쌉니다.

2. 결제 화면에서 카드 정보를 넣고 구독합니다. 요금이 달러(USD)로 청구되니 해외 결제가 되는 카드를 쓰세요.

저는 이미 유료 요금제를 쓰고 있어서 결제 화면은 찍지 않았습니다. 처음에는 무료로 시작하고, 한도에 자주 걸리거나 9장에 닿았을 때 바꿔도 늦지 않습니다. 요금은 자주 바뀌니 결제 전에 이 페이지에서 다시 확인하세요.

모델과 사용량: 이 책은 Pro로 따라 할 수 있게

Claude에는 실력과 속도가 다른 모델이 몇 가지 있습니다. 입력칸 아래 오른쪽의 모델 이름을 누르면 고를 수 있습니다(2026년 11월 기준).

모델	특징	이 책에서 쓰는 곳
Opus	생각이 깊고 꼼꼼합니다. 대신 사용량을 많이 씁니다	검토, 어려운 버그 찾기
Sonnet	실력과 사용량이 중간입니다. 대부분의 만들기에 충분합니다	16장부터 만들기
Haiku	빠르고 사용량을 적게 씁니다. 간단한 일에 맞습니다	14장의 작은 기능 만들기
Fable	가장 상위 모델입니다. Pro에서는 추가 결제(사용량 크레딧)로만 씁니다	쓰지 않음

무료 요금제에서는 Sonnet과 Haiku만 고를 수 있습니다. Pro부터 Opus도 쓸 수 있습니다.

유료 요금제라도 무제한은 아닙니다. 두 가지 한도가 함께 걸립니다.

- 5시간 한도: 처음 요청을 보낸 때부터 5시간 동안 쓸 수 있는 양입니다. 다 쓰면 5시간이 지나 다시 채워질 때까지 기다립니다.
- 주간 한도: 일주일 동안 쓸 수 있는 전체 양입니다. 5시간마다 채워져도 이 양을 넘을 수는 없습니다.

채팅과 Code 탭은 같은 한도를 나눠 씁니다. 채팅을 많이 하면 Code 탭에서 쓸 양이 줄어듭니다. 같은 일이라도 Opus는 Sonnet보다, Sonnet은 Haiku보다 한도를 빨리 씁니다. 그리고 대화가 길어질수록 Claude가 앞의 내용을 매번 다시 읽기 때문에 한 번 요청할 때 드는 양이 늘어납니다.

지금까지 얼마나 썼는지는 왼쪽 아래 내 이름을 누르고 '사용량'을 고르면 나오는 설정의 사용량 화면에서 볼 수 있습니다. 이번 주 사용량과 초기화 시각, 제품별(Claude Code, 채팅) 사용량이 보입니다. 9장부터 쓰는 Code 탭에서는 입력칸 근처의 '사용량' 단추에서 '주간 · 모든 모델 44%'처럼 바로 보입니다. 한도에 가까워지면 앱이 먼저 알려 줍니다. 한도에 닿으면 세 가지 중에 고릅니다.

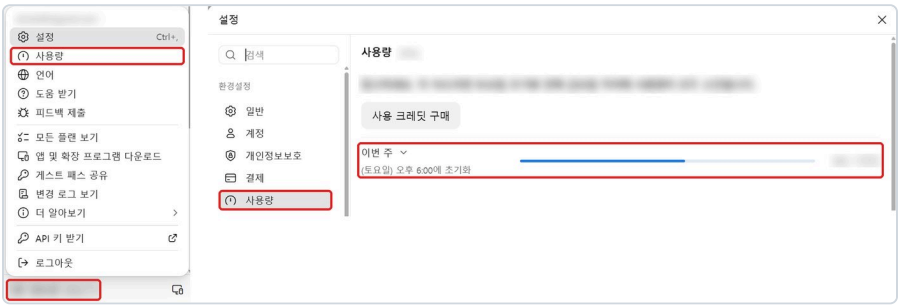


그림 5-4 왼쪽 아래 내 이름 → '사용량'. 이번 주에 얼마나 썼는지와 언제 다시 채워지는지가 보입니다.

1. 다시 채워질 때까지 기다리기. 가장 흔한 방법입니다.

2. 사용량 크레딧을 켜고 쓴 만큼 더 내기. 설정의 사용량 메뉴에서 켜고 끕니다.
3. 더 큰 요금제(Max)로 바꾸기.

> 주의: API 크레딧 안내는 다른 요금입니다. 한도에 닿았을 때 'Console API 크레딧'으로 계속하라는 안내가 나올 수 있습니다. 이것은 요금제와 따로, 쓴 만큼 API 가격으로 청구됩니다. 무엇인지 모르면 누르지 마세요.

저는 Ultra 요금제(예전 이름 Max 20배)로 실험했습니다. '20배'는 예전 이름에 붙은 숫자이고, 지금 Ultra의 주간 한도는 Pro의 10배입니다. 제가 쉽게 한 실험이 Pro에서는 한도에 걸릴 수 있어서, 이 책은 Pro로 따라 할 수 있게 맞췄습니다.

- 큰 프로젝트는 한 번에 맡기지 않고 단계로 나눕니다. 단계마다 새 대화(9장부터는 새 세션)를 엽니다.
- 만들기는 Sonnet, 꼼꼼히 봐야 하는 검토만 Opus를 씁니다.
- 16장부터 실습이 긴 장의 끝에는 '이 장의 사용량'을 적었습니다. 단계마다 Claude가 쓴 양(출력 토큰)과 걸린 시간입니다. 토큰이 Pro 한도의 몇 %인지는 공식 기준이 없어서, 단계끼리, 모델끼리 비교하는 데 씁니다. 내 한도에서 얼마나 쓰는지는 단계 전후에 '사용량' 단추를 보면 압니다.

사용량을 아끼는 요령은 14장과 부록 A.7에 더 모아 두었습니다. 요금제와 한도는 자주 바뀌니 `claude.com/pricing`에서 다시 확인하세요.

첫 화면 둘러보기: 요청은 어디에 쓰나

로그인하면 인사말 아래에 큰 입력칸이 있는 화면이 나옵니다(그림 5-5). 이 입력칸에 요청을 씁니다. 처음 보면 낯설지만 쓸 곳은 몇 군데뿐입니다.

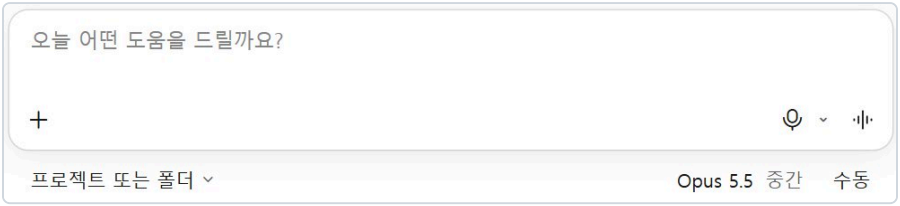


그림 5-5 새 대화의 입력칸. 여기에 요청을 쓰거나 붙여 넣습니다.

- 입력칸: 가운데 큰 칸입니다. 여기를 누르고 요청을 쓰거나 붙여 넣습니다.
- '+' 버튼: 입력칸 왼쪽 아래입니다. 사진이나 파일을 함께 보낼 때 씁니다. 6장에서 써 봅니다.
- 모델 이름: 입력칸 아래 오른쪽의 'Opus 5.5' 같은 글자는 지금 대답할 AI 모델입니다. 요금제에 따라 다르게 보일 수 있습니다. 모델마다 다른 점은 바로 앞의 '모델과 사용량'에 있습니다. 이 장에서는 그대로 두면 됩니다.
- '프로젝트 또는 폴더'와 '수동': 입력칸 아래에 있는 이 두 가지도 이 장에서는 그대로 둡니다(7장).
- 왼쪽 위 버튼들: 네모 버튼은 지난 대화 목록을 열고 닫습니다. 그 오른쪽의 말풍선과 `</>` 두 아이콘은 채팅과 Code 탭을 바꾸는 버튼입니다. 이 장에서는 말풍선(채팅)을 씁니다. 화면이 다르게 보이면 말풍선 아이콘을 누르세요.

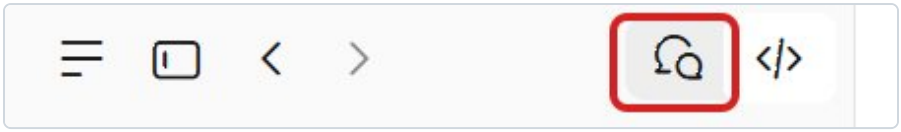


그림 5-6 창 왼쪽 위의 버튼들. 네모 버튼은 지난 대화 목록을 열고 닫고, 빨간 상자의 말풍선은 채팅, 그 오른쪽 `</>`는 Code 탭입니다.

요청을 보내는 방법은 이렇습니다.

1. 입력칸을 한 번 누릅니다.
2. 요청을 쓰거나, 복사해 둔 요청문을 붙여 넣습니다(Ctrl+V, Mac은 Cmd+V).
3. 글을 넣으면 입력칸 오른쪽 아래에 주황색 화살표 버튼이 생깁니다. 이 버튼을 누르거나 Enter를 누르면 전송됩니다.

> 주의: 줄을 바꾸려면 Shift+Enter. 입력칸에서 Enter만 누르면 쓰던 중이라도 바로 전송됩니다. 직접 여러 줄을 쓸 때는 Shift+Enter로 줄을 바꾸세요. 여러 줄을 한꺼번에 붙여 넣는 것은 괜찮습니다.

요청문의 4가지 재료

이제 무엇을 쓸지 정할 차례입니다. 좋은 요청은 네 가지를 담습니다. 무엇을 만들지, 어떤 내용을 넣을지, 어떤 모양이어야 하는지, 어떤 형태로 받을지입니다.

재료	의미	이번 실습에서는
1. 목표	무엇을, 누구를 위해 만드나	모임에서 나눠 낸 돈을 정산해 주는 한 장짜리 페이지

재료	의미	이번 실습에서는
2. 내용	페이지에 들어갈 칸과 결과	5.1 기획 시트의 '내 답'
3. 모양	느낌, 색, 화면 크기	분위기 표현 + 휴대폰에서도 보기 좋게
4. 받는 형태	결과물을 어떤 파일로 받을지	HTML 파일 하나, 이름은 index.html

초보자가 가장 많이 빠뜨리는 것이 4번입니다. 형태를 정해주지 않으면 AI가 디자인 파일, 기능 파일을 따로 나눠 줄 때가 있습니다. 파일이 여러 개면 저장하고 올리는 과정에서 엉키기 쉽습니다. 그래서 첫 실습은 '파일 하나에 다 넣어 달라'고 못박아 둡니다.

나쁜 요청과 좋은 요청

나쁜 요청

```
N빵 계산기 만들어줘
```

결과: 총액을 사람 수로 나누는 칸 하나만 나오거나, 원하지 않던 기능이 붙어 나옵니다. 디자인은 AI 마음대로이고, 파일이 여러 개로 나뉠 수도 있습니다.

좋은 요청 (복사해서 내 내용으로 바꿔 쓰세요)

```
모임 정산 웹페이지를 만들어줘.

[목표]
- 회식이나 여행에서 여러 사람이 나눠 낸 돈을 넣으면, 누가 누구에게 얼마를 보내면 되는지 알려 주는 한 장짜리 페이지

[들어갈 내용]
- 제목: 모임 N빵 정산기
- 함께한 사람 이름을 넣는 칸 (처음에는 하늘, 민수, 지영, 준호)
- 쓴 돈을 한 줄씩 넣는 칸: 어디에 썼는지, 누가 냈는지, 얼마인지
- 처음 열었을 때 예시로 들어 있을 것: 1차 고기집(하늘, 128,000원), 2차 노래방(민수, 40,000원), 택시(지영, 18,000원)
- 결과: 한 사람당 얼마인지, 그리고 "준호 → 하늘 46,500원"처럼 누가 누구에게 얼마를 보내면 되는지. 보내는 횟수는 되도록 적게
- 결과를 카카오톡에 붙여 넣을 수 있게 복사하는 버튼

[디자인]
- 깔끔한 느낌: 흰 배경, 여백 넓게, 포인트 색은 초록 하나
- 휴대폰에서도 보기 좋게

[결과물 형태]
- HTML 파일 하나에 디자인(CSS)과 기능까지 모두 넣어줘
- 외부 파일이나 이미지 없이 바로 열리게 해줘
- 파일 이름은 index.html로 해줘
```

대괄호 [ ]로 묶은 제목은 꼭 필요한 것은 아닙니다. 다만 이렇게 나눠 쓰면 AI도, 나도 빠진 게 없는지 한눈에 확인할 수 있습니다. 8장에서 만들 '기획서'의 아주 작은 버전이라고 생각하면 됩니다.

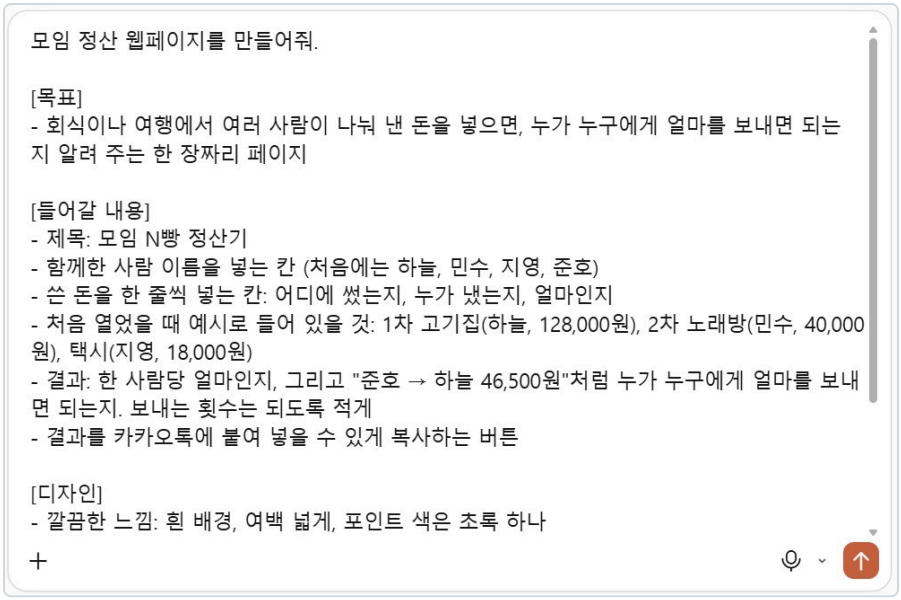


그림 5-7 입력칸에 요청문을 붙여 넣은 모습. 오른쪽 아래 주황색 화살표 버튼을 누르면 전송됩니다.

결과가 돌아오면 확인할 것

제 화면에서는 약 2분 걸렸습니다.

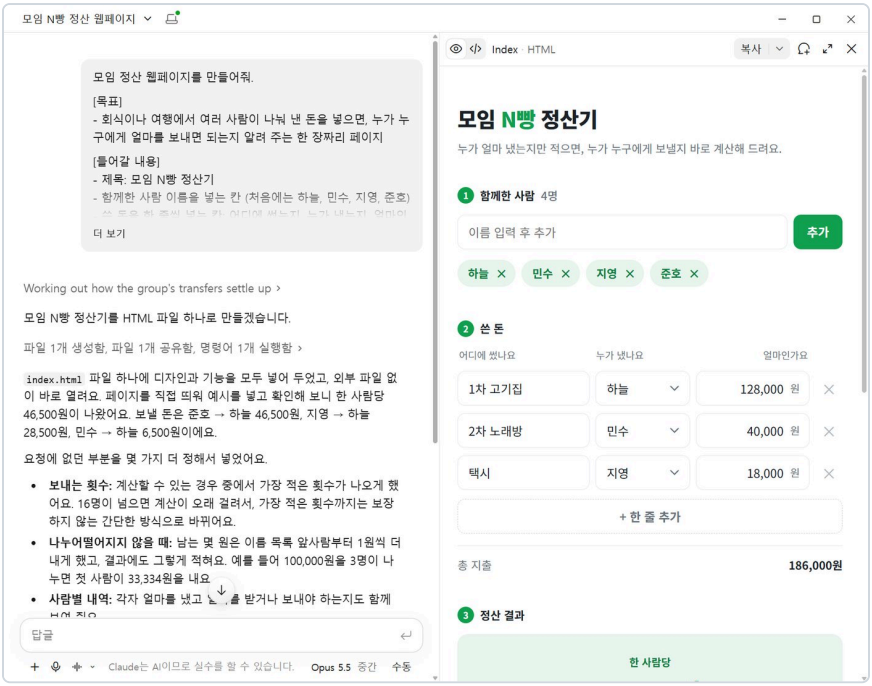


그림 5-8 답변 안에 파일 카드가 생기고, 오른쪽 창에 완성된 페이지가 미리 보입니다.

1. 미리보기가 보이나? Claude는 만든 페이지를 대화창 오른쪽에 바로 보여줍니다. 오른쪽 창이 닫혀 있다면 답변 안의 파일 카드('Index')를 누르면 열립니다.
2. 숫자가 맞나? 기획 시트에서 미리 계산해 둔 숫자와 맞춰 봅니다. 한 사람당 46,500원이고, 하늘이 128,000원을 냈으니 81,500원을 받아야 합니다. 준호(0원)는 46,500원, 지영(18,000원)은 28,500원, 민수(40,000원)는 6,500원을 보내면 46,500 + 28,500 + 6,500 = 81,500원으로 맞습니다. Claude도 "예시를 넣고 확인해 보니"라며 같은 숫자를 적었지만, 내 손으로 한 번 더 맞춰 보는 것이 이 책에서 계속 쓸 습관입니다.
3. AI가 정한 것이 있나? 답변에 "요청에 없던 부분을 몇 가지 더 정해서 넣었어요"라는 목록이 있었습니다. 금액이 사람 수로 나누어떨어지지 않을

때는 앞사람부터 1원씩 더 내게 했다는 것, 사람별로 얼마를 냈고 받거나 보내야 하는지도 보여 준다는 것 등입니다. 마음에 들지 않으면 바꿔 달라고 하면 됩니다. 이런 목록을 그냥 넘기지 말고 읽어 보세요.

4. 파일로 받을 수 있나? 미리보기 창 위쪽 '복사' 버튼 '옆의 ▾ 화살표'를 누르면 'HTML 형식으로 다운로드' 메뉴가 나옵니다. 코드만 길게 나오고 파일 카드가 없다면, "다운로드할 수 있게 index.html 파일로 만들어줘"라고 한 번 더 말하세요.

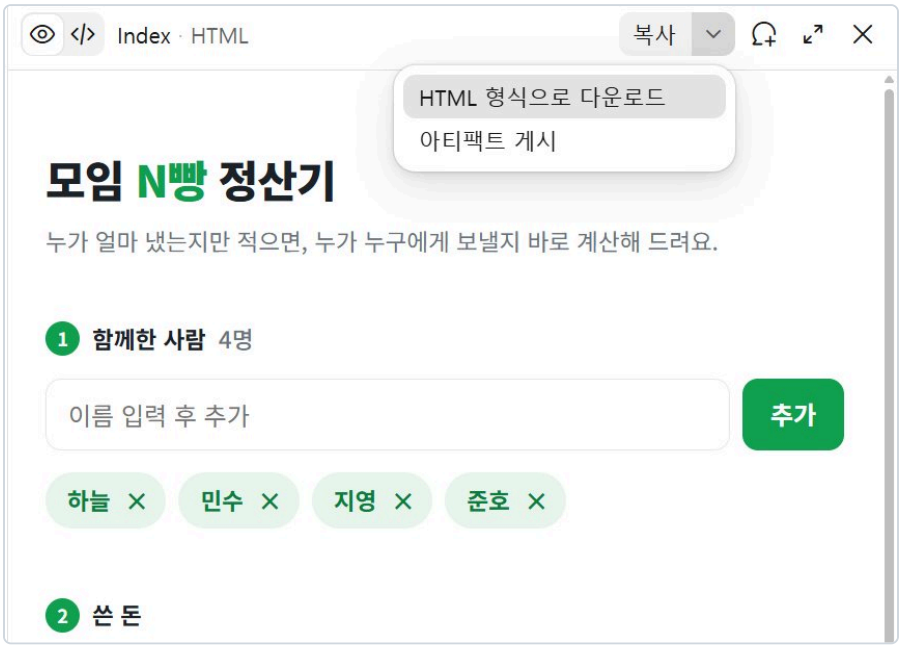


그림 5-9 '복사' 옆 ▾ → 'HTML 형식으로 다운로드'. 앱에서는 이어서 저장할 곳을 묻는 창이 뜹니다(5.3).

> 팁: 마음에 안 들어도 일단 받으세요. 첫 결과물은 초안입니다. 색이나 배치가 어색해도 파일로 받아서 다음 단계로 넘어가세요. 5.3에서 실제로 열어본 뒤에 고치는 것이 훨씬 정확합니다.

5.3 내 컴퓨터에서 열어보기

받은 파일을 전용 폴더에 index.html이라는 이름으로 저장하고
더블클릭하면, 인터넷에 올리기 전에도 내 컴퓨터에서 바로 열립니다.

따라 하기: 파일 저장부터 열기까지

1. 바탕화면에 새 폴더 만들기
   - 이름은 `my-first-site`처럼 영문 소문자와 하이픈(-)만 쓰세요.
   - 한글이나 띄어쓰기가 들어간 폴더 이름은 뒤에서 배울 개발 도구나 명령어에서 오류를 일으키기 쉽습니다. 처음부터 영문으로 짓는 습관을 들여 두세요. (이 장에서 쓰는 Netlify Drop은 폴더 이름과 상관없이 주소를 정해 줍니다.)
2. Claude에서 파일 저장하기
   - 그림 5-9의 'HTML 형식으로 다운로드'를 누르면 '다른 이름으로 저장' 창이 뜹니다.
   - 왼쪽에서 바탕화면을 누르고 1에서 만든 `my-first-site` 폴더를 연 다음, 파일 이름이 `index.html`인지 보고 '저장'을 누릅니다.
3. (웹에서 받았다면) 파일을 my-first-site 폴더로 옮기기

- 웹(claude.ai)에서 받으면 저장 창 없이 '다운로드' 폴더에 들어갑니다. 그 파일을 `my-first-site` 폴더로 옮기세요.

4. 파일 이름이 정확히 index.html인지 확인하기
   - `index (1).html`, `index.html.txt`처럼 되어 있다면 `index.html`로 고칩니다.
   - 파일 이름이 'index'로만 보이고 끝의 '.html'이 안 보인다면 확장자 표시가 꺼져 있는 것입니다. Windows는 처음 설치했을 때 이 상태입니다.
5. 더블클릭해서 열기
   - 크롬이나 엣지가 열리면서 정산기가 보이면 성공입니다. 이름 칸이나 금액을 하나 바꿔 보세요. 결과가 바로 다시 계산되면 기능까지 제대로 들어간 것입니다.

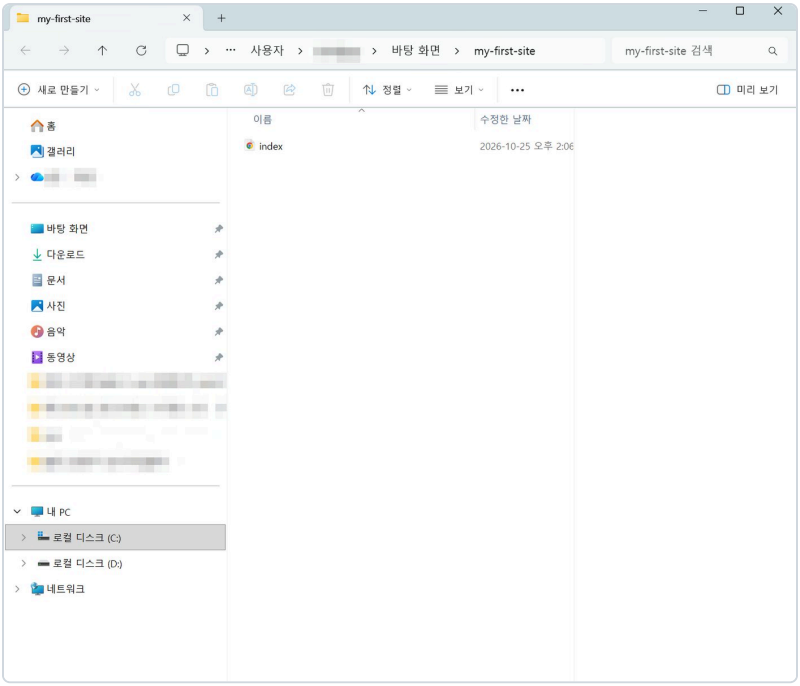


그림 5-10 확장자가 숨겨진 기본 상태. 파일이 'index'로만 보여서 진짜 이름이 index.html인지 index.html.txt인지 알 수 없습니다.

확장자 보이게 하기 (Windows 11)

파일 이름 끝을 확인하려면 확장자를 한 번 켜 두세요. 한 번만 바꾸면 계속 유지됩니다.

탐색기 위쪽 '보기 → 표시 → 파일 확장명'을 누릅니다.

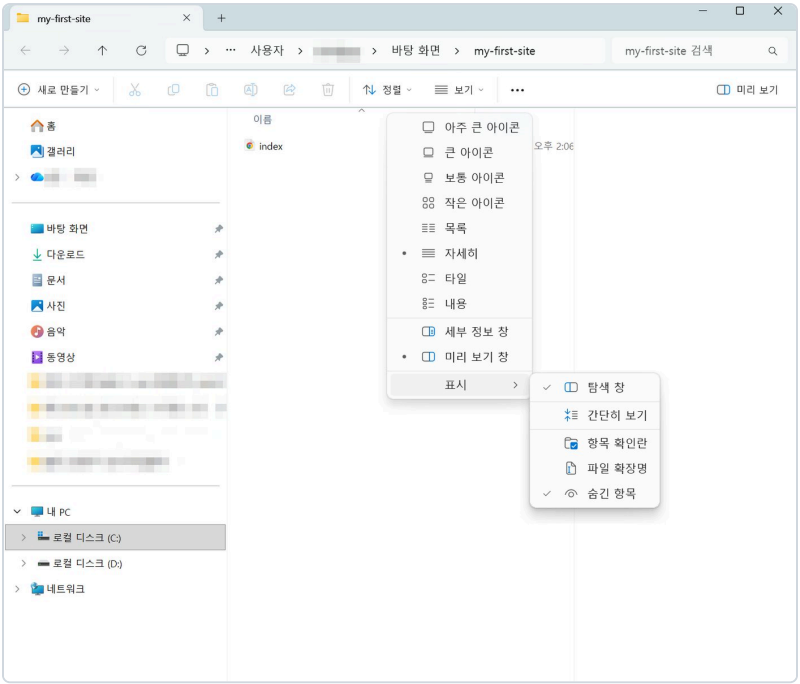


그림 5-11 보기 → 표시 → 파일 확장명.

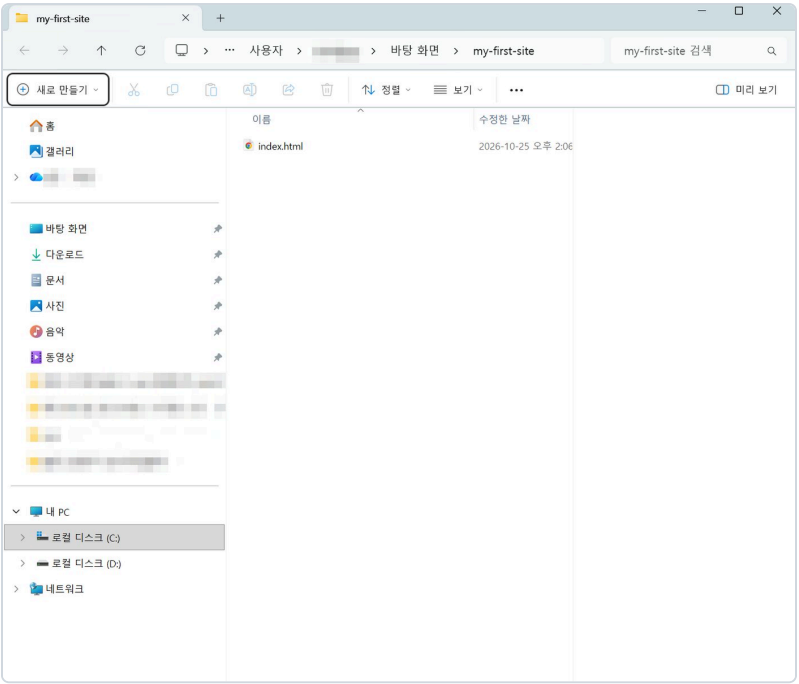


그림 5-12 이제 index.html로 정확히 보입니다. 폴더 안에는 이 파일 하나만 있으면 됩니다.

Windows 10과 Mac은 메뉴 위치가 다릅니다. 뒤의 '도구 따라 하기'를 보세요.

왜 하필 index.html일까

index.html은 웹에서 '첫 페이지'를 뜻하는 약속된 이름입니다. 누군가 주소만 치고 들어오면 서버는 자동으로 이 파일을 찾아 보여줍니다. 이름이 다르면 5.4에서 올렸을 때 '페이지를 찾을 수 없습니다'가 뜹니다.

주소창을 보면 지금 상태를 알 수 있다

index.html을 더블클릭하면 브라우저가 열립니다. 지금 주소창을 보세요. 5.4를 마치면 이 주소가 바뀝니다.

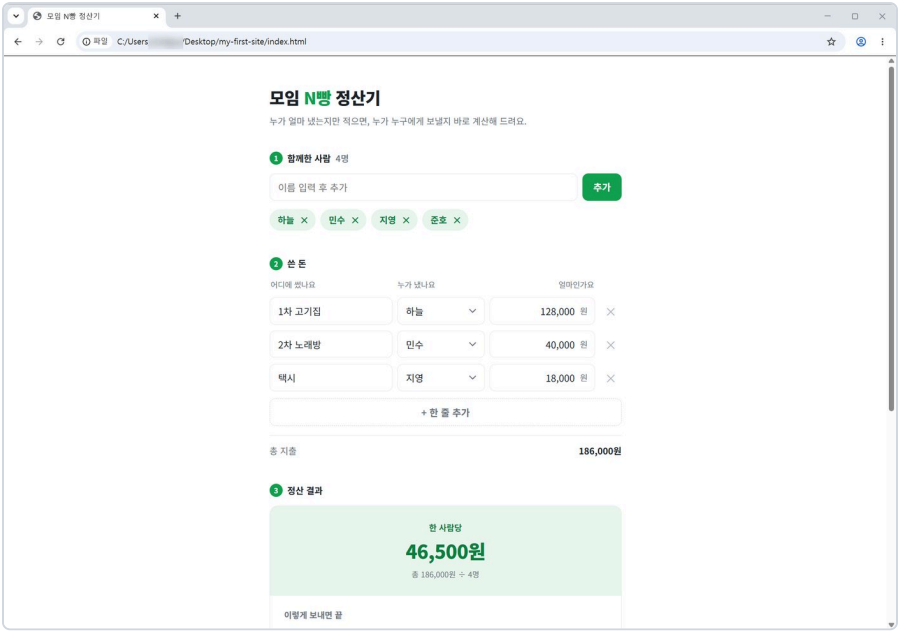


그림 5-13 내 컴퓨터의 파일을 연 상태. 크롬은 주소창 앞에 'ⓘ 파일'을 표시하고, 그 뒤에 파일이 있는 위치를 보여줍니다.

주소창이 이렇게 보이면	의미	누가 볼 수 있나
'ⓘ 파일' + `C:/Users/…` (또는 `file:///`)	내 컴퓨터 안의 파일을 열어 본 것	나만
`https://` 또는 사이트 주소	인터넷에 공개된 페이지	주소를 아는 누구나

지금 주소를 친구에게 보내도 친구는 열 수 없습니다. 파일이 내 컴퓨터에만 있기 때문입니다.

미니 실습: 한 군데만 고쳐보기

파일을 열어 보면 고치고 싶은 곳이 보일 겁니다. 한 번에 다 고치려 하지 말고 딱 한두 군데만 골라 요청해 보세요. 같은 대화창에 이어서 씁니다. 저는 정산 결과에서 가장 중요한 '보낼 사람과 금액'이 작아 보여서 이렇게 요청했습니다.

```
'이렇게 보내면 끝' 부분의 글자를 지금보다 크게 해줘.
보낼 사람 한 줄 한 줄을 각각 카드 모양으로 나눠줘.
나머지는 지금 그대로 두고, 수정된 index.html 파일로 다시 줘.
```

"나머지는 그대로"를 붙이는 이유: 이 말이 없으면 AI가 요청하지 않은 부분까지 바꿔버리는 경우가 있습니다. 마음에 드는 부분을 지키려면 꼭 붙이세요.

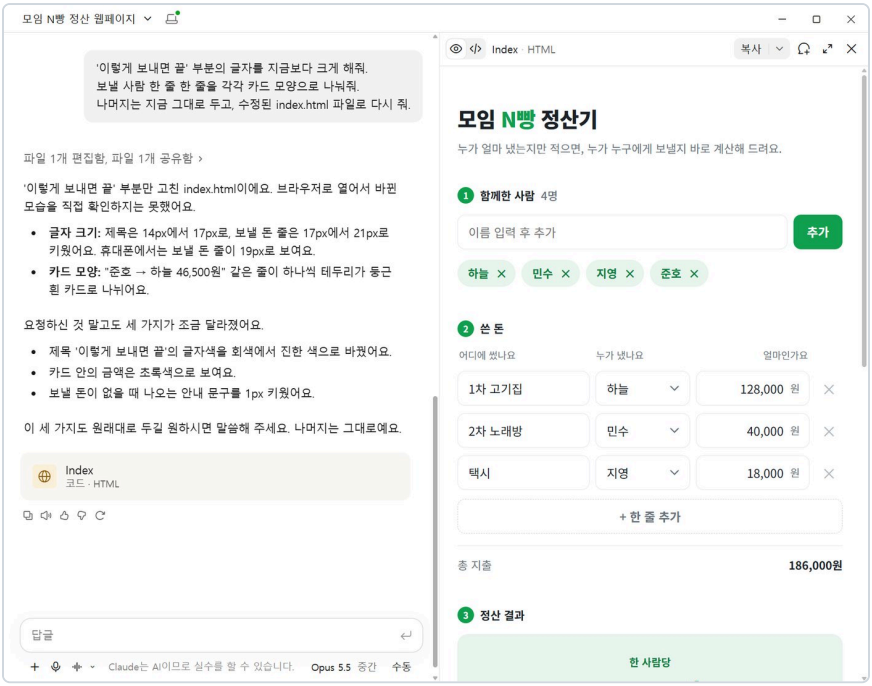


그림 5-14 Claude는 무엇을 어떻게 바꿨는지 설명하고, 같은 파일 카드에 수정본을 담아 줍니다.

"나머지는 그대로"라고 했는데도 답에는 "요청하신 것 말고도 세 가지가 조금 달라졌어요"라는 목록이 있었습니다. 제목 글자색, 카드 안 금액 색, 안내 문구 크기였습니다. 그리고 "브라우저로 열어서 바뀐 모습을 직접 확인하지는 못했어요"라고 했습니다. 그러니 다음 단계에서 내 눈으로 확인하는 것이 중요합니다. 바뀐 세 가지가 마음에 들지 않으면 "그 세 가지는 원래대로 해줘"라고 하면 됩니다.

수정본을 받으면 이렇게 반영합니다.

1. 새 파일을 저장합니다. 그림 5-9과 같은 방법으로, 저장 창에서 `my-first-site` 폴더의 index.html을 골라 덮어씁니다.

2. 웹(claude.ai)에서 받았다면 '다운로드' 폴더의 새 파일로 my-first-site 폴더의 index.html을 바꿉니다. (덮어쓰기)
3. 열려 있던 브라우저에서 새로고침합니다. Windows는 F5, Mac은 Cmd + R입니다.

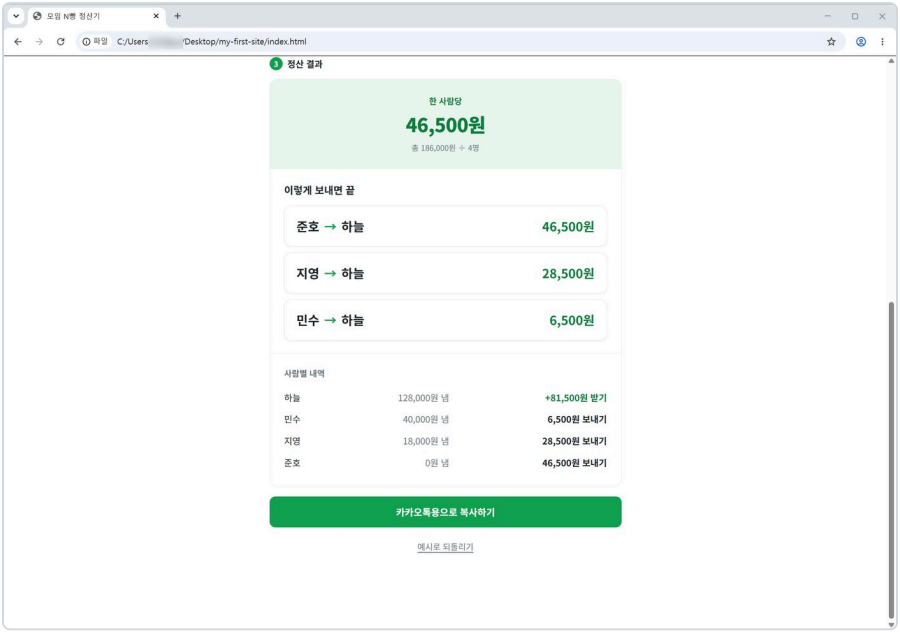


그림 5-15 새로고침하면 바뀐 모습이 보입니다. 보낼 사람이 카드 한 장씩으로 나뉘었습니다.

휴대폰 화면도 미리 확인하기

브라우저 창의 오른쪽 끝을 끌어서 폭을 좁게 줄여 보세요. 글자가 잘리거나 옆으로 밀려나가면 "화면 폭이 좁을 때 칸을 세로로 쌓아줘"처럼 요청하면 됩니다.

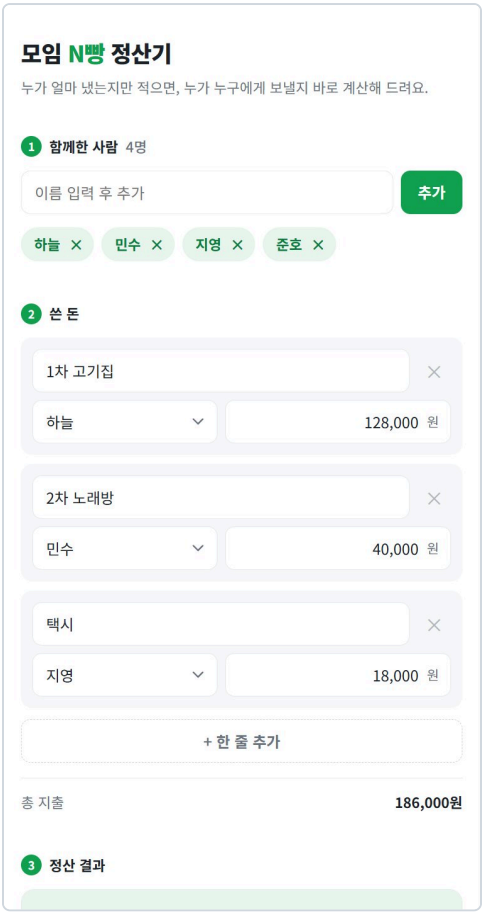


그림 5-16 창을 좁히면 쓴 돈 한 줄이 두 줄로 나뉘어 쌓입니다. 요청문에 '휴대폰에서도 보기 좋게'를 넣어 둔 덕분입니다.

다만 크롬 창은 일정 폭보다 더 좁아지지 않아서, 실제 휴대폰만큼 좁은 화면은 이 방법으로 확인할 수 없습니다. 실제로 이 실습 페이지도 5.4에서 휴대폰으로 열어 보니 문제가 하나 보였습니다(그림 5-24).

휴대폰 폭까지 미리 보고 싶다면 크롬에서 F12(Mac은 Cmd + Option + I)를 눌러 개발자 도구를 연 다음 Ctrl + Shift + M(Mac은 Cmd + Shift + M)을

누르세요. 화면 위쪽에서 휴대폰 기종을 골라 그 크기로 볼 수 있습니다.

5.4 무료로 인터넷에 올리고 주소 공유하기

Netlify Drop 페이지에 my-first-site 폴더를 끌어다 놓으면, 몇 초 뒤 https로 시작하는 인터넷 주소가 생깁니다. 가입 없이도 올릴 수 있고 비용도 들지 않습니다.

다만 가입 없이 올린 사이트는 1시간 동안만 유지되고, 비밀번호가 걸려 있습니다. 친구에게 보낼 주소를 만들려면 마지막에 무료 계정에 연결하는 단계까지 마쳐야 합니다.

'공개'란 무엇인가

웹페이지를 공개한다는 것은 내 파일을 24시간 켜져 있는 다른 컴퓨터(서버)에 맡겨 두는 일입니다. 내 일기장을 도서관 서가에 꽂아 두는 것과 비슷합니다. 내 책상 서랍에 있을 땐 나만 보지만, 도서관에 있으면 위치를 아는 누구나 찾아와 읽을 수 있습니다. 이때 서가 위치에 해당하는 것이 인터넷 주소입니다.

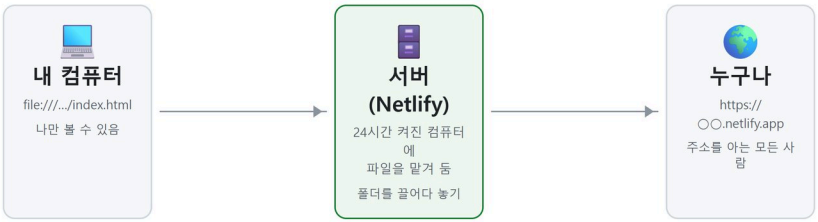


그림 5-17 공개의 원리. 내 컴퓨터 → 서버 → 누구나.

파일이 내 컴퓨터에만 있으면 나만 보고, 서버에 맡기면 주소를 아는 모든 사람이 봅니다. Netlify Drop은 이 '맡기는 일'을 끌어다 놓기 한 번으로 끝내 줍니다.

왜 Netlify Drop인가

무료로 웹페이지를 올릴 수 있는 곳은 여러 군데입니다. 첫 실습에서는 준비할 것이 가장 적은 곳을 골랐습니다.

서비스	올리는 방식	가입	이 책에서는
Netlify Drop	폴더를 끌어다 놓기	없이 시작, 유지하려면 나중에 연결	5장 (지금)
Vercel Drop	폴더를 끌어다 놓기	먼저 가입 필요	다루지 않음
GitHub Pages	저장소에 파일 올리기	필요, '저장소' 개념도 필요	이 책에서는 다루지 않음 (20장은 GitHub에 저장만)

따라 하기: 폴더 하나 끌어다 놓기

1. 크롬이나 엣지에서 `app.netlify.com/drop`을 엽니다.
    - 화면은 영어지만, 가운데 Drop here(여기에 놓기)만 찾으면 됩니다.

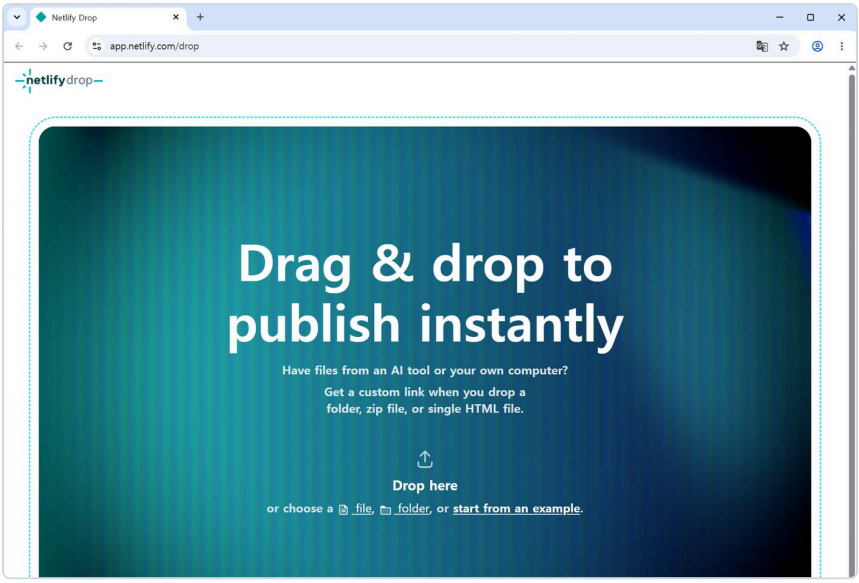


그림 5-18 Netlify Drop 첫 화면. 점선으로 둘러싸인 큰 영역 전체가 업로드 영역입니다.

2. 바탕화면의 my-first-site 폴더를 통째로 끌어다 업로드 영역에 놓습니다.
    - index.html 파일이 아니라 폴더를 놓습니다.
    - 탐색기 창과 브라우저 창을 나란히 띄워 두면 끌어다 놓기 쉽습니다.
    - 폴더를 열었을 때 index.html이 바로 보여야 합니다. 폴더 안에 폴더가 한 겹 더 있으면 주소를 열었을 때 페이지가 안 나옵니다.

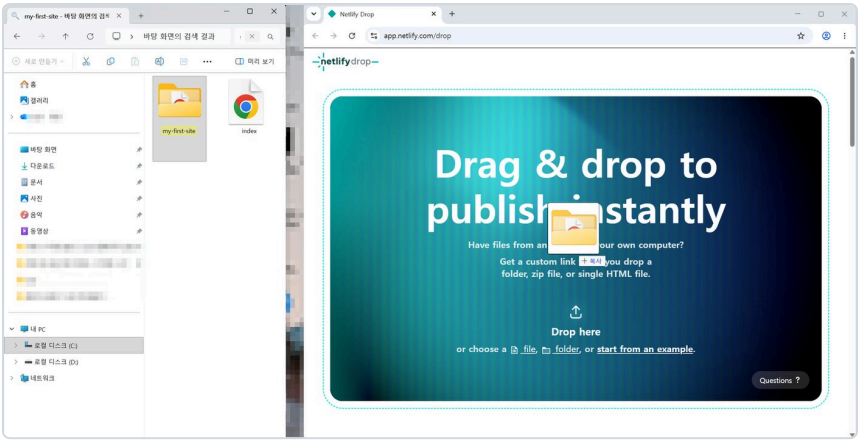


그림 5-19 왼쪽 탐색기에서 폴더를 집어 오른쪽 업로드 영역 위로 가져온 순간. 마우스 버튼을 놓으면 업로드가 시작됩니다.

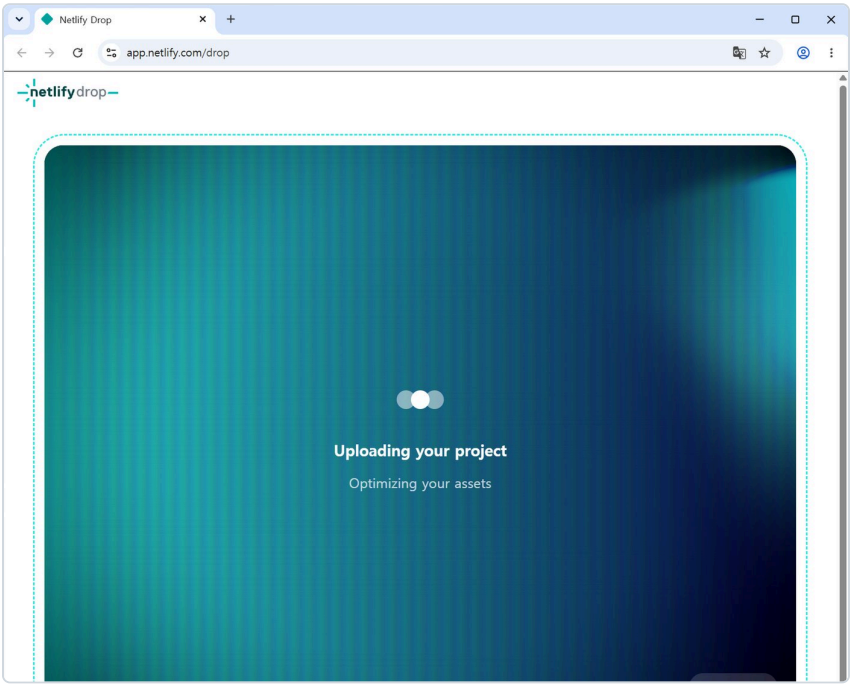


그림 5-20 업로드 중. 이 장의 파일은 작아서 몇 초면 끝납니다.

3. 업로드가 끝나면 주소가 나타납니다.

   - `https://무작위-단어-조합.netlify.app` 같은 모양입니다. 실습할 때는 `sprightly-gecko-047197.netlify.app`이 나왔습니다. 이상해 보여도 정상이며, 이름은 나중에 바꿀 수 있습니다.
   - 이 이름은 올릴 때마다 새로 무작위로 정해집니다. 이 장의 그림에서 주소 이름이 중간에 `profound-souffle-03ea40`, `mellifluous-shortbread-af6466`으로 바뀌는 것도, 실습하면서 폴더를 여러 번 올렸기 때문입니다.
   - 완료 화면에서 세 가지를 확인하세요.
     - 주소: 아래쪽 주소 칸. 옆의 Copy로 복사하고 Open으로 열어 볼 수 있습니다.
     - 남은 시간: 'Claim this site' 버튼 옆 숫자(59:37 등). 가입 없이 올린 사이트가 사라지기까지 남은 시간입니다.
     - 임시 비밀번호: 'My-Drop-Site'. 연결(Claim)하기 전까지 이 비밀번호를 넣어야 페이지가 열립니다.

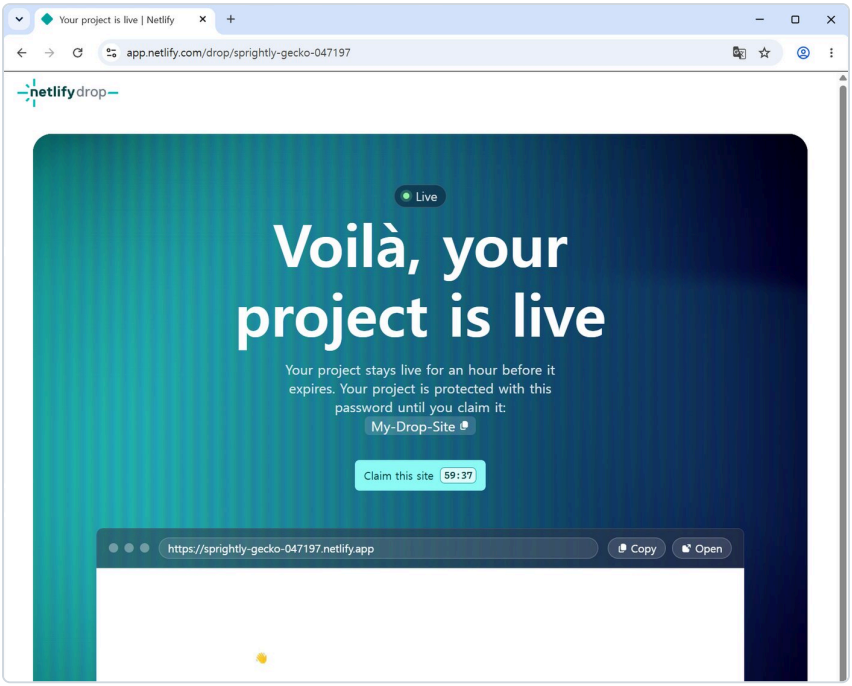


그림 5-21 업로드 완료 화면. "1시간 동안 유지되며, 연결(Claim)하기 전까지는 이 비밀번호로 보호된다"는 안내가 나옵니다.

> 주의: 1시간 안에 연결하세요. 가입 없이 올린 사이트는 1시간이 지나면 사라집니다. 아래 4~5단계로 확인을 마쳤다면 바로 6단계(연결)로 넘어가세요. 시간이 지나 사라졌다면 폴더를 다시 끌어다 놓고, 이번에는 바로 연결까지 마치면 됩니다.

4. 주소를 눌러 확인합니다.

   - 처음 열면 비밀번호를 묻는 화면이 나옵니다. 완료 화면에 있던 비밀번호를 넣고 Submit을 누르세요.

- 주소창에 사이트 주소(`oo.netlify.app`)가 보이면 인터넷에 공개된 것입니다.

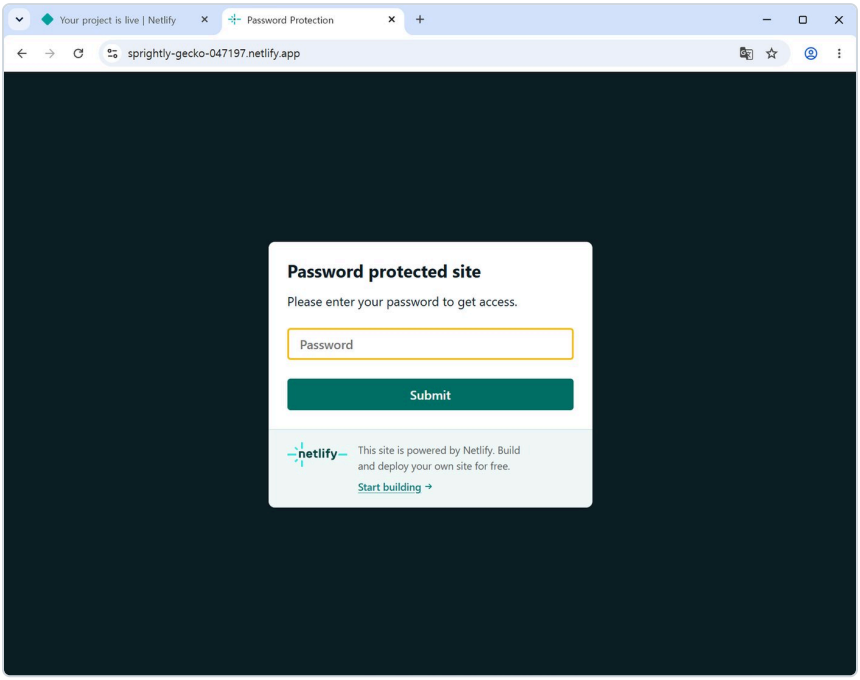


그림 5-22 연결(Claim)하기 전에는 방문자에게도 이 화면이 먼저 보입니다.

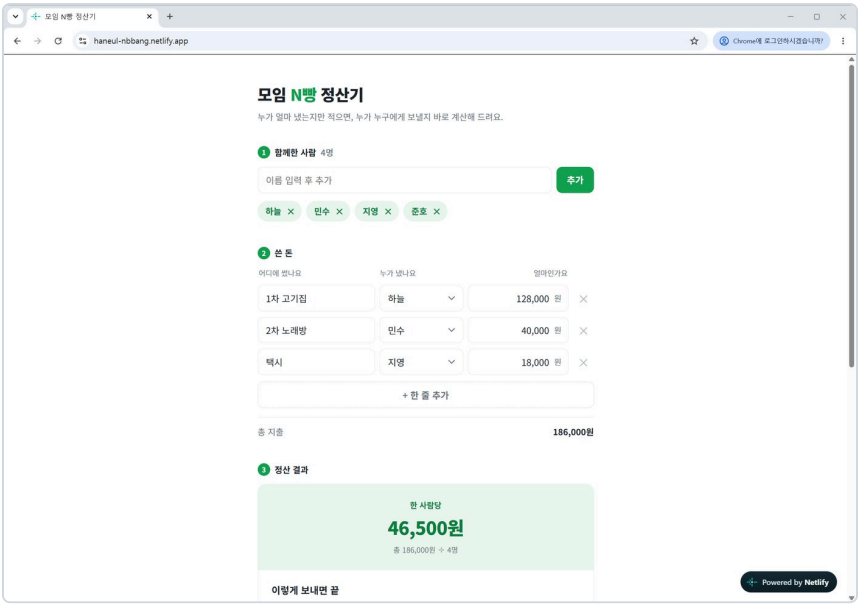


그림 5-23 같은 페이지가 이번에는 인터넷 주소로 열렸습니다. 그림 5-13의 주소창과 비교해 보세요. (이 그림은 7단계에서 이름을 바꾼 뒤의 주소입니다.)

5. 휴대폰으로도 열어 봅니다.

   - 나에게 카톡으로 주소를 보내고 눌러 보면 가장 간단합니다. 연결 전이라면 휴대폰에서도 비밀번호를 한 번 넣어야 합니다.

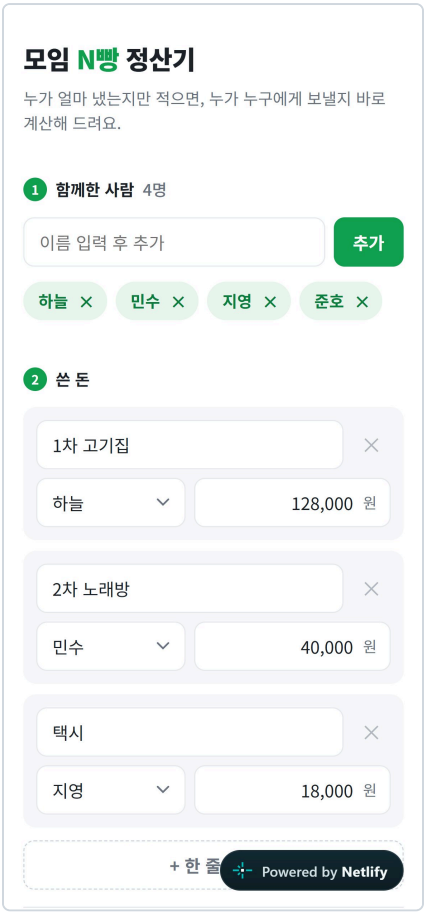


그림 5-24 휴대폰 화면 크기로 연 모습. PC에서는 괜찮았는데, 오른쪽 아래 Netlify 배지가 '+ 한 줄 추가' 단추를 가렸습니다.

- 이 배지는 Netlify가 공개한 페이지에 덧붙이는 것이라 index.html을 고쳐도 사라지지 않습니다. Netlify 설정에서 끌 수 있고, 끄는 법은 19장 공개 전 체크리스트에서 다룹니다. 지금은 화면을 조금 내리면 단추가 보이니 그대로 두어도 됩니다.

- 이렇게 PC에서는 안 보이던 것이 휴대폰에서 보이는 일이 흔합니다. 공개한 뒤에는 꼭 휴대폰으로 한 번 열어 보세요.

6. 무료 계정을 만들어 사이트를 내 것으로 연결합니다.

   - 완료 화면(그림 5-21)의 'Claim this site' 버튼을 누르면 가입 화면이 나옵니다. Google 계정이 있다면 'Sign up with Google'이 가장 빠릅니다. 이미 가입했다면 아래쪽 'Log in'을 누르세요.

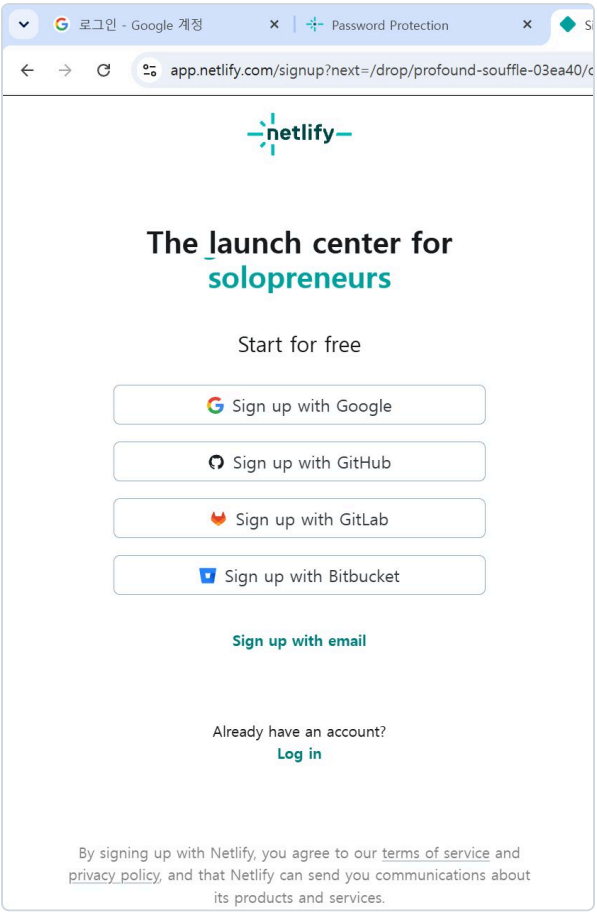


그림 5-25 Claim this site를 누르면 나오는 가입 화면. 가입이나 로그인을 마치면 방금 올린 사이트가 자동으로 내 계정에 연결됩니다.

- 연결이 끝나면 내 프로젝트 화면으로 넘어갑니다. 사이트 이름 옆에 Public이 보이면 성공입니다. 1시간 만료와 임시 비밀번호가 사라져서, 이제 누구나 비밀번호 없이 주소로 들어올 수 있습니다.
- 연결해 두면 나중에 수정본을 같은 주소로 다시 올릴 수 있습니다.

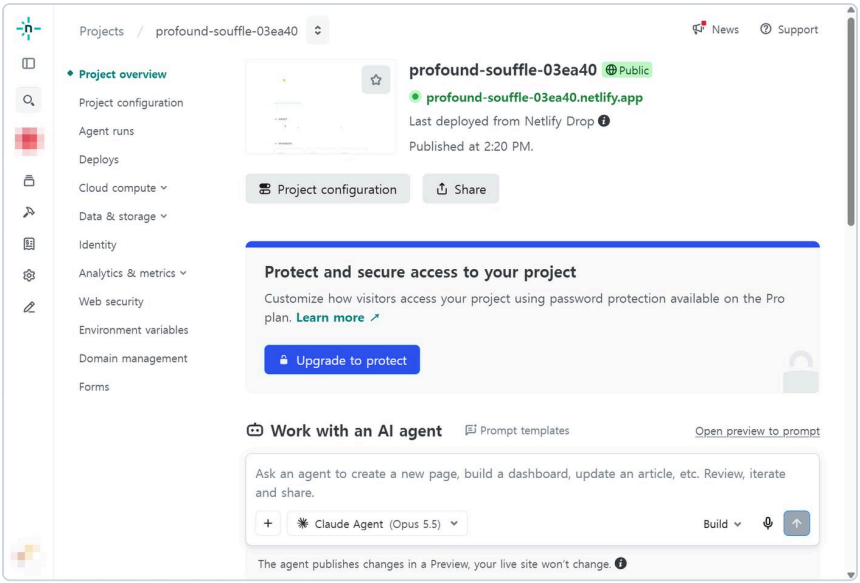


그림 5-26 내 계정에 연결된 사이트. 아래 'Protect and secure access' 상자는 Netlify의 유료 요금제(Netlify Pro)에서 쓰는 비밀번호 보호 안내라 넘어가도 됩니다. Claude의 Pro 요금제와는 관계없습니다.

7. (선택) 사이트 이름 바꾸기. `haneul-nbbang.netlify.app` 처럼 기억하기 쉬운 이름으로 바꿀 수 있습니다.

   - 프로젝트 화면에서 Project configuration → General → Project details로 가서 'Change project name'을 누릅니다.
   - 새 이름을 입력하면 바로 아래에 바뀔 주소가 미리 보입니다. Project name is available(쓸 수 있는 이름)이 뜨면 'Save'를 누르세요. 다른 사람이 이미 쓰는 이름은 고를 수 없습니다.
   - 영문 소문자, 숫자, 하이픈(-)만 쓸 수 있고 최대 37자입니다.

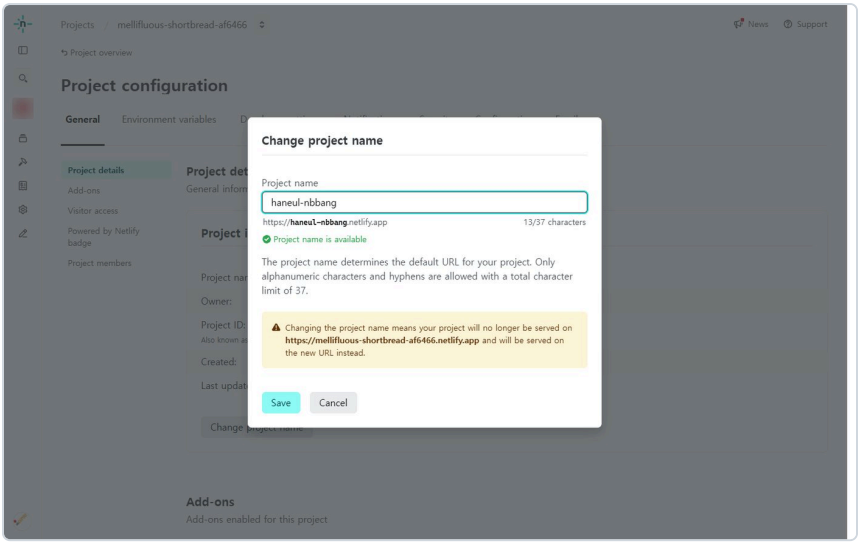


그림 5-27 이름 바꾸기 창. 노란 상자는 "이름을 바꾸면 예전 주소로는 더 이상 열리지 않는다"는 경고입니다.

> 주의: 이름을 바꾸면 예전 주소는 끊깁니다. 실습 중 앞서 올렸던 사이트의 이름을 바꾼 뒤 그 예전 주소(`profound-souffle-03ea40.netlify.app`)를 열어 보니 'Site not found'가 나왔습니다. 주소를 여기저기 공유하기 전에 이름부터 정해 두세요.

8. (선택) 수정본을 같은 주소로 다시 올리기. 정산 결과를 카톡에 붙여 넣어 보면 "어디로 보내?"라는 질문이 꼭 돌아옵니다. 받을 계좌를 함께 붙일 수 있게 고쳐서 다시 올려 봅시다.
    - 같은 Claude 대화창에 이어서 요청합니다.

```
카톡에 결과를 붙여 넣으면 다들 "어디로 보내?"라고 물어봐.
정산 결과 아래에 '받을 계좌' 칸을 하나 만들고, 적어 두면 카카오톡용으로
복사할 때 맨 끝에 같이 붙여줘.
비워 두면 지금처럼 계좌 없이 복사되게 해줘.
나머지는 지금 그대로 두고, 수정된 index.html 파일로 다시 줘.
```

- Claude는 계좌 칸을 넣고, 이번에도 요청 밖에서 정한 것을 밝혔습니다. 칸 이름 옆에 '(선택)'을 붙였고, 적은 계좌는 브라우저에 남아서 다시 열어도 그대로 있다는 것입니다(4장의 '브라우저 저장소'). 계좌는 내 휴대폰이나 PC의 브라우저에만 남고, 인터넷에 올린 파일에는 들어가지 않습니다.

- 받은 파일로 my-first-site 폴더의 index.html을 덮어씁니다. (5.3과 같은 방법)

- Netlify 프로젝트 화면 왼쪽 메뉴에서 'Deploys'를 누르고, 아래쪽 Need to update your project? 영역에 폴더를 다시 끌어다 놓습니다.

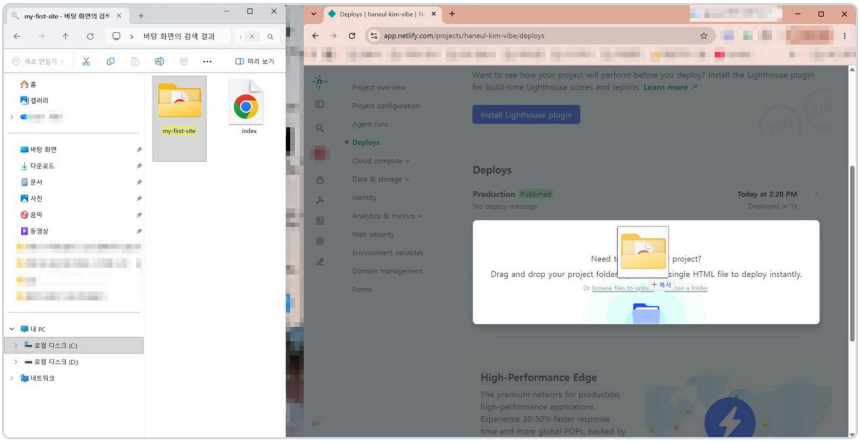


그림 5-28 처음 올릴 때와 똑같이 폴더째 끌어다 놓습니다. 업로드 영역이 하얗게 강조되면 놓으세요.

- 몇 초 뒤 배포 목록 맨 위에 새 항목이 Published로 올라옵니다. 주소는 그대로이고 내용만 바뀝니다.

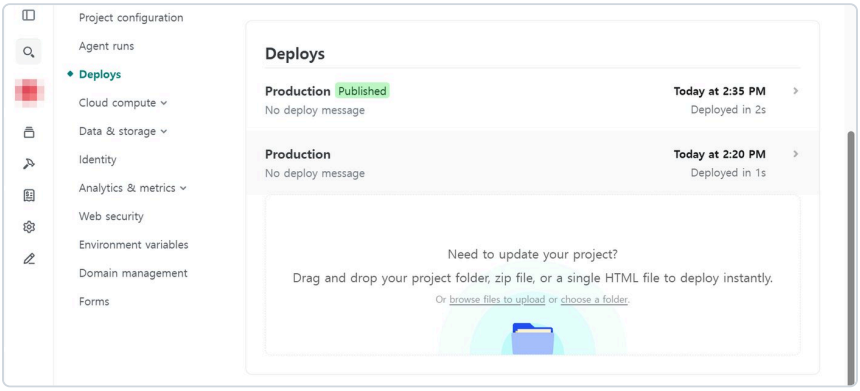


그림 5-29 새로 올린 수정본(맨 위)이 공개 중이고, 이전 버전은 아래에 기록으로 남습니다.

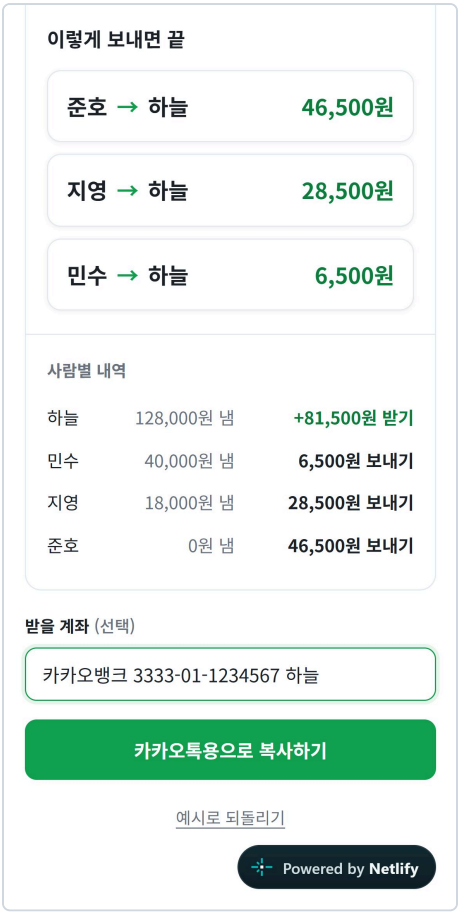


그림 5-30 같은 주소를 휴대폰 화면 크기로 다시 열었습니다. 주소는 그대로이고, 받을 계좌 칸이 생겼습니다. 그림의 계좌번호는 예시입니다.

알아두면 좋은 제한

- 폴더 전체는 50MB 이하로 올리는 것이 권장됩니다. 파일 하나는 10MB 이하가 권장됩니다. 이 장의 index.html은 수 KB 정도라 한참 여유가 있습니다.

- 사진이나 영상을 넣기 시작하면 용량을 확인해야 합니다.

주소 공유하기

계정 연결까지 마쳤다면 이제 이 주소를 모임 단톡방에 올려 두세요. 다음 정산 때 누구나 열어서 쓸 수 있습니다. 카톡에 보냈을 때 뜨는 미리보기 이미지와 제목을 바꾸는 법은 15장에서 다룹니다.

도구 따라 하기 (2026년 11월 기준)

본문에서 말한 원리는 바뀌지 않지만, 버튼 이름과 위치는 도구가 업데이트될 때마다 바뀝니다. 그런 부분만 이 박스에 모았습니다. 화면이 다르게 보이면 업데이트 페이지를 확인하세요.

> 이 표와 화면이 다르면 업데이트 페이지를 먼저 확인하세요:
> https://github.com/vibecoding-book/vibecoding-book

Claude 앱

하려는 일	방법	그림
앱 설치·로그인	claude.com/download → Windows용 다운로드(Mac은 macOS 다운로드) → 설치 → 구글·애플·이메일로 로그인	5-2
요금제 보기·바꾸기	주소창에 claude.com/pricing → 요금제 칸의 버튼 → 결제	5-3

하려는 일	방법	그림
채팅으로 바꾸기·새 대화	왼쪽 위 말풍선 아이콘(채팅) → 대화 목록의 '새로 만들기'	5-5
미리보기 열기	답변 안의 파일 카드('Index · 코드 · HTML')를 누르면 오른쪽 창에 열림	5-7
만든 파일 받기	미리보기 창 위쪽 '복사' 옆 ▾ → 'HTML 형식으로 다운로드' → 저장 창에서 폴더 고르기	5-8
수정 요청	같은 대화창에 이어서 입력. 새 대화를 열면 앞의 내용을 모름	5-13
코드만 보일 때	"index.html 파일로 만들어줘"라고 다시 요청	-

파일 확장자 보이게 하기

운영체제	메뉴 경로
Windows 11	파일 탐색기 → 보기 → 표시 → 파일 확장명 체크 (그림 5-11)
Windows 10	파일 탐색기 → 보기 탭 → 파일 확장명 체크
Mac	Finder → 설정 → 고급 → 모든 파일 확장자 보기 체크

Netlify Drop

하려는 일	방법	그림
처음 올리기	app.netlify.com/drop 접속 → 폴더를 업로드 영역(Drop here)에 끌어다 놓기	5-17, 5-18

하려는 일	방법	그림
주소·비밀번호 확인	완료 화면 아래쪽 주소 칸(Copy / Open), 가운데 임시 비밀번호	5-20
사이트 유지하기	완료 화면의 Claim this site → 가입 또는 로그인하면 자동 연결. 1시간 안에 해야 함	5-24, 5-25
사이트 이름 바꾸기	프로젝트 화면 → Project configuration → General → Project details → Change project name → Save	5-26
수정본 다시 올리기	프로젝트 화면 왼쪽 Deploys → 아래쪽 'Need to update your project?' 영역에 폴더를 다시 끌어다 놓기	5-27, 5-28
올리기가 안 될 때	최신 크롬·엣지로 바꾸거나, 폴더를 ZIP으로 압축해서 올리기	-

자주 막히는 문제

대부분의 문제는 파일 이름, 폴더 구조, 파일이 나뉘어 있는 것 세 가지 중 하나입니다.

이런 일이 생기면	원인	해결 방법
더블클릭했더니 메모장이 열려요	파일이 사실 index.html.txt이거나, 연결 프로그램이 메모장	확장자를 보이게 한 뒤 이름을 고치거나, 마우스 오른쪽 버튼 → 연결 프로그램 → 크롬

이런 일이 생기면	원인	해결 방법
글자만 나오고 디자인이 없어요	디자인(CSS)이 다른 파일로 나뉘어 있음	"디자인을 HTML 파일 안에 모두 넣어서 다시 줘"
한글이 이상한 기호로 보여요	글자 인코딩 설정 누락	"한글이 깨져. UTF-8 설정을 넣어서 다시 줘"
고쳤는데 화면이 그대로예요	새로고침 안 함, 또는 새 파일이 index (1).html로 저장됨	폴더 안 파일 이름 확인 후 새로고침 (F5 / Cmd + R)
업로드 영역에 놓아도 반응이 없어요	브라우저가 폴더 끌어놓기를 지원하지 않음	크롬이나 엣지 사용, 또는 폴더를 ZIP으로 압축해서 올리기
주소를 열면 비밀번호를 물어요	아직 계정 연결(Claim)을 하지 않음	완료 화면의 임시 비밀번호(My-Drop-Site) 입력. 다른 사람에게 보내려면 Claim까지 마치기
주소를 열면 '페이지를 찾을 수 없음'이 떠요	폴더 안에 index.html이 없거나, 이름이 다르거나, 폴더가 한 겹 더 감싸져 있음	폴더를 열었을 때 index.html이 바로 보이는지 확인 후 다시 올리기
주소가 이상한 영어 단어 조합이에요	자동으로 붙는 임시 이름	정상입니다. 계정 연결 후 이름 변경
이름을 바꿨더니 예전 주소가 'Site not found'예요	이름을 바꾸면 예전 주소는 더 이상 쓰이지 않음	새 주소를 다시 공유하기

이런 일이 생기면	원인	해결 방법
1시간쯤 뒤 사이트가 열리지 않아요	가입·연결 없이 임시로만 올린 상태	다시 올린 뒤 바로 계정 연결까지 마치기

이 장 마무리 체크리스트

☐ 기획 시트 7칸을 채웠다

☐ 실제 계좌번호나 휴대폰 번호는 요청문에 넣지 않았다

☐ 예시 숫자로 한 사람당 금액과 보낼 돈을 직접 검산했다

☐ 요청문에 목표·내용·모양·받는 형태 4가지를 넣었다

☐ my-first-site 폴더에 index.html을 저장했다

☐ 파일 확장자를 보이게 하고 이름이 index.html인지 확인했다

☐ 내 컴퓨터 브라우저에서 열어 확인했다

☐ "나머지는 그대로"를 붙여 한 번 이상 수정 요청을 해 봤다

☐ Netlify에 올려 https 주소를 받았다

☐ 1시간 안에 계정을 연결해 비밀번호 없이 열리게 했다

☐ 휴대폰으로 주소를 열어 봤다

이것으로 '기획 → 요청 → 확인 → 공개'를 한 바퀴 돌았습니다. 앞으로 만드는 프로젝트도 규모만 커질 뿐 모두 이 순서를 따릅니다. 6장에서는 "좀 더 예쁘게" 대신 AI가 정확히 알아듣는 수정 요청법을 배웁니다.

이 장의 요청문을 비롯해 이 책에 나오는 요청문은 부록 A에 모아 두었습니다.

참고 문헌

- Netlify Drop updates (2026-07-27) (2026년 10월 27일 확인)
- Netlify Drop vs Vercel Drop (2026년 10월 27일 확인)



여기까지가 맛보기예요.
내 링크, 하나 생기셨나요?

본편에는 고치는 법(6장), AI가 알고 시작하게 하는 법(7~10장),
일을 맡기고 검수하는 법(11~14장), 심리테스트·게임·AI 앱·
로그인 서비스 만들기(15~18장), 공개 전 체크리스트와
되돌리기(19~20장)까지 이어져요.

PDF 551쪽 **EPUB** **실습 예제 16개** **부록 5종**

vibecode.tmhub.co.kr

네이버 스마트스토어 · 크몽에서 구매할 수 있어요

이 PDF는 무료 맛보기본으로, 자유롭게 공유해도 됩니다. © 2026 김도균 · 투믹스소프트