

</> AI JAVAPARK

나만의 책만들기

Claude로 8주 안에 전자책 만들기

자바파커 · AI JAVAPARK

v0.1 (8주 빌드 인 퍼블릭 챕터판)

2026-06-14

blog.javapark.kr

차례

1장 — 시작 전 준비: 도구 4가지와 첫 폴더 만들기

2장 — 콘셉트 잡기: 누가 읽을지부터 정한다

3장 — 차례 짜기: outline이 진실의 원천이다

4장 — 자료 모으기: 흩어진 글을 책 자료로

5장 — 챗터 쓰기: 빌드 인 퍼블릭의 힘

6장 — 다듬기와 디자인: 표지·다이어그램·문체

7장 — 빌드와 배포: pandoc부터 무료 PDF까지

8장 — 마무리: 책 한 권의 의미, 그리고 다음 책

1장 — 시작 전 준비: 도구 4가지와 첫 폴더 만들기

"책 한 권 정도는 쓰고 싶은데, 시간이 없어요."

책을 써보고 싶다는 사람에게 가장 많이 듣는 말이다. 그런데 AI 도구가 글쓰기·구조 설계·디자인·빌드까지 함께 거들어주는 지금은, **8주에 한 챕터씩만** 쌓으면 한 권이 만들어진다.

이 책은 **자기 자신의 제작 과정을 보여주는 책**이다. 지금 읽고 있는 이 1장 자체가 그 책의 첫 장이고, 책 전체는 한 사람이 Claude 와 함께 8주 동안 전자책 한 권을 만들어가는 과정을 그대로 따라간다.

1장의 목표는 단순하다 — **오늘 폴더 하나를 만들고, Claude와 첫 대화를 끝내는 것**. 8주 챌린지의 출발선이자, 가장 작은 한 걸음 이다.

누구나 책을 쓸 수 있는 시대

5년 전이라면 책 한 권을 만든다는 건 적어도 다음을 의미했다.

- 50,000자 이상의 원고를 처음부터 끝까지 직접 쓰기
- 출판사 검토 → 편집 → 디자인 → 인쇄 → 유통 (수개월~수년)
- 표지·내지·일러스트 외주 또는 본인이 직접

지금은 다르다.

- **글쓰기 보조** — Claude가 초안·문체 정리·논리 검토를 동시에
- **구조 설계** — outline·차례·페르소나를 대화 한 번으로
- **디자인** — HTML+CSS로 표지·다이어그램을 30분 내 제작
- **빌드** — `pandoc` 한 줄로 마크다운 → PDF/EPUB
- **유통** — 무료 PDF 호스팅, KDP, 셀프 출판이 누구나 가능

여전히 **글의 본질(독자에게 무엇을 전할 것인가)**은 사람의 몫이지만, **그 외 모든 마찰 비용**이 사라졌다. 이 책이 보여줄 것은 그 흐름을 어떻게 한 사람이 8주 안에 끝낼 수 있는가다.

우리가 만들 책 — 「나만의 책만들기」

| 항목 | 값 |
|-------|--|
| 제목 | 나만의 책만들기 |
| 부제 가안 | "Claude로 8주 안에 전자책 만들기" |
| 분량 | 약 50p (전자책) |
| 대상 독자 | 혼합 — 글쓰기·출판이 처음인 누구나 + Claude를 더 잘 쓰고 싶은 개발자 |
| 출간 채널 | 무료 PDF 배포 (1차) |
| 메타 컨셉 | "이 책 자체가 자기 제작 과정을 보여주는 책" |

읽는 사람이 따라하면 자기만의 50p 전자책 한 권을 손에 쥐게 된다.

8주 챌린지 한눈에

| 주차 | 챕터 | 핵심 |
|----|------------|------------------|
| 1주 | #1 시작 전 준비 | 도구·폴더·마음가짐 |
| 2주 | #2 콘셉트 잡기 | 한 줄 요약·페르소나 |
| 3주 | #3 차례 짜기 | outline = 진실의 원천 |
| 4주 | #4 자료 모으기 | 블로그·노트 → 책 자료 |
| 5주 | #5 챕터 쓰기 | 빌드 인 퍼블릭의 힘 |
| 6주 | #6 다듬기·디자인 | 표지·다이얼로그·문체 |
| 7주 | #7 빌드·배포 | pandoc → 무료 PDF |
| 8주 | #8 마무리·부록 | 회고·다음 책 |

8주가 끝나면 발행한 챕터를 책 톤으로 정제하고 PDF로 묶는 작업이 이어진다.

필요한 도구 4가지

가장 자주 받는 질문 — "무엇부터 깔아야 하나?" — 에 답한다. 결론부터 말하면 딱 4가지다.

1. Claude — 글의 동료

claude.ai에 접속해 계정만 만들어도 충분하다. 이 책에서 Claude는 편집자·기획자·디자인 자문 역할을 한다. 비개발자도 대화창 하나로 모든 작업을 한다.

💡 글의 모든 결정(콘셉트·차례·표지 톤)은 사람이 내린다. Claude는 옵션을 빠르게 제시하고 일관성을 점검하는 도우미다.

2. Claude Code — 코드·자동화의 동료 (개발자용 사이드)

⚠️ 비개발자라면 이 절은 건너뛰어도 된다. 본문 전체는 Claude(웹)만으로 따라올 수 있게 구성했다.

Claude Code는 터미널에서 동작하는 Claude다. 책 작업에서는 **자동화** 영역을 담당한다.

- 블로그 글을 책 발췌(`drafts/from-blog/`)로 자동 미러
- 표지 HTML을 한 번에 PNG로 캡처(Playwright 연동)
- pandoc 빌드 명령을 wrapping

3. 마크다운 — 글의 형식

`.md` 확장자의 일반 텍스트 파일이다. 무거운 워드 프로세서 대신 **텍스트 + 약간의 기호**로 글을 쓴다.

```
# 제목
## 소제목

**굵게**, *기울임*, [링크](https://...)

- 목록 1
- 목록 2

> 인용문
```

이 책 자체도 마크다운으로 작성됐다. **포맷에 신경 쓰지 않고 글 자체에 집중**할 수 있는 게 가장 큰 장점이다.

4. pandoc — 빌드 도구

마크다운을 **PDF·EPUB·HTML로 한 번에 변환**해주는 도구다. 7장에서 본격적으로 다룬다. 지금은 "이런 게 있다" 정도만 기억하면 된다.

실습 1 — 작업 폴더 만들기

이 책 작업을 위한 디렉토리 구조다. **오늘 할 일은 이 폴더를 그대로 만드는 것이다.**

```

book/
├─ README.md      # 책 메타 정보
├─ outline.md     # 전체 차례 (진실의 원천)
├─ chapters/     # 정제된 챕터
├─ drafts/
│  └─ from-blog/ # 블로그 글 발췌
│     └─ reworked/ # 책 톤 재구성 작업장
├─ research/
│  └─ notes/      # 자유 메모
│     └─ case-studies/ # 실전 케이스
│        └─ references.md # 참고 자료
├─ assets/
│  └─ images/     # 표지·다이어그램 PNG
│     └─ diagrams/ # HTML 원본
├─ build/        # PDF·EPUB 산출물
└─ scripts/      # 자동화 스크립트 (개발자만)

```

비개발자 — 파일 탐색기로

1. 원하는 위치에 **book** 폴더 생성
2. 그 안에 위 구조대로 하위 폴더들을 손으로 만들기
3. **book/README.md** 를 메모장(혹은 Notion·VS Code)으로 새로 만들고 책 제목 한 줄만 적어두기

개발자 — 명령어 한 줄로

```

mkdir -p book/{chapters,drafts/{from-blog,reworked},research/{notes,case-
studies},assets/{images,diagrams},build,scripts}

```

오늘은 폴더만 비어 있으면 된다. **빈 그릇을 먼저 놓는 것이** 1장의 핵심이다.

실습 2 — Claude와 첫 대화: 책 콘셉트 인터뷰

폴더가 만들어지면, Claude를 열고 **딱 5개 질문**을 받아본다. 다음 프롬프트를 그대로 복사해 붙여 넣어보자.

나는 「나만의 책만들기」라는 전자책을 만들고 싶다. 8주에 걸쳐 한 챕터씩 발행하는 빌드 인 퍼블릭 방식이다.

다음 5가지 질문에 차례로 답할 수 있게 하나씩 물어봐 줘.

질문 하나에 답하면 다음 질문으로 넘어가는 식으로.

1. 이 책을 한 줄로 요약하면?
2. 누구를 위한 책인가? (구체적인 한 사람)
3. 이 책을 다 읽은 독자가 얻게 되는 것 한 가지는?
4. 비슷한 책과 어떻게 다른가?
5. 8주가 끝나는 시점에 어떤 모습이 되고 싶은가?

Claude가 한 번에 한 질문씩 던진다. **5개 답변을 다 마치는 것이** 오늘의 두 번째이자 마지막 미션이다. 답변은 길게 쓸 필요 없고, **떠오르는 그대로** 적어두면 된다.

답변이 끝나면 Claude에게 마지막으로 부탁한다.

"방금 5개 답변을 책 README의 첫 단락 형태로 정리해줘."

받은 결과를 `book/README.md` 에 그대로 복사해두면, 2장(컨셉트 잡기)의 출발점이 된다.

마음가짐 — 완벽보다 발행

빌드 인 퍼블릭의 본질은 **완벽한 글을 한 번에 내놓는 것이 아니라, 거친 글을 매주 내놓고 다듬는 것이다.** 이 책도 마찬가지다.

- 매주 한 챕터 **반드시 발행**
- 부족한 부분은 피드백으로 다듬기
- 8주 후 정제 단계에서 책 톤으로 재구성
- "완성 후 한 번에 공개"는 시도하지 않는다

매주 발행 약속이 챌린지의 거의 전부다. 글의 완성도는 8주 동안 자연스럽게 따라온다.

자주 묻는 질문

Q. 비개발자도 정말 따라할 수 있나? 가능하다. 본문은 모두 `claude.ai` 대화창 하나로 진행할 수 있게 구성했다. Claude Code·pandoc·자동화 스크립트는 "있으면 더 편하다" 정도이고, 부록·사이드바로 분리했다.

Q. 책의 저작권은 누구 것인가? 전적으로 본인이다. Claude는 도구이고, 글의 결정은 모두 사람이 내린다. 다만 출간 시 "AI 도구 활용 명시"를 권장한다(8장에서 다룬다).

Q. 8주 안에 진짜 만들어지나? 8주가 끝나는 시점에 50p 전자책 PDF가 손에 쥐어진다. 이 책이 그 증거다.

Q. 이미 다른 글을 쓰고 있는데, 그걸 책으로 바꿀 수 있나? 가능하다. 4장(자료 모으기)에서 기존 블로그·노트를 책 자료로 가져오는 워크플로우를 다룬다.

이 장의 정리

오늘 두 가지만 끝내면 된다.

- `book/` 폴더 만들기 (구조만, 비어 있어도 OK)
- Claude와 5개 질문 대화 → 결과를 `book/README.md` 에 붙여넣기

다음 장에서는 그 결과를 가지고 **책의 컨셉트를 단단히 잡는 작업**을 시작한다.

2장 — 콘셉트 잡기: 누가 읽을지부터 정한다

"쓰고 싶은 주제는 있는데, 막상 한 줄로 적어보라면 막혀요."

책 작업을 시작하면 가장 많이 마주하는 벽이다. 1장의 5개 질문에 답하다 보면 "내가 진짜 어떤 책을 쓰고 싶은 거지?"를 두세 번 다시 적게 된다. 자연스러운 일이다.

이 장의 목표는 책의 중심을 **단 세 줄로 못 박는 것**이다.

- 한 줄 요약** — 이 책은 무엇인가
- 페르소나** — 이 책을 읽는 한 사람은 누구인가
- 약속** — 다 읽은 독자가 무엇을 손에 쥐는가

이 세 줄이 흔들리면 8주가 흔들린다. 반대로 이 세 줄만 단단하면 나머지는 따라온다.

왜 콘셉트가 가장 먼저인가

빈 마크다운 파일 앞에 앉으면 바로 떠오르는 충동이 있다 — **차례부터 짜고 싶다**. "1장은 이거, 2장은 저거..." 한 번에 차례를 만들어 두면 뭔가 시작한 기분이 든다.

함정이다. 차례는 **콘셉트의 결과물**이지 출발점이 아니다.

| 콘셉트 없이 차례부터 | 콘셉트 먼저 |
|-------------------------------|-------------------------------|
| 챕터 제목이 모호함 ("AI 활용", "도구 소개") | 챕터마다 독자에게 줄 것이 명확 |
| 작업 후반에 절반은 "이거 빼야 하나?"로 회의 | 빼야 할 챕터가 처음부터 안 들어옴 |
| 표지·디자인 결정마다 "톤이 뭐였더라?" | 한 줄 요약이 디자인까지 결정 |
| 독자 피드백을 받아도 어디에 반영할지 모름 | 페르소나가 "이건 그 사람에게 필요한가?" 답을 준다 |

콘셉트 = 8주 동안 모든 결정의 기준선이다. 차례·문체·표지·심지어 피드백에 어떻게 답할지까지 여기서 갈린다.

1단계 — 한 줄 요약 (One-line Summary)

좋은 한 줄 요약의 조건

- 누가 읽으면 좋은지가 명시되어 있다
- 무엇을 다루는지가 한 단어로 압축된다
- 읽은 뒤 어떻게 변하는지가 암시된다

나쁜 예 vs 좋은 예

나쁜 예 ❌

"AI를 활용한 글쓰기에 관한 책"

— 누가 읽는지, 무엇을 얻는지 알 수 없다.

좋은 예 ✅

"한 번도 책을 써본 적 없는 사람이 Claude와 8주 동안 한 챕터씩 써가며 자기만의 50p 전자책을 손에 쥐는 책."

— 독자(처음 쓰는 사람), 도구(Claude), 기간(8주), 결과물(50p PDF)이 한 줄에 다 들어가 있다.

Claude 프롬프트 — 한 줄 요약 다듬기

1장에서 받은 5개 질문 답변을 그대로 들고 가서 다음 프롬프트를 던진다.

다음은 내가 만들고 싶은 책에 대한 답변이다.

[1장의 5개 답변 붙여넣기]

이 답변을 기반으로 책의 한 줄 요약 후보 5개를 만들어줘.

각 후보는 다음 조건을 모두 충족해야 한다:

1. 50자 이내
2. 누구를 위한 책인지 명시
3. 읽으면 무엇을 얻는지 암시
4. 같은 주제의 다른 책과 차별화 포인트 1개

5개를 표로 정리하고, 각 후보의 강점·약점도 한 줄씩 적어줘.

Claude가 5개 후보를 표로 준다. 그중 가장 마음에 드는 것 1개를 고르거나, 두 개를 합쳐 다시 다듬는다. 다듬은 결과를

`book/README.md` 최상단에 박는다.

💡 5개를 한 번에 받아 비교하는 게 핵심이다. "괜찮은 한 줄"을 단번에 만들려 하지 말고, 선택지를 펼쳐놓고 고르는 작업으로 바꾼다. Claude가 가장 잘하는 일이 바로 이것이다.

2단계 — 페르소나 (한 사람 정하기)

"모두를 위한 책"의 함정

처음 책을 쓰는 사람이 가장 흔히 빠지는 함정 — "누구나 읽을 수 있는 책" 으로 만들려는 것이다. 결과는 정반대다. 누구를 위한지 모르는 책은 누구도 끝까지 읽지 않는다.

해법은 단순하다 — 한 사람을 떠올리고 그 사람에게 말하듯 쓴다.

페르소나 카드 템플릿

book/research/notes/persona.md 파일을 만들고 다음을 채운다.

```
# 페르소나 - 이 책의 첫 독자 한 사람

## 기본 정보
- 이름(가명): 김지영
- 나이: 34세
- 직업: 마케터 (회사원 7년 차)
- 거주: 서울

## 책과 관련된 상황
- 글쓰기 경험: 회사 보고서·제안서 정도. 개인 글쓰기는 SNS만.
- AI 도구 경험: ChatGPT는 이메일 초안에 가끔 사용. Claude는 처음.
- 책을 쓰고 싶은 이유: 자기 분야 노하우를 정리해서 이직·강의 자료로
- 가장 큰 두려움: "분량을 채울 수 있을까?", "AI가 다 써버린 책이라고 욕먹지 않을까?"

## 일주일 시간 가용성
- 평일 저녁: 30분 ~ 1시간 (피곤할 때는 패스)
- 주말: 2~3시간 (둘째 주말은 대부분 외출)

## 이 책에 대한 기대
- 매주 1챕터씩, 부담 없이 따라할 수 있길
- 코드 같은 거 없이 대화창만으로 끝낼 수 있길
- 8주 후 정말 PDF가 손에 들리길
```

Claude 프롬프트 — 페르소나 만들기

내가 쓰려는 책의 한 줄 요약은 이거야:
"[1단계에서 정한 한 줄 요약]"

이 책의 첫 독자가 될 한 사람을 페르소나 카드 형태로 만들어줘.

조건:

- 가상의 인물이지만 구체적 (이름·나이·직업·거주지)
- 글쓰기/AI 도구 경험 수준 명시
- "이 책을 왜 쓰고 싶은가"의 동기 1~2개
- "가장 큰 두려움" 1~2개
- 일주일 시간 가용성 (평일/주말 각각)
- 이 책에 거는 기대 3가지

출력은 위 항목을 갖춘 마크다운 페르소나 카드.

받은 결과를 `persona.md` 에 그대로 저장한다. 이후 모든 챕터를 쓸 때 "이 문장이 이 사람에게 닿는가?" 를 자문하는 게 페르소나의 유일한 역할이다.

⚠ 페르소나는 현실의 특정 인물을 그대로 복사하지 않는다. 가상의 합성 인물이 안전하고 솔직한 글을 만든다.

3단계 — 약속 (Promise)

약속이란?

"이 책을 다 읽은 독자가 이 한 가지는 반드시 손에 쥘다." 라는 한 문장이다. 마케팅 카피가 아니라 **저자가 자신과 맺는 계약**이다.

이 약속이 흐릿하면 챌린지 중간에 "이 챕터를 굳이 써야 하나?" 같듯이 생길 때 기준이 없다. 명확하면 — 약속에 기여하지 않는 챕터는 **빠는 게 맞다**는 결정을 빠르게 내릴 수 있다.

약속의 좋은 예

| 책 종류 | 약속 |
|----------------|-------------------------------|
| 이 책 「나만의 책만들기」 | 8주 후 자기만의 50p 전자책 PDF |
| 가상 예 — 운동 책 | 12주 뒤 푸시업 0개 → 30개 |
| 가상 예 — 요리 책 | 4주 뒤 손님 4명을 30분 만에 차려낼 코스 1세트 |
| 가상 예 — 글쓰기 책 | 한 달 뒤 매일 200자씩 멈추지 않고 쓰는 사람 |

공통점 — **검증 가능하다**. "글을 잘 쓰게 됩니다" 같은 모호한 표현이 아니라, 다 읽은 사람이 **자기가 약속을 받았는지 안 받았는지 스스로 판단**할 수 있어야 한다.

Claude 프롬프트 — 약속 다듬기

내 책 정보:

- 한 줄 요약: "[1단계 결과]"
- 페르소나 핵심: "[2단계 페르소나의 핵심 동기·기대 3줄]"

이 책의 "약속(promise)"을 만들어줘.

조건:

- 다 읽은 독자가 손에 쥐는 구체적 결과 1가지
- 검증 가능 (독자가 스스로 "받았다/못 받았다" 판단 가능)
- 분량/기간이 들어가면 더 좋음
- 후보 3개를 강도 순으로 정렬, 각 후보의 검증 방법도 함께

출력은 후보 3개 + 추천(왜 그것을 추천하는지 이유).

후보 3개 중 하나를 골라(또는 다듬어서) `book/README.md` 에 "약속" 섹션으로 박는다. 이후 챕터 검토 시 항상 이 약속을 기준으로 한다.

종합 — 콘셉트 시트 한 페이지

세 단계가 끝나면 한 페이지짜리 **콘셉트 시트**가 나온다. 이걸 `book/README.md` 최상단에 둔다.

책 콘셉트 시트

한 줄 요약

> [1단계 결과 - 50자 이내]

페르소나

- 이름: [가명]
- 직업·나이: [예: 마케터, 34세]
- 핵심 동기: [한 줄]
- 가장 큰 두려움: [한 줄]
- 시간 가용성: [평일 N분 / 주말 N시간]

약속

> [3단계 결과 - 검증 가능한 한 문장]

좋은 콘셉트 시트의 체크리스트

- 한 줄 요약을 처음 보는 사람에게 보여줘도 "어, 그거 누가 읽으면 좋겠다" 떠오르는가?
- 페르소나가 가족·친구 중 한 명을 떠올리게 하는가?
- 약속을 8주 후 스스로에게 "지켰다/못 지켰다" 판정할 수 있는가?
- 세 줄을 합쳐서 한 친구에게 30초 안에 설명할 수 있는가?

체크가 다 붙으면 3장(차례 짜기)으로 넘어갈 준비가 끝났다. 하나라도 안 붙으면 — 다음 장으로 넘어가지 말고 그 한 항목을 다듬는 데 시간을 더 쓴다. 콘셉트가 흔들리는 채로 차례를 짜면 무조건 다시 짬다.

함정 4가지 — 콘셉트 단계에서 흔히 빠지는 실수

1. "AI를 활용한 OO" 같은 메타 표현

"AI를 활용한 글쓰기", "AI로 만드는 OO" — 어디서나 들리는 표현이라 차별화가 안 된다. AI 도구는 수단이지 주제가 아니다. 약속한 줄에서 'AI'라는 단어를 빼고도 매력이 남는지 확인한다.

2. 페르소나를 "본인 자신"으로 정하기

"내가 모르던 시절의 나를 위한 책" — 흔한 실수다. 본인의 과거는 너무 가까워 객관화가 어렵고, 결과적으로 **나만 이해할 수 있는 책**이 된다. 가족·친구·동료 중 **현재 그 상황에 있는 한 사람**을 떠올리는 편이 훨씬 강하다.

3. 약속의 단위가 너무 큼

"인생이 바뀝니다", "전문가가 됩니다" — 검증 불가능하다. 8주가 끝나는 날 독자가 **물리적으로 손에 들고 보일 수 있는 결과** 또는 **그 자리에서 보여줄 수 있는 행동**이 좋은 약속이다.

4. 콘셉트 시트를 다른 사람에게 안 보여주기

머릿속에서는 명확해도 글로 적으면 모호해진다. **콘셉트 시트를 한 사람에게 보내고 한 줄 반응을 받는다.** 가장 빠른 검증이다.

자주 묻는 질문

Q. 한 줄 요약이 도저히 안 나온다. 그냥 넘어가도 되나? 아니다. 여기서 막히는 건 작은 문제가 아니라 큰 신호다. 두 가지 가능성이 크다 — (1) 주제 자체가 너무 광범위함, 또는 (2) 본인이 정말 쓰고 싶은 게 무엇인지 아직 모름. 어느 쪽이든 **한 주 더 쓰는 게 8주 전체를 살린다**. Claude에게 "내가 떠올린 답변에서 모순되는 지점을 짚어줘"라고 부탁해보자.

Q. 페르소나를 여러 명 만들어도 되나? 가능하지만 추천하지 않는다. 페르소나는 의사결정 도구다. 둘 이상이 되면 갈등하는 순간 결정이 안 난다. **주 페르소나 1명 + 부 페르소나 1명(선택)** 이 최대치다.

Q. 약속을 너무 좁게 잡으면 책이 작아 보이지 않을까? 오히려 반대다. 좁은 약속이 진짜 가치를 준다. "글쓰기 잘하기"보다 "30일 동안 매일 200자씩 멈추지 않고 쓰기"가 훨씬 강한 책이다. 좁힌 약속이 실현되면 독자는 **그 너머도 알아서 간다**.

Q. 한 줄 요약이 다른 책과 비슷해 보이는데? 비슷해도 괜찮다. 비슷한 책이 있다는 건 시장이 있다는 뜻이다. 차별화는 한 줄이 아니라 **페르소나 + 약속 + 톤**의 조합에서 나온다.

이 장의 정리

오늘 세 가지만 끝내면 된다.

- Claude와 함께 **한 줄 요약** 후보 5개 → 1개 확정 → **README.md** 최상단
- **페르소나 카드** 작성 → **research/notes/persona.md**
- **약속** 한 문장 → **README.md** 에 박기
- (보너스) 콘셉트 시트를 한 사람에게 보내 한 줄 반응 받기

세 줄이 단단해졌다면 8주의 가장 어려운 고비를 넘긴 셈이다. 다음 장에서는 이 콘셉트 시트를 들고 **차례(outline)** 를 짠다.

3장 — 차례 짜기: outline이 진실의 원천이다

"차례를 짤 적이 없어서, 어디서부터 손을 대야 할지 모르겠어요."

책을 처음 쓰는 사람에게 가장 자주 듣는 말이다. 솔직히 누구나 매번 헛갈린다 — "한 챕터에 어디까지 담아야 하지?", "정해진 기간 안에 진짜 50p가 나올까?" 같은 질문 말이다.

1장에서 도구와 작업 폴더를 잡고, 2장에서 콘셉트를 단단히 했다면, 이번 장은 그 콘셉트를 정해진 기간 안에 어떻게 펼칠지 차례로 옮기는 단계다.

결론부터 말하면 — **outline(차례)**은 단순한 목차가 아니라, 책 작업 모든 결정의 진실의 원천(single source of truth)이다. 챕터를 추가하거나 옮기거나 합칠 때 항상 **outline에서 먼저 결정**하고, 파일과 문단은 그 결정을 따라온다. 이 순서가 흔들리면 책 전체가 흔들린다.

왜 outline이 "진실의 원천"인가

책 작업을 처음 시작하면 흔히 **outline 없이 챕터 파일부터** 만든다. 결과는 예상대로다 — 작업 중반쯤 되면 같은 내용이 두 챕터에 겹쳐 있고, 어떤 챕터는 1페이지짜리, 어떤 챕터는 15페이지짜리가 되어 있다. "이 책의 전체 모습이 머리에 안 그려진다"는 게 가장 큰 문제다.

outline 한 장이 있으면 다음 세 가지가 한 번에 해결된다.

1. 전체 분량이 한눈에 보인다

8개 챕터 × 평균 6페이지 = **50p**. 책의 크기가 차례에서 결정된다. 한 챕터가 두꺼워질 조짐이면, 다른 챕터를 줄이거나 아예 챕터를 분할해야 한다는 신호다.

2. 중복과 누락이 사전에 잡힌다

"이 얘기는 3장에서 했나, 5장에서 할까?" 같은 고민이 사라진다. **차례가 답을 미리 갖고 있기** 때문이다.

3. 챕터 작업 순서가 자유로워진다

차례가 안 잡히면 무조건 1장부터 써야 한다. 차례가 잡히면 **준비된 챕터부터** 써도 책 전체가 흔들리지 않는다.

💡 **빌드 인 퍼블릭의 안전장치** — 매주 한 챕터씩 공개로 발행할 때, outline이 있으면 "이 글이 책 어디에 들어갈지" 독자도 알 수 있다.

분량 잡기 — 50p, 8장이 왜 적정선인가

분량의 기준선

| 분량 | 형태 | 작업 기간 | 적합한 첫 책 |
|---------------|---------------|--------------|------------------|
| 20p 이하 | 리포트·소책자 | 1~2주 | △ (책이라 부르기 애매) |
| 30~80p | 전자책 표준 | 6~10주 | ◎ (가장 추천) |
| 100p 이상 | 본격 단행본 | 6개월~ | × (첫 책으로 부담) |

첫 책은 50p 안팎이 가장 안전하다. 8주에 한 챕터씩 약 6페이지를 쓰는 페이스가 — 일하면서, 다른 글도 쓰면서 — 무리 없이 굴러가는 거의 유일한 속도다.

8장이라는 숫자

8장은 마법의 숫자가 아니다. 8주 챌린지에 1챕터 = 1주를 매핑한 결과다. 5주 챌린지면 5장, 12주면 12장이 된다. 시간 단위가 챕터 단위가 되는 것이 가장 단순하고 어기지 않는 방식이다.

8주가 부담스러우면 — 8장에서 다루는 "8일 스프린트 모드"가 대안이다. 매일 1챕터씩 8일에 끝내는 압축 변형이다.

챕터 분할의 황금률 — "한 챕터 = 한 메시지"

차레를 짤 때 가장 자주 빠지는 함정은 "챕터 = 주제"로 묶는 것이다. 더 나은 기준은 "챕터 = 한 줄로 말할 수 있는 메시지"다.

나쁜 예 vs 좋은 예

| 나쁜 챕터 제목 (주제) | 좋은 챕터 제목 (메시지) |
|---------------|-------------------------|
| 도구 소개 | 시작 전 준비 — 도구 4가지면 충분하다 |
| 컨셉트 | 컨셉트 잡기 — 누가 읽을지부터 정한다 |
| 차레 | 차레 짜기 — outline이 진실의 원천 |
| 자료 | 자료 모으기 — 흩어진 글을 책 자료로 |

좋은 챕터 제목은 그 챕터의 주장을 한 줄에 담고 있다. 챕터를 다 읽지 않아도 차레만 봐도 책이 무슨 말을 하려는지 알 수 있다.

한 챕터에 메시지가 둘이라면?

둘로 나눈다. 이 책의 8장은 원래 "마무리 + 다음 책"이라는 두 메시지였는데, 부록 E (8일 스프린트 모드)를 분리해서 하나로 정리했다. 분리하지 않으면 챕터가 두꺼워지고 독자가 길을 잃는다.

이 책의 outline (실제 사례)

이 책이 따라가는 outline이다. 지금 읽고 있는 이 장이 3장 자리에 들어가 있다.

| # | 챕터 | 한 줄 메시지 |
|----------|--------------|--------------------------|
| 0 | 머리말 | 왜 지금 책인가 |
| 1 | 시작 전 준비 | 도구 4가지면 충분하다 |
| 2 | 컨셉트 잡기 | 누가 읽을지부터 정한다 |
| 3 | 차례 짜기 | outline이 진실의 원천이다 |
| 4 | 자료 모으기 | 흩어진 글을 책 자료로 |
| 5 | 챕터 쓰기 | 빌드 인 퍼블릭의 힘 |
| 6 | 다듬기·디자인 | 표지·다이어그램·문체 |
| 7 | 빌드·배포 | pandoc 한 줄로 PDF까지 |
| 8 | 마무리·부록 | 회고와 다음 책 + 8일 스프린트 |

표 자체가 메타 사례다 — 이 책의 차례를 보면, 이 책이 무슨 말을 할지 처음부터 다 알 수 있다. 그게 좋은 outline의 증거다.

실습 1 — 빈 outline.md 만들기

1장에서 만든 `book/` 폴더 안에 `outline.md` 파일을 만든다. 첫 버전은 거칠어도 된다. 다음 템플릿을 그대로 복사하자.

```
# Outline - [내 책 제목]

## 책 한 줄 요약
> [2장에서 정한 한 줄 요약]

## 대상 독자
- [페르소나]

## 챕터 구성

| # |  가제  | 한 줄 메시지 |
|---|-----|-----|
| 0 | 머리말 | (왜 이 책인가) |
| 1 | (제목) | (메시지) |
| 2 | (제목) | (메시지) |
| 3 | (제목) | (메시지) |
| 4 | (제목) | (메시지) |
| 5 | (제목) | (메시지) |
| 6 | (제목) | (메시지) |
| 7 | (제목) | (메시지) |
| 8 | 마무리 | (회고·다음 단계) |

## 부록 (선택)
- A. ...
- B. ...
```

비개발자 — 메모장·Notion으로

위 템플릿을 메모장이나 Notion 한 페이지에 그대로 옮겨둔다. 표 형태가 어려우면 단순 목록으로 시작해도 된다. 형식은 나중에 정돈해도 늦지 않다.

개발자 — VS Code 한 줄로

```
cd book && touch outline.md
```

이 파일이 앞으로 모든 챕터 결정의 시작점이 된다.

실습 2 — Claude로 outline 검증하기

빈 칸을 다 채웠다면, Claude 대화창을 열고 다음 프롬프트를 그대로 붙여넣는다.

다음은 내가 8주 안에 쓰려는 전자책의 outline이다.

[outline.md 전체 내용 붙여넣기]

다음 5가지 관점에서 검토해줘.

1. 챕터 8개의 메시지가 서로 겹치지 않는가?
2. 한 챕터에 메시지가 둘 이상 들어 있는 챕터는?
3. 8주 만에 완성하기에 분량이 무리인 챕터는?
4. 누락된 단계나 챕터가 있다면?
5. 챕터 순서를 바꾸면 더 자연스러워질 부분은?

각 항목에 대해 구체적인 챕터 번호와 함께 답해줘.

Claude가 항목별로 짚어주면 — 모두 받아들이지는 않는다. 이 책의 outline도 Claude가 "8장이 너무 두껍다"고 지적해서, **부록 E (8일 스프린트 모드)** 를 분리한 결과로 지금 모습이 됐다. 검토를 반영할지는 사람의 결정이다.

outline 운영 원칙 — 변경은 항상 outline에서 먼저

작업하는 동안 outline은 한 번도 안 바뀌지 않는다. 반드시 바뀐다. 중요한 건 변경의 순서다.

✅ 옳은 순서

1. outline.md 표를 먼저 수정
2. 그에 맞춰 chapters/, drafts/ 파일을 옮기거나 합치거나 분리
3. 작업 일지에 변경 사실 한 줄 기록

❌ 흔한 실수

1. 새 챗터 파일을 먼저 만들기
2. outline.md는 나중에 갱신
→ 두 곳이 어긋나기 시작 → 어느 쪽이 진실인지 헷갈림

outline.md가 진실, 파일은 그 그림자다. 이 한 문장만 지켜도 책 작업이 무너지지 않는다.

마음가짐 — 완벽한 outline은 없다

✅ 1차 outline은 30분 안에 끝낸다

8개 챗터의 주제와 한 줄 메시지를 30분 안에 채운다. **완벽하지 않아도 된다.** outline은 작업 내내 매주 한 번씩 다듬는다는 전제로 작성한다.

✅ 매주 한 번 outline 훑기

작업 일지를 쓰는 날, outline.md를 한 번 다시 본다. **이번 주 챗터 작업이 다음 챗터의 메시지를 바꿨다면,** 그 자리에서 outline을 갱신한다.

❌ 첫 outline부터 완벽하게 잡으려 하지 않기

처음에 완벽하게 잡으려다 첫 챗터를 한 달 동안 못 쓰는 경우가 흔하다. **거친 outline + 매주 발행이, 완벽한 outline + 무발행보다 항상 낫다.**

자주 묻는 질문

Q. 챗터 수가 꼭 8개여야 하나? 아니다. 8주 챌린지에 맞춘 결과다. 5주면 5장, 12주면 12장. **시간 단위 = 챗터 단위**가 가장 무리 없이 굴러간다.

Q. 중간에 챗터가 늘어나면 일정이 망가지지 않나? outline에서 다른 챗터를 통합하거나 부록으로 옮기면 챗터 수는 유지된다. "총 8개"라는 외부 약속은 지키되, 내부 구성은 자유롭게 바꿀 수 있다.

Q. outline을 꼭 마크다운으로 써야 하나? Notion·구글 문서 어느 곳이어도 된다. 핵심은 **한 곳에 모이고, 변경 이력이 남는 것**이다. 마크다운은 7장(빌드)에서 pandoc으로 PDF까지 그대로 변환되는 장점이 있다.

Q. 챗터 한 줄 메시지가 잘 안 떠오른다. "이 챗터를 다 읽은 독자가 다른 사람에게 한 줄로 뭐라고 말할까?" 를 자문한다. 답이 한 문장으로 안 나오면, 그 챗터는 메시지가 둘 이상 섞여 있다는 신호다.

이 장의 정리

이 장에서 할 일은 두 가지다.

- `book/outline.md` (또는 노트앱 한 페이지)에 챕터 8개 가제 + 한 줄 메시지 채우기
- Claude에게 outline 검토 5가지 질문 → 받은 피드백 중 반영할 것 1~2개만 골라 적용

차례가 잡히면, 다음 장에서는 그 **outline에 자료를 채우는 작업**을 시작한다. 흠어진 글·메모·블로그 포스트가 어떻게 책의 한 챕터가 되는지를 4장에서 다룬다.

4장 — 자료 모으기: 흩어진 글을 책 자료로

"막상 쓰려니 빈 화면만 깜빡거려요. 50페이지를 어떻게 다 채우죠?"

책 작업에서 가장 먼저 마주하는 감정이다. 1장에서 도구와 폴더를 잡고, 2장에서 콘셉트를, 3장에서 차례(outline)까지 세웠다. 이제 그 빈 차례 칸을 **글로 채울 차례**인데 — 이 장은 그 첫 삽을 "**새로 쓰기**"가 아니라 "**모으기**"로 뜨는 이야기다.

결론부터 말하면 — **책의 첫 자료는 새로 쓰는 게 아니라, 이미 써둔 것을 모으는 데서 나온다.** 누구나 생각보다 많은 글을 이미 갖고 있다. 블로그 글, 메모, 누군가에게 보낸 긴 답장, SNS에 끄적인 생각. 흩어져 있을 뿐이다. 이 장에서 할 일은 그것들을 **한곳에 모으고, 어느 챕터의 재료인지 표시**하는 것이다.

왜 "새로 쓰기"보다 "모으기"가 먼저인가

빈 화면이 무서운 이유는 단순하다. **0에서 1을 만들어야 하기 때문이다.** 그런데 책 한 권의 대부분은 0에서 시작하지 않는다. 어떤 주제로 책을 쓰기로 했다는 건, 이미 그 주제를 **여러 번 말하고 써봤다**는 뜻이다.

이 책 「나만의 책만들기」의 절반도 **이미 블로그에 흩어져 있던 글**이었다.

- 작업 폴더 세팅 → 예전 "개발 환경" 글
- 빌드 인 퍼블릭 운영 → 매주 쓰던 **작업 일지**
- 자동화 워크플로우 → 30분 만에 만든 자동화 사례 글

새로 쓴 게 아니라, **이미 쓴 것을 책의 언어로 옮긴** 것이다. "모으기"를 먼저 하면 빈 화면이 **반쯤 찬 화면**으로 바뀐다. 1에서 2를 만드는 일은 0에서 1보다 훨씬 쉽다.

핵심: **수집은 창작보다 심리적 저항이 낮다.** 가장 막막한 첫 단계를, 가장 쉬운 작업으로 시작하는 전략이다.

발췌와 재구성을 분리하라 — 폴더 두 개의 원칙

자료 모으기에서 가장 중요한 한 가지를 먼저 말한다. "**가져오는 일**"과 "**고쳐 쓰는 일**"을 **절대 한 번에 하지 않는다.** 두 작업은 머리를 쓰는 방식이 완전히 다르다. 섞으면 둘 다 느려진다.

그래서 자료 폴더를 둘로 나눈다.

```
book/drafts/  
├── from-blog/    # 1단계: 발췌 - 원본을 그대로 복사 (손대지 않음)  
└── reworked/    # 2단계: 재구성 - 책 톤으로 다시 쓰는 작업장
```

1단계 — 발췌(extract): 원본 그대로, 출처만 남긴다

발체는 **편집하지 않는다**. 원본 글을 그대로 복사해 오고, 맨 위에 **출처와 가져온 날짜**만 주석으로 단다. 이 단계의 목적은 단 하나 — **"이 챕터에 쓸 재료가 여기 있다"**는 사실을 확보하는 것이다.

왜 안 고치냐고? 발체하면서 문장을 다듬기 시작하면, 자료 10개를 모으는 데 하루가 다 간다. 모으기는 **빠르게, 양으로** 가야 한다. 다듬기는 그다음이다.

2단계 — 재구성(rework): 책의 언어로 다시 쓴다

`reworked/` 에서 비로소 책 톤으로 고친다. 블로그 글은 "그때 그 순간"의 호흡이고, 책은 "처음 보는 독자"의 호흡이다. 시점을 바꾸고, 앞뒤 챕터와 연결하고, 블로그 특유의 시의성("지난주에")을 걷어낸다. 이 작업은 **5장(챕터 쓰기)** 에서 본격적으로 다룬다.

왜 굳이 폴더를 나누나

- **원본 보존** — 재구성하다 길을 잃어도 `from-blog/` 의 원본은 그대로 있다. 언제든 처음으로 돌아갈 수 있다.
- **진척이 눈에 보인다** — `from-blog/` 에 파일이 쌓이는 것만으로 "자료가 모이고 있다"는 실감이 난다. 빌드 인 퍼블릭의 동력이다.
- **출처 추적** — 나중에 부록 D(발행 글 링크 모음)를 만들 때, 어떤 챕터가 어느 원본에서 왔는지 한눈에 보인다.

자료는 어디서 모으나 — 네 곳

빈손이라고 느껴져도, 보통 이 네 곳에 재료가 숨어 있다.

1. **내 블로그·SNS 글** — 가장 큰 광맥. 이미 다듬어진 문장이다.
2. **메모·노트앱** — Notion, 메모장, 카톡 "나에게 보내기". 정제는 안 됐어도 **생각의 원석**이다.
3. **남에게 보낸 긴 답변** — 질문에 진심으로 답한 글은 이미 "독자를 향한" 글이다. 책 본문에 가장 가깝다.
4. **외부 자료·인용** — 공식 문서, 통계, 인용하고 싶은 책. 단, 출처를 반드시 `research/references.md` 에 기록한다.

3장에서 만든 outline의 **"발체 후보"** 칸을 기억하는가. 지금이 그 칸을 채울 때다. 모은 자료를 챕터 번호에 하나씩 붙여 나간다.

실습 1 (개발자) — `mirror-from-blog.sh` 한 줄로 발체

개발자라면 발체를 손으로 복사·붙여넣기 할 필요가 없다. 원본 경로만 넘기면 출처 주석을 달아 `from-blog/` 에 떨어뜨리는 스크립트 하나면 된다.

```
# 블로그 글 한 편을 책 작업 폴더로 발체
./scripts/mirror-from-blog.sh contents/posts/claude-code/qa-automation-guide/index.md

# 출력: book/drafts/from-blog/qa-automation-guide-extract.md
# - 상단에 "출처 경로 · 미러 시각" 주석 자동 추가
# - 본문은 원본 그대로 (편집 없음)
```

스크립트가 하는 일은 세 가지뿐이다 — ① 원본을 읽고, ② 맨 위에 출처·시각 주석을 붙이고, ③ `from-blog/` 에 복사. **원본은 절대 건드리지 않는다**. "발체는 편집하지 않는다"는 원칙이 스크립트로 강제되는 셈이다.

스크립트 전문은 **부록 C**에 그대로 실행한다. 직접 만들고 싶다면 Claude에게 "원본 마크다운을 출처 주석과 함께 다른 폴더로 복사하는 bash 스크립트"를 요청하면 5분이면 나온다.

실습 2 (비개발자) — 폴더 두 개와 "출처 한 줄"

스크립트가 없어도 원칙은 똑같다. 손으로 하면 된다.

1. 작업 폴더에 하위 폴더 두 개를 만든다 — **발췌**, **재구성** (영어로 **from-blog**, **reworked** 라도 좋다).
2. 책에 쓸 만한 글을 찾으면 **그대로 복사해 발췌** 폴더에 새 문서로 붙여넣는다.
3. 문서 맨 위 한 줄에 출처를 적는다 — **출처: 2025-11-03 블로그 "ooo" / 가져온 날: 오늘**.
4. 끝. 다듬고 싶은 충동이 들어도 참는다. 다듬기는 **재구성** 폴더에서, 다음 장에서.

Notion이라면 **데이터베이스 한 개**에 "원문 / 출처 / 매핑 챕터 / 상태(발췌·재구성·완료)" 열을 두는 것으로 충분하다.

모으기의 함정 — 이것만 피하기

❌ 1. 원본을 고치기 시작한다

발췌하다 "이 문장 어색한데" 하고 손대는 순간, 모으기가 글쓰기로 바뀌고 속도가 죽는다. **발췌 폴더에선 읽기만, 쓰기는 재구성 폴더에서.**

❌ 2. 출처를 안 남긴다

3주 뒤 "이 단락 어디서 가져왔지?"가 반드시 온다. 출처 한 줄이 부록 D와 인용 표기를 살린다. **가져올 때 바로** 적는다. 나중은 없다.

❌ 3. 모으기만 하고 안 쓴다

수집은 중독성이 있다. 자료가 20개 쌓이면 "더 모아야지"가 아니라 **"이제 충분하다"**가 정답이다. 한 챕터에 자료 2~3개면 차고 넘친다.

자주 묻는 질문

Q. 모을 글이 정말 하나도 없으면? 그럼 "쓴 글"이 아니라 "한 말"을 모은다. 누군가에게 이 주제를 설명했던 카톡, 댓글, 음성 메모를 텍스트로 옮기면 그게 1차 자료다. 완전한 백지에서 시작하는 책은 거의 없다.

Q. 남의 글·자료를 발췌해도 되나? 인용은 되고, 복제는 안 된다. 남의 글은 "출처를 밝힌 짧은 인용"까지만.

`research/references.md`에 URL과 인용 범위를 기록하고, 본문에선 내 언어로 요약한다. 통째로 옮기는 건 발췌가 아니라 표절이다.

Q. 발췌와 재구성을 꼭 따로 해야 하나? 나눌수록 빨라진다. 한 번에 하면 글 한 편당 30분씩 걸려 첫 단계에서 지친다. 발췌는 양으로 빠르게(편당 2분), 재구성은 질로 천천히(편당 30분)로 분리하면 진척이 훨씬 잘 보인다.

Q. 블로그 글을 책에 그대로 써도 SEO에 문제 없나? 책(PDF/EPUB)과 블로그는 검색 인덱스가 분리되어 중복 문제는 거의 없다. 다만 책에 실을 땐 어차피 `reworked/` 에서 톤을 바꾸므로 문장이 달라진다. 같은 내용을 **다른 매체에 맞게 두 번 쓰는 것**이라고 보면 된다.

이 장의 정리

오늘 두 가지만 하면 된다.

- 작업 폴더에 `from-blog · reworked` (발체·재구성) 폴더 두 개 만들기
- 책에 쓸 글 **3개**를 발체 폴더로 가져오고, 각 문서 맨 위에 **출처 한 줄** 적기

딱 3개면 된다. 완벽한 자료가 아니라, **"모으기 근육"**을 **한 번 써보는 것**이 이 장의 목표다. 다음 장에서는 그 발체가 어떻게 "책의 한 장"으로 바뀌는지를 다룬다.

5장 — 챗터 쓰기: 빌드 인 퍼블릭의 힘

"자료는 모았는데, 이걸 어떻게 '책 문장'으로 바꾸죠? 블로그 글을 그냥 붙이면 안 되나요?"

4장에서 자료를 모은 사람이 곧바로 마주하는 질문이다. 발췌 폴더(`from-blog/`)에 재료가 쌓였으니, 이제 `reworked/` 에서 한 챗터를 실제로 써낼 차례다.

결론부터 말하면 — 챗터 쓰기는 "새 글짓기"가 아니라 "발췌를 처음 보는 독자의 호흡으로 다시 쓰는 일"이다. 그리고 한 가지 더 — 그 일을 혼자 조용히 하지 않고, 공개하며 한다. 이 책의 모든 장이 그렇게 쓰였다. 왜 공개하며 쓰는 게 글을 끝까지 끌고 가는지, 이 장에서 함께 풀어본다.

블로그 글과 책 문장은 호흡이 다르다

발췌한 블로그 글을 그대로 책에 붙이면 어딘가 어색하다. 이유는 읽는 사람의 위치가 다르기 때문이다. 블로그 독자는 "이번 주 연재"를 따라오는 사람이고, 책 독자는 아무 맥락 없이 한 페이지를 펼친 사람이다. 세 가지를 바꿔야 한다.

1. 시의성을 걷어낸다

블로그는 시간 위에 떠 있다 — "지난주에", "요즘", "최근 업데이트로". 책은 1년 뒤에 읽혀도 어색하지 않아야 한다. "지난주에 겪은 일"은 "한번은 이런 일이 있었다"로 시점을 풀어준다.

2. 빠진 맥락을 채운다

연재 독자는 3장을 읽고 4장을 본다. 책 독자는 그 장만 펼칠 수도 있다. 그래서 블로그에선 생략했던 전제를 한 문장으로 다시 깔아 줘야 한다. "(앞에서 만든 outline 기억하시죠)"가 아니라 "책의 차례는 작업의 기준점이 된다 — 앞 장에서 만든 그것 말이다"처럼.

3. 앞뒤 챗터와 잇는다

책은 장과 장이 강물처럼 이어진다. 챗터 끝에 다음 장으로 넘어가는 한 문장, 시작에 앞 장을 받는 한 문장을 둔다. 이 연결고리가 "글 모음"을 "책"으로 만든다.

핵심: 블로그 글은 재료이지 완성품이 아니다. 같은 내용을 다른 독자에게 다시 들려준다고 생각한다.

`reworked` 한 챗터를 쓰는 4단계

막막할 땐 절차로 만든다. 한 챗터를 다음 순서로 쓴다.

- 발췌 다시 읽기 — `from-blog/` 의 원본을 한 번 읽되, 고치지 않고 읽는다. "이 챗터의 핵심이 뭐였지"만 잡는다.
- 한 줄 메시지 재확인 — 3장에서 만든 outline에서 이 챗터의 한 줄 메시지를 다시 본다. 쓰다가 길을 잃으면 항상 여기로 돌아온다.
- 골격부터 — 소제목만 먼저 — 본문을 쓰기 전에 `##` · `###` 소제목을 4~6개 박는다. 골격이 서면 절반은 끝난 것이다.

4. **살 붙이기** — 소제목 사이를 채운다. 한 번에 완벽하게가 아니라, **거칠게 끝까지** 먼저. 다듬기는 6장에서.

이 4단계의 핵심은 **"완성"이 아니라 "초고 완료"가 목표**라는 것이다. 한 챕터의 초고가 화면에 다 떠 있는 것과, 머릿속에만 있는 것은 하늘과 땅 차이다.

빌드 인 퍼블릭 — 왜 공개하며 쓰는가

이 책의 진짜 엔진은 여기 있다. 챕터를 **다 쓴 뒤** 공개하는 게 아니라, **공개하기로 했기 때문에 쓴다**. 순서가 반대다.

- **마감이 생긴다** — "정해진 날 발행"이라는 약속이 매주 한 챕터를 강제로 끝내게 한다. 혼자 쓰는 책의 90%가 미완으로 끝나는 이유는 마감이 없어서다.
- **피드백 루프가 돈다** — 발행하면 댓글·질문이 온다. 그 질문이 다음 챕터의 FAQ가 되고, 빠진 설명을 채워준다. 혼자선 절대 못 보는 시각지대다.
- **자산이 누적된다** — 발행한 글은 사라지지 않는다. 검색에 쌓이고, 독자가 모이고, 그 자체가 책의 마케팅이 된다. **글 한 편 한 편 이 책의 예고편인 셈이다**.

이 책 자체가 증거다. 각 장은 따로 쓴 원고가 아니라, 매주 연재로 발행된 글이 곧 그 장의 초고가 됐다. **연재를 쓰는 일이 곧 책을 쓰는 일이었다**.

물론 공개에는 비용이 있다 — 완벽하지 않은 글을 내보내는 부담, 반응이 없을 때의 허전함. 그래서 약속한다. **"완벽한 글"이 아니라 "이번 주의 최선"을 낸다**. 다듬기는 연재가 끝난 뒤 `reworked` → `chapters` 승격 단계에서 한 번 더 한다.

실습 1 — Claude로 "블로그 톤 → 책 톤" 변환

발췌 글을 통째로 붙이고 다음처럼 요청한다. (개발자·비개발자 공통)

```
아래는 내 블로그 글이야. 이걸 전자책의 한 챕터 초고로 바꿔줘.
조건:
1. "지난주", "최근" 같은 시의성 표현을 걸어내고 시간에 안 묶이게.
2. 이 장만 펼친 독자도 이해하도록 빠진 전제를 한 문장씩 보완.
3. 이 챕터의 한 줄 메시지는 "ooo" — 여기서 벗어나는 문단은 빼자고 제안해줘.
4. 문체는 ~합니다체, 1인칭, 담백하게.
내 문장을 최대한 살리되, 바꾼 곳은 이유를 한 줄로 알려줘.
```

[발췌 글 붙여넣기]

핵심은 마지막 줄 — **"바꾼 곳은 이유를 알려줘"** 다. Claude가 통째로 다시 쓰면 내 목소리가 사라진다. **내 문장을 살리고, 바꾼 부분만 검토**하는 방식이야 "내 책"이 된다.

실습 2 — 한 챕터 공개하고, 피드백 "1개만" 반영

1. 위 4단계로 한 챕터 초고를 끝낸다. (완벽 금지, 초고면 충분)
2. 블로그·SNS·뉴스레터 어디든 **공개**한다. 비공개 메모장에 두지 않는다.

3. 들어온 반응 중 **반영할 것 딱 1개**만 고른다. 전부 반영하려 하면 영원히 못 끝낸다.

발행 채널이 아직 없다면, 가까운 사람 한 명에게 보내는 것도 "공개"다. **읽어줄 사람이 생기는 순간 글의 밀도가 달라진다.**

챕터 쓰기의 함정 — 이것만 피하기

❌ 1. 완벽해질 때까지 발행을 미룬다

빌드 인 퍼블릭의 반대말은 "완벽주의 서랍"이다. 초고를 서랍에 넣는 순간 그 챕터는 죽는다. **80%면 내보낸다.**

❌ 2. 피드백을 전부 반영한다

열 명이 열 가지를 말한다. 다 따르면 책이 누더기가 된다. **한 줄 메시지에 맞는 피드백 1~2개만.** 나머지는 정중히 흘려보낸다.

❌ 3. 블로그 글을 그대로 복붙한다

가장 흔한 실수다. 발체는 재료일 뿐, **시점·맥락·연결**을 손보지 않으면 "책"이 아니라 "블로그 묶음"이 된다.

자주 묻는 질문

Q. 글을 공개할 채널이 없다. 그래도 빌드 인 퍼블릭이 되나? 된다. 핵심은 "관객의 존재"이지 "큰 플랫폼"이 아니다. 친구 한 명, 가족, 스터디 단톡방이면 충분하다. **읽어줄 한 사람만** 있으면 마감과 책임감이 생긴다.

Q. 한 챕터에 시간이 너무 오래 걸린다. 소제목부터 박는다(4단계의 3번). 빈 화면에 문장을 짜내려 하면 한없이 늘어난다. 골격(소제목 4~6개)을 먼저 세우면 "칸 채우기"가 되어 속도가 붙는다.

Q. Claude가 고친 문장이 더 매끄러운데, 그냥 그걸 쓰면 안 되나? 매끄러움보다 내 목소리가 먼저다. AI 문장은 평균적으로 깔끔하지만 개성이 없다. Claude는 **검토자**로 쓰고, 최종 문장은 내가 고른다. 그래야 1년 뒤에도 "내가 쓴 책"이라 말할 수 있다.

Q. 발행한 글을 나중에 또 고쳐도 되나? 얼마든지. 다만 큰 수정은 연재가 끝난 뒤 **reworked → chapters** 승격 때 한 번에 하는 걸 권한다. 매주 고치기 시작하면 새 챕터를 못 쓴다.

이 장의 정리

오늘 두 가지만 하면 된다.

- 발체 글 1편을 위 **4단계**로 "한 챕터 초고"까지 끌고 가기 (완벽 금지)
- 그 초고를 **어디든 공개**하고, 들어온 반응 중 **1개만** 반영하기

초고 하나를 끝까지 써서 세상에 내보내는 경험 — 그게 이 장의 진짜 목표다. 완성도가 아니라 **"끝내고 공개하는 근육"** 말이다. 초고를 8주에 걸쳐 다 쌓았다면, 다음 장에서는 그 더미에 **웃을 입힌다.**

6장 — 다듬기와 디자인: 표지·다이어그램·문체

"내용은 다 썼는데, 왜 아직 '책 같지'가 않을까요?"

초고를 다 쌓은 사람이 공통으로 느끼는 감정이다. 5장에서 한 챕터를 써내는 법까지 왔다면, 이제 손에는 **초고 더미**가 있다. 내용은 다 들어 있는데 어딘가 허전하다. 이 장은 그 초고에 **옷을 입히는** 이야기다 — 문체, 표지, 다이어그램.

결론부터 말하면 — **사람은 책을 읽기 전에 먼저 본다**. 내용이 아무리 좋아도, 챕터마다 말투가 다르고 표지가 없으면 독자는 "대충 만든 글"로 느낀다. 반대로 **일관된 문체 + 표지 한 장만** 있어도 같은 내용이 "한 권의 책"으로 격상된다. 다듬기는 꾸밈이 아니라 **신뢰의 문제**다.

왜 다듬기가 내용만큼 중요한가

독자가 책을 만났을 때 순서는 이렇다 — ① **표지를 본다** → ② **몇 줄 읽어본다** → ③ **말투가 일관되면 끝까지 읽는다**. 내용 평가는 마지막이다. 그 앞 두 관문을 통과 못 하면 좋은 내용도 읽히지 않는다.

특히 **무료 PDF**처럼 진입장벽이 낮은 책일수록 첫인상이 전부다. 돈을 낸 책은 아까워서라도 읽지만, 공짜 책은 표지에서 한 번, 첫 페이지에서 또 한 번 걸러진다. **다듬기는 "끝까지 읽히게 만드는" 작업**이다.

핵심: 디자인은 예쁨이 아니라 **"이 사람이 정성을 들였다"**는 **신호**다. 그 신호가 신뢰가 된다.

1. 문체 통일 — 8주를 한 사람이 쓴 것처럼

8주에 걸쳐 쓴 글은 **8주만큼 말투가 다르다**. 첫 주는 의욕 넘쳐 길고, 바쁜 주는 뚝뚝 끊기고, 어떤 챕터는 반말이 섞인다. 독자는 이 미세한 불일치를 **무의식적으로** 느낀다.

말투 기준 한 줄을 먼저 정한다

다듬기 전에 규칙을 한 줄로 박는다. 예: **"합니다체, 1인칭(저/제), 한 문단 34문장, 전문용어는 첫 등장 시 한 번 풀이."** 이 한 줄이 8개 챕터를 같은 옷으로 맞추는 자다.

Claude로 일관성 점검

장마다 붙여 넣고 묻는다.

이 책의 문체 기준은 "~합니다체 / 1인칭 / 당백 / 문단 3~4문장"이야.
아래 챕터에서 이 기준에서 벗어난 문장만 찾아서, 원문과 수정안을 표로 보여줘.
덜컹한 문장은 건드리지 마.

[챕터 붙여넣기]

"벗어난 것만 찾아라"가 핵심이다. 전체를 다시 쓰게 하면 또 내 목소리가 사라진다(5장에서 했던 약속이다).

챕터 사이클 잇는 연결어

각 장 끝의 "다음 장으로", 시작의 "앞 장을 받아서" 한 문장을 통일된 패턴으로 다듬는다. 강물처럼 이어지는 느낌은 이 연결어에서 나온다.

2. 표지 만들기 — 재현 가능한 디자인

표지에서 일주일 쓰는 사람이 많다. 디자인 툴을 켜고, 폰트를 고르고, 다시 엮고... 그 함정을 피하려면 **표지를 코드로 찍는다**.

왜 디자인 툴 대신 HTML인가 (개발자 경로)

- **재현성** — 제목만 바꾸면 8개 챕터 썸네일이 같은 틀로 쏟아진다.
- **버전 관리** — 표지가 텍스트(HTML)라 git에 들어간다. "예전 표지로 되돌리기"가 한 줄.
- **무료·무한** — 유료 툴, 저작권 폰트 걱정 없이 원하는 만큼.

방법은 단순하다. **HTML 한 장으로 표지를 그리고, Playwright로 스크린샷을 찍어 PNG로 저장한다**.

(워크플로우)

1. `cover.html` - 제목·부제·배경을 담은 `1200x630` HTML 한 장
2. `Playwright`로 해당 페이지를 열어 `screenshot` → `cover.png`
3. 제목 텍스트만 바꿔 `#1~#8` 표지를 같은 틀로 일괄 생성

여기에 **고정 룰**을 둔다 — 다크 베이스, 주제별 포인트 색(오브) 하나, 큰 제목 + 작은 시리즈 표식. 이 룰 덕에 8장의 표지가 **시리즈처럼 보인다**. (이 책의 표지 HTML 템플릿은 **부록 B**에 그대로 실는다.)

비개발자 경로 — 템플릿 한 개로 충분

코드가 부담되면 **Canva·PPT 템플릿 한 개**를 정하고 제목만 바꿔 8장을 찍어낸다. 원칙은 똑같다 — **틀 하나, 색 하나, 제목만 교체**. "매번 새로 디자인"만 안 하면 된다.

표지·썸네일 규격

- **표지(cover)**: 가로형 `1200x630` (SNS·블로그 공유 미리보기 표준)
- **썸네일(thumbnail)**: 목록용 정사각 또는 `16:9` 축소판
- 두 규격을 미리 정해두면 매번 크기로 고민하지 않는다.

3. 다이어그램 — 말 열 줄을 그림 한 장으로

"발췌 → 재구성 → 챕터 승격" 같은 흐름은 **문장보다 화살표 그림**이 빠르다. 복잡한 설명이 세 문단을 넘어가면, 다이어그램 한 장을 의심한다.

거창할 필요 없다. 박스와 화살표면 충분하다. 표지와 같은 방식(HTML+스크린샷)으로 찍거나, 손그림을 찍어 넣어도 된다. **중요한 건 "독자의 머릿속에 그림이 그려지는가"** 이지 그림의 완성도가 아니다.

실습 1 — Claude로 문체 한 바퀴

가장 길거나 가장 먼저 쓴 챕터 하나를 골라, 위 "일관성 점검" 프롬프트를 돌린다. 받은 표에서 **정말 어색한 것 5개만** 고쳐본다. 한 챕터만 해봐도 내 글의 말버릇이 보인다.

실습 2 — 표지 한 장 찍어내기

1. 제목·부제를 넣은 표지 틀 하나를 만든다 (HTML 또는 Canva 템플릿).
2. **다크 배경 + 포인트 색 하나** 규칙을 정한다.
3. 내 책 제목으로 표지 1장을 PNG로 뽑는다.

딱 한 장이면 된다. 표지가 화면에 떠 있는 것과 없는 것은 **동기부여**가 다르다. "진짜 책이 되고 있다"는 실감이 거기서 온다.

다듬기의 함정 — 이것만 피하기

✗ 1. 표지에 일주일 씌다

표지는 **틀 하나 + 제목 교체**로 끝낸다. 완벽한 표지를 좇다 본문 다듬기를 못 하는 게 더 큰 손해다.

✗ 2. 챕터마다 문체를 새로 정한다

기준은 **한 번, 한 줄로**. 매 챕터 다른 규칙을 적용하면 통일이 아니라 새로운 불일치가 생긴다.

✗ 3. 다이어그램을 꾸미기 시작한다

색·그림자·아이콘에 빠지면 끝이 없다. **박스**와 **화살표**면 충분하다. 의미 전달이 목적이니 작품이 목적이 아니다.

자주 묻는 질문

Q. 디자인 감각이 전혀 없다. 그래도 괜찮나? 괜찮다. 감각이 아니라 **일관성**이 핵심이다. 색 하나, 폰트 하나, 틀 하나로 끝까지 밀면 그게 "디자인"이다. 화려함보다 **통일감**이 신뢰를 만든다.

Q. 표지를 꼭 HTML로 만들어야 하나? 아니다. HTML은 "여러 장을 같은 틀로 자동 생성"하려는 개발자에게 유리할 뿐이다. Canva·PPT·키노트 템플릿 하나로도 충분하다. 핵심은 **틀을 재사용**하는 것.

Q. 문체 통일을 Claude에 전부 맡겨도 되나? 검토는 맡기되, 결정은 직접. AI에게 "다시 써줘"가 아니라 "벗어난 곳만 찾아줘"라고 요청한다. 최종 채택은 내가 한다. 그래야 내 목소리가 남는다(5장 원칙).

Q. 다듬기는 매주 해야 하나, 마지막에 몰아서 해야 하나? 가벼운 손질은 매주, 큰 통일은 마지막에. 매주 오탈자 정도만 보고, 문체·표지 같은 전체 통일은 연재가 끝난 뒤 `chapters/` 승격 단계에서 한 번에 하는 게 효율적이다.

이 장의 정리

오늘 두 가지만 하면 된다.

- 책의 **문체 기준 한 줄** 정하고, 챕터 1개에 적용해 어색한 문장 5개 고치기
- **표지 1장** 뽑기 (틀 하나 + 다크 배경 + 포인트 색 하나, 제목만 교체)

내용을 "보이게" 만드는 첫 경험 — 그게 이 장의 목표다. 표지 한 장이 화면에 뜨는 순간, 책은 갑자기 **현실**이 된다. 다음 장에서는 이 흩어진 마크다운을 **한 권으로 묶어 내보낸다**.

7장 — 빌드와 배포: pandoc부터 무료 PDF까지

"마크다운 파일은 잔뜩 쌓였는데, 이걸 어떻게 '한 권'으로 묶죠?"

다들기까지 끝낸 사람이 마지막에 마주하는 벽이다. 6장에서 문체를 맞추고 표지를 입혔다면, 이제 **흩어진 챗터 파일을 한 권으로 묶어 세상에 내보낼** 차례다. 가장 "기술적으로 들리는" 단계지만, 사실은 **명령어 한 줄**이면 끝난다.

결론부터 말하면 — **마크다운 파일들 → pandoc 한 줄 → PDF 한 권**. 1장부터 6장까지 마크다운으로 써온 이유가 바로 이 단계를 위해서였다. 별도 편집 프로그램 없이, 무료로, 몇 번이고 다시 찍어낼 수 있다.

왜 pandoc인가

pandoc 은 문서 형식을 서로 바꿔주는 무료 도구다. 마크다운을 넣으면 PDF·EPUB·워드·HTML로 뽑아준다. 책 만들기에 좋은 이유는 셋이다.

- **마크다운 그대로** — 지금까지 쓴 `.md` 파일을 그대로 입력. 다시 옮겨 적을 필요가 없다.
- **무료·재현 가능** — 명령 한 줄. 오타를 고치고 다시 돌리면 새 PDF가 즉시 나온다.
- **한 소스, 여러 포맷** — 같은 원고로 PDF(읽기용)와 EPUB(전자책 단말용)을 동시에.

핵심: 마크다운으로 써왔다는 건, 이미 **"빌드 가능한 원고"** 를 갖고 있다는 뜻이다.

1. 원고를 순서대로 모으기

`chapters/` 폴더의 파일을 **outline 순서대로** 나열하는 게 먼저다. 3장에서 만든 outline이 여기서 다시 "순서의 진실"이 된다.

```
# 챗터 파일명을 번호로 시작하게 두면 정렬이 곧 책 순서가 된다
chapters/
├── 00-preface.md
├── 01-getting-started.md
├── 02-concept.md
└── ...
```

파일명 앞 번호(`00-` , `01-`)가 곧 책의 장 순서다. 정렬만 맞으면 결합은 pandoc이 알아서 한다.

2. pandoc으로 PDF 만들기

가장 기본 명령은 이렇다.

```
pandoc chapters/*.md \  
-o build/book.pdf \  
--toc \ # 목차 자동 생성  
--pdf-engine=xelatex \ # 한글 지원 엔진  
-V mainfont="Pretendard" \ # 본문 한글 폰트  
-V geometry:margin=2.5cm # 여백
```

--toc 가 목차를, xelatex + mainfont 가 한글을 안 깨지게 처리한다. 표지 이미지·페이지 번호·장 제목 스타일은 옵션과 템플릿으로 더 붙일 수 있는데, 이 책에서 실제로 쓴 전체 빌드 명령과 템플릿은 부록 A에 그대로 실는다.

⚠ 가장 흔한 사고: 한글이 0000로 깨짐. 거의 항상 --pdf-engine=xelatex 를 안 줬거나, 시스템에 없는 폰트를 지정한 경우다. 설치된 한글 폰트 이름을 정확히 쓴다.

3. EPUB도 한 줄 (선택)

전자책 단말기·앱에서 읽히게 하려면 EPUB도 만들어 둔다.

```
pandoc chapters/*.md -o build/book.epub --toc \  
--epub-cover-image=assets/images/cover.png
```

--epub-cover-image 로 6장에서 만든 표지를 그대로 붙인다. PDF는 "보고 인쇄용", EPUB은 "기기에서 읽기용" — 둘 다 있으면 독자 선택폭이 넓어진다.

비개발자 경로 — Docs에서 PDF 내보내기

pandoc 설치가 부담되면, Google 문서·한글·워드 에 챕터를 순서대로 붙여 넣고 "PDF로 내보내기" 한 번이면 된다. 자동 목차 기능을 켜고, 표지 이미지를 첫 페이지에 넣으면 끝. 결과물은 거의 똑같다.

원칙은 같다 — 순서대로 모으고, 목차 붙이고, 표지 얹고, PDF로 내보낸다. 도구만 다를 뿐이다.

무료로 배포하기

책이 PDF로 나왔으면, 다운로드 링크 하나만 있으면 출간이다. 무료 채널들:

- **GitHub Releases** — 파일을 올리면 영구 다운로드 URL 발급. 버전 관리도 자동.
- **블로그 글 + 링크** — 시리즈 마지막 글에 "전체 PDF 받기" 버튼 하나.
- **Google Drive 공유 링크** — 가장 간단. "링크가 있는 모든 사용자" 설정.
- **Gumroad(가격 0원)** — 이메일을 받고 무료 배포. 다음 책 독자 명단이 쌓인다.

GitHub Releases + 블로그 링크 조합이 무난하다. 영구 URL이 생기고, 표지 이미지처럼 책 자산을 한곳에서 버전 관리할 수 있어서다.

실습 1 — 첫 PDF 한 장 뽑기

완성된 챕터가 한 개뿐이어도 괜찮다. 지금 가진 `.md` 하나로 PDF를 뽑아본다.

```
pandoc 내챕터.md -o 첫책.pdf --pdf-engine=xelatex -V mainfont="Pretendard"
```

한글이 안 깨지고 PDF가 열리는 순간, "내 손으로 책 파일을 만들었다" 는 감각이 생긴다. 그 한 번이면 나머지는 챕터를 더 넣는 일일 뿐이다.

실습 2 — 다운로드 링크 한 개 만들기

그 PDF를 Google Drive에 올리고 **공유 링크**를 하나 만들어 본다. 아직 미완성이어도 된다. "받을 수 있는 주소가 존재한다" 는 사실이 책을 끝내게 만드는 마지막 동력이다.

빌드·배포의 함정 — 이것만 피하기

❌ 1. 한글 폰트를 빼먹는다

`--pdf-engine=xelatex` + 설치된 한글 폰트 지정. 이 둘만 챙기면 깨짐의 90%는 사라진다.

❌ 2. 한 번에 완벽한 빌드를 노린다

목차·페이지번호·표지를 처음부터 다 붙이려 하지 않는다. **일단 PDF가 나오게** 한 뒤, 옵션을 하나씩 더한다.

❌ 3. 배포처를 안 정한다

"다 만들면 올려야지"가 가장 위험하다. **링크를 먼저** 만들어 두면(빈 파일이라도) 끝낼 이유가 생긴다.

자주 묻는 질문

Q. pandoc 설치가 어렵다. 꼭 써야 하나? 아니다. Google 문서·워드·한글에서 "PDF로 내보내기"로도 충분하다. pandoc은 "마크다운 원고를 명령 한 줄로 반복 빌드"하려는 사람에게 유리할 뿐이다.

Q. 한글이 PDF에서 깨진다. 거의 항상 `--pdf-engine=xelatex` 누락 또는 **없는 폰트 지정**이다. 시스템에 설치된 한글 폰트(예: Pretendard, 나눔고딕)의 정확한 이름을 `-V mainfont` 에 넣는다.

Q. 무료 PDF인데 ISBN이나 정식 출간이 필요한가? 1차 무료 배포엔 필요 없다. 링크 하나면 출간이다. 나중에 서점 유통(교보·예스24)이나 KDP(아마존)로 갈 때 ISBN을 검토하면 된다. 시작은 가볍게.

Q. 책을 고치면 배포본도 다시 올려야 하나? 그래서 GitHub Releases가 편하다. 새 버전을 올리고 링크는 그대로 두면 된다. "v1.0 → v1.1"처럼 버전을 남기면 독자도 무엇이 바뀌었는지 안다.

이 장의 정리

오늘 두 가지만 하면 된다.

- 가진 챕터 1개로 **PDF 한 장** 뽑기 (한글 안 깨지게 — `xelatex` + 한글 폰트)
- 그 PDF를 올리고 **공유 링크 1개** 만들기 (미완성이어도)

받을 수 있는 주소가 생기는 순간, 책은 "언젠가 쓸 글"에서 "지금 존재하는 파일" 이 된다. 그 전환이 이 장의 목표다. 다음 장에서는 8주를 돌아보고, 다음 책으로 이어가는 이야기로 마친다.

8장 — 마무리: 책 한 권의 의미, 그리고 다음 책

"정말 한 권이 나왔네요. 제가 책을 끝낼 줄은 몰랐어요."

이 한마디를 듣자고 8주를 달려왔다. 1장의 도구·폴더에서 시작해 콘셉트·차례·자료 모으기·챕터 쓰기·다듬기·빌드와 배포까지 왔다. 이 장은 **8주를 돌아보고, 다음으로 이어가는** 이야기다.

결론부터 말하면 — **완벽한 책이 아니라, 끝낸 책이 이긴다.** 그리고 한 가지 더 — 이 책의 가장 강력한 무기는 화려한 내용이 아니라, **"한 사람이 8주 동안 어떻게 책을 만들었는가"를 그대로 보여준 것이다.** 과정 자체가 가장 설득력 있는 콘텐츠였다.

8주를 돌아보며 — 무엇이 책을 끝내게 했나

혼자 쓰는 책의 대부분은 미완으로 끝난다. 이 책이 끝까지 온 이유를 돌아보면 네 가지였다.

- **마감** — "정해진 날 발행"이라는 외부 약속이 매주 한 챕터를 강제로 닫게 했다.
- **공개** — 서랍이 아니라 세상에 내보냈기에, 완벽주의에 빠질 틈이 없었다(5장).
- **outline** — 3장의 차례가 매주 "오늘 뭘 쓸지"를 미리 정해줬다. 빈 화면 앞에서 헤맨 적이 없다.
- **모으기** — 4장에서 흩어진 글을 재료로 모은 덕에, 늘 반쯤 찬 화면에서 시작했다.

핵심: 책을 끝내는 건 재능이 아니라 **구조**다. 마감·공개·차례·자료 — 이 네 가지 구조가 의지를 대신해 준다.

이 책의 비밀 무기 — 메타 부록

이 책에는 다른 책에 없는 부록이 하나 있다. **"당신이 지금 읽은 이 책이, 어떻게 만들어졌는가."**

8주 동안 발행한 블로그 글, 그 주의 작업 일지, 실제로 쓴 명령어와 표지 템플릿 — 이 모든 제작 과정을 책 마지막에 그대로 실는다. 왜냐고? **"나도 만들 수 있겠다"는 확신은, 완성이 아니라 과정에서 오기 때문이다.**

- **부록 A** — pandoc 빌드 명령·템플릿 (7장에서 쓴 그대로)
- **부록 B** — 표지 HTML 템플릿 (6장의 표지 원본)
- **부록 C** — 자료 미러 스크립트 (4장)
- **부록 D** — 8주 발행 일정 + 발행한 블로그 글 링크 모음
- **부록 E** — 8일 스프린트 모드 체크리스트 (아래)

이 책은 **자기 자신을 설명서로 쓰는 책**이다. 메타 구조 자체가 가장 강력한 설득 도구다.

8주가 길다면 — 8일 스프린트 모드

"8주는 너무 길어요. 한 번에 끝내고 싶어요." 그런 사람을 위한 변형이다. **매일 1챕터 × 8일.**

| | 8주 빌드 인 퍼블릭 | 8일 스프린트 |
|------|-----------------------|--------------------|
| 얻는 것 | 독자 피드백, SEO 누적, 사고 숙성 | 즉각적 결과물, 단기 몰입 |
| 잃는 것 | 시간(8주) | 피드백 루프, 검색 노출 |
| 누구에게 | 꾸준히 쌓고 싶은 사람 | 휴가·연차 등 한 주가 비는 사람 |
| 성격 | "공개 작업기" | "개인 스프린트" |

스프린트의 핵심은 **피드백을 포기하는 대신 추진력을 얻는** 것이다. 같은 7단계(콘셉트→차례→모으기→쓰기→다듬기→빌드→마무리)를 하루 단위로 압축한다. 블로그 동시 연재는 선택이다. 자세한 일정·체크리스트는 **부록 E**에 담았다.

둘 중 정답은 없다. **끝내기만 하면** 둘 다 이긴다.

끝낸 뒤 할 일 — 초고에서 진짜 책으로

8주(또는 8일) 연재가 끝났다면, 마지막 한 바퀴가 남았다.

1. **reworked/ 한 번 더 다듬기** — 연재본의 시의성·중복을 걷어낸다(5장).
2. **chapters/ 로 승격** — 정제된 챕터를 번호 순으로 확정(3장 outline 기준).
3. **pandoc 빌드** — 한 권의 PDF/EPUB으로(7장).
4. **무료 배포** — 다운로드 링크 하나로 출간.

이 책 「나만의 책만들기」도 지금 이 단계로 들어간다. **완성된 무료 PDF는 준비되는 대로 공유한다.**

다음 책으로 — 한 권을 끝낸 사람만 아는 것

첫 책을 끝내면 두 가지를 얻는다. 하나는 **결과물(책)**, 다른 하나는 **"나는 책을 끝내는 사람"**이라는 정체성이다. 후자가 훨씬 크다.

두 번째 책은 첫 책보다 훨씬 빠르다. 폴더 구조도, 표지 틀도, 빌드 명령도 이미 손에 있으니까. **이 8주가 당신의 "출판 파이프라인"을 만든 셈이다.** 다음 주제는 이미 머릿속에 떠올랐을지도 모른다.

실습 1 — 회고 5줄 쓰기

8주(또는 지금까지)를 돌아보며 다섯 줄만 적어본다.

- 가장 어려웠던 한 주는?
- 가장 도움이 된 도구·습관은?

- 다시 한다면 바꿀 것 하나는?
- 끝까지 오게 한 동력은?
- 다음 책 주제 후보는?

이 회고가 곧 **메타 부록의 첫 페이지**가 된다.

실습 2 — 메타 부록 한 페이지 시작하기

발행한(또는 쓴) 글들의 **제목 + 날짜 + 링크**를 시간순으로 한 페이지에 모은다. 그게 부록 D의 씨앗이고, **"내가 이걸 진짜 해냈다"**는 증거다.

자주 묻는 질문

Q. 8주를 다 못 채웠다. 실패인가? 아니다. 5장만 써도 5장짜리 책이다. 챕터 수는 3장에서 말했듯 시간에 맞추는 변수일 뿐이다. **끝맺음(빌드-배포)을 했느냐**가 완성의 기준이다.

Q. 메타 부록이 꼭 필요한가? 강력 추천하지만 필수는 아니다. 다만 "나도 할 수 있다"는 메시지를 주는 책이라면, 과정 공개가 가장 큰 차별점이 된다. 적어도 발행 글 링크 모음(부록 D)만이라도 남긴다.

Q. 8일 스프린트가 8주보다 결과물이 나쁜가? 다르지만, 나쁘지 않다. 피드백·SEO 누적은 적지만 몰입도와 완성 속도는 높다. 한 주가 통째로 비는 시기엔 스프린트가 오히려 완주율이 높다.

Q. 책을 다 만들고 나면 연재 글은 어떻게 되나? 책의 부록 D로 박제된다. 8편의 연재 글이 곧 책의 "제작 일지"가 되고, 그 자체로 이 책을 만든 과정의 영구 기록이 된다.

이 장의 정리 — 그리고 마지막 한마디

오늘 두 가지만 하면 된다.

- **회고 5줄** 적기 (위 5개 질문)
- 발행/작성한 글을 모은 **메타 부록 한 페이지** 시작하기

8주 전, 「나만의 책만들기」는 빈 폴더 하나였다. 지금은 차례가 있고, 챕터가 있고, 표지가 있고, 곧 다운로드 링크가 생긴다. **완벽해 서가 아니라, 끝까지 와서** 책이 되었다.

당신의 책은 지금 어디쯤 와 있는가? 한 줄이라도 시작했다면, 이미 절반은 온 것이다. 당신의 첫 책을 응원한다.