바이브뉴스 전용 자료 · 무료

AI를 직원처럼 부리는 구조

바이브뉴스 실전 팁 vol.1

혼자서 프로젝트 9개 굴리는 사람들의 공통 구조 — 복붙 프롬프트 포함

바이브뉴스 by 바이브 메이커

cafe.naver.com/vibemaker

CONTENTS

목차

이 자료의 모든 팁은 원문 딥리딩·실제 운영 검증을 거쳐 정리했습니다. 복붙 박스(회색 코드 상자)는 그대로 복사해 붙여넣으면 작동하는 형태입니다.

PART 1 전체 4장

혼자 프로젝트 9개 굴리는 구조

혼자서 서비스 9개를 동시에 운영하는 개발자가 있습니다. 초등학교 학급 관리 서비스(교사 80명, 학생 1천 명 이상), AI 진정서 작성 도구, 스팀에 출시한 게임까지 — 전부 한 사람이 클로드 코드로 굴립니다.

비결을 뜯어보면 재능이 아니라 **구조**입니다. 핵심은 딱 세 가지예요.

1.1 중앙 허브 — CLAUDE.md 파일 하나

이 분 컴퓨터의 홈 폴더에는 `CLAUDE.md` 라는 파일이 하나 있습니다. 모든 프로젝트의 목차이자 공용 규칙집이에요.

- 지금 진행 중인 프로젝트가 뭔지
- 일을 시킬 때 지키는 절차 (리서치 → 검증 → 실행)
- 반복되는 실수와 금지사항

이 파일 덕분에 어떤 프로젝트를 열어도 AI가 **먼저 맥락을 읽고 시작**합니다. 매번 "우리 프로젝트는 이런 거고..." 설명을 반복할 필요가 없어져요.

> 💡 복붙해서 시작하세요. 홈 폴더에 `CLAUDE.md` 파일을 만들고 아래를 채우면 됩니다.

MD

```
# 내 작업 허브

## 진행 중인 프로젝트
- {프로젝트 A}: {한 줄 설명} — 폴더 {경로}
- {프로젝트 B}: {한 줄 설명} — 폴더 {경로}

## 일하는 규칙
- 새 작업은 먼저 계획을 보여주고, 내가 승인하면 실행한다
- 파일을 지우거나 돈이 나가는 일은 반드시 나에게 먼저 묻는다
- 모르는 건 아는 척하지 말고 모른다고 말한다

## 자주 하는 실수 (하지 말 것)
- {예: 한글 파일명을 영어로 바꿔버리는 것}
```

1.2 인수인계 — 세션 끝날 때 "handoff" 한 마디

AI와의 대화(세션)는 끝나면 기억이 사라집니다. 그래서 이 분은 세션을 닫기 전에 **"handoff"** 라고 입력합니다. 그러면 AI가 `HANDOFF.md` 파일에 오늘 한 일·현재 상태·다음 할 일을 자동으로 정리해요.

다음 날 새 세션을 열면 그 파일부터 읽게 하고 — 어제 멈춘 지점에서 바로 이어갑니다. 세션을 바꾸는 비용이 사실상 0이 되는 거죠.

> 💡 복붙 프롬프트 — 세션을 닫기 전에 이렇게 시키세요.

```TEXT
오늘 세션을 마무리하자. HANDOFF.md 파일에 다음을 정리해줘:
1) 오늘 한 일 요약
2) 지금 상태 (되는 것 / 안 되는 것)
3) 다음 세션에서 이어서 할 일 순서대로
다음 세션은 이 파일을 먼저 읽고 시작할 거야.
```

> ⚡ 한 단계 더 — 요약 명령보다 인수인계 파일이 나은 이유: 대화 압축 기능은 출력이 길어지면 세션이 멈추는 일이 있습니다. 파일로 남기면 세션이 몇 개든, 다음 주에 열든 똑같이 이어집니다.

1.3 욕심 금지 — 일주일에 2개만

9개를 운영하지만, **한 주에 집중하는 프로젝트는 정확히 2개**입니다. 본인 표현으로는 "AI의 도움을 받아도 사람이 운영까지 동시에 감당하는 데는 분명한 한계"가 있어서요.

나머지 7개는 멈춰 있는 게 아니라, CLAUDE.md 와 HANDOFF.md 덕분에 **언제든 이어받을 수 있는 상태로 대기**하는 겁니다. 이게 9개 병렬의 진짜 정체예요 — 9개를 동시에 달리는 게 아니라, 어떤 것이든 즉시 재개할 수 있는 구조.

✦ 이 PART의 핵심 정리

- 홈 폴더에 CLAUDE.md 허브 하나 — 프로젝트 목록 + 일하는 규칙 + 금지사항
- 세션 끝 = "handoff" — HANDOFF.md 로 다음 세션에 인수인계
- 집중은 주 2개까지 — 나머지는 "즉시 재개 가능" 상태로 대기

PART 2 전체 4장

에이전트를 직원처럼 뽑는 법

국내의 한 1인 사업가는 AI 에이전트 5개로 회사를 돌립니다. AI 뉴스를 긁어 카카오톡 방 1,300명에게 브리핑하는 에이전트, 커뮤니티 관리 에이전트, 주식 매매 판단을 보조하는 에이전트 — 도메인별로 하나씩, 직원 뽑듯이요.

이 분이 팟캐스트에서 공개한 운영 원칙이 아주 실용적입니다.

2.1 대원칙 — 정해진 일은 문서로, 기억만 AI에게

이 분의 한 문장: **"결정론적인 건 밖에서 다 만들고, 기억 부분만 에이전트에 맡긴다."**

풀어 쓰면 이렇습니다.

- **패턴이 뚜렷한 반복 작업** (뉴스 수집, 요약 형식, 발송 절차) → 스킬·문서로 고정합니다. 매번 다르게 하면 안 되는 일이니까요.
- **취향과 판단 기준** ("이런 글은 빼줘", "이 톤이 좋아") → 에이전트의 기억에 쌓습니다. 쓸수록 나에게 맞춰지는 부분이죠.

좋은 에이전트의 3요소를 이 분은 "좋은 모델, 좋은 도구, 좋은 기억"으로 정리하는데 — 모델은 돈으로 사고, 도구는 문서로 만들고, **기억만이 시간이 만들어주는 자산**입니다.

2.2 신입 온보딩 — 핸드오프 문서 한 장

새 에이전트에게 일을 맡길 때 가장 좋은 방법으로 이 분이 꼽은 팁:

먼저 기존 AI에게 "온보딩용 핸드오프 문서"를 만들게 하고, 그 문서를 새 에이전트에게 물리는 것.

> 💡 복붙 프롬프트 — 일을 넘기기 전에 이렇게 시키세요.

TEXT

```
새로운 AI 에이전트가 이 일을 이어받을 거야.
온보딩용 핸드오프 문서를 만들어줘:
1) 이 일의 목적과 결과물이 뭔지
2) 반드시 지켜야 하는 규칙 (형식·톤·금지사항)
3) 지금까지 내가 준 피드백 중 중요한 것
4) 자주 하는 실수와 피하는 법
새 에이전트가 이 문서만 읽고도 바로 일할 수 있게 써줘.
```

문서를 받으면 새 에이전트에게 "이거 보고 온보딩해. 기억할 건 기억하고" 라고 주면 끝. 사람 신입한테 인수인계 문서 주는 것과 똑같습니다.

2.3 스킬 만드는 공식 — 잘된 것 10개 → 일반화

반복 작업을 스킬(저장해두고 명령 한 번으로 부르는 작업 절차)로 만들 때, 이 분의 공식이 명확합니다.

1. **잘 터진 결과물 10개를 모은다** — 글 자동화라면 반응 좋았던 글 10개
2. AI에게 **"이걸 보고 일반화해봐"** 라고 시켜 규칙을 뽑게 한다
3. 그 규칙으로 만든 스킬을 **새 대화에서 테스트**한다 (75점짜리일지 85점짜리일지는 돌려봐야 안다)
4. 이후엔 피드백을 기억에 쌓아 **점수를 조금씩 올린다**

> 💡 복붙 프롬프트 — 반복 작업을 스킬로 굳힐 때.

TEXT

```
여기 내가 만족했던 결과물 {N}개를 줄게.
이것들의 공통 패턴을 일반화해서, 앞으로 같은 작업을
똑같은 품질로 반복할 수 있는 작업 절차(스킬)로 정리해줘.
- 입력으로 뭘 받는지 / 출력 형식이 뭔지
- 단계별 절차
- 하지 말아야 할 것
새 대화에서 이 절차만 보고 작업해도 같은 결과가 나와야 해.
```

> 💡 💡 핵심은 순서입니다. "좋은 레퍼런스 → 일반화"지, 맨바닥에서 "스킬 만들어줘"가 아니에요. 쓰레기를 넣으면 쓰레기가 나온다(GIGO)의 반대 — 훌륭한 예시를 넣으면 훌륭한 규칙이 나옵니다.

✦ 이 PART의 핵심 정리

- 반복 패턴은 문서·스킬로 고정, 취향·판단은 기억으로 축적
- 새 에이전트 = 온보딩 핸드오프 문서 한 장으로 인수인계
- 스킬 공식: 잘된 것 10개 → "일반화해봐" → 새 대화 테스트 → 기억으로 개선

PART 3 전체 4장

생각의 세기 다이얼 — 같은 AI, 비용은 4배 차이

요즘 AI 모델에는 "생각의 세기"를 조절하는 다이얼이 있습니다. 정식 명칭은 추론 노력(reasoning effort) — 빠르고 싼 답과, 느리지만 정확한 답 중에 **쓰는 사람이 고르는** 기능이에요.

문제는 대부분이 이 다이얼의 존재를 모르고, 간단한 질문에도 최고 세기로 돌리며 돈과 시간을 태운다는 것.

3.1 얼마나 차이 나길래

실측 기준으로, 같은 질문에 생각 세기만 바꿨을 때:

- **중간 세기**: 최저 대비 비용 2~~3배, 정확도 +5~~10%
- **최고 세기**: 비용 4~~6배, 정확도 +10~~20%

포인트는 **정확도가 어느 지점부터 포화**된다는 겁니다. 간단한 일에 최고 세기를 써도 답이 더 좋아지지 않아요 — 돈만 더 나갑니다.

3.2 판단 기준표 — 언제 높이고 언제 낮추나

💡 복붙 기준표. 헷갈리면 이 표대로 하세요.

TEXT

```
[세게 생각시킬 때 — 정확도가 돈보다 중요할 때]
- 수학·계산이 들어간 일
- 코드 작성·버그 찾기
- 답이 맞았는지 검증 가능한 일
- 계약서·정책처럼 틀리면 손해가 큰 문서 검토

[가볍게 시킬 때 — 속도·비용이 중요할 때]
- 요약, 맞춤법·표현 다듬기
- 간단한 사실 질문
- 실시간으로 주고받는 대화
- 같은 작업을 수십 번 반복하는 배치 작업
```

3.3 실전 — 어디서 어떻게 조절하나

- **챗지피티 등 채팅 서비스**: 모델 선택 메뉴에서 세기 옵션을 고릅니다 (Light~Ultra 단계형).
- **클로드 코드**: 지시문에 키워드를 넣으면 됩니다 — 어려운 문제엔 "천천히 깊게 생각해서(think hard)", 단순 작업엔 별도 지시 없이.
- **API·자동화를 쓴다면**: 요청마다 추론 켜기/끄기 파라미터가 있습니다. 반복 배치 작업은 끄고 돌리는 것만으로 비용이 크게 줄어요.

> 💡 시작은 중간에서. 중간 세기로 시작해서 답이 부족하면 한 단계 올리는 습관이, 처음부터 최고 세기로 시작하는 것보다 훨씬 경제적입니다.

⚡ 한 단계 더 — 모델별 이름은 계속 바뀝니다(2026년 7월 기준: GPT 계열은 effort 단계 선택, Qwen 계열은 thinking 켜기/끄기, 클로드는 think 키워드). 이름이 바뀌어도 원리는 같아요: "생각 토큰을 얼마나 쓰게 할 것인가"를 사용자가 정하는 겁니다.

✦ 이 PART의 핵심 정리

- AI 비용의 상당 부분은 "생각 토큰" — 다이얼은 내가 쥐고 있다
- 수학·코딩·검증형 = 세게 / 요약·반복·실시간 = 가볍게
- 중간에서 시작해 필요할 때만 올리기 — 최고 세기는 정확도 포화 지점 이후 낭비

PART 4 전체 4장

에이전트 보안 3원칙 — 사고 나기 전에

에이전트 자동화를 도입한 기업의 절반 이상(54%)이 이미 보안 사고를 겪었다는 조사가 있습니다. 더 놀라운 건 — 그런데도 다수가 여전히 에이전트에게 **계정 비밀번호와 키를 통째로** 주고 있다는 것.

개인이라고 다르지 않습니다. 에이전트에게 내 본계정을 물려주고 자동화를 돌리다 계정이 잠기면, 자동화로 번 시간보다 복구에 쓰는 시간이 더 큽니다. 저희도 자동 발행 시스템을 운영하면서 이 세 가지를 철칙으로 지킵니다.

4.1 원칙 ① — 에이전트 전용 계정을 따로 판다

자동화가 활동할 계정은 **버려도 되는 전용 서브 계정**으로 분리하세요. 본계정(특히 광고 계정·결제와 연결된 계정)은 절대 자동화에 물리지 않습니다.

이유는 단순합니다. 플랫폼이 자동화 트래픽을 봇으로 판정하는 순간, 그 계정에 연결된 모든 것이 같이 잠깁니다. 서브 계정이면 잃는 게 계정 하나지만, 본계정이면 사업이 멈춰요.

4.2 원칙 ② — 권한은 일에 필요한 만큼만

읽기만 필요한 일에는 읽기 권한만. 결제·삭제·발송 권한은 애초에 주지 않습니다.

"어차피 내 에이전트인데"라고 다 열어주면, 에이전트의 실수 한 번이 곧 내 계정의 실수가 됩니다. 사람 직원에게도 입사 첫날 법인카드와 마스터키를 주진 않잖아요.

4.3 원칙 ③ — 되돌리기 어려운 건 사람이 마지막에 누른다

돈이 나가는 일, 지우는 일, 대량 발송 — 이 세 종류는 에이전트가 준비까지만 하고, **최종 실행은 사람이 확인 후** 누르게 하세요. 에이전트가 아무리 똑똑해도, 되돌릴 수 없는 버튼의 주인은 사람이어야 합니다.

> 💡 복붙 스니펫 — 프로젝트의 CLAUDE.md(또는 에이전트 지시문)에 이 조항을 넣어두세요. 저희가 실제로 쓰는 형태입니다.

MD

```
## 보안 규칙 (반드시 지킬 것)

- 자동화 활동은 지정된 전용 계정으로만 한다.
  본계정 로그인·인증이 필요한 상황이 오면 즉시 멈추고 나에게 알린다.
- 권한은 지금 작업에 필요한 최소한만 사용한다.
  결제·삭제·대량 발송 권한이 필요해 보이면 임의로 진행하지 말고 먼저 묻는다.
- 되돌리기 어려운 작업(결제, 삭제, 외부 발행, 대량 발송)은
  실행 직전 상태까지만 준비하고, 최종 실행 전 반드시 내 확인을 받는다.
- 짧은 시간에 요청을 몰아치지 않는다 (플랫폼 봇 감지 회피 목적이 아니라,
  정상 사용 패턴 유지가 계정을 지키는 길이기 때문).
```

💡 💡 이 세 원칙은 귀찮음이 아니라 보험입니다. 세팅에 10분 쓰면, 계정 복구에 쓸 일주일을 아낍니다.

✦ 이 PART의 핵심 정리

- 전용 서브 계정 분리 — 본계정·광고계정은 자동화에 절대 안 물린다
- 최소 권한 — 읽기면 읽기만, 결제·삭제는 애초에 제외
- 되돌리기 어려운 실행은 사람이 최종 확인 — 에이전트는 준비까지만