

바이브 메이커 · 무료 자료

클로드코드 플러그인

설치는 한 줄.
안 먹히는 건 거의 폴더 위치입니다.

담기는 것 | 설치 | 폴더 함정 | 내 것 만들기

CONTENTS · 목차

6 PAGES

목차

1 · 플러그인이 뭔가	기능 추가가 아니라 세팅을 옮기는 단위입니다	03
2 · 설치	공식 마켓은 이미 등록돼 있습니다	04
3 · 가장 흔한 실수	폴더를 잘못 두면 에러도 없이 안 먹힙니다	05
4 · 내 세팅을 플러그인으로	지금 쓰는 .claude 를 그대로 옮기면 됩니다	06

기능 추가가 아니라 세팅을 옮기는 단위입니다

클로드코드를 쓰다 보면 스킬·에이전트·훅·도구연결이 폴더 여기저기에 쌓입니다. 새 프로젝트를 열면 그걸 또 옮기죠. **플러그인은 그 뭉치를 한 폴더에 담아 통째로 들고 다니는 것**입니다. 팀에 그대로 뿌릴 수도 있고요.

플러그인 한 폴더에 담기는 것

1. **skills/**
   스킬. **<이름>/SKILL.md** 폴더들. 설치하면 **/플러그인이름:스킬이름** 으로 불립니다
2. **agents/**
   서브에이전트 정의
3. **hooks/hooks.json**
   이벤트 훅. 설정에 흩어져 있던 hooks 를 파일 하나로
4. **.mcp.json · .lsp.json**
   도구 연결(MCP)과 코드 인텔리전스(LSP) 서버
5. **.claude-plugin/plugin.json**
   이름표. **name · description · version · author**. 이 파일만 **.claude-plugin 안**에 들어갑니다

📌 **이름은 겹치지 않습니다.** 플러그인 스킬은 항상 **/플러그인이름:스킬이름** 으로 불려서, 여러 플러그인이 같은 이름 스킬을 가져도 충돌하지 않습니다.

공식 마켓은 이미 등록돼 있습니다

따로 추가할 필요 없습니다. 클로드코드를 **처음 대화형으로 실행할 때** 공식 마켓이 자동 등록됩니다.

설치 — 클로드코드 안에서 SLASH COMMAND

쓰고 싶은 플러그인 이름만 넣으면 됩니다.

```
/plugin install code-review
```

자동 등록이 안 됐다면 (대화형으로 연 적이 없을 때)

마켓 직접 등록 — 터미널 CLI

공식 마켓은 터미널에서, 커뮤니티 마켓은 클로드코드 안에서 붙입니다.

```
claude plugin marketplace add anthropics/claude-plugins-official
/plugin marketplace add anthropics/claude-plugins-community
```

✅ **됐는지 확인** — 설치 요약에 **「Run /reload-plugins to activate.」**가 뜨면 그 명령을 한 번 더 칩니다. 그다음 **/help** 의 커스텀 명령 탭에 플러그인 이름으로 묶여 보이면 끝입니다.

🔎 2026년 9월 기준 공식 마켓에 올라온 플러그인은 **292개**. 앤트로픽이 직접 만든 것과 검증된 외부 연동이 섞여 있습니다.

폴더를 잘못 두면
에러도 없이 안 먹힙니다

.claude-plugin 폴더 안에 들어가는 건 plugin.json 하나뿐입니다. skills · agents · hooks · commands 를 그 안에 같이 넣으면 클로드코드가 못 읽어요. 그런데 **에러가 안 납니다.** 그래서 「왜 안 되지」로 한참 헤매는 자리입니다.

✕ 이렇게 두면 안 먹습니다

```
my-plugin/
└── .claude-plugin/
    ├── plugin.json
    └── skills/ ← 여기가 아닙니다
        └── my-skill/SKILL.md
```

✓ 이게 맞습니다

```
my-plugin/
├── .claude-plugin/
│   └── plugin.json
├── skills/ ← 플러그인 맨 위
│   └── my-skill/SKILL.md
├── agents/ · hooks/
└── .mcp.json
```

헷갈릴 때 외울 한 줄

.claude-plugin 안에는 plugin.json 하나. 나머지는 전부 플러그인 맨 위.

⚠️ 여기서 말하는 「플러그인 맨 위」는 그 플러그인 **자기 폴더**지 **~/.claude 가 아닙니다.** 예를 들어 ~/.claude/.mcp.json 은 읽히지 않습니다.

💡 **혼자 쓸 거면 더 짧은 길도 있습니다. claude plugin init 이름** 을 치면 ~/.claude/skills/ 아래에 뼈대가 알아서 생기고, 다음 세션부터 **마켓도 설치도 없이** 그대로 로드됩니다. 폴더 위치를 신경 쓸 일 자체가 없어져요.

지금 쓰는 .claude 를 그대로 옮기면 됩니다

새로 만드는 게 아니라 **있는 걸 옮기는 일**입니다.

1 **폴더와 이름표를 만든다**
my-plugin/.claude-plugin/plugin.json 에 name · description · version 을 적습니다

2 **있던 것을 복사한다**
.claude/ 의 **commands · agents · skills** 를 플러그인 **맨 위**로 복사

3 **훅을 옮긴다**
settings.json 의 **hooks 객체를 그대로 hooks/hooks.json** 으로 (형식이 같습니다)

4 **켜서 확인한다**
아래 명령으로 설치 없이 바로 띄워봅니다. 고칠 때마다 **/reload-plugins**

5 **원본을 지운다**
원본을 남겨두면 **그쪽이 이깁니다.** 지워야 플러그인 판이 먹혀요

설치 없이 바로 켜보기 · 내보내기 전 검사 CLI

만드는 중엔 설치 없이 띄우고, 남에게 주기 전엔 검사를 돌립니다. 통과하면 ✔ Validation passed 가 뜹니다.

```
claude --plugin-dir ./my-plugin
claude plugin validate ./my-plugin
```