

2026 개발 도구 TOP 8

설치부터 쓰는 법까지 한 장에

영상에서 소개한 깃허브 개발 도구 여덟 개를 8위부터 1위까지, 클로드에게 말 한 번으로 설치하는 방법과 쓰는 법을 정리했어요.

8개 모두 깃허브에 무료 공개

클로드 코드에서 사용

말 한 번으로 설치

한눈에 보는 여덟 개

순위는 영상 순서예요. 전부 따로 설치하는 도구라, 필요한 것만 골라 깔아도 돼요.

8위 아키펠파이

별 8.5만

코드가 어디에 연결됐는지 헷갈릴 때 — 말로 설명하거나 프로젝트를 보여 주면, 눌러 보며 살펴보는 구조 그림을 웹페이지 한 장으로 만들어 줘요.

7위 버셀 리액트 베스트 프랙티스

별 3.2만

AI로 만든 웹사이트가 느릴 때 — 리액트·넥스트(웹사이트 만드는 도구)로 만든 사이트를 빠르게 짜는 버셀의 요령을 클로드에게 넣어 줘요.

6위 애디 오스마니 에이전트 스킬

별 10.8만

돌아가는 코드, 그대로 써도 될지 불안할 때 — 기획 → 계획 → 개발 → 테스트 → 검토 → 출시를 실무 개발자처럼 단계별로 하게 만드는 스킬 25개 묶음이에요.

5위 맷 포콕 스킬

별 29.6만

AI가 내 말을 멋대로 알아들을 때 — AI가 엉뚱한 걸 만들지 않도록, 만들기 전에 질문을 쏟아내 빠뜨린 조건을 끄집어내 주는 스킬 모음이에요. 영상의 '그릴 미'가 여기 있어요.

4위 에브리씽 클로드 코드

별 28.3만

개발 환경, 뭘 설정해야 할지 막막할 때 — 개발·리뷰 도우미, 테스트·보안 점검 스킬, 자동화 설정까지 클로드 코드에 한 번에 엮은 큰 세트예요.

3위 지스택

별 13.7만

혼자 기획·디자인·버그 확인까지 다 해야 할 때 — 클로드 코드를 대표·설계·QA·보안 담당이 있는 가상 개발팀처럼 굴리는 스킬 세트예요.

2위 슈퍼파워스

별 30.2만

시킨 대로 만들었는데 원하던 결과가 안 나올 때 — 바로 코드를 짜지 않고 질문 → 설계 → 계획 → 테스트 순서로 일하게 만드는 개발 방법 스킬이에요.

1위 레아

별 32.2만

다른 앱의 기능이 어떻게 만들어졌는지 궁금할 때 — 소스 코드가 없어도 앱의 속 구조와 동작을 분석해서, 그 기능이 어떻게 돌아가는지 따라가 주는 분석 도구예요.

목차

0 시작 전 준비

클로드 코드 · 노트 · 깃

7 버셀 리액트 베스트 프랙티스

AI로 만든 웹사이트가 느릴 때

5 맷 포콕 스킬

AI가 내 말을 멋대로 알아들을 때

3 지스택

혼자 기획·디자인·버그 확인까지 다 해야 할 때

1 레아

다른 앱의 기능이 어떻게 만들어졌는지 궁금할 때

? 자주 묻는 질문

뭐부터 깔까 · 돈 드나요 · 윈도우

8 아키파이

코드가 어디에 연결됐는지 헷갈릴 때

6 애디 오스마니 에이전트 스킬

돌아가는 코드, 그대로 써도 될지 불안할 때

4 에브리씽 클로드 코드

개발 환경, 뭘 설정해야 할지 막막할 때

2 슈퍼파워스

시킨 대로 만들었는데 원하던 결과가 안 나올 때

! 꼭 지킬 것

한 번에 하나씩 · 겹쳐 깔지 않기

시작 전 준비

- 클로드 코드(Claude Code, 컴퓨터에서 쓰는 클로드)가 있어야 해요. 여덟 개 모두 클로드 코드에 얹어 쓰는 도구예요.
- 대부분 노드(Node.js, 프로그램을 돌리는 도구)와 깃(Git, 깃허브에서 내려받는 도구)이 필요해요. 없으면 클로드에게 "노드랑 깃 설치를 도와주세요"라고 말하면 돼요.
- 여덟 개 모두 깃허브에 무료로 공개된 도구예요. 4위 ECC 만 회사용 유료 기능이 따로 있고, 우리가 쓰는 공개 버전은 무료예요.

설치는 전부 이렇게, 클로드에게 말하세요

클로드 코드를 열고 각 도구의 노란 칸을 [복사]해서 입력창에 붙여 넣으면, 클로드가 안내문을 읽고 설치해 줘요. 직접 쳐야 하는 줄이 있으면 클로드가 알려 줘요.

클로드 코드에게 그대로 말하기

<https://github.com/obra/superpowers>

이 주소의 안내대로 설치해 주세요. 제가 직접 입력해야 하는 명령이 있으면 한 줄씩 알려 주세요.

위 칸은 예시예요. 아래 1~8번 칸에 도구마다 주소를 바꿔 넣어 두었어요.

8 8위 아키파이

Archify · 깃허브 별 8.5만

말로 설명하거나 프로젝트를 보여 주면, 눌러 보며 살펴보는 구조 그림을 웹페이지 한 장으로 만들어 줘요.

언제 쓰면 좋아요

코드가 어디에 연결됐는지 헷갈릴 때

1 클로드 코드를 열어요.

2 아래 칸의 [복사]를 눌러요.

3 입력창에 붙여 넣고 보내요.

클로드 코드에게 그대로 말하기

<https://github.com/tt-a1i/archify>

이 주소의 안내대로 아키텍처 스킴을 설치해 주세요. 설치가 끝나면 지금 프로젝트의 구조를 아키텍처로 그려서 보여 주세요.

직접 설치하는 방법 보기

저장소 주소

<https://github.com/tt-a1i/archify>

직접 설치하고 싶다면 아래처럼 하면 돼요. 안내문에 적힌 방법이에요.

터미널에 한 줄

```
npx skills add tt-a1i/archify -g
```

팁 노드(Node.js, 프로그램을 돌리는 도구)가 필요해요. 없다고 나오면 클로드에게 "노드 설치를 도와주세요"라고 말하세요.

이렇게 되면 성공 클로드가 웹페이지 한 장(그림)을 만들어 주고, 상자를 누르면 설명이 펼쳐지면 성공이에요.

쓰는 법 이렇게 말하면 돼요: "아키텍처로 우리 사이트 구조를 그려 주세요. 화면 → 서버 → 데이터베이스 순서로요."

7 7위 버셀 리액트 베스트 프랙티스

Vercel React Best Practices · 깃허브 별 3.2만

리액트·넥스트(웹사이트 만드는 도구)로 만든 사이트를 빠르게 짜는 버셀의 요령을 클로드에게 넣어 줘요.

언제 쓰면 좋아요

AI로 만든 웹사이트가 느릴 때

1 클로드 코드를 열어요.

2 아래 칸의 **[복사]**를 눌러요.

3 입력창에 붙여 넣고 보내요.

클로드 코드에게 그대로 말하기

<https://github.com/vercel-labs/agent-skills>

이 주소의 안내대로 버셀 스킬을 설치해 주세요. 그중 리액트 베스트 프랙티스 스킬로 제 사이트에서 느린 곳을 찾아서 고쳐 주세요.

직접 설치하는 방법 보기

저장소 주소

<https://github.com/vercel-labs/agent-skills>

직접 설치하고 싶다면 아래처럼 하면 돼요. 안내문에 적힌 방법이에요.

터미널에 한 줄

```
npx skills add vercel-labs/agent-skills
```

팁 이 주소에는 버셀 스킬이 여러 개 들어 있어서 한꺼번에 깔려요. 리액트 스킬 이름은 vercel-react-best-practices 예요.

이렇게 되면 성공 설치하면 저절로 켜져요. "이 화면이 왜 느린지 봐 주세요"라고 했을 때 다시 그리기·기다리기 같은 원인을 짚어 주면 돼요.

쓰는 법 리액트로 만든 사이트를 열어 둔 채로 "성능 문제를 찾아서 고쳐 주세요"라고 말하면 돼요.

6 6위 애디 오스마니 에이전트 스킬

Addy Osmani Agent Skills · 깃허브 별 10.8만

기획 → 계획 → 개발 → 테스트 → 검토 → 출시를 실무 개발자처럼 단계별로 하게 만드는 스킬 25개 묶음이에요.

언제 쓰면 좋아요

돌아가는 코드, 그대로 써도 될지 불안할 때

1 클로드 코드를 열어요.

2 아래 칸의 [복사]를 눌러요.

3 입력창에 붙여 넣고 보내요.

클로드 코드에게 그대로 말하기

<https://github.com/addyosmani/agent-skills>

이 주소의 안내대로 에이전트 스킬을 설치해 주세요. 제가 직접 입력해야 하는 명령이 있으면 한 줄씩 알려주세요.

직접 설치하는 방법 보기

저장소 주소

<https://github.com/addyosmani/agent-skills>

직접 설치하고 싶다면 아래처럼 하면 돼요. 안내문에 적힌 방법이에요.

클로드 코드 입력창에 한 줄씩(추천)

```
/plugin marketplace add addyosmani/agent-skills  
/plugin install agent-skills@addy-agent-skills
```

터미널에 한 줄(가장 빠른 방법)

```
npx skills add addyosmani/agent-skills
```

팁 등록할 때 오류가 나면 첫 줄 대신 `/plugin marketplace add https://github.com/addyosmani/agent-skills.git` 을 입력해 보세요.

이렇게 되면 성공 입력창에 `/` 를 치면 `/spec · /plan · /build · /test · /review · /ship` 같은 명령이 보이면 성공이에요.

쓰는 법 `/spec` 으로 만들 것을 정리 → `/plan` 으로 계획 → `/build · /test · /review` 순서로 쓰면 돼요. 기본 `/review` 와 겹치면 `/agent-skills:review` 로요.

5 5위 매트 포콕 스킬

Matt Pocock Skills · 깃허브 별 29.6만

AI가 엉뚱한 걸 만들지 않도록, 만들기 전에 질문을 쏟아내 빠뜨린 조건을 끄집어내 주는 스킬 모음이에요. 영상의

'그릴 미'가 여기 있어요.

언제 쓰면 좋아요

AI가 내 말을 멋대로 알아들을 때

- 1 클로드 코드를 열어요.
- 2 아래 칸의 **[복사]**를 눌러요.
- 3 입력창에 붙여 넣고 보내요.

클로드 코드에게 그대로 말하기

<https://github.com/mattpocock/skills>

이 주소의 안내대로 매트 포콕 스킬을 설치해 주세요. 설치가 끝나면 /grill-me 가 잘 되는지 확인해 주세요.

직접 설치하는 방법 보기

저장소 주소

<https://github.com/mattpocock/skills>

직접 설치하고 싶다면 아래처럼 하면 돼요. 안내문에 적힌 방법이에요.

터미널에 한 줄(추천 · 저절로 업데이트)

```
claude plugin install mattpocock-skills@claude-plugins-official
```

터미널에 한 줄(다른 방법)

```
npx skills@latest add mattpocock/skills
```

주의 두 방법을 같이 쓰면 스킬이 두 번씩 생겨요. 하나만 고르세요.

팁 설치한 뒤 프로젝트마다 한 번 /setup-matt-pocock-skills 를 입력하면, 문서를 어디에 둘지 같은 걸 물어보고 맞춰 줘요.

이렇게 되면 성공 입력창에 /grill-me 를 치면 클로드가 내 계획에 대해 질문을 하나씩 던지기 시작하면 성공이에요.

쓰는 법 그릴 미는 저절로 켜지지 않아요. 계획을 말한 뒤 직접 /grill-me 를 입력하세요. 코드 작업용은 /grill-with-docs 예요.

4 4위 에브리싱 클로드 코드

ECC (Everything Claude Code) · 깃허브 별 28.3만

개발·리뷰 도우미, 테스트·보안 점검 스킴, 자동화 설정까지 클로드 코드에 한 번에 엮는 큰 세트예요.

언제 쓰면 좋아요

개발 환경, 뭘 설정해야 할지 막막할 때

1 클로드 코드를 열어요.

2 아래 칸의 [복사]를 눌러요.

3 입력창에 붙여 넣고 보내요.

클로드 코드에게 그대로 말하기

<https://github.com/affaan-m/ECC>

이 주소의 안내대로 ECC를 설치하려고 해요. 설치 도중 고르는 칸이 나오면 저한테 하나씩 물어봐 주세요.

직접 설치하는 방법 보기

저장소 주소

<https://github.com/affaan-m/ECC>

직접 설치하고 싶다면 아래처럼 하면 돼요. 안내문에 적힌 방법이에요.

터미널에 한 줄(추천)

```
npx ecc-universal@2.2.3 setup
```

클로드 코드 입력창에 한 줄씩(다른 방법)

```
/plugin marketplace add https://github.com/affaan-m/ECC
/plugin install ecc@ecc
```

주의 설치하는 중에 고르는 칸이 나와요. 내 컴퓨터 하나에서 쓸 거면 'Global user', 자동 규칙은 'Standard'를 고르세요. 설치 방법을 두 가지 섞으면 겹쳐서 생겨요.

팁 노드 18 이상·깃-클로드 코드 2.1 이상이 필요해요. 기능이 많아 대화 비용(토큰)이 늘 수 있으니, 안내문의 절약 설정도 같이 켜 두세요.

이렇게 되면 성공 새 대화를 열고 /plugin list 를 쳤을 때 ecc@ecc 가 켜져 있으면 성공이에요.

쓰는 법 /ecc:plan "만들 기능 설명" 으로 계획, /code-review 로 검토, /build-fix 로 빌드 오류 고치기.

3 3위 지스택

gstack · 깃허브 별 13.7만

클로드 코드를 대표·설계·QA·보안 담당이 있는 가상 개발팀처럼 굴리는 스킬 세트예요.

언제 쓰면 좋아요

혼자 기획·디자인·버그 확인까지 다 해야 할 때

1 클로드 코드를 열어요.

2 아래 칸의 [복사]를 눌러요.

3 입력창에 붙여 넣고 보내요.

클로드 코드에게 그대로 말하기

<https://github.com/garrytan/gstack>

이 주소의 안내대로 지스택을 설치해 주세요. 제 컴퓨터는 윈도우예요. 막히는 곳이 있으면 쉬운 말로 알려주세요.

직접 설치하는 방법 보기

저장소 주소

`https://github.com/garrytan/gstack`

직접 설치하고 싶다면 아래처럼 하면 돼요. 안내문에 적힌 방법이에요.

클로드 코드에 붙여 넣기(안내문 방식)

```
git clone --single-branch --depth 1 https://github.com/garrytan/gstack.git ~/.claude/skills/gstack && cd ~/.claude/skills/gstack && ./setup
```

윈도우 파워셸에서 마지막 단계

```
.\setup.ps1
```

주의 윈도우 '스마트 앱 컨트롤'이 켜져 있으면 지스택 프로그램이 막혀요. 설정 → 개인 정보 및 보안 → Windows 보안 → 앱 및 브라우저 컨트롤에서 확인하세요.

팁 깃·번(Bun, 프로그램을 돌리는 도구) 1.4.2 이상, 윈도우는 노드도 필요해요. 업데이트한 뒤엔 setup 을 한 번 더 돌려야 해요.

이렇게 되면 성공 입력창에 /office-hours 를 쳤을 때 만들 것을 묻기 시작하면 성공이에요.

쓰는 법 /office-hours 로 만들 것 설명 → /plan-ceo-review 로 방향 검토 → /review → /qa 로 브라우저에서 직접 눌러 보며 문제 찾기.

2 2위 슈퍼파워스

Superpowers · 깃허브 별 30.2만

바로 코드를 짜지 않고 질문 → 설계 → 계획 → 테스트 순서로 일하게 만드는 개발 방법 스킬이에요.

언제 쓰면 좋아요

시킨 대로 만들었는데 원하던 결과가 안 나올 때

1 클로드 코드를 열어요.

2 아래 칸의 **[복사]**를 눌러요.

3 입력창에 붙여 넣고 보내요.

클로드 코드에게 그대로 말하기

<https://github.com/obra/superpowers>

이 주소의 안내대로 슈퍼파워스를 설치하고 싶어요. 제가 클로드 코드 입력창에 직접 쳐야 하는 명령을 알려주세요.

직접 설치하는 방법 보기

저장소 주소

<https://github.com/obra/superpowers>

직접 설치하고 싶다면 아래처럼 하면 돼요. 안내문에 적힌 방법이에요.

클로드 코드 입력창에 한 줄(가장 간단)

```
/plugin install superpowers@claude-plugins-official
```

다른 방법(한 줄씩)

```
/plugin marketplace add obra/superpowers-marketplace
```

```
/plugin install superpowers@superpowers-marketplace
```

팁 /plugin 으로 시작하는 줄은 클로드 코드 대화 입력창에 직접 쳐야 해요. 클로드에게 부탁하는 것보다 직접 한 줄 치는 게 가장 확실해요.

이렇게 되면 성공 설치한 뒤 "로그인 기능 만들고 싶어요"라고 하면, 바로 코드를 짜지 않고 질문부터 하면 성공이에요.

쓰는 법 따로 명령 없이 저절로 켜져요. 이상하면 "이번 대화에서 슈퍼파워스가 뭘 잘못했는지 찾아 주세요"라고 말하세요.

1 1위 레아

REA (Reverse Engineer Anything) · 깃허브 별 32.2만

소스 코드가 없어도 앱의 속 구조와 동작을 분석해서, 그 기능이 어떻게 돌아가는지 따라가 주는 분석 도구예요.

언제 쓰면 좋아요

다른 앱의 기능이 어떻게 만들어졌는지 궁금할 때

1 클로드 코드를 열어요.

2 아래 칸의 [복사]를 눌러요.

3 입력창에 붙여 넣고 보내요.

클로드 코드에게 그대로 말하기

<https://github.com/morluto/rea>

이 주소의 안내대로 레아를 설치하려고 해요. 설치 도중 승인할 것이 나오면 무엇인지 쉬운 말로 설명하고 저한테 물어봐 주세요.

직접 설치하는 방법 보기

저장소 주소

<https://github.com/morluto/rea>

직접 설치하고 싶다면 아래처럼 하면 돼요. 안내문에 적힌 방법이에요.

터미널에 한 줄(공식)

```
npx rea-agents setup
```

주의 내가 만든 프로그램이나 분석을 허락받은 것에만 쓰세요(안내문에 적힌 조건). 비슷한 이름의 가짜 저장소도 조심하세요 — 주소는 morluto/rea 예요.

팁 노드 22.19 이상(또는 24.11 이상)이 필요해요. 깊은 분석에 쓰는 외부 프로그램 중 Hopper 는 윈도우에서 안 되고, Ghidra 는 아직 실험 단계예요. 자바스크립트.NET 앱은 추가 프로그램 없이 돼요.

이렇게 되면 성공 설치가 끝나고 클로드를 다시 켜 뒤, 레아로 분석해 달라고 했을 때 분석 결과와 근거를 보여 주면 성공이에요.

쓰는 법 이렇게 말하면 돼요: "레아로 메모 앱의 검색 기능이 어떻게 동작하는지 근거와 함께 보여 주고, 제 프로젝트에 비슷한 기능을 만들어 주세요."

꼭 지킬 것

- 한 번에 하나씩 깔고, 잘 되는지 확인한 다음 다음 것을 깔아요. 여러 개를 한꺼번에 깔면 어디서 막혔는지 찾기 어려워요.
- 같은 도구를 두 가지 방법으로 깔지 마세요. 맷 포콕·ECC 는 겹쳐서 두 번씩 생긴다고 안내문에 적혀 있어요.
- 설치 중에 승인·선택 칸이 나오면 무엇인지 클로드에게 물어보고 고르세요(4위 ECC, 1위 레아).
- 1위 레아는 내 것이거나 허락받은 프로그램에만 쓰세요.

자주 묻는 질문

여덟 개 다 깔아야 하나요? 뭐부터 깔까요?

아니요, 필요한 것만 골라 깔면 돼요. 처음이라면 2위 슈퍼파워스(일하는 순서를 잡아 줌)와 5위 맷 포콕의 그릴 미(만들기 전에 질문으로 정리)부터 권해요. 둘 다 쓰는 법이 간단해요.

돈이 드나요?

여덟 개 모두 깃허브에 무료로 공개돼 있어요. 다만 기능이 많은 도구(ECC·지스택)는 대화량이 늘어서 클로드 사용량이 더 들 수 있어요. 클로드 요금은 쓰는 요금제에 따라 달라요.

윈도우에서도 되나요?

대부분 돼요. 3위 지스택은 윈도우 '스마트 앱 컨트롤'이 켜져 있으면 막히고, 1위 레아의 깊은 분석용 외부 프로그램 중 일부(Hopper)는 윈도우에서 안 돼요. 막히면 노란 칸 문장에 "제 컴퓨터는 윈도우예요"를 붙여서 클로드에게 물어보세요.

/plugin 으로 시작하는 줄은 어디에 쳐요?

클로드 코드 대화 입력창에 그대로 쳐요. npx · git 으로 시작하는 줄은 터미널(검은 창)에 쳐요. 헛갈리면 노란 칸을 클로드에게 보내면 클로드가 알맞은 곳에서 해 줘요.

설치했는데 명령이 안 보여요

클로드 코드를 완전히 꺾다가 다시 켜 보세요. 그래도 안 되면 "방금 설치한 ○○가 안 보여요, 확인해 주세요"라고 말하면 클로드가 설치 위치를 확인해 줘요.