

Superpowers + Claude Code

AI 개발 생산성 완전 가이드

obra/superpowers 분석 · Claude Code 입문 핵심 자료 · 추천 리소스

★ 153K Stars

Shell

2025.10 출시

이 문서에 담긴 내용

1. obra/superpowers 저장소 심층 분석
2. 핵심 스킬 시스템 & 6단계 개발 워크플로
3. Claude Code 초보자 필수 개념 & 설치 가이드
4. 무료 학습 리소스 큐레이션
5. Superpowers + Claude Code 통합 활용법 & 요약

Superpowers 저장소 분석

프로젝트 개요

Superpowers는 Claude Code, Cursor, Codex, OpenCode 등 AI 코딩 에이전트를 위한 플러그인 시스템이자 소프트웨어 개발 방법론입니다. "즉시 코딩"이 아니라 "먼저 무엇을 원하는지 파악"하는 구조적 워크플로를 제공합니다.

Stars	153,252 ★ (2026-04-15 기준)
언어	Shell (설치 스크립트 + Markdown 기반 스킴)
출시일	2025년 10월 9일
설명	An agentic skills framework & software development methodology that works.
지원 플랫폼	Claude Code, Cursor, Codex, OpenCode, GitHub Copilot CLI, Gemini CLI

6단계 개발 워크플로

Superpowers의 핵심은 다음 6단계로 구성된 구조화된 개발 프로세스입니다:

- ▶ **Design (설계)** 소크라테스식 질문으로 요구사항 정제 후 섹션별 스펙 승인
- ▶ **Preparation (준비)** Git worktrees로 격리된 개발 브랜치 생성 + 깨끗한 테스트 기준선 확인
- ▶ **Planning (계획)** 2~5분 단위 태스크로 분해 + 파일 경로·코드·검증 단계 명시
- ▶ **Execution (실행)** 서브에이전트가 개별 태스크 처리 + 2단계 리뷰(스펙 준수 → 코드 품질)
- ▶ **Implementation (구현)** TDD RED-GREEN-REFACTOR 사이클 강제: 실패 테스트 먼저
- ▶ **Completion (완료)** 자동 검증 + 머지 결정 워크플로 + 정리

스킬(Skills) 목록

14개 스킬이 skills/ 디렉토리에 폴더 단위로 구성되어 있습니다:

테스트 & 디버깅

test-driven-development systematic-debugging verification-before-completion

협업 & 계획

brainstorming writing-plans executing-plans subagent-driven-development dispatching-parallel-agents
requesting-code-review receiving-code-review using-git-worktrees finishing-a-development-branch

메타

writing-skills using-superpowers

설치 방법

```
bash
# Claude Code (공식 마켓플레이스)
/plugin install superpowers@claude-plugins-official

# Claude Code (커스텀 마켓플레이스)
```

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

핵심 철학 3원칙

1. 테스트 주도 개발(TDD) — 필수적 실천, 선택이 아님
2. 직관보다 체계적 프로세스 — 일관된 품질 확보
3. 복잡성 감소 우선 — 단순함이 최우선 설계 원칙

Claude Code 초보자 가이드

Claude Code란?

Claude Code는 Anthropic이 만든 AI 기반 코딩 어시스턴트입니다. 코드베이스 전체를 이해하고 여러 파일에 걸쳐 기능 구현, 버그 수정, 테스트 작성, 커밋·PR 생성까지 자동으로 처리합니다. 터미널, VS Code, JetBrains, 데스크탑 앱, 브라우저 등 다양한 환경에서 실행됩니다.

설치 방법

Windows (PowerShell)

powershell

```
irm https://claude.ai/install.ps1 | iex

# 또는 winget 사용
winget install Anthropic.ClaudeCode
```

macOS / Linux / WSL

bash

```
curl -fsSL https://claude.ai/install.sh | bash
```



Windows 네이티브 설치에는 Git for Windows가 필요합니다. 네이티브 설치의 자동 업데이트를 지원하며, winget/Homebrew는 수동 업그레이드가 필요합니다.

프로젝트 시작

bash

```
cd 내-프로젝트
claude
```

처음 실행 시 로그인 프롬프트가 나타납니다. Claude 구독 또는 Anthropic Console 계정이 필요합니다.

핵심 기능 7가지

- ◆ **코드 작성 & 버그 수정** 자연어로 설명하면 여러 파일에 걸쳐 코드를 작성하고 버그를 추적·수정합니다.
- ◆ **Git 자동화** 변경 사항 스테이징, 커밋 메시지 작성, 브랜치 생성, PR 오픈까지 자동 처리합니다.
- ◆ **CLAUDE.md 메모리** 프로젝트 루트의 CLAUDE.md 파일을 매 세션 시작 시 읽어 코딩 표준·아키텍처·라이브러리 설정을 유지합니다.
- ◆ **MCP 통합** Model Context Protocol로 Google Drive, Jira, Slack 등 외부 도구와 연결됩니다.
- ◆ **커스텀 명령(/skills)** 팀이 공유하는 반복 워크플로를 /review-pr, /deploy-staging 같은 명령으로 패키징합니다.
- ◆ **멀티 에이전트** 병렬로 여러 Claude Code 에이전트를 실행해 대규모 태스크를 분산 처리합니다.
- ◆ **스케줄 & 자동화** 정기 PR 리뷰, 야간 CI 분석, 주간 의존성 감사 등을 자동화할 수 있습니다.

bash

```
# 로그 분석
tail -200 app.log | claude -p "이상 징후가 있으면 알려줘"

# 번역 자동화
claude -p "새 문자열을 프랑스어로 번역하고 PR 올려줘"

# 보안 리뷰
git diff main --name-only | claude -p "변경 파일의 보안 이슈 리뷰"
```

추천 학습 리소스

공식 문서 (필수)

공식 문서 홈

code.claude.com/docs

Claude Code 전체 문서. 퀵스타트부터 고급 기능까지.

퀵스타트

code.claude.com/docs/en/quickstart

첫 번째 실제 태스크 — 코드베이스 탐색부터 버그 수정 커밋까지.

CLAUDE.md 메모리

code.claude.com/docs/en/memory

영구 지시사항과 자동 메모리 설정 방법.

공통 워크플로

code.claude.com/docs/en/common-workflows

Claude Code 활용 패턴과 모범 사례.

설정

code.claude.com/docs/en/settings

워크플로에 맞게 커스터마이징하는 방법.

트러블슈팅

code.claude.com/docs/en/troubleshooting

자주 발생하는 문제와 해결책.

Superpowers 공식 리소스

- ▶ **GitHub 저장소** — github.com/obra/superpowers — 소스 코드, 이슈 트래커, 릴리즈 노트
- ▶ **공식 웹사이트** — primeradiant.com — 릴리즈 발표 및 커뮤니티 정보
- ▶ **디스코드** — Discord 커뮤니티 — 실시간 지원 및 사용자 Q&A
- ▶ **플러그인 마켓** — code.claude.com/plugins — Claude Code 공식 플러그인 마켓플레이스

초보자 학습 로드맵

1. Claude Code 설치 후 간단한 프로젝트에서 "claude" 명령 실행 체험
2. 공식 Quickstart 따라 첫 버그 수정 커밋 만들어보기
3. 프로젝트에 CLAUDE.md 파일 추가하여 코딩 표준 설정
4. Superpowers 플러그인 설치 → /brainstorming 스킬로 기능 설계 연습
5. test-driven-development 스킬로 TDD 워크플로 적용
6. subagent-driven-development 스킬로 복잡한 기능 병렬 처리 경험
7. writing-skills 스킬로 팀 맞춤형 커스텀 스킬 직접 제작

Superpowers + Claude Code 통합 활용 팁

- ▶ CLAUDE.md에 Superpowers 스킬 트리거 조건을 명시하면 에이전트가 자동으로 적절한 스킬을 선택합니다.
- ▶ Git worktrees 스킬을 사용하면 실험적 기능과 메인 브랜치를 완전히 격리하여 안전하게 개발할 수 있습니다.
- ▶ dispatching-parallel-agents로 대규모 리팩토링 시 여러 모듈을 동시에 수정하면 시간을 대폭 절약합니다.

- ▶ systematic-debugging 스킬은 Claude Code의 코드베이스 탐색 능력과 결합하면 근본 원인을 빠르게 추적합니다.
- ▶ verification-before-completion 스킬로 PR 생성 전 자동 검증을 강제하여 코드 품질을 일관되게 유지하세요.

Superpowers 한눈에 보기



Superpowers = AI 코딩 에이전트를 위한 "구조화된 개발 방법론 플러그인"

14개 스킬 × 6단계 워크플로 × 멀티 플랫폼 지원

→ 즉흥 코딩에서 체계적 설계·구현·검증 프로세스로 전환

Claude Code 한눈에 보기



Claude Code = 터미널·IDE·웹 통합 AI 코딩 어시스턴트

코드 작성 / Git 자동화 / 멀티 에이전트 / MCP 통합 / 스케줄 자동화

→ CLAUDE.md로 영구 지시사항 유지, /skills로 팀 워크플로 표준화

시작하기 위한 3단계

- 설치: `irm https://claude.ai/install.ps1 | iex` (Windows)
`curl -fsSL https://claude.ai/install.sh | bash` (Mac/Linux)
- Claude Code에서: `/plugin install superpowers@claude-plugins-official` 실행
- 첫 사용: `cd 프로젝트폴더 → claude → /brainstorming` 으로 설계 시작

참고 URL 모음

Superpowers GitHub	https://github.com/obra/superpowers
Claude Code 공식 문서	https://code.claude.com/docs/en/overview
Claude Code 퀵스타트	https://code.claude.com/docs/en/quickstart
CLAUDE.md 메모리 가이드	https://code.claude.com/docs/en/memory
Claude Code 베스트 프랙티스	https://code.claude.com/docs/en/best-practices
플러그인 마켓플레이스	https://claude.ai/marketplace