

# Claude Code 빠른 참조

CLI 명령어, 워크플로우, MCP, hooks, 설정

## 시작하기

### 설치 & 실행

```
npm install -g @anthropic-ai/claude-code
claude # start interactive session
claude --version # check version
claude update # update to latest
```

### 인증

```
claude login # browser OAuth
export ANTHROPIC_API_KEY="sk-ant-..."
claude logout # clear session
```

## 슬래시 명령어

### 세션 명령어

```
/help 사용 가능한 명령어 표시
/clear 대화 히스토리 지우기
/compact 토큰 절약을 위해 컨텍스트 압축
/cost 토큰 사용량 및 비용 표시
/status 세션 정보 및 모델 표시
/new 새 대화 시작
/config 설정 열기 또는 보기
/doctor 설정 문제 진단
/init 프로젝트용 CLAUDE.md 생성
/login Anthropic 인증
/logout 인증 해제
```

### 워크플로우 명령어

```
/bug 버그 리포트 제출
/review 코드 리뷰 요청
/pr 풀 리퀘스트 생성 또는 업데이트
/commit 스테이저된 변경사항 커밋
```

## 키보드 단축키

```
Ctrl+C 현재 생성 취소
Ctrl+D Claude Code 종료 (EOF)
Escape 현재 입력 / 편집 취소
Tab 파일 경로 및 명령어 자동완성
Up/Down 입력 히스토리 탐색
```

## CLI 플래그

### 주요 플래그

```
-p, --print 비대화형 (헤드리스) 모드
--model 모델 설정: opus, sonnet, haiku
--output-format 출력 형식: text, json, stream-json
--allowedTools 도구 제한: Edit, Read, Bash
--max-turns N 대화 턴 수 제한
--debug 디버그 로깅 활성화
-n, --name 세션 이름 지정
```

### 헤드리스 / 스크립팅

```
claude -p "explain this error" < log.txt
claude -p "list todos" --output-format json
echo "fix typos" | claude -p
```

## 설정 파일

### CLAUDE.md 계층 구조

```
./CLAUDE.md 프로젝트 수준 지침 (레포에 커밋)
./.claude/settings.json 프로젝트 설정 (퍼미션, hooks)
~/.claude/CLAUDE.md 사용자 전역 지침 (모든 프로젝트)
~/.claude/settings.json 사용자 전역 설정
~/.claude/projects/*/CLAUDE.md 프로젝트별 사용자 지침 (레포 외)
```

### 환경 변수

```
ANTHROPIC_API_KEY 직접 인증용 API 키
CLAUDE_MODEL 기본 모델 재정의
CLAUDE_CONFIG_DIR 커스텀 설정 디렉토리 경로
```

## 퍼미션

### settings.json

```
{
  "permissions": {
    "allow": ["Read", "Glob", "Grep"],
    "deny": ["Bash(rm *)"]
  }
}
```

### 퍼미션 모드

```
default 위험한 도구 사용 전 확인
--dangerously-skip-permissions 모든 도구 허용 (CI/스크립트 전용)
--allowedTools CLI 플래그로 도구 집합 제한
```

## MCP 서버

### MCP 서버란?

Model Context Protocol 서버는 데이터베이스, API, 서비스 등 커스텀 도구로 Claude Code를 확장합니다. CLI 또는 `.mcp.json`으로 관리합니다.

### CLI 관리

```
claude mcp add server-name -- cmd arg1 arg2
claude mcp list
claude mcp remove server-name
```

### .mcp.json 설정

```
{
  "mcpServers": {
    "my-db": {
      "command": "python",
      "args": ["-m", "mcp_server_db"],
      "env": { "DB_URL": "${DATABASE_URL}" }
    }
  }
}
```

## 범위

```
.mcp.json 프로젝트 수준 MCP 서버
~/.claude/mcp.json 사용자 전역 MCP 서버
claude mcp add -s user 사용자 범위에 추가
claude mcp add -s project 프로젝트 범위에 추가
```

## hooks

### hook 이벤트

```
PreToolUse 도구 실행 전 실행
PostToolUse 도구 완료 후 실행
Notification Claude가 알림 전송 시 실행
Stop Claude가 응답 완료 시 실행
```

### settings.json 예시

```
{
  "hooks": {
    "PreToolUse": [{
      "matcher": "Bash",
      "hooks": [{
        "type": "command",
        "command": "check-command.sh $INPUT"
      }]
    }]
  }
}
```

## hooks 동작

```
exit 0 도구 실행 허용
exit 2 도구 실행 차단
stdout Claude에게 JSON 피드백 전달
```

## SDK & 자동화

### 헤드리스 스크립팅

```
# Single prompt, get JSON output
claude -p "summarize main.py" \
  --output-format json

# Pipe input
git diff | claude -p "review this diff"
```

### CI / GitHub Actions

```
- uses: anthropics/claude-code-action@v1
  with:
    claude_args: >
      --max-turns 5
      --model claude-sonnet-4-6
```

## 팁

자동화에서 비용 제한에는 `--max-turns`를 사용하세요. 결과를 프로그래밍 방식으로 파싱하려면 `--output-format json`을 사용하세요. 안전을 위해 `--allowedTools`와 함께 조합하세요.

## 모델

### 모델 선택

```
claude --model opus # most capable
claude --model sonnet # balanced (default)
claude --model haiku # fastest, cheapest
```

## 모델 단축키

```
opus Claude Opus — 최고 성능
sonnet Claude Sonnet — 기본값, 균형
haiku Claude Haiku — 빠르고 저렴
--model full-name 예: claude-sonnet-4-6
```

## 모범 사례

### CLAUDE.md 작성

프로젝트 관례, 기술 스택, 빌드 명령어, 테스트 지침을 CLAUDE.md에 작성하세요. 간결하게 유지하세요 — Claude가 매 세션 읽습니다. ``/init``으로 스케폴딩할 수 있습니다.

### 비용 관리

```
/cost 현재 토큰 사용량 확인
/compact 컨텍스트가 커지면 압축
--max-turns 자동화에서 턴 수 제한
sonnet / haiku 단순 작업에는 저렴한 모델 사용
```

### 효과적인 프롬프팅

```
Be specific "auth.ts:42의 null 체크 수정" > "버그 수정"
Give context 파일, 함수, 오류 메시지 참조
Use @file 파일 직접 참조: @src/main.ts
Use images 시각적 컨텍스트를 위해 스크린샷 끌어다 놓기
Iterate 후속 질문으로 개선; /clear로 초기화
```