

# Conventional Commits 빠른 참조

커밋 메시지 형식, 타입, 스코프, 브레이킹 체인지

## 형식

### 커밋 메시지 구조

```
<type>[optional scope]: <description>
[optional body]
[optional footer(s)]
```

### 구성 요소 설명

<b>type</b>	변경 범주 (feat, fix 등)
<b>scope</b>	영향받는 코드베이스 영역 (선택)
<b>description</b>	명령형 어조의 짧은 요약
<b>body</b>	상세 설명 (선택, 빈 줄 후 작성)
<b>footer</b>	BREAKING CHANGE 또는 이슈 참조 등 메타데이터

### 규칙

<b>Imperative mood</b>	"add feature" not "added feature"
<b>Lowercase type</b>	feat: 이며 Feat: 가 아님
<b>No period</b>	설명 끝에 마침표 없음
<b>Blank line</b>	설명과 본문/푸터 사이 빈 줄

## 타입

### 핵심 타입 (SemVer 관련)

<b>feat</b>	새 기능 (MINOR 버전 업)
<b>fix</b>	버그 수정 (PATCH 버전 업)

### 확장 타입 (일반 관례)

<b>build</b>	빌드 시스템 또는 외부 의존성
<b>chore</b>	유지보수 작업 (소스/테스트 변경 없음)
<b>ci</b>	CI 설정 및 스크립트
<b>docs</b>	문서만 변경
<b>perf</b>	성능 개선
<b>refactor</b>	버그 수정도 기능 추가도 아닌 코드 변경
<b>revert</b>	이전 커밋 되돌리기
<b>style</b>	포매팅, 공백 (CSS 스타일링 아님)
<b>test</b>	테스트 추가 또는 수정

## 스코프

### 스코프 사용

```
feat(auth): add OAuth2 login flow
fix(parser): handle empty input gracefully
docs(readme): update installation steps
refactor(api): extract validation middleware
```

### 스코프 가이드라인

<b>Module name</b>	feat(auth); fix(parser):
<b>Layer name</b>	feat(api); fix(db):
<b>Feature area</b>	feat(search); fix(checkout):
<b>Dependency</b>	build(deps); chore(npm):
<b>Omit if broad</b>	횡단 관심사 변경 시 스코프 생략

## 브레이킹 체인지

### 브레이킹 체인지 표시

```
feat!: remove deprecated login endpoint
feat(api)!: change response format to JSON:API
fix!: drop Node 14 support
```

### 푸터 방식 브레이킹 체인지

```
feat(api): change user endpoint response
BREAKING CHANGE: response now returns array
instead of object. Update client parsing.
```

## 규칙

<b>! after type/scope</b>	브레이킹 체인지 약속 표시
<b>BREAKING CHANGE:</b>	푸터 토큰 (항상 대문자)
<b>BREAKING-CHANGE:</b>	하이픈 형식도 유효
<b>SemVer impact</b>	MAJOR 버전 업 트리거
<b>Any type</b>	브레이킹 체인지는 모든 타입에 적용 가능

## 예시

### 간단한 커밋

```
feat: add email notifications
fix: prevent race condition in checkout
docs: correct typo in contributing guide
style: format with prettier
refactor: simplify error handling logic
```

### 스코프 포함

```
feat(blog): add comment threading
fix(auth): refresh token before expiry
test(api): add missing edge case coverage
ci(github): add Node 20 to test matrix
```

### 본문 및 푸터 포함

```
fix(parser): handle nested quotes correctly

Previously, nested quotes caused the parser
to enter an infinite loop. This adds a depth
counter to prevent unbounded recursion.

Closes #234
```

## 푸터

### 푸터 토큰

<b>BREAKING CHANGE:</b>	브레이킹 API 변경 설명
<b>Closes #123</b>	머지 시 이슈 자동 닫기
<b>Fixes #456</b>	이슈 자동 닫기 (수정 참조)
<b>Refs #789</b>	닫지 않고 이슈 참조
<b>Reviewed-by: name</b>	리뷰어 표시
<b>Co-authored-by: name</b>	공동 작성자 표시

### 복수 푸터

```
feat(api)!: redesign authentication flow

Migrate from session-based to JWT auth.

BREAKING CHANGE: /auth endpoints changed
Closes #101
Refs #98, #99
```

## 도구

### 커밋 린팅

```
npm install -D @commitlint/cli \
  @commitlint/config-conventional
echo "module.exports = { extends: \
  ['@commitlint/config-conventional'] }" \
  > commitlint.config.js
```

## 주요 도구

<b>commitlint</b>	커밋 메시지 컨벤션 린팅
<b>husky</b>	Git 훅 (커밋 시 commitlint 실행)
<b>commitizen (cz)</b>	대화형 커밋 메시지 작성기
<b>standard-version</b>	자동 changelog + 버전 업
<b>semantic-release</b>	완전 자동 릴리즈 파이프라인
<b>release-please</b>	Google의 릴리즈 자동화 도구

### Commitizen 설정

```
npm install -D commitizen \
  cz-conventional-changelog
npx commitizen init cz-conventional-changelog
# Use: npx cz (or git cz with alias)
```

## 일반 패턴

### 버전 업 매핑

<b>fix:</b>	PATCH (1.0.0 → 1.0.1)
<b>feat:</b>	MINOR (1.0.0 → 1.1.0)
<b>BREAKING CHANGE</b>	MAJOR (1.0.0 → 2.0.0)
<b>docs:, style:, etc.</b>	버전 업 없음

### Changelog 그룹화

```
## [1.2.0] - 2026-03-27
### Features
- add email notifications (abc1234)
### Bug Fixes
- prevent race condition (#123) (def5678)
```

### 리버트 형식

```
revert: feat(blog): add comment threading
This reverts commit abc1234def5678.
```