

CONVENTIONAL COMMITS クイックリファレンス

コミットメッセージの形式、タイプ、スコープ、破壊的変更

フォーマット

コミットメッセージの構造

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

各部の説明

- type** 変更のカテゴリ (feat, fix など)
- scope** 影響を受けるコードベースのセクション (省略可)
- description** 命令形での短い要約
- body** 詳細な説明 (省略可、空行の後に記述)
- footer** BREAKING CHANGE やイシュー参照などのメタデータ

ルール

- 命令形** "add feature" であり "added feature" ではない
- type は小文字** feat: であり Feat: ではない
- ピリオドなし** 説明は "" で終えない
- 空行** 本文/フッターと説明を空行で区切る

タイプ

コアタイプ (SemVer 関連)

- feat** 新機能 (MINOR バージョンアップをトリガー)
- fix** バグ修正 (PATCH バージョンアップをトリガー)

拡張タイプ (一般的な慣習)

- build** ビルドシステムや外部依存の変更
- chore** メンテナンスタスク (src/test の変更なし)
- ci** CI 設定とスクリプト
- docs** ドキュメントのみの変更
- perf** パフォーマンス改善
- refactor** 修正でも機能追加でもないコード変更
- revert** 以前のコミットを元に戻す
- style** フォーマット、空白 (CSS スタイルではない)
- test** テストの追加または修正

スコープ

スコープの使用例

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

スコープのガイドライン

- モジュール名** feat(auth):、fix(parser):
- レイヤー名** feat(api):、fix(db):
- 機能領域** feat(search):、fix(checkout):
- 依存関係** build(deps):、chore(npm):
- 広範な場合は省略** 横断的な変更にはスコープを使わない

破壊的変更

破壊的変更のマーク方法

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

フッター形式の破壊的変更

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

ルール

- type/scope の後の !** 破壊的変更の短縮マーカー
- BREAKING CHANGE** フッタートークン (常に大文字)
- BREAKING-CHANGE** ハイフン形式も有効
- SemVer への影響** MAJOR バージョンアップをトリガー
- 任意のタイプ** 破壊的変更は任意のタイプに適用可能

例

シンプルなコミット

```
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
```

スコープ付き

```
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
```

本文とフッター付き

```
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
```

フッター

フッタートークン

- BREAKING CHANGE:** API の破壊的変更を説明
- Closes #123** マージ時にイシューを自動クローズ
- Fixes #456** イシューを自動クローズ (修正参照)
- Refs #789** クローズせずにイシューを参照
- Reviewed-by: name** レビューアのクレジット
- Co-authored-by: name** 共同著者のクレジット

複数フッター

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

Migrate from session-based to JWT auth.

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

ツール

コミットのリンティング

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

人気のツール

- commitlint** コミットメッセージを規約に沿ってリント
- husky** Git フック (コミット時に commitlint を実行)
- commitizen (cz)** 対話形式のコミットメッセージビルダー
- standard-version** 自動チェンジログとバージョンバンプ
- semantic-release** 完全自動化されたリリースパイプライン
- release-please** Google のリリース自動化ツール

Commitizen のセットアップ

```
npm install -D commitizen \
  cz-conventional-changelog
npx commitizen init cz-conventional-changelog
# 使用方法: npx cz (またはエイリアスで git cz)
```

よくあるパターン

バージョンバンプのマッピング

- fix:** PATCH (1.0.0 → 1.0.1)
- feat:** MINOR (1.0.0 → 1.1.0)
- BREAKING CHANGE** MAJOR (1.0.0 → 2.0.0)
- docs:, style: など** バージョンバンプなし

チェンジログのグループ化

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

リバートの形式

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