Référence rapide Conventional Commits

Format de message de commit, types, portées, changements majeurs

Format

Structure du message de commit

```
<type>[portée optionnelle]: <description>
[corps optionnel]
[pied de page optionnel(s)]
```

Explication des parties

type	Catégorie de changement (feat, fix, etc.)
scope	Section du code affectée (optionnel)
description	Résumé court à l'impératif
body	Explication détaillée (optionnel, après une ligne vide)
footer	Métadonnées comme BREAKING CHANGE ou références d'issues

Règles

Mode impératif	"add feature" pas "added feature"
Type en minuscules	feat: pas Feat:
Sans point final	La description ne se termine pas par "."
Ligne vide	Séparer corps/pied de page de la description

Types

Types principaux (pertinents pour SemVer)

feat	Nouvelle fonctionnalité (déclenche une montée MINOR)
fix	Correction de bug (déclenche une montée PATCH)

Types étendus (convention commune)

build	Système de build ou dépendances externes
chore	Tâches de maintenance (sans changement src/test)
ci	Configuration et scripts CI
docs	Changements de documentation uniquement
perf	Amélioration des performances
refactor	Changement de code sans correction ni ajout
revert	Annule un commit précédent
style	Formatage, espaces blancs (pas le style CSS)
test	Ajout ou correction de tests

Portée

Utilisation de la portée

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

Conseils sur la portée

Nom de module	feat(auth):, fix(parser):
Nom de couche	feat(api):, fix(db):
Zone fonctionnelle	feat(search):, fix(checkout):
Dépendance	build(deps):, chore(npm):
Omettre si large	Pas de portée pour les changements transversaux

Changements majeurs

Marquer les changements majeurs

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

Changement majeur en pied de page

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

Règles

! après type/portée	Raccourci pour marquer un changement majeur
BREAKING CHANGE:	Token de pied de page (toujours en majuscules)
BREAKING-CHANGE:	Aussi valide (forme avec tiret)
Impact SemVer	Déclenche une montée MAJOR
Tout type	Les changements majeurs s'appliquent à tout type

Exemples

Commits simples

```
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
```

Avec portée

```
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
```

Avec corps et pied de page

```
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
```

Pied de page

Tokens de pied de page

BREAKING CHANGE:	Décrit un changement d'API majeur
Closes #123	Fermer automatiquement l'issue à la fusion
Fixes #456	Fermer automatiquement l'issue (référence fix)
Refs #789	Référencer l'issue sans la fermer
Reviewed-by: nom	Attribution du relecteur
Co-authored-by: nom	Attribution du co-auteur

Pieds de page multiples

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

Migrate from session-based to JWT auth.

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

Outillage

Validation de commits

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

Outils populaires

commitlint	Valider les messages de commit contre la convention
husky	Hooks git (exécuter commitlint au commit)
commitizen (cz)	Constructeur interactif de messages de commit
standard-version	Changelog auto + montée de version
semantic-release	Pipeline de release entièrement automatisé
release-please	Outil d'automatisation de release de Google

Configuration Commitizen

```
npm install -D commitizen \
  cz-conventional-changelog
npx commitizen init cz-conventional-changelog
# Utiliser : npx cz (ou git cz avec alias)
```

Patterns courants

Correspondance montée de version

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:, etc.	Pas de montée de version

Groupement dans le changelog

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

Format de revert

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