

# Conventional Commits

# Qu'est-ce qu'un Conventional Commit ?

- **Définition :**

- Un format standardisé pour les messages de commit Git.
- Aide à rendre l'historique de commits lisible et compréhensible.

- **Structure :**

```
<type>[optional scope]: <description>
```

```
[optional body]
```

```
[optional footer(s)]
```

# Pourquoi utiliser Conventional Commits ?

- **Avantages :**

- Facilite la génération automatique de journaux de changements (*changelogs*).
- Aide à maintenir des versions (*semantic versioning*) cohérentes.
- Améliore la lisibilité des contributions au projet.
- Utile pour les équipes collaboratives.

# Les types principaux

1. **feat** : ajout d'une nouvelle fonctionnalité.
2. **fix** : correction d'un bug.
3. **docs** : modification ou ajout de documentation.
4. **style** : modifications qui n'affectent pas la logique. (ex: indentation)
5. **refactor** : modification de code sans ajout de fonctionnalité ou correction de bug.
6. **test** : ajout ou mise à jour de tests.
7. **chore** : tâches diverses (ex : mise à jour de dépendances).

## Exemple d'un bon commit

```
feat(login): add OAuth2 authentication
```

```
Added support for OAuth2 authentication. This change includes updates  
to the login service and associated tests.
```

```
BREAKING CHANGE: The login endpoint now requires an 'authType' field.
```

## Scope : À quoi ça sert ?

- **Scope** : optionnel, décrit la partie impactée du projet.
  - Exemples : `api`, `ui`, `auth`.
- Utile pour :
  - Contextualiser les modifications.
  - Faciliter la lecture dans de grands projets.

## Pieds de message (footer)

- Utilisé pour des informations additionnelles :
  - **BREAKING CHANGE** : indique une modification majeure.
  - Références aux tickets ou bugs (ex : Fixes #123 ).

# Pourquoi utiliser les Commits conventionnels

- Générer automatiquement des CHANGELOGs.
- Déterminer automatiquement un saut de version sémantique (basé sur les types de commits reçus).
- Communiquer la nature des changements aux coéquipiers, au public et aux autres parties prenantes.
- Déclencher des processus de compilation et de publication.
- Faciliter la contribution des personnes à vos projets en leur permettant d'explorer un historique des livraisons plus structuré.



## Liens et ressources utiles

- [Conventional Commits](#)
- [Semantic Versioning](#)
- [Git Commit Messages](#)