

06. Contribuer en pratique

Objectifs de la séance

- Savoir cloner, forker, créer une branche et ouvrir une pull request.
- Produire une contribution simple (code ou documentation).

1. Workflow Git : fork → branch → commit → PR

Le processus standard de contribution à un projet open source se déroule en plusieurs étapes :

1. Fork

- Sur GitHub/GitLab, cliquez sur *Fork* pour créer une copie du dépôt dans votre espace personnel.
- Cela vous permet de travailler librement sans affecter le dépôt officiel.

2. Clone

- Clonez votre fork en local :

```
git clone https://github.com/votre-utilisateur/nom-du-projet.git
cd nom-du-projet
```

3. Branch

- Créez une nouvelle branche dédiée à votre modification :

```
git checkout -b fix-typo-readme
```

4. Commit

- Effectuez vos modifications, puis validez-les avec un message clair :

```
git add .
git commit -m "docs: corrige une faute de frappe dans le README"
```

5. Push

- Envoyez votre branche vers votre fork :

```
git push origin fix-typo-readme
```

6. Pull Request (PR)

- Depuis GitHub/GitLab, ouvrez une PR vers le dépôt original.
- Donnez un titre explicite et décrivez vos changements dans la description.

2. Rédaction d'un bon message de commit

Un bon commit message est **court, précis et normé**.

- Format recommandé : type: description courte au présent
- Types fréquents :
 - feat : nouvelle fonctionnalité
 - fix : correction de bug
 - docs : documentation
 - test : ajout ou correction de tests
 - refactor : refonte sans changement fonctionnel

Exemple

```
fix: corrige l'erreur 500 lors de la création d'un utilisateur sans email
```

Astuce : le premier mot doit être à l'impératif, comme si vous donniez un ordre au code.

3. Revue de code et feedback

Une pull request passe généralement par une phase de revue :

- Mainteneurs : relisent votre code, posent des questions, demandent des modifications.
- Vous : répondez aux commentaires, ajustez votre PR, gardez un ton respectueux.
- Cas possibles :

- La PR est mergée : félicitations ! 🎉
- La PR est fermée sans merge : prenez le feedback comme un apprentissage, pas comme un échec.

Bonnes pratiques

- Soyez réactif sans vous précipiter.
- Répondez aux commentaires un par un.
- Mettez à jour la documentation si nécessaire.
- N'hésitez pas à remercier les relecteurs.

4. Mini-atelier

1. Forkez un petit projet open source (ex. : une bibliothèque utilitaire, une doc de framework).
2. Apportez une contribution simple (ex. correction d'une faute de frappe, ajout d'un exemple de code).
3. Créez une branche, faites votre commit et ouvrez une PR.
4. Notez le retour reçu et préparez-vous à y répondre.

5. Livrable de la séance

Une première contribution (PR ou issue) dans un projet open source.

Le livrable doit contenir :

- Le lien vers la PR ou l'issue ouverte.
- Une brève description de la modification apportée.
- (Optionnel) Le feedback reçu ou attendu.

6. Pour aller plus loin

- **Pro Git** (Scott Chacon, Ben Straub) – Chapitre sur GitHub/Git.
- **Conventional Commits** : <https://www.conventionalcommits.org>
- **GitHub Docs** – About pull requests.

- Exemples de workflows :
 - **Django** : PR avec review obligatoire.
 - **Symfony** : validation par un core team member avant merge.

7. Mini-FAQ

- Dois-je écrire mes commits en anglais ?

C'est préférable, surtout pour les projets internationaux. Mais certains projets francophones acceptent le français.

- Que faire si je fais une erreur dans ma PR ?

Pas de panique : corrigez votre branche et poussez à nouveau, la PR sera mise à jour automatiquement.

- Et si ma contribution est jugée inutile ?

C'est normal, cela arrive. Vous aurez tout de même appris le workflow et les bonnes pratiques.