

# **GITHUB ?**

## **DÉPOT GIT**

# **GITHUB ?**

## **FORGE LOGICIEL**

- Gestion de tickets (bug, nouvelles fonctionnalités ...)
- Outils de gestion de projets
- ...

# GITHUB ?

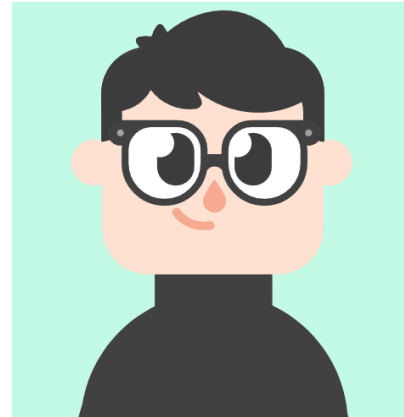
## LIÉ À L'OPEN SOURCE

- Rachat par Microsoft
- GitLab, <https://bitbucket.org/> (Jira) ...

# OPENSOURCE FLOW

Comment organiser le travail en équipe

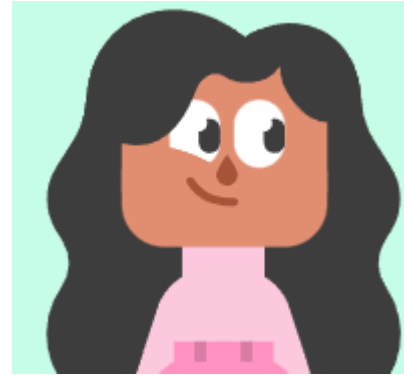
Pas le seul : [Git Flow](#), Trunk base developpement ...



Benjamin  
**Contributeur**

De passage sur le projet  
A du temps en ce moment

Améliorer le produit  
Propose des contributions



Elise  
**Mainteneur**

Passe régulièrement sur le projet  
N'as pas beaucoup de temps

Accompagner les contributeurs  
Vérifier/Valider les contributions

# OPENSOURCE FLOW

**Benjamin (Contributeur)**

Hey, la feature XXX m'intéresse bien, je peux m'en occuper ?

**Elise (Mainteneur)**

Ouai, c'est bon vas y.

# OPENSOURCE FLOW

**Benjamin (Contributeur)**

Tiens je t'ai fait une PR tu peux me la valider

**Elise (Mainteneur)**

Le code est pas mal mais ta documentation c'est pas ouf, tu me re-travaille ça



# OPENSOURCE FLOW

**Benjamin (Contributeur)**

Je t'ai fait une nouvelle version.

**Elise (Mainteneur)**

C'est nickel, je valide, bon travail 😊

**Robot**

\*Les commits ont été ajouté au dépôt\*

# AUTHENTIFICATION GITHUB

1. [Créer une clé SSH](#)
2. [Ajouter la clé publique à GitHub](#)

**POUR ELISE (MAINTENEUR)**



1. Créer un dépôt Github
2. Pousser notre historique local
3. Créer une issue

# AJOUTER LE DÉPÔT DISTANT

```
git remote add origin "url"
```

- origin convention
- Ajoute la section [remote "origin"] dans .git/config

# RENOME LA BRANCHE EN "MAIN"

```
git branch -M main
```

# ENVOYER LES COMMITS SUR LE DÉPÔT

```
git push -u origin main
```

- Ajoute la section [branch "main"] dans `.git/config`
- Crée une branche dans `.git/refs/remotes/origin/main`

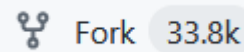
# POUR BENJAMIN (CONTRIBUTEUR)





# DUPLIQUER LE DEPOT GIT SUR SON ESPACE PERSONNEL

1. On se connecte sur [Github](#)
2. On recherche le dépôt sur lequel on veut contribuer
3. On appuie sur



# CRÉER UNE PR

1. Créer une branche
2. Pousser les modifications
3. Faire une PR
4. Mettre à jour la PR
5. Associé la PR à notre issue

**POUR ELISE (MAINTENEUR)**



# VALIDER LA PR

1. Valider la PR
2. Vérifier que l'issue est fermée
3. Vérifier l'historique du projet

# GESTION DES CONFLITS



# ON VA SUR LA BRANCHE PRINCIPAL

```
git switch main
```

# ON RÉCUPÈRE LES MODIFICATIONS

```
git pull
```

1. fetch: récupère les nouveau commit du dépôt sur refs/remotes/origin/main
2. merge sur notre branche principale (main)

# ON REBASE NOTRE BRANCHE FEATURE

```
git switch branch_name  
git rebase main branch_name
```



# ON POUSSE NOTRE BRANCHE FEATURE

```
git push -f
```

-f: force

# EXTENSIONS GITHUB

- [Cloner depuis VSCode](#)
- [Valider les PR depuis VSCode](#)

# BONNES PRATIQUES

1. **Faire de petites PR** pour faciliter la relecture et éviter les conflits
2. **Avoir un historique propre** pour faciliter la relecture

# SOMMAIRE

1. Créer un historique local
2. Travailler à plusieurs avec GitHub
3. [Utiliser la puissance de GIT](#)
4. [Pour aller plus loin](#)