

Initiation à Github



Présentation par [Nicolas Karasiak](#)

Réalisé en markdown avec [MARP](#)

Git ?

Github

Créer un dépôt / ou cloner



Cloner cette présentation :

<https://github.com/lennekade/GitPresentation>



Mettez-vous dans le repertoire de votre choix et :

```
git clone https://github.com/lennekade/GitPresentation
```



Cela créera un dossier nommé GitPresentation. Mais vous pouvez aussi choisir le nom du dossier :

```
git clone https://github.com/lennekade/GitPresentation  
SuperbePresentationGitDeNicolas
```

Maintenant que je sais cloné, je fais quoi ?

- Je regarde mon clonage, c'est pas tous les jours qu'on clone !

Mais si je veux corriger une erreur ou mettre à jour, je fais comment ?

- Pas comme ça !



Fourche en anglais ? Fork !

Un fork est une duplication sur votre compte Git d'un dépôt d'une personne tiers.

!? Pourquoi faire ?

Pour mettre à jour, modifier, ajouter des éléments, bref **travailler en groupe** sans demander au créateur*!

*On peut ajouter manuellement des contributeurs à son Git, mais les *fork* permettent à gentilles personnes d'améliorer/corriger des erreurs et de nous tenir au courant.

Après le clonage, on fourche, puis on clone encore !

Retourner sur le git :

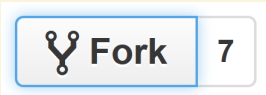
<https://github.com/lennepkade/GitPresentation>

Mais cette-fois fait un fork : 

Et après ? On reclone ! 

Et maintenant qu'on a cloné, forké, cloné ? on refork ?

Quand même pas 😊!

- On peut ajouter une image au dossier images 📁 📷
- On peut ajouter son nom au fichier [README.md](#)
- On peut ajouter l'image au fichier [README.md](#) avec les balises :
 - ``
 - 

J'ai fini ! Je fais comment pour mettre à jour ?

D'abord on met à jour **pour soi** 😎 :

 Pour voir les mises à jour, on tape dans le dossier de son git (terminal) :

```
git status
```

+ Pour ajouter les modifications, soit on ajoute un à un les fichiers

`git add monfichier.py`, soit on ajoute tout : `git add --all`

 On explique ce qu'on a fait : `git commit -m 'jai ajouté une image de chat'`

 On envoie ! `git push origin master`*

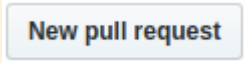
origin car c'est son propre dépôt, *master* car c'est la branche par défaut.

Plus pratique

Si on ajoute à chaque fois tous les fichiers. On peut faire en une seule ligne le `git add --all` et le `git commit -m 'mon message'` :

```
git commit -a -m 'ADD : Tous mes fichiers'
```

Pour partager

- Je partage avec quiconque mes modifications,
- Je peux demander à Nicolas de mettre à jour son dépôt avec mes modifications : . Mais attention, il peut y avoir des conflits :
 - Si plusieurs personnes modifient un fichier au même endroit
 - Si ma version est en retard sur celle du créateur

Conclusion

Ce tutorial est conçu et partagé sur Github.

Mais le protocole git est universel :



Il vaut surement mieux créer un git sur [Framagit](#) 👍 !