

GIT

Kévin NIEL





Qu'est-ce que GIT ?





1 - Installation

<https://git-scm.com/book/fr/v1/D%C3%A9marrage-rapide-Installation-de-Git>



2 - Concepts de base

Versionning

Sauvegarde

Travail à plusieurs



Les repository

Les repository sont l'équivalent des dossiers. C'est le terme utilisé au sein de GIT pour symboliser le dossier de base, celui qui pourra contenir d'autres dossiers ou d'autres fichiers, mais qui sera le principale.

Les repository peuvent être soit local, soit distant. Le premier correspond à un repository sur votre machine, l'autre correspond à un distant, hébergé sur une tierce partie.



GITHUB

Github est un système d'hébergement de fichiers, destiné à être utilisé à l'aide de GIT. C'est l'interface que nous allons utiliser pour l'apprentissage de Git, et je vous invite donc dès lors à y faire un tour et à vous y créer un compte dès à présent !

<https://github.com>

3 - Commandes





3.1 - git init

permet d'initialiser un repository en local, sur votre machine. Rendez vous dans le dossier concerné avant de lancer la commande, cela signifie qu'à partir de ce moment, GIT sera pris en considération dans votre dossier et vous pourrez l'utiliser complètement.



3.2 git remote add origin <url>

Cette commande permet de renseigner vers quel repository distant le repository local doit se synchroniser. Placez vous dans le dossier préalablement initialisé, puis lancez cette commande pour synchroniser les deux repository.

Paramètres :

- url : l'url du repository distant



Git clone <url> <name>

Cette commande permet de créer un repository local à partir d'un repository distant. Ainsi, les deux repository seront automatiquement reliés et il n'y aura aucune nécessité d'utiliser la commande "git remote add".

Paramètres :

- <url> : l'URL du repository distant
- <name> : le nom du dossier qui sera créé et lié avec le repository distant



git add <file>

Cette commande s'effectue en local, elle permet de signifier à GIT la nouvelle prise en compte du ou des fichiers spécifié(s)

Paramètres :

- file : Chemin / nom du fichier à prendre en compte
- astuce : vous pouvez spécifier un “.” à la place du nom de fichier, cela aura pour effet de systématiquement prendre en compte l'intégralité des fichiers changés ou ajoutés.



git commit -m <message>

Cette commande permet de réaliser un “screenshot” de votre repository à un instant “t”. Tous les changements apportés, et ayant fait l’objet de la commande “git add” seront alors instantanément sauvegardé **en local**.

Paramètres :

- <message> : message permettant de renseigner un libellé pour qualifier les modifications apportées lors du commit



git push

Cette commande permet de synchroniser les commits existants en local avec ceux existants sur le repository distant. De la sorte, l'ensemble des modifications seront envoyés sur le repository distant, et seront dans notre cas donc visibles sur le repository github.



git status

Cette commande affiche plusieurs choses :

- Les différences entre le dernier commit distant et le dernier commit local
- Les différences entre les fichiers du dernier commit distant et ceux du dernier commit local
- Met en évidence les fichiers qui ne sont pas suivi par GIT



git log

Cette commande affiche la liste de tous les commits que vous avez effectué avec dans l'ordre :

- un identifiant unique
- un auteur
- une date
- la description renseignée lors du commit (le message)



git reset --hard origin master

C'est la commande de reset par excellence de git. Si vous avez fait des modifications sur vos fichiers, depuis votre dernier commit, mais que vous ne souhaitez pas garder ces modifications, cette commande vous permettra de remettre l'ensemble de vos fichiers à l'état où il étaient lors du dernier commit.



git branch <nom>

Permet de créer une nouvelle branche. Une branche permet par exemple de décorer une fonctionnalité que vous êtes en train de développer, de votre code principal. De la sorte, en cas de nécessité de mise en production de votre branche principale, les modifications apportées sur votre nouvelle branche ne seront pas prises en compte et n'affecteront pas le code principal.

Paramètres :

- <nom> : le nom que vous souhaitez donner à votre nouvelle branche



git checkout <branch_name>

Commande permettant de se déplacer vers la branche “branche_name”.

Paramètres :

- <branch_name> : nom de la branche vers laquelle vous souhaitez vous déplacer



git merge <branch_name>

Cette commande permet de fusionner plusieurs branches. Par défaut, la branche “branche_name” sera fusionnée avec la branche sur laquelle vous vous trouvez.

paramètres :

- <branch_name> : le nom de la branche à fusionner avec celle courante



.gitignore

Le fichier “.gitignore” permet d’ignorer volontairement certains fichiers. Par exemple lorsque vous créez un fichier avec des données sensibles, il est commun de ne pas les intégrer à GIT pour des raisons évidentes de confidentialité.

Pour ajouter des fichiers ou des dossiers, il vous suffit d’ajouter leurs chemins par rapport à la base du repository (où se trouve également le fichier .gitignore”. Un chemin par ligne.



Le FORK

Le fork est un système permettant de “dupliquer” un repository distant qui ne vous appartient pas, vers un repository distant qui sera créé et vous appartiendra. Ce dernier sera donc une copie du repository original sur lequel vous pourrez travailler de la même manière que si vous l’aviez créé vous-même.

Cette action s’effectue directement sur l’interface de gestion des repository distants.



Les **PULL REQUESTS**

Les Pull Requests (PR) fonctionnent de pair avec les forks. Le principe est de demander au propriétaire du repository distant original de fusionner votre fork avec ledit repository original.

Cette action s'effectue directement sur l'interface de gestion des repository distants.



readme.md

Ce fichier est au format Markdown. Il se situe à la racine de votre projet et sera généralement affiché lorsque vous afficherez la page de votre repository distant (sur github par exemple).

Il contient en générale :

1. une description de votre projet
2. les prérequis à l'installation
3. un tuto d'installation step-by-step
4. les explications pour le paramétrage
5. ce qui est nécessaire en plus