Git - GitHub

...

Qu’est-ce que Git ?

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. En 2016, il s’agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes.
C'est un outil de bas niveau, qui se veut simple et performant, dont la principale tâche est de **gérer l'évolution du contenu d'une arborescence.**

Git est un logiciel de versioning créé en 2005 par Linus Torvalds, le créateur de Linux.

Un logiciel de versioning, ou logiciel de gestion de version est un logiciel qui permet de conserver un historique des modifications effectuées sur un projet afin de pouvoir rapidement identifier les changements effectuées et de revenir à une ancienne version en cas de problème.

Les logiciels de gestion de versions sont quasiment incontournables aujourd’hui car ils facilitent grandement la gestion de projets et car ils permettent de travailler en équipe de manière beaucoup plus efficace.

Parmi les logiciels de gestion de versions, Git est le leader incontesté et il est donc indispensable pour tout développeur de savoir utiliser Git.

Quelques liens utiles.

Télécharger Git :
https://gitforwindows.org/

Aller sur GitHub :
https://github.com/

Pour aller plus loin :
https://www.pierre-giraud.com/git-github-apprendre-cours/presentation-git-github
https://openclassrooms.com/fr/courses/5641721-utilisez-git-et-github-pour-vos-projets-de-developpement/5641728-decouvrez-la-magie-du-controle-de-versions

Pour aller plus loin :

https://www.git-tower.com/windows

https://www.gitkraken.com/

https://github.com/stevemao/awesome-git-addons

A quoi ça sert un logiciel de versioning ?

Le versioning va avoir plusieurs atouts :

- *Travailler à plusieurs sur un projet / sur les mêmes fichiers.*
- *Posséder un historique complet des modifications d'un projet.*
- *Modifier un projet sans impacter le rendu client.*
- Posséder plusieurs "sauvegardes des versions de travail".
- Pouvoir récupérer son travail sur n'importe quel support depuis une commande.

Avant tout, un peu de “vocabulaire”

Git dispose notamment des commandes suivantes :

- `git init` crée un nouveau dépôt ;
- *`git clone` clone un dépôt distant ;*
- `git add` ajoute de nouveaux objets *blob*s dans la base des objets pour chaque fichier modifié depuis le dernier *commit*. Les objets précédents restent inchangés ;
- `git commit` intègre la somme de contrôle SHA-1 d'un objet *tree* et les sommes de contrôle des objets *commit*s parents pour créer un nouvel objet *commit* ;
- `git branch` liste les branches ;
- *`git merge` fusionne une branche dans une autre ;*
- *`git rebase` déplace les commits de la branche courante devant les nouveaux commits d’une autre branche ;*
- *`git log` affiche la liste des commits effectués sur une branche ;*
- `git push` publie les nouvelles révisions sur le *remote*. (La commande prend différents paramètres) ;
- `git pull` récupère les dernières modifications distantes du projet (depuis le *Remote*) et les fusionne dans la branche courante ;
- *`git stash` stocke de côté un état non commité afin d’effectuer d’autres tâches.*
- `git statut` donne l’état de votre “branche”

Installer Git.

1 - Créer un profil sur github.com
Cela représentera un “bureau virtuel” où vous pourrez déposer/modifier vos travaux.

2 - Installer git.
https://gitforwindows.org/

Utiliser le GUI (Graphical User Interface) ou les lignes de commande ?
La question est donc : pourquoi utiliser la console plutôt que notre interface graphique utilisateur ? Tout simplement car certaines opérations sont beaucoup plus simples à réaliser via la console.

Initialisation de votre Git

Après l'installation, il est nécessaire de paramétrer certaines informations, directement via Git Bash.

Nous allons notamment ici renseigner un nom d'utilisateur et une adresse mail que Git devra utiliser ensuite.

On va donc taper les commandes suivantes :

```
git config --global user.name "Xxxx XXX"
git config --global user.email xxxxx@xxxx.com
```

Pour vérifier vos infos, il suffit de taper :

```
git config user.name
git config user.email
```

Démarrer un dépôt Git

Un “dépôt” correspond à la copie et à l’importation de l’ensemble des fichiers d’un projet dans Git.
Il existe deux façons de créer un dépôt Git :
- On peut importer un répertoire déjà existant dans Git
- On peut cloner un dépôt Git déjà existant.

La gestion des informations selon Git

Git pense les données à la manière d’un flux d’instantanés ou “snapshots”. A chaque fois qu’on va valider ou enregistrer l’état d’un projet dans Git, il va prendre un instantané du contenu de l’espace de travail à ce moment et va enregistrer une référence à cet instantané pour qu’on puisse y accéder par la suite.

Chaque instantané est stocké dans une base de donnée locale, c’est-à-dire une base de donnée située sur notre propre machine. Le fait que l’on dispose de l’historique complet d’un projet localement fait que la grande majorité des opérations de Git peuvent être réalisées localement, c’est-à-dire sans avoir à être connecté à un serveur central distant. Cela rend les opérations beaucoup plus rapides et le travail de manière générale beaucoup plus agréable.

Maintenant. Il est temps de créer le vôtre.

...