

GIT

Quelques liens.

<http://fr.openclassrooms.com/informatique/cours/gerez-vos-codes-source-avec-git>

<http://www.grafikart.fr/tutoriels/internet/git-github-131linux>

<https://doc.ubuntu-fr.org/git>

<http://www.moussu.fr/git/>

I- Les caractéristiques de GIT.

A - Qu'est-ce?

Git est un logiciel de gestion de versions.

« Un **logiciel de gestion de versions** (ou **VCS** en [anglais](#), pour *Version Control System*) est un [logiciel](#) qui permet de stocker un ensemble de [fichiers](#) en conservant la chronologie de toutes les modifications qui ont été effectuées dessus. Il permet notamment de retrouver les différentes [versions](#) d'un lot de fichiers connexes. » **wikipedia**

Il existe d'autres outils de gestion de versions tels que CVS/SVN – mercurial – TFS (Team Foundation Server) ...

B - Les principes.

Le dépôt.

GIT est un système distribué et non centralisé. Chaque personne a son propre dépôt. On peut avoir un dépôt central sur un serveur. Un dépôt est une copie du projet, il contient tous les fichiers ainsi que **l'historique** des changements.

Le principe est que chaque développeur a son dépôt local sur sa machine. Il travaille sur son projet et 'commit' localement les modifications.

Lorsqu'il souhaite que ses modifications soient vues par les autres, il 'push' son projet sur le dépôt central.

(Espace de travail – Dépôt local – Dépôt central)

II- Installation et configuration sur les postes de développement.

Installation de git sur votre poste de développement :

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
<https://git-scm.com/book/fr/v2/Démarrage-rapide-Installation-de-Git>

Configuration de git sur votre poste de développement :

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
<https://git-scm.com/book/fr/v2/Démarrage-rapide-Paramétrage-à-la-première-utilisation-de-Git>

Sur windows, on installe 'msys' qui permet d'émuler et donne une console dans laquelle on pourra taper les commandes linux (cd, mkdir, ls, etc...) ainsi que les commandes git.

- Sur votre poste de développement, créer un répertoire puis clic droit git bash.
- Configurer votre git local :
 - git config --global user.name "xxxx"
 - git config --global user.email xxxx@xxxx.xxx
- Pour vérifier la config : git config --list

Création du dépôt central sur Bitbucket (ou github) puis du dépôt local sur chaque machine développeur.

- Créer un compte pour chaque développeur sur Bitbucket.org – préciser team de 5 personnes maxi
- Un développeur crée un nouveau dépôt (repository) avec par exemple comme nom premierTest.
Inclure un fichier README.
Ce dépôt est pour l'instant quasi vide. Il n'existe que sur le dépôt central de Bitbucket.org.
- Ajouter les autres utilisateurs sur ce dépôt (Settings)
- Cloner ce dépôt sur chaque ordinateur :
 - Sur le site de bitbucket, Clic sur Clone pour copier le lien indiqué.
 - Sur chaque ordinateur, placez-vous dans un dossier:
 - Clic droit : Git Bash Here
 - Coller le lien qui permet de cloner puis valider : un dossier est alors créé.
Ce dossier porte le nom du projet.
(Vous obtenez le lien en cliquant sur clone depuis la page web de bitbucket)
Vous avez maintenant un dépôt local sur chaque machine.
- Modifier le dépôt local :
 - Créer un fichier dans ce dossier sur l'ordinateur local (index.html par exemple)
 - Placez-vous dans ce dossier depuis le terminal Git Bash (cd ...)
 - git status permet de voir l'état du dépôt local.
 - git add leFichier **puis** git status pour vérifier l'état
 - git commit -m 'ajout de fichier' **puis** git status pour vérifier l'état
- Mettre à jour le dépôt distant sur le serveur :
 - git remote add origin https://xxxx@bitbucket.org/xxxx/premierTest.git
permet de donner un alias (ici origin) à l'url complète.
 - git pull origin master : permet de récupérer sur le dépôt local les modifications éventuelles faites sur le dépôt central.
 - git push origin master : permet de mettre à jour le dépôt central.

Utilisation de git.

- Vous placer dans le dossier premierTest.
- Créer un fichier dans le dossier premierTest, par exemple contact.html
- git status : Pour _____
- git add contact.html Pour _____
- git status : Pour _____
- git commit Pour _____
- pour rendre disponible aux autres développeurs vos modifications :
 - git pull // pour récupérer les modifications éventuelles des autres développeurs
 - git push // pour valider vos modifications sur le dépôt central.

GIT – PROCEDURE.

Sur le serveur (bitbucket ou gitHub)

- Créer un compte pour chaque développeur.
- Créer un repository avec un des comptes comme indiqué précédemment.

Sur le client.

- Cloner ce repository pour les autres comptes comme indiqué précédemment.
- Eventuellement:
git remote add origin <https://xxxxxxxx/projet.git>

Récupération du projet sur le poste local :
- git clone

Modification du projet en local (Dans l'espace de travail):
- git add nomFichier puis git commit -m "...."
Ou git commit -a -m "....." // si modification de fichier
Sans -m, un éditeur de texte s'ouvre pour nous permettre d'écrire le message lié au commit. Dans ce cas, i pour écrire et :wq ! Pour quitter en sauvegardant
- git status // pour vérifier que tout est ok localement

Pull pour récupérer les modifications effectuées par les autres développeurs
- git pull origin master
Push (lorsque cela est nécessaire)
- git push origin master // le mot de passe est demandé

- si besoin, pour enlever le lien origin vers l'url :
git remote rm origin
- git remote : liste les dépôts distants
git remote -v : liste les dépôts distants avec leur url.