GESTIONNAIRE DE VERSIONS GIT

Git est un gestionnaire de version. Il permet :

- d'historiser les états successifs de chaque fichier
- des retours en arrière possibles
- le travail simultané par plusieurs personnes sur un seul fichier
- le travail en parallèle sur plusieurs branches

Un cours est disponible ici

Installer GIT

Windows

Installer Git. Penser à indiquer l'éditeur (vim par default, conseillé notepad++).

Linux

Avec les droits administrateur :

```
apt-get install git
```

Configuration

Avant d'utiliser Git, il faut entrer les commandes suivantes (git bash sous Windows, terminal sous Linux) :

```
git config --global user.name <nomUtilisateur>
```

```
git config --global user.email <mailUtilisateur>
```

Confusion de comptes git sur le poste :

Lors de la publication (push) des ressources sur gitHub, le nom de l'utilisateur stocké sur le poste peut ne pas être le bon. Git enregistre un fichier sous Windows.Il faut donc supprimer ce fichier.

`Aller dans le panneau de configuration | Comptes Utilisateurs | Gérer vos informations d'identification`.

Supprimer alors le fichier concernant git dans les informations d'identifications Windows.

Initialiser un dépot

Nouveau projet

Dans le répertoire contenant le projet :

```
git init
```

Astuce : Sous Windows on peut faire un clic droit → Git Bash Here pour se positionner directement dans le répertoire.

Création d'un dossier caché .git (ne pas toucher!).

Clone projet existant distant

```
git clone <url>
```

Cela crée le projet dans un sous-dossier du même nom. **Attention** : si le projet ne vous appartient pas, se référer à la partie ***remote***.

Commandes

Ajout de fichier

Pour ajouter un seul fichier :

```
git add <cheminDuFichier>
```

Pour ajouter tous les fichiers d'un coup :

```
git add -A
```

Commit

Voir les modifications en cours

Lister les fichiers modifiés depuis le dernier commit :

```
git status
```

Voir toutes les lignes modifiées sur le projet depuis le dernier commit :

```
git diff [cheminFichier]
```

Pour afficher les modifications d'un seul fichier, préciser le chemin.

Faire un commit

Remarque : Le commit concerne tout les fichiers précédemment ajoutés

```
git commit
```

cela ouvre l'éditeur de texte enregistré par défaut. On enregistre alors un message pour préciser les changements effectués.

```
git commit -m "<messageCommit>"
```

Cette commande permet de directement ajouter le message du commit sans passer par l'éditeur de texte.

Lister les commits

Voir les différents commits effectués sur la branche actuelle :

```
git log
```

Voir l'historique de tous les commits sur les différentes branches :

```
git log --graph --oneline --all
```

Annuler les modifications

Sur un fichier

Restaurer un fichier à son état lors du dernier commit :

```
git checkout -- <nomFichier>
```

Cycle de travail git

Pour le travail en équipe, un serveur git distant est nécessaire. GitHub permet de centraliser tous les versions réalisées par les différents membres d'une équipe. Chaque membre doit donc disposer d'un compte github. Afin d'éviter les conflits, une organisation de travail est impérative.

Avant même de pouvoir récupérer les modifications présentes en ligne vous allez devoir ajouter

l'ensemble de vos modifications puis créer le paquet correspondant.

```
git add .
```

puis

```
git commit -m "Votre message de commit" »
```

Attention ne bien mettre un commentaire explicite et crédible !

Vous pouvez maintenant récupérer les fichiers distants avec la commande :

```
git pull
```

Attention à bien vérifier que vos fichiers et lignes ne rentrent pas en collision avec les fichiers et lignes distants.

Incroyable maintenant vous pouvez Push vos fichiers.

```
git push
```

Sur un commit

Gérer une branche

Créer

Supprimer

Changer

Fusionner

Envoyer / récupérer changements

Remote

tag

show

autres

gitk

.gitignore

Gestion des conflits

GitHub

README

LICENCE

From:
https://wiki.sio.bts/ - **WIKI SIO : DEPUIS 2017**

Permanent link:
https://wiki.sio.bts/doku.php?id=git

Last update: **2022/06/21 17:16**