
TP_AP4A_C++

2.3 Explication de commandes Git

•Git config

Permet de configurer Git. Cette commande permet autant de configurer des informations de l'utilisateur que de modifier l'interface. Pour l'utilisateur on peut par exemple modifier l'adresse email de l'utilisateur avec `user.email`, pour l'interface on peut par exemple modifier les couleurs de Git avec `color.ui`. Ces configurations s'effectuent autant au niveau local que global.

•Git init

Créer un nouveau projet github vide ou étant une copie d'un projet existant

•Git status

Permet d'afficher l'état du répertoire de travail. Donne uniquement des informations, cette commande n'apportera aucun changement. Cette commande permet notamment de savoir où notre « HEAD » se situe (dans quel branche nous sommes actuellement. Et montre les autres branches du projet.

•Git add

Permet de placer une copie d'un fichier dans notre dossier de branche, on pourra ensuite l'ajouter à github en validant les modification puis en validant les modifications grâce à push.

•Git push

Envoie-le contenu de la branche actuelle ou spécifiée sur le clone de l'arbre. Cette commande permet notamment de valider des modifications effectuées sur un repo distant.

•Git merge

Crée un commit spécial qui a deux parents. Cette commande permet de « fusionner » deux commits en ajoutant les modifications d'une branche, apportées depuis la divergence avec la branche principale, dans la dernière version de la branche spécifiée.

•Git diff

Pour utiliser cette commande on lui passe deux fichiers en argument. La commande retourne les différences entre les deux fichiers. Cette commande fonctionne aussi si on lui précise un seul fichier. Dans ce cas il montrera la différence entre ce fichier et la version précédente de ce fichier.

•Git blame

Permet d'afficher l'historique d'un fichier en montrant notamment qui a mis à jour le fichier et à quel heure

