
RAPPORT COMMANDES GIT

git config : Commande globale qui permet de modifier certains paramètres de Git. Permet par exemple de customiser les couleurs du terminal grâce à `color.ui`. Nous pouvons également créer des alias permettant de réaliser des raccourcis de commandes comme la commande « `git config --global alias.ci commit` » permettant de réduire le mot « `commit` » en « `ci` » lors de l'appel d'une commande.

git init : Commande permettant de créer un nouveau projet. Elle permet ainsi de générer un dépôt de travail et un fichier `.git` ainsi que des sous-répertoires contenant toutes les données Git nécessaires. Cela assure ainsi la mise en place de son environnement de travail et assure également le bon fonctionnement de ce répertoire.

git status : Cette commande permet d'afficher des informations lorsque l'on a un doute sur l'état de notre environnement de travail. Cette commande permet ainsi d'afficher si l'on a oublié d'ajouter un document par la commande `add` avant de le `commit`. Elle permet également d'afficher sur quelle branche ou commit la HEAD se situe actuellement.

git add : Cette commande permet de mettre en premier plan un commit/document afin d'expliquer à Git que l'on souhaite que ce dernier soit transmis au dépôt distant lors de l'exécution de la commande « `git push` ».

git push : Il s'agit de la commande opposée de `git pull`. Tandis que `git pull` est utilisée pour récupérer le travail distant sur son travail local, la commande `git push` permet d'envoyer du travail sur le dépôt distant. Ainsi, toutes les personnes ayant accès au projet pourront accéder aux mises à jour du répertoire.

git merge : Cette commande permet de combiner le contenu de deux branches en créant une nouvelle branche qui possède toutes les caractéristiques des commits de ses branches parents.

git diff : Permet de visualiser sur le terminal Git la différence entre notre travail actuel local et le travail présent sur le dépôt distant. Elle permet ainsi de comparer des documents/commits dans des branches.

git blame : Cette commande permet de connaître l'auteur de la modification d'un document, l'heure où le document a été push/modifié, ainsi que d'examiner le contenu d'un fichier en l'analysant lignes par lignes. De manière générale, elle permet de dresser un historique des modifications qui ont été effectuées dans un dépôt, ce qui peut s'avérer utile lors d'un travail collaboratif.

