Les commandes push, pull et revert (2)

Auteur : Romain NOEL	**Date : 02/03/2020**	**Version : 1**
Périmètre de l'opération : Travailler à plusieurs, publier des sauvegardes et gérer les conflits		
Prérequis : Poste de travail configuré		
Outil : Git	**Durée :** 45min	

Opérations (ce qu'il faut faire)	**Points clés** (à quoi il faut faire attention)	**Argumentaire** (pourquoi c'est important)
Partie 1 : Push sans conflits		
1/ Attendre que votre binôme termine l'action 1.1		➢ Votre binôme publie ses commits
2/ Récupérer le dépôt de votre binôme cd .. git clone <http://repoGitBinome.git> cd <repoGitBinome>	Copie du dépôt de votre binôme sur votre machine	➢ Pour pouvoir travailler à deux sur une même base de code.
3/ Faire un commit touch <votreprenom> git add <votreprenom> git commit -m "Commit de <votreprenom>"		➢ Votre binôme ne bénéficiera pas de votre dernier commit puisqu'il n'existe que sur votre machine pour le moment.
4/ Attendre que votre binôme termine son action 1.4	Votre binôme publie un commit avant vous	➢ Votre dépôt local est donc en retard d'un commit par rapport au dépôt distant.
5/ Publier son commit git push	Cette commande retourne une erreur	➢ Vous n'êtes pas à jour avec le dépôt distant. Git refuse de publier votre commit et vous demande de vous mettre à jour.
6/ Se mettre à jour avec le repo distant et publier son commit git pull git push	Il n'y a pas de conflits.	➢ L'absence de conflits vous permet de publier votre code sans soucis pour peu que votre dépôt soit à jour.

Partie 2 : Pull avec conflits		
1/ Attendre que votre binôme effectue l'action 2.1		➢ Votre binôme génère un fichier servira à générer un conflit
2/ Récupération du jeu de données git pull git log ls	Vous venez de récupérer le jeu de données généré par votre binôme	
3/ Ajouter son prénom dans le fichier et le publier echo "Marie-Yasmina-Thu" >> prenoms.txt git add prenoms.txt git commit -m "Ajout de Marie-Yasmina-Thu" git push	Vous ajoutez votre prénom sur la première ligne du fichier.	➢ Votre binôme va en faire autant. C'est ce qui va générer un conflit !
4/ Suivre et aider votre binôme pour le reste de l'exercice		
Partie 3 : Pull avec conflits		
1/ Créer un fichier « animaux.txt » et le publier touch animaux.txt git add animaux.txt git commit -m "Ajout d'un fichier animaux.txt" git push		➢ Création d'un jeu de données qui va devenir conflictuel.
2/ Attendre que votre binôme effectue l'action 3.2	Votre binôme récupère le fichier que vous venez de créer	
3/ Ajouter un animal dans le fichier et faire un commit echo "Fou à pieds bleus" >> animaux.txt git add animaux.txt git commit -m "Ajout de Fou à pieds bleus"		
4/ Attendre que votre binôme effectue l'action 3.3		Votre binôme écrit également son prénom sur la première ligne de ce fichier et le publie.

5/ Publier son animal git push	La publication ne passe pas car votre dépôt n'est pas à jour.	
6/ Mettre à jour son dépôt git pull	Git vous indique que vous avez des conflits.	
7/ Identifier et résoudre les conflits cat animaux.txt sed -i "1d;3d;5d" animaux.txt cat animaux .txt	Git marque les conflits par les caractères suivants : <<<<<<<<<, ==========, et >>>>>>>>>. La commande sed supprime ces lignes (1, 3 et 5 étant les numéros des lignes qui marquent le conflit).	Ici, nous cherchons à conserver les deux lignes. Il suffit juste de supprimer les marqueurs de conflits. Dans certains cas, il faudra étudier quel bloc conserver, ou encore effectuer les fusions manuellement.
8/ Publier le code fusionné git add animaux.txt git commit -m "Conflit résolu. Conservation des deux animaux" git push	Création d'un nouveau commit résultat de la fusion des deux modifications. Et publication de ce dernier.	Le conflit est résolu
Partie 4 : Revert		
1/ Ajoutez et publiez une ligne dans le fichier animaux echo "tabouret" >> animaux.txt git add animaux.txt git commit -m "Ajout de tabouret" git push	Vous vous êtes trompés dans votre commit (un tabouret n'est pas un animal). Votre commit est déjà publié.	➢ La publication d'un commit est irréversible. Il va falloir réparer votre erreur.
2/ Créer un commit inverse git log copier le # du dernier commit git revert <#commit> echap -> :q -> entrée (sortir de vi) git log	Création d'un commit inverse (suppression de la ligne que vous vener d'ajouter)	➢ Cette commande permet de créer un commit inverse. Il aurait été possible de le l'effectuer à la main mais il vaut mieux faire confiance à git pour gérer des cas plus complexes.
3/ Publier son commit inverse git push	Votre erreur est réparée	➢ Ce mode de réparation d'erreur se voit dans les logs de git.