# TD - Gestion de versions avec git

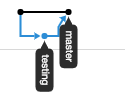
1. Installez git et configurez le nom et l'email.

Commandes:

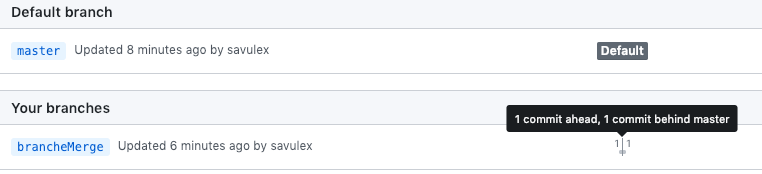
git config --global user.name "votre\_pseudo"

git config --global user.email moi@email.com

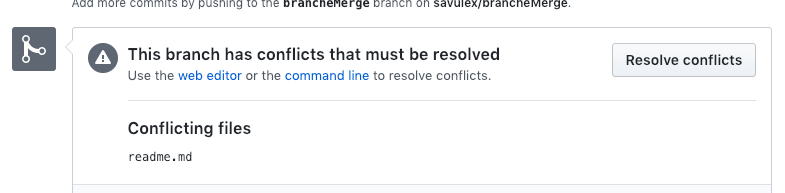
1. Démarrez un nouveau dépôt
   * Créez un nouveau dépôt vide
   * Ajoutez des fichiers dans l'index
   * Vérifiez le statut actuel du répertoire de travail
   * Commitez avec un message
   * Vérifiez l'histoire
     + Commande: **git log**
2. Nouveau commit
   * Modifiez les fichiers que vous venez de commiter
   * Regardez les modifications apportés à votre répertoire de travail
     + Commande: **git diff**
   * Ajoutez les fichiers modifiés à l'index
   * Commitez avec un message
   * Vérifiez dans l'histoire vos modifications des fichiers
     + Commande: **git show**
3. Github: publiez votre dépôt
   * Créez un nouveau projet sur Github
   * Dans le répertoire locale ajoutez comme remote votre projet
     + Commande: **git remote add**
   * Publiez votre répertoire
     + Commande: **git push**
   * Vérifiez l'historique de vos commits avec l'interface Github
4. Supprimez votre dépôt local
5. Github: clone
   * Clonez votre projet Github
     + Commande: **git clone**
   * Vérifiez l'historique localement
6. Nouvelle branche
   * Dans votre répertoire de travail clonée, créez une nouvelle branche
   * Faites un checkout de la nouvelle branche
   * Vérifiez que vous êtes sur la nouvelle branche
     + Commande: **git status**
   * Modifiez un des fichiers
   * Commitez les changements
   * Publiez votre nouvelle branche sur Github
     + Commande: **git push**
7. Requete pull
   * Dans l'interface Github, synchronize la branche master avec les changements de la branche que vous avez créée
     + Appuyez sur "Compare & pull request" sur la page Code ou sur la page des branches
     + Vérifiez votre modifications
     + Appuyez sur create pull request
     + Sur la page Pull requests acceptez le pull request en cliquant sur "Merge pull request" et après "Confirm"
     + Vérifiez l'historique de façon network:
       - Page Insights -> Network



1. Synchronization remote
   * Dans votre répertoire local, faites un checkout du master
   * Synchronizes votre depot
     + Commande: **git fetch**
   * Actualizes le master avec les changements du origin/master (le commit merge de l'exercice 8)
     + Commande: git pull origin master
   * Supprimez l'autre branche
     + Commande : **git branch -d**
2. Github: Fork
   * Faites un Fork sur Github du projet: <https://github.com/savulex/brancheMerge>
   * Analysez les deux branches : **master** + **brancheMerge**
     + Utilizes aussi la page Insights -> Network
   * Les deux branches ont divergé:



* + Creez un nouveau pull request. La fusion ne peut pas être réalisée .



Y a un conflit, car **master** a un nouveau commit qui a touche readme.md et **bracheMerge** a touche egalement a readme.md



1. Realisez la fusion en locale
   * Clonez le depot en locale
   * Faites un checkout du **brancheMerge**
   * Faites un merge avec **master**
     + Commande: **git merge**
   * Réalisez les modifications pour résoudre les conflits
     + Editez **readme.md**
   * Ajoutez **readme.md** a l'index et puis commitez
   * Publiez **brancheMerge** sur Github et faites un pull request sur **master**
   * Maintenant le merge est possible. Acceptez les modifications et puis regardez le Insights->Network
2. **Optionnel:** refaites exercice 11 mais cette fois ci, réalisez le merge avec l'interface Github (supprimez votre fork et refaites-le) .
3. Outil git
   * Installez un outil git de votre choix:
     + Github desktop
     + GitKraken
     + Sourcetree
     + Etc.
   * Refaites les exercices de 2 à 11 en utilisant l'outil que vous venez d'installer (pas d'interface en ligne de commande)