Formation sur Git

GLO-2004/IFT-2007

C’est quoi Git?

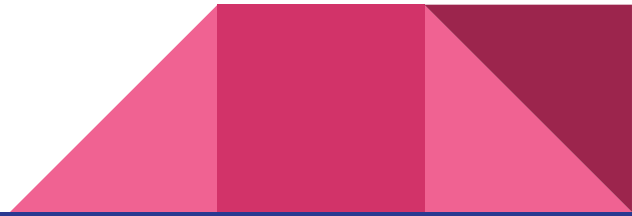
- Système distribué de contrôle de version
- Un des plus utilisé actuellement
- Relativement facile d’utilisation

Pourquoi une formation sur Git?

- Les échanges de fichiers par courriel/dropbox/google drive/pigeon voyageur/clé USB échangée par Fedex c'est non!
- Ça facilite le travail d'équipe
- Si quelqu'un fait tout planter, on peut revenir à une version précédente
- Je répète: Pas de pigeon voyageur pour partager le code

Installation

Instructions ici: http://bit.ly/1WQ50nb



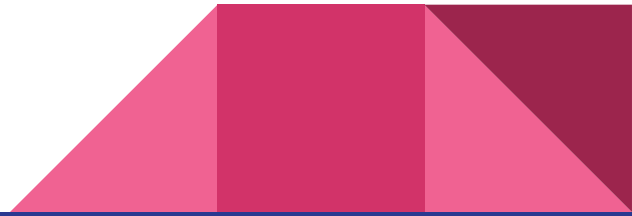
Avez-vous un repo?

Si non, allez sur github.com

Créez un compte si vous n’en avez pas et créez vous un repo.

GUI ou Command Line?

- À votre convenance.
- Si c’est un GUI, comprenez quand même ce qui se passe
- Formation d’aujourd’hui = command line



Configuration de votre repo

- Initiation du repo
  - En command line, allez dans le dossier parent d'où vous voulez placer le repo
  - git clone url_de_votre_repo_fsg
  - Voilà vous avez un repo!
- Vous ajoutez en tant qu'utilisateur avec vos credentials ulaval
  - cd mon_repo
  - git config user.email "toto.foo@ulaval.ca"
  - git config user.name "IDUL"
- Créer le gitignore
  - gitignore.io

Commandes importantes

- git status
  - Donne l'état du repo local
- git add nom_fichier
  - git add -A (inclut fichier non ajouté à git)
  - git add . (tous les fichiers déjà dans git qui ont eu des modifications)
  - Permet d'ajouter un fichier à la liste des fichiers suivis et/ou à un commit
- git commit -m"mon_message"
  - git commit -am"mon_message" (même chose que git add -A suivi de git commit -m"mon_message"
  - Permet de commiter les fichiers prêt à être envoyés sur git
- git pull origin ma_branche
  - Amène les modifications distantes sur la copie locale
  - TOUJOURS faire un pull avant de commencer à travailler!

Commandes importantes

- git checkout -b ma_branche
  - Change de branche vers une nouvelle branche qui s’appelle ma_branche
- git checkout ma_branche
  - Change de branche vers une branche existante qui s’appelle ma_branche
- git push origin ma_branche
  - Permet d’envoyer les commits sur git et de les partager aux autres sur la branche ma_branche
- git merge ma_branche
  - Permet de merger (fusionner) la branche ma_branche dans la branche actuelle
- git tag -a 1.0 -m”mon message”
  - Créer un tag nommé 1.0 avec comme indication “mon message”

Pour le cours...

- Les remises se font avec des tags!
- Respectez le nom du tag.
  - 1.0.1 != 1.0
  - Attention aux espaces avant et après
- Contribution au travail d’équipe == commit sur le repo
  - Votre setting de nom et de courriel est important.
  - Nous ne pouvons pas savoir qui est pieuvrePoilue192.
- Votre utilisation de Git est évalulée!

Gestion de conflit de merge

- Un conflit arrive lorsque 2 personnes ont modifié le même fichier et que Git ne peut pas déterminer quoi garder et ne pas garder.
- 2 options:
  - Utilisation d'un GUI
  - Résolution du conflit "à bras"

Quelques règles sur git en général!

- Ne JAMAIS utiliser --force sur un push
- S'il y a un conflit, il faut le résoudre.
- Toujours des fins de lignes Linux sur Git

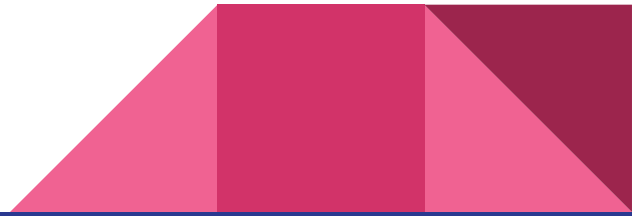
Amusons-nous!

Exercice de gestion de conflits

- Sur votre repo d'équipe:
  - Créer une branche awesome_feature
  - Créer un fichier README.md et ajouter votre nom sur la première ligne
  - Commit et push
  - Checkout master && git merge awesome_feature && git push origin master
  - Créer une branche another_awesome_feature
  - Créer un fichier README.md et ajouter votre nom sur la première ligne
  - Commit et push
  - Attendre que le merge soit fait pour awesome_feature
  - git checkout master && git merge another_awesome_feature
  - git pull origin master
  - Gérer le merge!

Outils utiles

- GitKraken (https://www.gitkraken.com/)
- Sourcetree (https://www.sourcetreeapp.com/)
- TortoiseGit (https://www.tortoisegit.org/)



Références utiles

- Documentation officielle: https://git-scm.com/docs
- (À faire!)Tutoriel: https://try.github.io
- (À faire!)Tutoriel sur les branches: http://learngitbranching.js.org/
- StackOverflow reference: http://stackoverflow.com/questions/315911/git-for-beginners-the-definitive-practical-guide
- Git for n00bs: http://rogerdudler.github.io/git-guide/