Git

pbn 2023

- **Bien démarrer**
  - Vérifier que git est installé
    - `git --version`
    - `which git`
  - Installer git
    - Site officiel — `https://git-scm.com/`
    - Dans Linux
      - `sudo apt-get update && sudo apt-get upgrade && sudo apt-get dist-upgrade`
      - `alias update='sudo apt-get update && sudo apt-get upgrade && sudo apt-get dist-upgrade'`
      - `sudo apt install git`
    - Sources de git — `https://github.com/git/git/tree/master`
  - Configurer git
    - Vérifier la configuration — `git config --list --show-origin`
    - Renseigner votre identité
      - `git config --global user.name "John Doe"`
      - `git config --global user.email johndoe@example.com`
    - Visual studio code par défaut — `git config --global core.editor "code --wait"`
    - Changer la branche par défaut — `git config --global init.defaultBranch main`
    - Vérifier les réglages utilisateur — `git config --list`
  - Documentation officelle (fr)
    - En ligne — `https://git-scm.com/book/fr/v2`
    - Hors ligne — `https://github.com/progit/progit2-fr/releases/download/2.1.71/progit.pdf`
  - Documentation officieuse (en) — `https://books.goalkicker.com/GitBook/`
- **Etudier les commandes**
  - Lister toutes les commandes — `git help -a`
  - Aide sur une commande
    - Longue — `git <command> —help`
      - `git status —help`
      - `git help status`
    - Courte — `git <command> -h` — `git status -h`
  - man linux — `man <git-command>` — Par exemple : `man git-add`
- **Dépot local**
  - 1) Créer le dossier — `mkdir /path/my_project`
  - 2) Se déplacer dans le dossier — `cd /path/my_project`
  - 3) Initialiser le dépôt — `git init`
  - 4) Vérifier l'état du dépôt
    - `git status`
    - `git status -v`
    - `git status -vv`
  - 5) Valider les modifications du dépot — `git commit -m "Initial commit"`
  - 6) Changer le message du dernier commit
    - `git commit --amend`
    - `git commit --amend -m "New commit message"`
- **Dépot distant**
  - Associer un dépôt local à un dépôt distant — `git remote add main https://gitlab.ictge.ch/owner/repository.git`
  - Cloner un dépôt distant
    - `cd \<path where you would like the clone to create a directory\>`
    - `git clone https://gitlab.ictge.ch/username/projectname.git`
  - Cloner un dépôt distant dans un dossier différent — `git clone https://github.com/username/projectname.git MyFolder`
  - Flux (fetch/push) — Vérifier les flux — `git remote -v`
- **Gérer les fichiers**
  - Remarque importante — *Contrairement à d'autres systèmes de gestion de versions, lorsque vous exécutez git commit, Git n'enregistre pas les modifications depuis le répertoire de travail mais depuis « l'index », une zone tampon servant à préparer le prochain commit. Les commandes git add et git rm permettent d'intégrer certains changements du répertoire de travail dans l'index.*
  - Créer un fichier — `touch README.md`
  - Editer un fichier
    - `code README.md`
    - `vim README.md`
  - Ajouter un fichier à l'index
    - `git add README.md`
    - `git stage README.md`
  - Ajouter plusieurs fichiers à l'index — `git add \<file/directory name #1\> \<file/directory name #2\> \< ... \>`
  - Ajouter tous les fichiers à l'index — `git add .`
  - `git add -p fichier` / `git stage -p fichier` — *Ajoute à l'index un sous-ensemble des changements présents dans le fichier. Les changements individuels sont extraits du patch global (cf. git diff) et sont proposés de manière interactive.*
  - Ajouter certains fichiers
    - `git add *.txt`
    - `git add Documentation/\*.txt`
  - Supprimer un fichier du répertoire de travail et de l'index — `git rm fichier`
  - Supprimer un fichier de l'index tout en le conservant dans le répertoire de travail — `git rm --cached fichier`
  - Valider les modifications de l'index dans le dépôt — `git commit -m "Initial commit"`
  - Exclure des fichiers de l'index — `.gitignore`
- **Historique**
  - Parcourir l'historique — `git log`
  - Limiter aux n derniers push — `git log -n`
  - Détaillé et joli — `git log --decorate --oneline --graph`
  - Alias
    - Créer un alias — `git config --global alias.lol "log --decorate --oneline --graph"`
    - Utiliser l'alias
      - `\# history of current branch :`
      - `git lol`
      - `\# combined history of active branch (HEAD), develop and origin/master branches :`
      - `git lol HEAD develop origin/master`
      - `\# combined history of everything in your repo :`
      - `git lol --all`
  - En couleurs — `git log --graph --pretty=format:'%C(red)%h%Creset -%C(yellow)%d%Creset %s %C(green)(%cr) %C(yellow)<%an>%Creset'`
  - En bref — `git log --oneline`
  - Combiné — `git log -2 --oneline`
  - Groupé par contributeurs — `git shortlog`
  - Afficher le détail des modifications — `git log --stat`
  - Afficher le détail d'un seul « commit » — `git show 48c83b3`
- **Les branches**
  - Documentation officielle — `https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Les-branches-en-bref`
  - Créer une nouvelle <branche> dans le commit courant — `git branch \<testing\>`
  - Basculer vers une <branche> — `git checkout \<testing\>`
  - Liste toutes les branches locales — `git branch`
  - Liste toutes les branches distantes — `git -r branch`
- **Les différences**
  - Afficher la différence entre original et new — `git diff original new`
  - Afficher tous les changements entre original et new — `git diff original..new`
  - Afficher les différences avec la dernière version — `git diff HEAD^ HEAD`
- **Commandes pratiques**
  - Remotes — Ajouter l'URL d'un deuxième dépôt distant pour que le push mette à jour les deux dépôts.
    - `git remote set-url --add`
    - `<remote_name>`
    - `<remote_url>`
  - Annuler des changements
    - Supprimer le dernier commit et retrouver les fichiers du commit précédent — `git reset --soft HEAD^`
    - Revenir à trois commits en arrière — `git reset HEAD^3`
    - Supprimer les changements d'un fichier dans l'index courant — `git reset <filename>`
    - Annuler les modifications d'un fichier dans le dossier de travail
      - `git checkout <filename>`
      - `git reset --hard <filename>`