GIT BASH CHEAT SHEET

BASIC BASH COMMANDS

Basic commands to get you started with Git Bash

- **cd [directory] or cd [path]** : change directory to [directory]
- **ls [directory] or ls [path]** : list directory contents
- **mkdir [directory]** : make directory
- **touch [file_name]** : create a new file
- **vim [file_name]** : Open a file in vim editor **cat [file_name]** : Display the contents of a file

SETUP

Configuring user information used across all local repositories

- **git config --global user.name "[firstname lastname]"** : set a name that is identifiable for credit when review version history
- **git config --global user.email "[valid-email]"** : set an email address that will be associated with each history marker
- **git config --global color.ui auto** : set automatic command line coloring for Git for easy reviewing

BASE COMMANDS

Commands to get you started with Git and Github

- **git init :** initialize an existing directory as a Git repository
- **git remote add origin [url] :** add a remote repository
- **git add [file] or git add :** add one or more files to staging area
- **git commit -m "[Type in the commit message]" :** commit changes to head (but not yet to the remote repository)
- **git push origin [branch] :** send changes to the master branch of your remote repository

TRACK YOUR PROJECT

Commands to help you track your project

- **git status** : list which files are staged, unstaged (added), and untracked
- **git log :** show all commits, starting with newest
- **git diff :** show changes to files that are not yet added
- **git diff -- staged** : show changes to files that are added but not yet committed

CLONING AND COLLABORATION

Commands to help you clone and collaborate with others

- **git clone [url] :** clone a repository that already exists on GitHub, including all of the files, branches, and commits
- **git pull :** fetch and merge changes on the remote server to your working directory

BRANCHES

Commands to help you manage branches

- **git branch [branch-name] :** create a new branch at the current commit
- **git checkout [branch-name] :** switch to another branch and check it out into your working directory
- **OU git checkout -b [branch-name]** : same as the TWO previous commands
- **git merge [branch]:** merge the specified branch's history into the current one

COMPTES

Pour permettre le multi account, lancer en premier lieu dans un bash git, la commande suivante : git config --list

Trouver la ligne : **credencial.helper=wincred**

Si elle ne s'y trouve pas, taper la commande :

git config --global credential.helper wincred

et maintenant activons le multi credential pour git avec :

git config --global credential.useHttpPath true