# Résumé des commandes Git et GitHub

## Création du projet sur le GitHub

Caractéristiques du projet  
**Nom du projet, version publique ou privée, ...**

## Connexion avec le git hub (localement)

Mise en relation avec le GitHub …

**>>** **git config --global user.name tux  
>>** **git config --global user.email tux@cmaisonneuve.qc.ca**

Exécution des instructions proposées pour un nouveaux projet (localement) …

**>>** **echo "# demohtaccess" >> README.md  
>>** **git init  
>>** **git add README.md  
>>** **git commit -m "Confirmation-01"  
>>** **git branch -M main  
>>** **git remote add origin <URL>** -- **origin correspond au nom que l'on veut utiliser pour le dépôt (généralement, on utilise origin)** -- **<URL> est le lien https vers le dépôt sur GitHub (un fichier git)** -- **Exemple ... git remote add origin "htpps://github.com/tux/demohtaccess"  
>>** **git push -u origin main** -- **origin spécifie le dépôt à mettre à jour** -- **main spécifie le nom de la branche qui doit être mise à jour**

## Modification au projet (localement)

On effectue les modifications au projet …

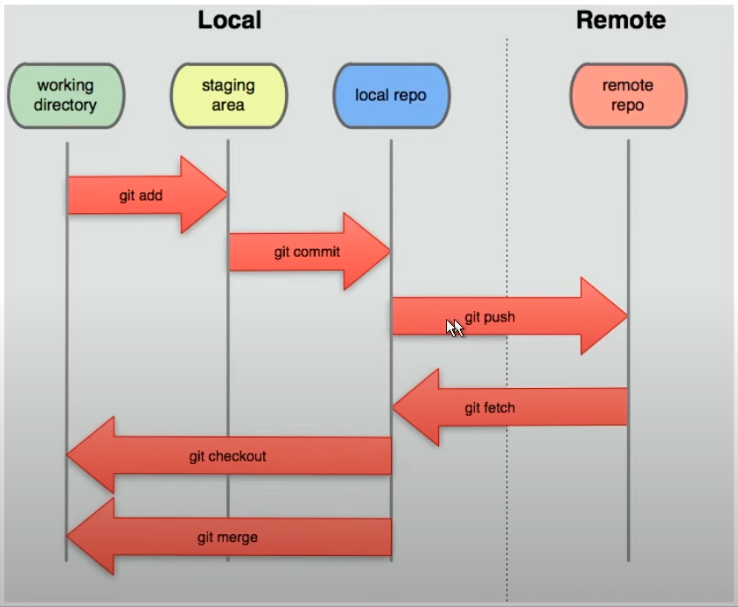
**Pour vérifier les modifications  
>>** **git status**   
**Pour valider (*staging*) les modifications  
>>** **git add . (git add --all)  
Pour vérifier les modifications  
>>** **git status   
Pour confirmer (*commit*) les modifications  
>>** **git commit -m "Confirmation-02"  
Pour téléverser les modifications au référentiel (dépôt)  
>>** **git push -u origin main**

**Remarque** …  
Il est aussi possible de télécharger les éléments du projet résident sur le référentiel distant de GitHub.  
Pour ce faire, on utilise la **commande git clone** …  
**Pour télécharger les éléments depuis le référentiel (dépôt)  
>>** **git clone <URL>** -- **Exemple ... git clone "htpps://github.com/tux/demohtaccess"**

## En bref

Voici les trois espaces de Git …

* Espace de travail ;
* Espace de validation (*staging*) ;
* Espace confirmé (*commit*).



## Création et fusion de branches

Mise en place de la branche …

**Pour vérifier l’état du projet  
>>** **git status**   
**Pour vérifier les confirmations (commit) et pour identifier les condensés  
>>** **git log   
Pour créer la nouvelle branche à partir d’un condensé**  **>>** **git branch brancheA 1788c**   
**Pour basculer vers la branche   
>>** **git checkout brancheA  
Pour visualiser la branche sur laquelle on travaille  
>>** **git branch  
Pour vérifier l’état du projet  
>>** **git status**   
**On effectue les modifications …   
Pour valider (*staging*) les modifications  
>>** **git add . (git add --all)  
Pour vérifier les modifications  
>>** **git status   
Pour confirmer (*commit*) les modifications  
>>** **git commit -m "Confirmation-A01"  
Pour revenir sur la branche principale  
>>** **git checkout main  
Pour fusionner le contenu de la brancheA à la branche principale  
>> git merge <branche> -m <message>  
 Exemple … git merge brancheA -m "Intégration des modifications de brancheA"   
Pour vérifier les modifications  
>>** **git status  
Pour valider (*staging*) les modifications  
>>** **git add . ou (git add --all)  
Pour confirmer (*commit*) les modifications  
>>** **git commit -m "Confirmation-03"  
Pour téléverser les modifications au référentiel (dépôt)  
>>** **git push -u origin main**