Commandes de base



Git config

Comme vous avez pu l’entrevoir dans [Démarrage rapide](https://git-scm.com/book/fr/v2/ch00/ch01-getting-started), vous pouvez spécifier les paramètres de configuration de Git avec la commande git config. Une des premières choses que vous avez faites a été de paramétrer votre nom et votre adresse de courriel :

//Configurer git pour tous les projets

$ git config --global user.name "sirine sabbeh"

$ git config --global user.email "[sabbehsyrine@gmail.com](mailto:sabbehsyrine@gmail.com)"

//pour contrôler

$ git config --global --list

Git init

git-init - Créer un dépôt Git vide ou réinitialiser un dépôt existant :

un repository git :ajouter un dossier caché .git dans laquelle gère les différents version de notre programme (c’est la base de donnée local de git)

Git status

Pour voir les fichiers non tarcké.

Pour voir l’état actuel des fichiers pour git.

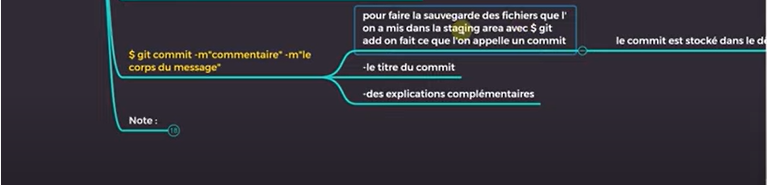
Git add nomdefichier

Git add . //ajouter tous les fichiers



Git reset nomdefichier

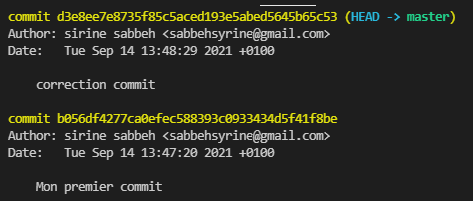
Git reset .//tous les fichier a comité



Git commit -m “déscription“ -m “déscription“

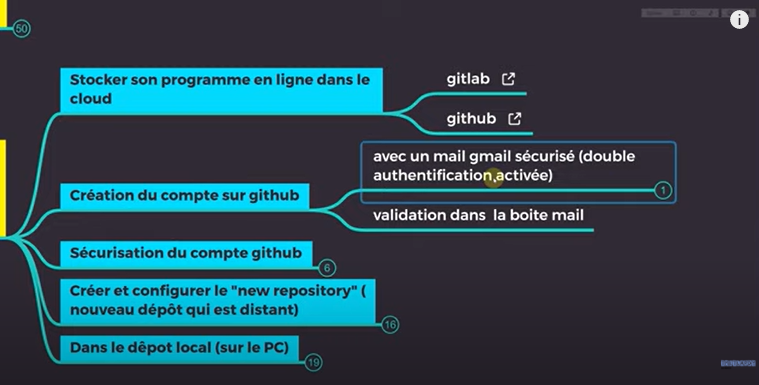
git commit -m "formation web septembre 2021"

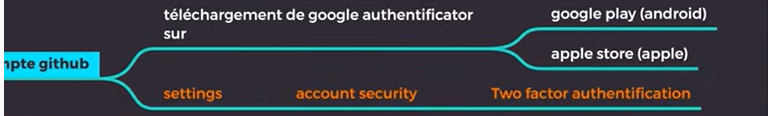
git log //avoir les nombre de commit que j’ai fait

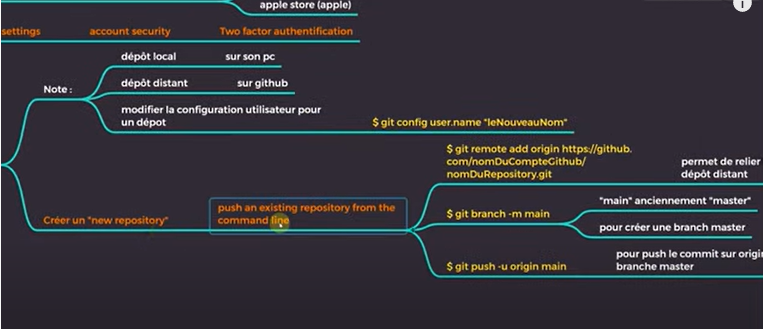


**git –help**

# Mettre son programme sur GitHub-Pousser son premier COMMIT







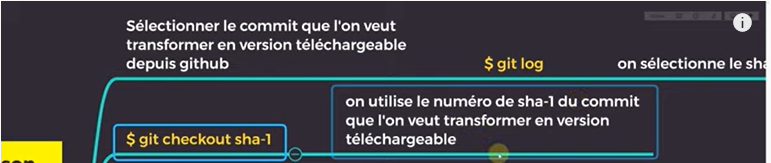
* git remote add origin <https://github.com/SirineSabbeh/Base-Git.git>
* git remote –v
* git branch -M main
* git branch -M master
* git push -u origin main/(master)
* git remote set-url origin https://github.com/SirineSabbeh/testgit.git



* **git fetch**
* **git remote show origin**

# Créer une version téléchargeable de son programme sur github-une release

* **git log**
* **git checkout 8dbedeabb718175c24088bfbf6ab457133d987f9**



* **git tag version0.1 -m "version sans formation react js"**



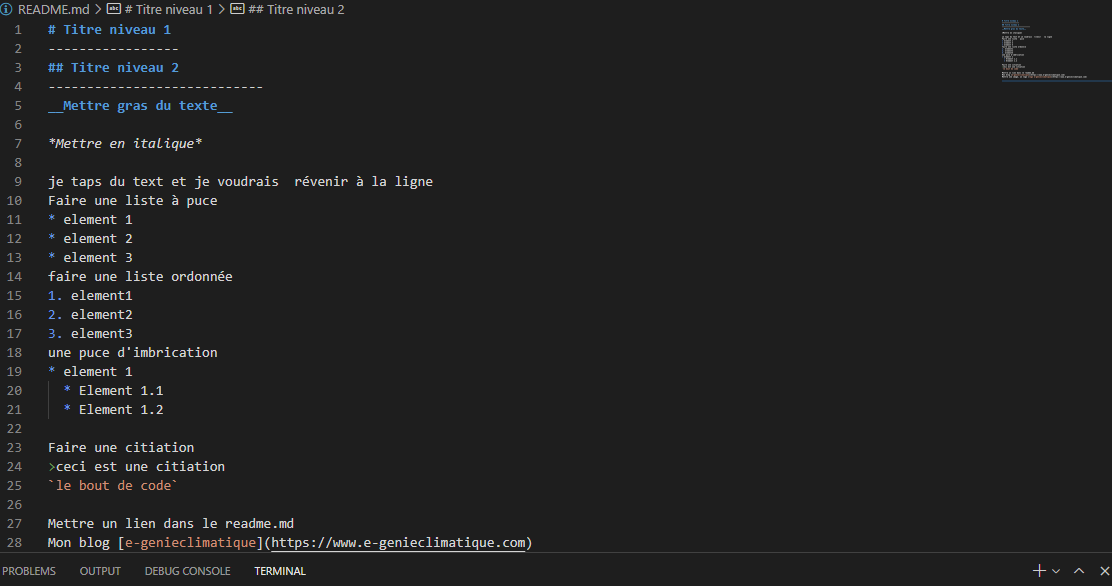
* **git push –tags**

Revenir à la dernière version du programme qui a été commit avec git reset --hard

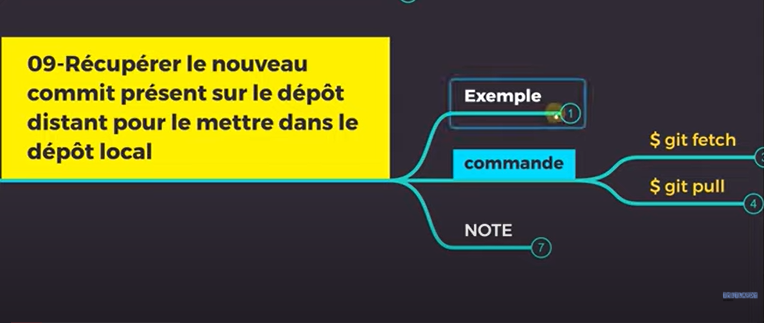
* **Git reset –hard**



Mettre un README.md sur son dépôt distant et faire la mise en page avec Markdown



Git08-Transformer un répertoire GitHub en serveur web avec Github Pages pour l'affichage en page web

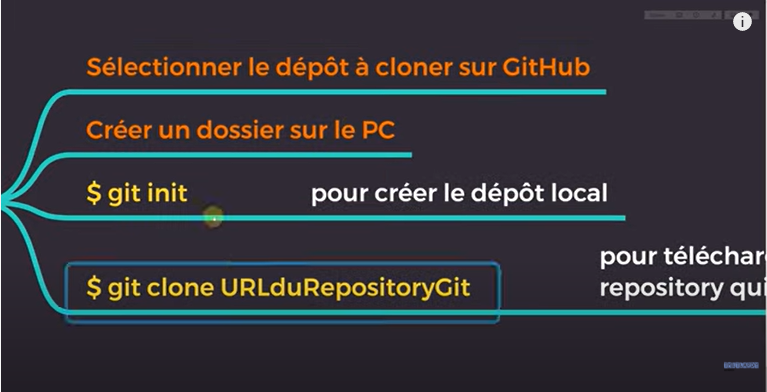


**Git fetch :** **Téléchargement d’objets et de références depuis un autre dépôt**

**Git pull - Récupération et intégration avec un autre dépôt ou une branche locale**

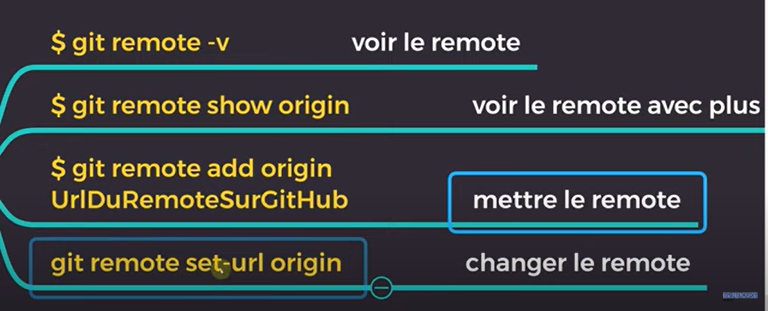
Lors de la comparaison de Git pull vs fetch, Git fetch est une alternative plus sûre car il récupère tous les commits de votre distant mais n’apporte aucune modification à vos fichiers locaux. ... Git fetch est un peu différent ; vous pouvez utiliser la commande Git fetch pour voir toutes les modifications de la télécommande sans les appliquer.

Cloner un dépôt distant sur GitHub pour le mettre dans le dépôt local du folder dans son pc



**git clone** [**https://github.com/SirineSabbeh/formationweb2021.git**](https://github.com/SirineSabbeh/formationweb2021.git)

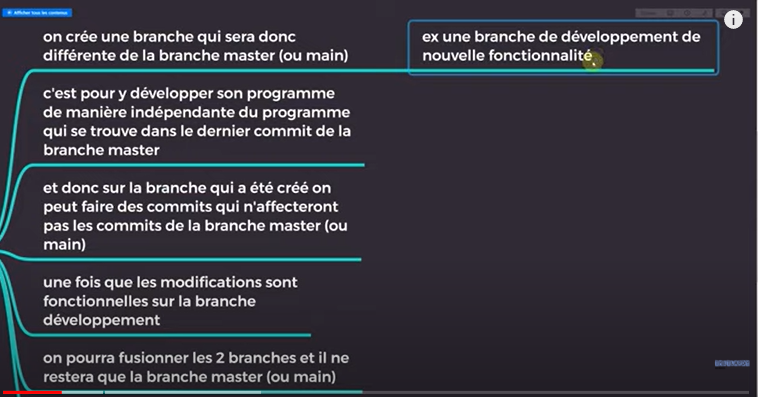
Changer l'url des remotes parce qu'ils ne sont pas bon par exemple suite à un git clone

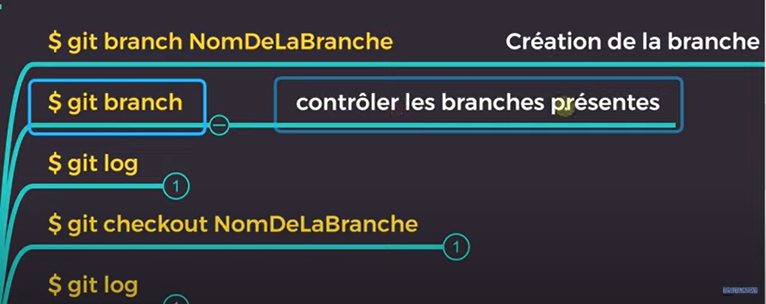


Copier un repository d'un autre compte GitHub sur son Github en faisant un FORK

**Fork command**

Créer une branche pour y mettre des commits qui ne modifiera pas le dernier commit de master









**$ git status**

**$ git branch dev**

**$ git branch**

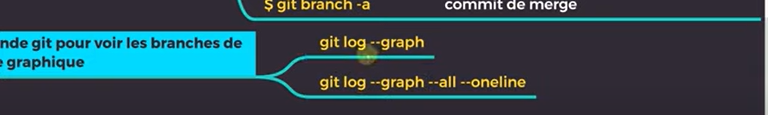
**$git push --set-upstream origin dev**

**$git push**

Fusionner les deux branches en faisant un merge-la branche master et la branche développement







# Git16-Supprimer une branche de son dépôt local et de son dépôt distant



**$git branch**

**$git branch -d dev**

**$git branch**

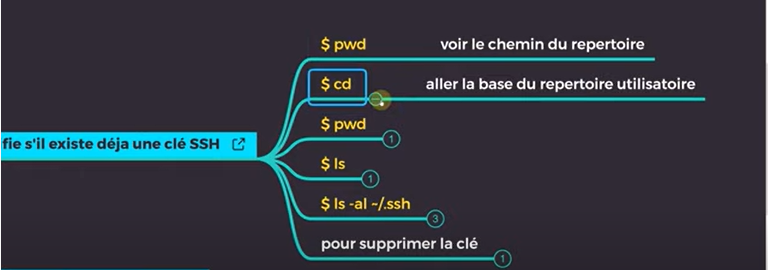
**$git branch -a**

**$ git push origin --delete dev**

**$git branch -a**

# Se connecter à Github par le biais d'une clé SSH-plus besoin du windows credential manager

# 

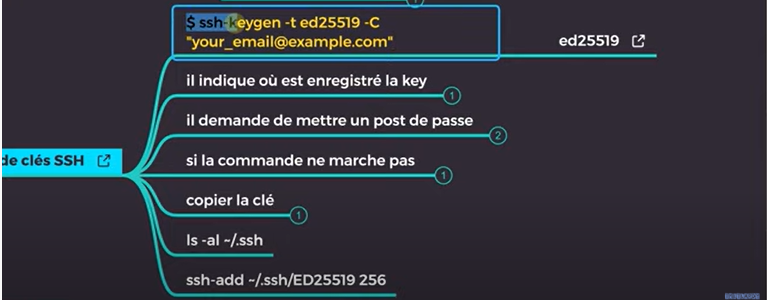


**$ pwd**

**$ cd**

**$pwd**

**$ls**



# J’ai oublié un fichier dans dernier commit

git commit --amend -m "Votre nouveau message de commit"

git add FichierOublie.txt git commit --amend --no-edit

# git reset --merge

# Git20-Raccourci pour faire un commit de manière plus rapide (git add et git commit en même temps)

**$ git commit -am "raccourci commit add"**

# Git21-Modifier le commentaire du dernier commit (suite à une erreur) s'il n'a pas été push

**$ git log -2**

**$git log -2 –stat**

**$git commit –amend -m”commentaire ”**

# Git22-Modifier ou ajouter un fichier sans refaire un nouveau commit dans le dernier commit

**$git commit –amend -m”commentaire ” avant push**

**$git log -3**

**$git revert HEAD^**