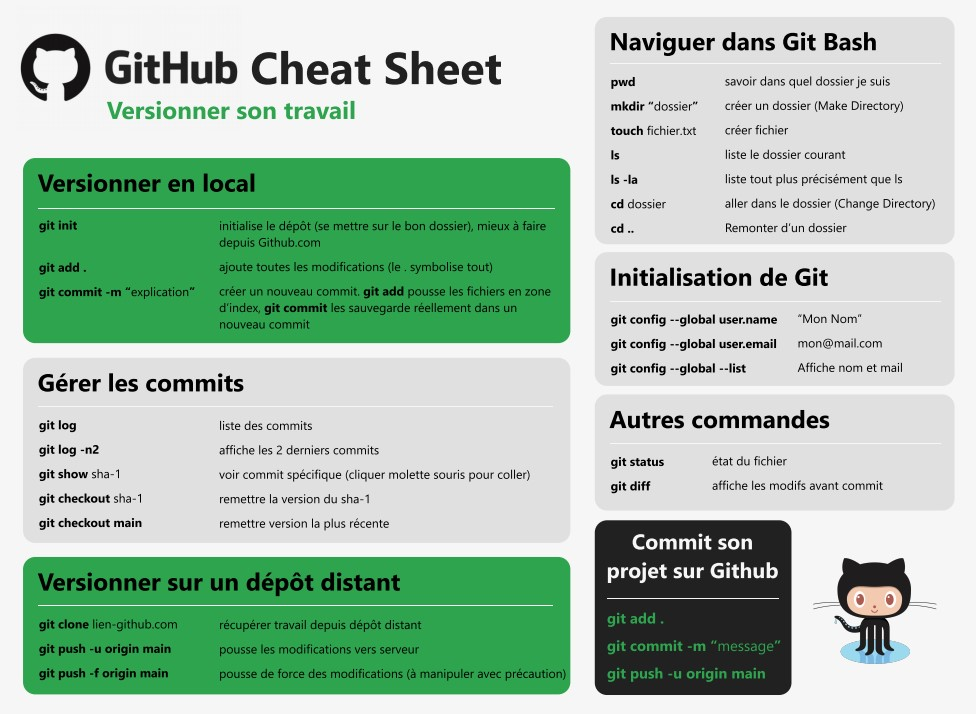
**GitHub**

**GitHub** permet de créer des versions de son travail et de l’envoyer dans le cloud de GitHub où l’on peut partager les avancées du projet.

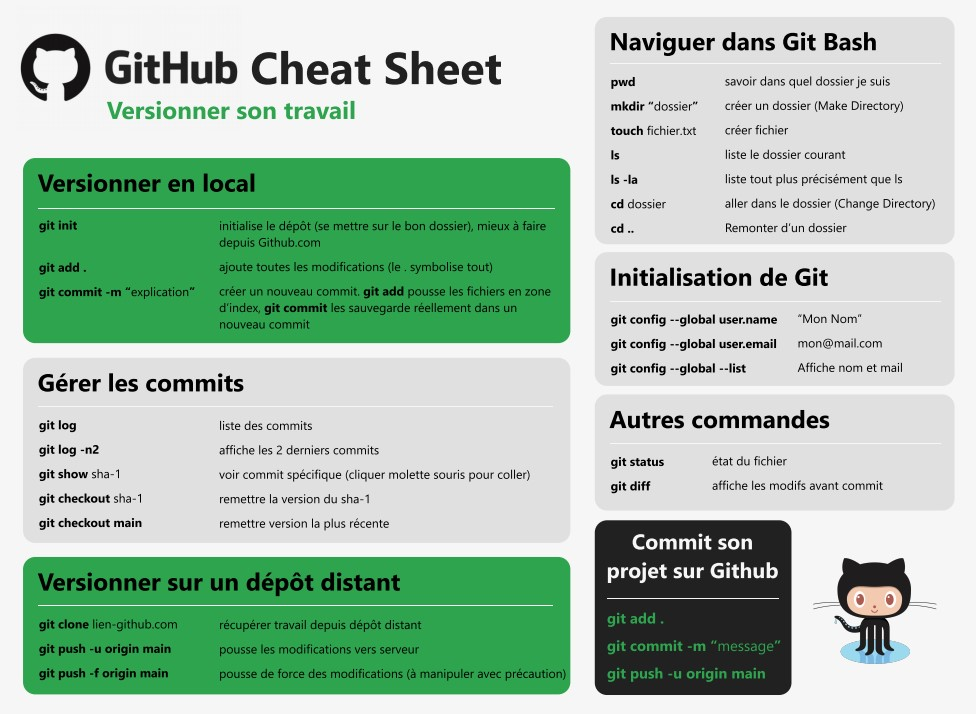
**Initié le Git**, configurer le git pour se connecter lors des envois.

*git config --global user.name <Nom>* : permet d’ajouter le nom du compte.

*git config --global user.email <email>* : permet d’ajouter une adresse mail du compte.

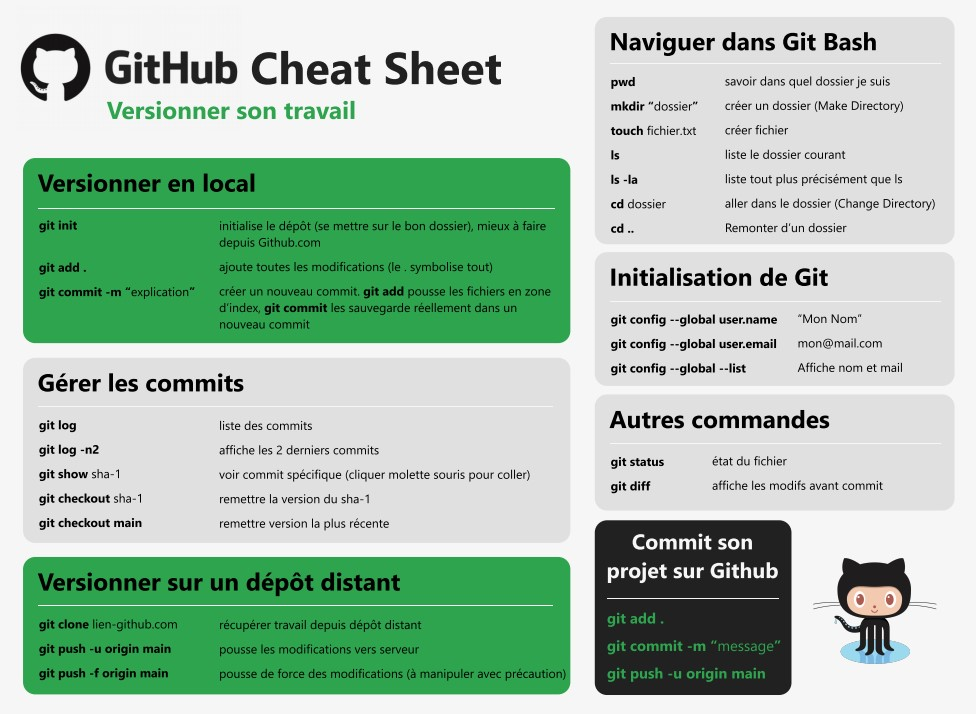
*git config --global --list* : affiche le nom & le mail.

**Créer des versions en local**, pour des sauvegardes de version en offline.

*******git init* : prépare le dossier pour futur dépôt.

git add . : ajoute toutes les modifications et mise en tampon ( le . symbolise l’entièreté du dossier) .

*git commit -m ‘’explication‘’* : créer un commit (une version) avec le **git add** et **git commit** les sauvegarde en local avec le commentaire.

**Créer des versions en ligne**, pour partager et sécurisé les données.

*echo "# Git" >> README.md* : Créer un fichier de présentation.

*git init* : prépare le dossier pour futur dépôt / envoie.

*git add README.md* : ajoute le fichier dans le git (ne fonctionne que si le fichier existe et que **git init** a été fait)

*git add .* : ajoute toutes les modifications (pré-sauvegarde) ( le . symbolise l’entièreté du dossier) .

*git commit -m ‘’explication‘’* : créer un commit (une version) depuis le **git add** et **git commit** confirme la sauvegarde en local avec son commentaire. (-m signifie qu’il y a un message)

*git remote add origin <lien du projet git>* : ajoute la destination d’envoie dans le GitHub

*git push -u origin main* : envoie les sauvegardes vers GitHub.

*git pull* : permet de récupérer la version récente depuis le dépôt GitHub

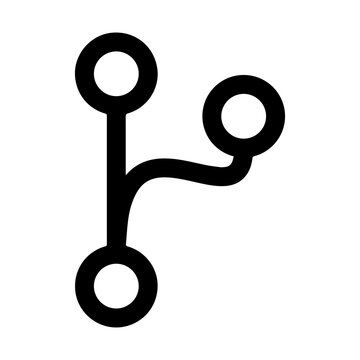
**Les outils de git ou terminal**, libre comme l’air utiliser afin d’utiliser comme on le souhaite.

*git clone <lien git>* : clone le dossier à partir lien GitHub.

*git status* : permet de connaitre les fichiers dans le **git add** (pré-sauvegarde)

*git diff* : permet de voir les modifications avant commit

*git rm --cached <le fichier>* : enlever des fichiers du **git add** (pré-sauvegarde)



**Utiliser des branches**, cela permet faire des développements en parallèle.

*git checkout <nom>* : permet de basculer vers une branche du nom spécifié

/ !\ noter que avec les options

-b cela permet de créer et basculer vers une nouvelle branche

--list donne la liste des branches

-R tout contenu dans le répertoire

<SHA-1> récupère la version antérieure d’un commit

main récupère la plus version récente

Un fichier .gitignore permet de ne pas prendre en compte les fichiers lors d’une pré-sauvegarde.

En conclusion, **GitHub** est une plateforme puissante et largement utilisée pour la gestion de projets de développement de logiciels, que ce soit pour des projets open source ou des projets privés.

Elle facilite la collaboration, la gestion de versions, la publication de documentation, et offre de nombreuses fonctionnalités pour simplifier le processus de développement de logiciels en équipe.