# Aide-mémoire Git

## Git est le système de contrôle de version distribué open source sur votre ordinateur portable ou de bureau.

## Cette aide-mémoire résume les instructions de ligne de commande Git couramment utilisées pour une référence rapide.

### Installer Créer des référentiels

GitHub [pour Windows](https://desktop.github.com/)

## https://windows.github.com

[GitHub pour Mac](https://desktop.github.com/)

## https://mac.github.com

Git [pour toutes les plateformes](https://git-scm.com/)

[http://git­](http://git/)scm.com

Les distributions Git pour les systèmes Linux et POSIX sont disponibles sur le site Web officiel de Git SCM.

Lorsque vous démarrez avec un nouveau référentiel, vous ne devez le faire qu'une seule fois ; soit localement, puis push vers GitHub, soit en clonant un référentiel existant.

#### $ git init

Transformez un répertoire existant en référentiel git

#### $ git clone [url]

Cloner (télécharger) un référentiel qui existe déjà sur GitHub, y compris tous les fichiers, branches et commits

### Configurer les outils

Configurer les informations utilisateur pour tous les référentiels locaux

#### $ git config ­­global user.name "[nom]"

Définit le nom que vous souhaitez attacher à vos transactions de validation

### Le fichier .gitignore

Parfois, il peut être judicieux d’exclure les fichiers du suivi avec Git. Cela se fait généralement dans un fichier spécial nommé

#### .gitignore . Vous pouvez trouver des modèles utiles pour .gitignore

fichiers sur github.com/github/gitignore.

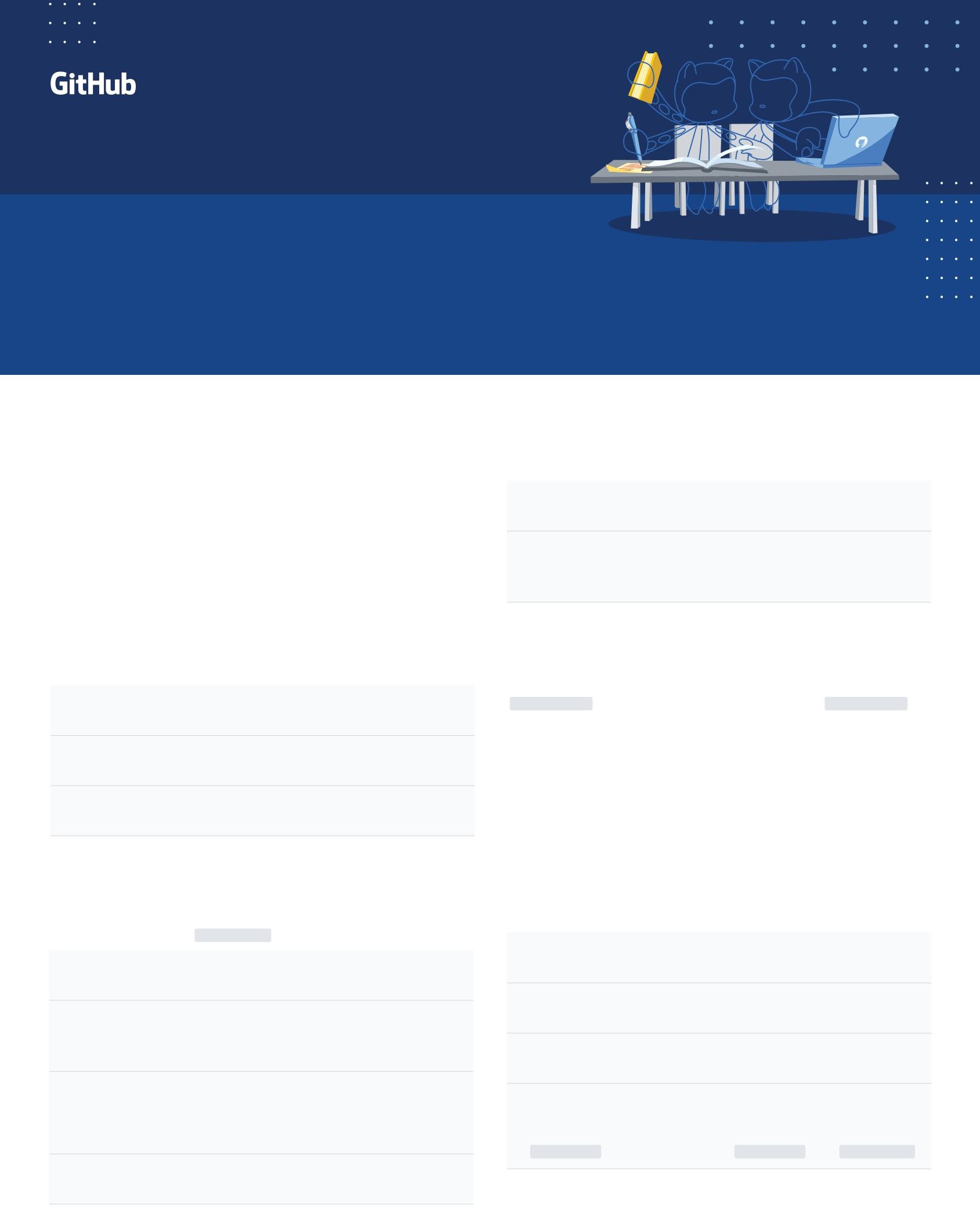
#### $ git config ­­global user.email "[adresse e­mail]"

Définit l'e­mail que vous souhaitez joindre à vos transactions de validation

#### $ git config ­­global color.ui auto

Permet une colorisation utile de la sortie de la ligne de commande

### Branches Synchroniser les modifications

Les branches sont une partie importante du travail avec Git. Tous les commits

que vous effectuez seront effectués sur la branche sur laquelle vous êtes actuellement

« extrait ». Utilisez git status pour voir de quelle branche il s’agit.

#### $ branche git [nom­branche]

Crée une nouvelle branche

#### $ git checkout [nom de la branche]

Bascule vers la branche spécifiée et met à jour le répertoire de travail

#### $ git merge [branche]

Combine l'historique de la branche spécifiée dans la branche

actuelle. Cela se fait généralement dans le cadre de requêtes pull, mais il s’agit d’une opération Git importante.

#### $ git branch ­d [nom­branche]

Supprime la branche spécifiée

Synchronisez votre référentiel local avec le référentiel distant sur GitHub.com

#### $ git fetch

Télécharge tout l'historique des succursales de suivi à distance

#### $ git merge

Combine la branche de suivi à distance dans la branche locale actuelle

#### $ git push

Télécharge tous les commits de branche locale sur GitHub

#### $ git pull

Met à jour votre branche de travail locale actuelle avec tous les nouveaux commits de la branche distante correspondante sur GitHub. git pull est une

combinaison de git fetch et git merge

Aide­mémoire Git

Faire des changements

Parcourez et inspectez l’évolution des dossiers de projet

### Refaire les commits

Effacer les erreurs et créer l'historique des remplacements

#### $ git log

Répertorie l'historique des versions pour la branche actuelle

#### $ git reset [commit]

Annule tous les commits après [commit], en préservant les modifications localement

#### $ git log ­­follow [fichier]

Répertorie l'historique des versions d'un fichier, y compris les renommages

#### $ git diff [première branche]...[deuxième branche]

Affiche les différences de contenu entre deux branches

#### $ git show [commit]

Produit les métadonnées et les modifications de contenu du commit spécifié

#### $ git reset ­­hard [commit]

Supprime tout l'historique et revient au commit spécifié

PRUDENCE! Changer l’histoire peut avoir des effets secondaires désagréables. Si vous devez modifier les commits qui existent sur GitHub (le distant), procédez avec prudence. Si vous avez besoin d'aide, contactez github.community ou

contactez l'assistance.

#### $ git add [fichier]

Ajoute le fichier dans la zone de gestion de version

#### $ git commit ­m "[message descriptif]"

Enregistre les instantanés de fichiers de manière permanente dans l'historique des versions

### Flux GitHub

branche 'maître'

Créer une branche « fonctionnalité » à partir de « maître »

Fusionner la branche « fonctionnalité » dans « maître »

Valider les modifications

Soumettre une demande de validation par les pairs

Discutez des modifications proposées et prenez davantage de engagements

### Glossaire

git : un système de contrôle de version distribué et open source

GitHub : une plateforme d'hébergement et de collaboration sur les référentiels Git

commit : un

objet Git, un instantané de l'intégralité de votre référentiel compressé dans une branche SHA : un pointeur léger et mobile vers un clone de commit : une version locale d'un référentiel,

incluant tous les commits et branches distant : un référentiel commun sur GitHub que tous les membres

de l'équipe utilisent pour échanger leurs modifications. fork : une copie d'un référentiel sur GitHub appartenant à un autre utilisateur pull request : un endroit pour comparer et discuter des différences introduites sur une

branche avec des critiques, des commentaires, tests intégrés, et plus encore

HEAD : représentant votre répertoire de travail actuel, le pointeur HEAD peut être déplacé vers différentes branches, balises ou commits lors de l'utilisation de git checkout

