

Git Cheatsheet

Configurer Git

Git config permet de configurer les informations utilisateur et les options globales.

- Configurer l'utilisateur globalement :

git config --global user.name « prénom »

git config --global user.email « monemail@gmail.com »

- Voir toutes les configurations globales :

git config --global --list

Initialiser un dépôt

- Git init permet de créer un dépôt Git dans le répertoire courant.

git init

Gérer les fichiers

- Voir l'état des fichiers dans le dépôt :

git status

- Ajouter tous les fichiers dans la zone d'indexation :

git add .

- Enlever un fichier de la zone d'indexation :

git reset monFichier.html

- Envoyer les modifications au dépôt local :

git commit -m « mon message »

- Voir les fichiers modifiés depuis le dernier push :

git diff

Voir/gérer les commits

- Voir les commits effectués :

git log

git log --graph

- Voir tous les commit local et distant

git log --all

- Voir toutes les actions qui ont affecté HEAD :

git reflog

- Récupérer les commits, branches, tags distants sans les appliquer :

git fetch

- Modifier l'historique des commit

Ici 3 représente le nombre de commits situés avant le pointeur HEAD sur lesquels on veut effectuer l'action (! cette commande ouvre une interface vim)

Git rebase -i HEAD~3

- Annuler le dernier commit sans supprimer les modifications

Ici 1 représente le numéro du commit situé avant le pointeur HEAD sur lequel on veut effectuer l'action (ici c'est le commit juste avant head)

Git reset --soft HEAD~1

- Annuler le dernier commit et supprimer les modifications

Git reset --hard HEAD~1

- Créer un commit d'annulation sans supprimer l'historique des commit

Git revert commit_du_sha1

Naviguer entre commits et branches

- Se déplacer sur un autre commit :

git checkout NuméroSHA-1

- Se déplacer sur une branche :

git switch nom_de_ma_branche

Gérer les tags

- Lister tous les tags :

git tag

- Créer un nouveau tag :

git tag nomDuTag -m « mon message »

- Supprimer un tag :

git tag --delete nomDuTag

- Envoyer les tags sur le dépôt distant

Git push --tag

GitHub

Gérer les branches

- Créer une nouvelle branche :

git branch <Nom_de_la_branche>

- Créer une branche et y basculer (checkout est l'ancienne version):

git checkout -b nouvelle_branche

git switch -c nouvelle_branche

- Lister toutes les branches (locales et distantes) :

git branch -a

- Changer de branche

Git switch Nom_de_la_branche

- Pousser une branche locale sur le dépôt distant pour la première fois :

git push -u origin Nom_De_La_Branche

- Envoyer des modification sur une branche distante déjà existante

Git push

- Supprimer une branche locale :

git branch -d Nom_De_La_Branche

- Supprimer une branche distante :

git push origin --delete Nom_De_La_Branche

- Supprimer une branche local et distante en une ligne

git branch -d Nom_De_La_Branche && git push origin --delete Nom_De_La_Branche

Fusions, Rebase et Cherry-pick

- Fusionner une branche dans la branche actuelle :

git merge Nom_De_La_Branche_A_Merge

- Rebase d'une branche sur une autre (réécriture de l'historique) :

git rebase nom_branche_sur_laquelle_rebase

- Appliquer un commit spécifiques d'une autre branche :

git cherry-pick commit_hash

- Fusionner des branches qui proviennent de repositories différents :

git merge depot-B/master --allow-unrelated-histories

Gérer les stash

- Mettre de côté des modifications non commités :

git stash push -m « mon message »

- Appliquer le dernier stash sans le supprimer :

git stash apply

- Appliquer un stash spécifique sans le supprimer :

git stash apply stash@{1}

- Afficher les modifications d'un stash

Git stash show stash@{2}

- Supprimer un stash spécifique :

git stash drop stash@{0}

- Supprimer tous les stashes :

git stash clear

- Appliquer et supprimer le dernier stash

Git stash pop

- Voir la liste des stashes :

git stash list

Collaborer avec GitHub

- Voir le dépôt distant associé :

git remote -v

- Ajouter un dépôt distant :

git remote add origin UrlRepo

- Supprimer un dépôt distant :

git remote remove origin

- Cloner un dépôt distant :

git clone UrlRepo

- Récupérer les éléments d'un dépôt cloné

git pull

Raccourcis Vim

1. Modes de base dans Vim

Esc : Quitter le mode insertion et revenir en mode normal.

i : Passer en mode insertion à l'endroit où se trouve le curseur.

I : Passer en mode insertion au début de la ligne actuelle.

a : Passer en mode insertion après le curseur (ajouter après).

A : Passer en mode insertion à la fin de la ligne actuelle.

v : Passer en mode visual pour sélectionner du texte.

V : Passer en mode visual line pour sélectionner des lignes entières.

R : Passer en mode remplacement pour écraser du texte.

2. Navigation dans Vim

Déplacer le curseur (gauche, bas, haut, droite).

gg : Aller au début du fichier.

G : Aller à la fin du fichier.

0 : Aller au début de la ligne.

\$: Aller à la fin de la ligne.

w : Aller au mot suivant.

b : Aller au mot précédent.

Ctrl + f : Faire défiler vers le bas d'une page.

Ctrl + b : Faire défiler vers le haut d'une page.

H : Aller en haut de l'écran (haut de la fenêtre).

M : Aller au milieu de l'écran.

L : Aller en bas de l'écran (bas de la fenêtre).

3. Manipulation des lignes

dd : couper la ligne actuelle.

yy ou **Y** : Copier la ligne actuelle.

p : Coller après le curseur (ou après la ligne courante pour une ligne entière).

P : Coller avant le curseur (ou avant la ligne courante pour une ligne entière).

u : Annuler la dernière action.

Ctrl + r : Rétablir une action annulée.

x : Supprimer le caractère sous le curseur.

X : Supprimer le caractère avant le curseur.

4. Recherche et remplacement

/mot : Rechercher un mot dans le fichier.

n : Aller à la prochaine occurrence du mot recherché.

N : Aller à la précédente occurrence du mot recherché.

:%s/mot1/mot2/g : Remplacer toutes les occurrences de mot1 par mot2 dans tout le fichier.

:%s/mot1/mot2/gc : Remplacer toutes les occurrences avec confirmation.

5. Sauvegarder et quitter

:w : Sauvegarder le fichier.

:q : Quitter Vim.

:wq ou **ZZ** : Sauvegarder et quitter.

:q! : Quitter sans sauvegarder.

:e nomdefichier : Ouvrir un fichier spécifique.

:x : Quitter et sauvegarder (équivalent à :wq).

6. Sélection et manipulation du texte

v : Commencer la sélection en mode visual.

V : Sélectionner la ligne entière.

Ctrl + v : Passer en mode visual block (sélectionner un bloc de texte).

d : Supprimer le texte sélectionné.

y : Copier le texte sélectionné.

c : Supprimer et entrer en mode insertion.

>G : Indenter toutes les lignes jusqu'à la fin du fichier.

<G : Désindenter toutes les lignes jusqu'à la fin du fichier.

7. Manipulation des fenêtres et des onglets

:sp ou **:split** : Diviser la fenêtre en deux (horizontalement).

:vsp ou **:vsplit** : Diviser la fenêtre en deux (verticalement).

Ctrl + w puis **w** : Naviguer entre les fenêtres ouvertes.

:tabnew : Ouvrir un nouveau fichier dans un nouvel onglet.

gt : Aller à l'onglet suivant.

gT : Aller à l'onglet précédent.

8. Commandes pour le rebase et Git

dd : Couper une ligne (utilisé dans Vim pour supprimer une ligne dans un rebase interactif).

p : Coller la ligne coupée après la ligne courante.

P : Coller la ligne avant la ligne courante.

:wq : Sauvegarder et quitter après avoir réorganisé les commits dans un rebase.