Rebase

`git rebase master`
se place au dernier commit de la branche `master` et ré-applique les commits de la branche courante

`git rebase -i HEAD~3`
liste les 3 derniers commits avant de les ré-appliquer, permettant de les modifier

`git rebase --abort`
annule le rebase en cours

Tag

`git tag`
liste les tags du dépôt local

`git tag <new tag>`
crée un tag sur le dernier commit

Bisect

`git bisect start`
démarre une recherche dichotomique pour trouver l'introduction d'un bug

`git bisect bad`
indique que le dernier commit est mauvais

`git bisect good <commit>`
indique qu'un commit est bon

`git bisect log`
affiche le résultat de la recherche

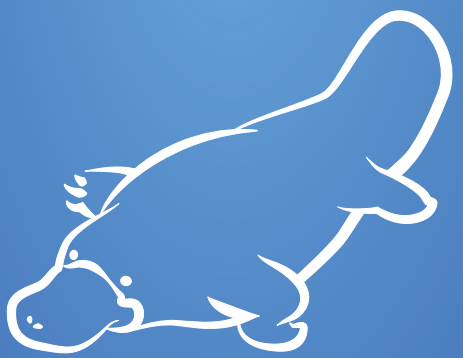
`git bisect view`
affiche les commits suspectés d'avoir introduit le bug

`git bisect reset`
retourne à l'état de départ

Cette cheat sheet a été imprimée grâce au soutien de

gandi.net

git
cheat sheet



CLERMONT'ECH

Association de développeurs auvergnats

Création

`git init`
initialise un dépôt local

Information

`git status`
affiche l'état de l'espace de travail et de la zone indexée

`git diff [<file>]`
affiche les différences entre l'espace de travail et la zone indexée

`git diff --cached [<file>]`
affiche la différence entre la zone indexée et le dépôt local

`git diff <commit> [<file>]`
affiche la différence entre l'espace de travail et le commit

`git log -p [<file>]`
affiche l'historique avec les diff

`git blame <fichier>`
affiche les dernières modifications et auteurs ligne par ligne

Branche

`git checkout <branch>`
bascule l'espace de travail sur une branche

`git checkout -b <new branch>`
crée et bascule sur la nouvelle branche

`git branch -d <old branch>`
supprime l'ancienne branche, `-r` pour supprimer sur le serveur en même temps

`git merge <branch>`
fusionne la branche `<branch>` dans la branche courante

`git merge --no-ff <branch>`
fusionne en forçant la création d'un commit de fusion

Remisage

`git stash save <message>`
remise les modifications de l'espace de travail et fait un `reset --hard`

`git stash pop`
dépile les modifications de la remise dans l'espace de travail

`git stash list`
liste le contenu de la remise

Zone indexée

`git add <file>`
ajoute un fichier à la zone indexée

`git add -p`
ajoute de manière intéractive les modifications dans la zone indexée

`git reset <file>`
enlève le fichier de la zone indexée

Commit

`git commit`
valide la zone indexée dans le dépôt local

`git commit -p`
valide de manière intéractive les modifications de l'espace de travail

`git reset --hard`
remet l'espace de travail ainsi que la zone indexée dans l'état du dépôt et donc efface les modifications non-validées

`git revert <commit>`
défait le commit et réalise un nouveau commit du résultat

`git cherry-pick <commit id>`
applique un commit sur une autre branche

Dépôt distant

`git remote add <remote> <url>`
ajoute un dépôt distant nommé <remote>

`git fetch <remote>`
synchronise un dépôt distant et ses références locales

`git pull <remote> <branch>`
récupère une branche du dépôt distant et la fusionne dans la branche courante

`git pull --rebase <remote> <branch>`
idem mais en rebasant la branche courante au lieu de fusionner

`git push <remote> <branch>`
envoie les commits de la branche sur le dépôt distant

`git push <remote> <local-branch>:<remote-branch>`
envoie `<local-branch>` locale vers la branche `<remote-branch>` de `<remote>`

`git push <remote>:<old branch>`
supprime la branche du dépôt distant

`git push --tags`
envoie les tags vers le dépôt distant