Git basics commands Cheat Sheet

by [deleted] via cheatography.com/26084/cs/7195/

Workflow

La règle d'or est de ne jamais travailler sur master et toujours la garder fonctionnelle.
Implémentation d'une nouvelle feature:
1) Créer une nouvelle branche
$ git checkout -b <branchName>
2) Une fois le travail terminé
$ git add <file(s)> && git commit -m 'message explicite'
3) Récupérer les changements sur master depuis le début de ma feature
$ git checkout master && git pull
4) Ajouter les changements à ma branche
$ git checkout <branchName> && git rebase -i master
Profitez de cette étape pour mettre un clean dans les commit et ne faire qu'un seul commit structuré. (option 'squash' dans l'editeur qui rebase)
5) TESTER QUE TOUT FONCTIONNE
6) RETESTER QUE TOUT FONCTIONNE
7) Repasser sur master, vérifier qu'aucun changement n'a été effectué durant les tests
$ git checkout master && git pull
Si il y a eu des changements répéter les étapes 4, 5, 6 et 7 sinon continuer a l'étape suivante
8) Ajouter ma feature sur master
$ git merge <branchName>
Les changements sont maintenant sur master.
9) Détruire la branche en local et sur le serveur
$ git branch -D <branchName> && git push origin :<branchName>

Merge / Rebase

Fusionne <branch> avec la branche courante

$ git merge <branch>

Rebase la branche courante avec <branch>

$ git rebase <branch>

Rebase en mode interactif

$ git rebase -i <branch>

Annuler un rebase en cours

$ git rebase --abort

Ajoute les fichiers pour continuer le rebase (après avoir réglé les conflits)

$ git add <file>

Valider le rebase après avoir résolu les confits et add les fichiers

$ git rebase --continue

Tools

Atom avec les plugins:
Merge conflict
git control
git log
shell
http://ohmyz.sh/
client graphique git
sourcetree (mac)
gitk (linux)

Save time

```
git config --global editor.<editorName>
git config --global alias.co checkout
git config --global alias.ci commit
git config --global alias.st status
git config --global alias.lg "log --oneline --
decorate --graph --all"
git config --global alias.u nstage " reset HEAD --
"
git config --global alias.u nc ommit " reset --
soft HEAD^"
Ajouter un .gitignore a la racine de votre dépôt
et ajouter les classiques
*.o
*~
*.txt
le nom du binaire et ce que ne voulez pas rendre
en général.
```

Divers

Statut des fichiers dans le dossier courant

$ git status

Afficher l'historique des commits

$ git log

Afficher les différences depuis le précédent commit

$ git diff ou git diff <name> pour spécifier un fichier

Se déplacer sur un commit en particulier

$ git checkout <SHA_1>

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Branch

Créer une branche
$ git branch <branch_name>
Effacer une branche en local
$ git branch -D <branch_name>
Effacer une branche sur le serveur
$ git push origin :<branch_name>
Afficher la liste des branches
$ git branch
Changer de branche
$ git checkout <branch_name>
Créer et changer de branche
$ git checkout -b <branch_name>

Panic button

Enlever un fichier indexé
$ git reset HEAD <fileName>
Déplacer la HEAD sur le commit précédent
$ git reset --soft HEAD^
Editer le précédent commit
$ git commit --amend
Annuler tous les changement sur un fichier depuis le précédent commit
$ git checkout -- <fileName>
Restaurer l'état du dossier et faire une copie temporaire des changement en cours
$ git stash
Afficher la liste des sauvergardes temporaires
$ git stash list
Récupérer une sauvegarde temporaire
$ git stash drop / git stash drop stash@{2}
Supprimer la liste des sauvegardes temporaires
$ git stash clear

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com