Comment éviter les conflits, cas d'école

Utilisation du pull --rebase

Cas d'utilisation : *J'ai fait des **comits localement** que je n'ai **pas pushés**. Des **commits ont été pushés entre temps** sur le dépôt distant. Je veux donc **rendre mes commits compatibles** avec ce qui a déjà été pushé avant de pouvoir les pusher.*

```
$ git pull --rebase
$ git push
```

Avec `git pull --rebase`, les commits distants sont récupérés et appliqués dans mon historique des commits. S'il n'y a pas de conflit, j'obtiens un nouvel état dans lequel je peux pusher mes commits comme s'ils avaient été faits après les commits du dépôt distant.

Avec `git push`, j'envoie mes commits *rebasés* sur le dépôt distant.

Utilisation du stash

Cas d'utilisation : *J'ai fait **des modifications localement**, mais **pas de commits non pushés** parce que je suis au milieu de quelque chose. Cependant, **je veux faire un pull** pour récupérer les mises à jour faites sur le dépôt distant.*

```
$ git stash
$ git pull
$ git stash pop
```

Avec `git stash`, le dépôt est ramené dans l'état du dernier commit et toutes mes modifications sont stockées sous forme de *patch*.

Avec `git pull`, les derniers commits publiés sur le serveur sont récupérés et appliqués en local.

Avec `git stash pop`, mon *patch* est appliqué au dépôt local mis à jour.

Combinaison de stash et pull --rebase

Cas d'utilisation : *J'ai du **code non-commité** et des **commits non pushés**. Je veux envoyer mes commits mais pas mon code en travaux.*

```
$ git stash
$ git pull --rebase
$ git push
$ git stash pop
```

Que faire en cas de conflit ?

Dans les cas où une de ces opérations entraînerait malgré tout des conflits, il faut les résoudre à la main.

Les conflits sont représentés dans les codes sources par des

```
<<<<<<< version bla
Code de la version bla
======
code de la version bli
>>>>>>> version bli
```

À corriger à la main, à la suite de quoi,

- si vous étiez en train de faire un `pull --rebase`, continuez l'opération de `rebase`:

```
$ git rebase --continue
```

- si vous étiez en train de faire un `stash pop`, continuez votre édition tranquillement :)

From:
http://debian-facile.org/ - **Documentation - Wiki**

Permanent link:
http://debian-facile.org/atelier:git-df:tp-merge

Last update: **17/05/2016 11:18**