Stashing

Stash permet de mettre de côté les modifications pas encore commitées d'un répertoire de travail .

Ce faisant , il permet de quitter une branche sans avoir à la committer pour revenir dessus plus tard.

git stash sans aucun argument est équivalent à git stash push. Un remisage est par défaut listé comme "WIP sur nom-de-branche ...", mais vous pouvez donner un message plus descriptif sur la ligne de commande lorsque vous en créez un. Elle est préférée à git stash save

stash@{0} est le dernier remisage créé, stash@{1} est celui qui le précède

git stash list permet de lister les stash existants

git stash show show permet d'inspecter un stash

git stash pop supprime un stash et l'applique le stash au working directory

git stash apply applique le stash au working directory, mais ne supprime pas le stash

git stash drop supprime un stash

Exemple de use case : interruption du travail en cours

... travail, travail, travail ...

$ git stash

$ édition de la correction en urgence

$ git commit -a -m "Correction en urgence"

$ git stash pop

... poursuite de travail ...

Simple Stash - Exemple - illustration du Use Case interruption du travail en cours.

```
pwd
***** Etape 1 *****
git checkout master
git status
git checkout -b mystashbranch

***** Etape 2 *****
***** Creation d un fichier et mise en remisage du working directory
npp mystash.txt # Ajouter du texte
git status
git stash -u # Crée un stash sans nom en prenant en compte les fichiers non trackés
ls -l # le fichier mystash.txt disparait du working directory car il est remisé
git status


```

```
***** Etape 3 *****
***** Il faut procéder à une correction d urgence
#Se placer dans la branche master
git checkout master
npp fixinproduction.txt #Ajouter du texte
npp README.md #Ajouter du texte
git status

git add .
git commit -m "Quick fix in production"

```

```
***** Etape 4 *****
***** Le fix est terminé
***** Restauration du contenu de l espace de travail
# Se placer dans la branche de travail d'origine (mystashbranch)
git checkout mystashbranch
git stash list
git stash apply # le dernier stash est restauré dans l espace de travail mais n est
pas supprimé
ls -l #le fichier mystash.txt apparait dans le working directory
git status # Le working directory est dans le même état qu'au moment du stash.

```

Lister les fichiers non suivis du stash le plus récent (l'option -u (untracked) affiche les fichiers non suivis)

```
git stash show -u 0
```