

Git Stashing

Simple Example

How to simple path for stashing in Git.

Command Listing

```
pwd
cd projects/starter-web
pwd
git status
clear
ls
mate simple.html
git status
git stash save the changes without commit so we can get them later, It works with the tracked files only
git status
mate simple.html
clear
ls
mate README.md
git status
git commit -am "Quick fix in production to improve copyright notice"
git status
clear
git stash apply get back the changes that we stashed them before
mate simple.html
git commit -am "Done with simple.html updates"
git status
clear
git stash list get list of stashes
git stash drop drops the last stash
clear
```

Stashing Untracked Files / Using Stash Pop

Git does not include untracked (new) files with stash by default, however, we change change that. Also, we cover a new way to apply the most recent stash.

Command Listing

```
pwd
git status
clear
git ls-files list all tracked files \(files that already added\)
mate humans.txt
git status
mate ANewFile.txt
git status
git stash
git status
git stash apply
git stash drop
git stash list
clear
git status
git stash -u stash the untracked file -u
clear
git stash list
mate README.md
git commit -a
```

```
~
git status
clear
git stash pop apply then drop the stash in one command
clear
git status
rm ANewFile.txt
git status
git commit -am "Updates to humans file after stash pop"
git status
```

Managing Multiple Stashes

Git's stash feature can support multiple stashes.

Command Listing

```
pwd
git status
clear
ls
mate simple.html
git stash save "simple changes" make a stash for a change with stash message
clear
mate index.html
git stash save "index changes" make a stash for another change with stash message
mate README.md
git stash save "Readme changes" make a stash for another change with stash message
git stash list
git stash show stash@{1} show a specific stash stash@{1} is the stash reference
clear
git status
git stash list get the list of stashes
git stash apply stash@{1} apply a specific stash
clear
git status
git stash list
git stash drop stash@{1} drop a specific stash
git stash list
clear
git stash list
git stash clear drop all stash
git stash list
clear
```

Stashing into a Branch

We can use stash to move accidental changes into a feature branch.

Command Listing

```
pwd
git status
git stash list
clear
mate
mate simple.html
mate humans.txt
git status
git add index.html
git status
mate new.md
git status
git stash -u stash all changes for tracked and untracked files
```

```
git status
clear
git stash branch newchanges  apply the changes to a new branch (this will create a new branch then checkout to that branch then apply the stashes to this branch
                              and finally drop all the stashes )
git stash list
git status
rm new.md
git add .
git status
git commit
git checkout master
git merge newchanges  Merge with main branch then delete the branch
git branch -d newchanges
git branch
clear
```

Section Cleanup

Each section, we synchronize our changes with GitHub.

Command Listing

```
pwd
git status
git pull origin master
git push origin master
```