Useful commands in git

# Basic commands

**Initiate bare repository (remote)**

*git init --bare name.git*

**Initiate repository (local)**

*git init*

**Create first commit**

*git add \* (or git add <filename1> <filename2> to add only specific files)*

*git commit -m “first\_commit”*

**Push first commit**

*git push --set-upstream origin master*

**Push commit**

*git push*

**Push branch to origin repository**

*git push origin <branch\_name>*

**Pull commit**

*git pull*

**Download changes from remote repository**

*git fetch*

**Download all changes from remote repository**

*git fetch –all*

**Create new branch and checkout (keeps modified files but switch to a new branch)**

*git checkout -b <branchname>*

**Checkout branch**

*git checkout <branchname>*

**Checkout branch, forcing overwrite of local changes**

*git checkout -f <branchname>*

**Reverse local changes and pull:**

*git reset --hard*

*git pull*

**View nice graph of commit history:**

*git log --all --graph --oneline --decorate*

**Tag commit with software version**

*git tag -a v1.4 -m "my version 1.4"*

**List tags with message**

*git tag -n5*

# Advanced commands

**Amend commit (to change message)**

*git commit --amend*

**Checkout single file from other commit**

*git checkout <branch> -- <file>*

**List remotes**

*git remote -v*

**Add origin**

*git remote add origin PATH*

**Remove origin**

*git remote remove origin*

**Change url of origin**

*git remote set-url origin new\_url*

**Compare file with file from other commit**

*git diff <branch> <filename>*

**Delete local branch**

*git branch -d <branchname>*

**Delete remote branch**

*git push --delete <remote\_name> <branch\_name>*

**Commit changes outside of the branch context (can be convenient if working on wrong branch)**

*git stash*

**Apply last stashed commit**

*git stash apply*

**Clean the stash**

*git stash clear*

**List all files in commit**

*git ls-tree --name-only -r <commit-ish>*

**List all changed files in commit**

*git diff-tree --no-commit-id --name-only -r <commit-ish>*

**Merge changes**

*git merge <branch>*

**Continue merging**

*git merge –continue*

**Fix merge manually**

*Open meld and manually move lines*

**Merge specific file from branch B into branch A**

*git checkout A*

*git checkout --patch B file1*

**Incorporate specific file from branch B into branch A**

*git checkout A*

*git checkout B file1*

**Remove specific files or folders from entire branch history on local copy then push (careful, can be an intensive operation)**

*git filter-branch --index-filter "git rm -rf --cached --ignore-unmatch path\_to\_file\_or\_folder" HEAD*

*git push –all –force*

*git push origin --force --tags*

**Remove specific files or folders from entire repository history on local copy then push (careful, can be an intensive operation)**

*git filter-branch --index-filter "git rm -rf --cached --ignore-unmatch path\_to\_file\_or\_folder" --prune-empty --tag-name-filter cat -- --all*

*git push origin --force –all*

*git push origin --force --tags*

**Using the above 2 operations will leave the files or folders in the compressed git repository until they are garbaged. To force this, run these commands on both local and remote repositories**

*git for-each-ref --format='delete %(refname)' refs/original | git update-ref --stdin*

*git reflog expire --expire=now --all*

*git gc --prune=now*

**Move remote repository to a new remote repository**

**In the remote location, do**

*git init –bare new\_repository.git*

**In a local location, do**

*git clone –mirror old\_repository.git*

*cd old\_repository*

*git push –mirror new\_repository.git*