

HOW TO CREATE A REPOSITORY

`$ git init [project name]`

Create a New local Repository

`$ git clone my_url`

Download an existing Repository

MAKING A CHANGE

`$ git add [file name]`

Stages the file, ready for commit

`$ git add`

Stage all changed files, ready for commit

`$ git commit -m "add commit message"`

Commit all staged files to versioned history

`$ git commit -am "add commit message"`

Commit all your tracked files to versioned history

`$ git reset [file name]`

Un-stages file, keeping the file changes

`$ git reset --hard`

Revert everything to the last commit

DISPLAY YOUR REPOSITORY

`$ git status`

Display new or modified files not yet committed

`$ git diff`

Display the changes to files not yet staged

`$ git diff --cached`

Display the changes to files

DISPLAY YOUR REPOSITORY (cont)

`$ git diff HEAD`

Display all staged and unstaged file changes

`$ git diff commit1 commit2`

Display the changes between two commit ids

`$ git blame [file name]`

Display the changed dates and authors for a file

`$ git show [commit]:[file name]`

Display the file changes for a commit id and/or file

`$ git log`

Display full change history

`$ git log -p [file name/directory]`

Display change history for file/directory including diffs

SYNCHRONIZE

`$ git fetch`

Display the latest changes from origin

`$ git pull`

Display the latest changes from origin and merge

`$ git pull --rebase`

Display the latest changes from origin and rebase

`$ git push`

Pushes the local changes to the origin

BRANCHES

`$ git branch`

Display all local branches

`$ git branch -av`

Display all branches, local and remote

`$ git checkout myBranch`

Switch to a branch, myBranch, and update working directory

`$ git branch newBranch`

Create a new branch called newBranch

`$ git branch -d myBranch`

Delete the branch called myBranch

`$ git checkout branchB $ git merge branchA`

Merge branchA into branchB

`$ git tag myTag`

Tag the current commit myTag

IF IN DOUBT - HELP !

`$ git command --help`

