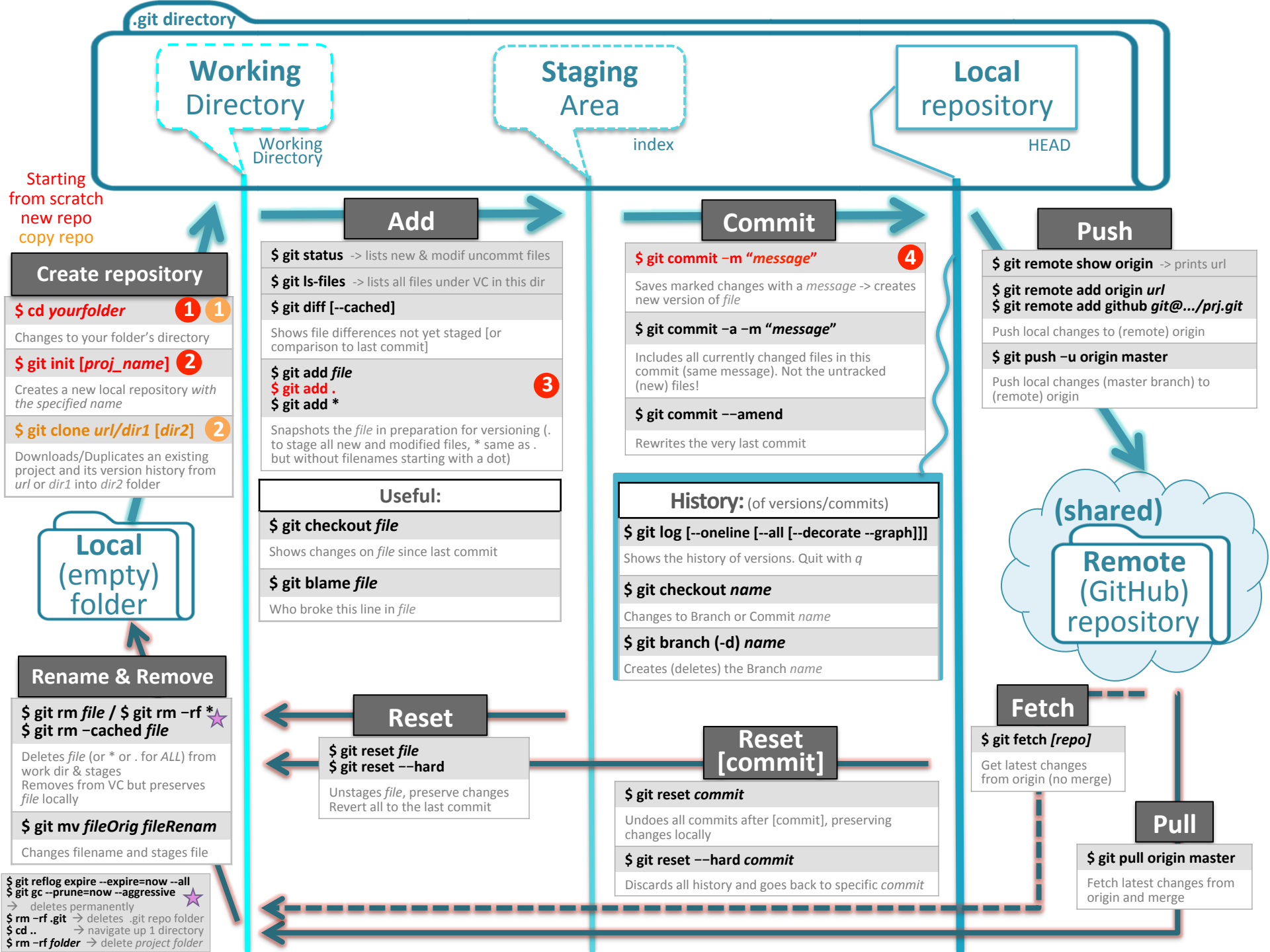
.git directory
Working Directory
Working Directory
Staging Area
index
Local repository
HEAD
Starting from scratch
new repo
copy repo
Create repository
$ cd yourfolder 1 1
Changes to your folder's directory
$ git init [proj_name] 2
Creates a new local repository with the specified name
$ git clone url/dir1 [dir2] 2
Downloads/Duplicates an existing project and its version history from url or dir1 into dir2 folder
Local (empty) folder
Rename & Remove
$ git rm file / $ git rm -rf *
$ git rm -cached file
Deletes file (or * or . for ALL) from work dir & stages
Removes from VC but preserves file locally
$ git mv fileOrig fileRenam
Changes filename and stages file
$ git reflog expire --expire=now --all
$ git gc --prune=now --aggressive
→ deletes permanently
$ rm -rf .git → deletes .git repo folder
$ cd .. → navigate up 1 directory
$ rm -rf folder → delete project folder
Add
$ git status -> lists new & modif uncommt files
$ git ls-files -> lists all files under VC in this dir
$ git diff [--cached]
Shows file differences not yet staged [or comparison to last commit]
$ git add file
$ git add .
$ git add *
3
Snapshots the file in preparation for versioning (. to stage all new and modified files, * same as . but without filenames starting with a dot)
Useful:
$ git checkout file
Shows changes on file since last commit
$ git blame file
Who broke this line in file
Reset
$ git reset file
$ git reset --hard
Unstages file, preserve changes
Revert all to the last commit
Commit
$ git commit -m "message" 4
Saves marked changes with a message -> creates new version of file
$ git commit -a -m "message"
Includes all currently changed files in this commit (same message). Not the untracked (new) files!
$ git commit --amend
Rewrites the very last commit
History: (of versions/commits)
$ git log [--oneline [--all [--decorate --graph]]]
Shows the history of versions. Quit with q
$ git checkout name
Changes to Branch or Commit name
$ git branch (-d) name
Creates (deletes) the Branch name
Reset [commit]
$ git reset commit
Undoes all commits after [commit], preserving changes locally
$ git reset --hard commit
Discards all history and goes back to specific commit
Push
$ git remote show origin -> prints url
$ git remote add origin url
$ git remote add github git@.../prj.git
Push local changes to (remote) origin
$ git push -u origin master
Push local changes (master branch) to (remote) origin
(shared)
Remote (GitHub) repository
Fetch
$ git fetch [repo]
Get latest changes from origin (no merge)
Pull
$ git pull origin master
Fetch latest changes from origin and merge