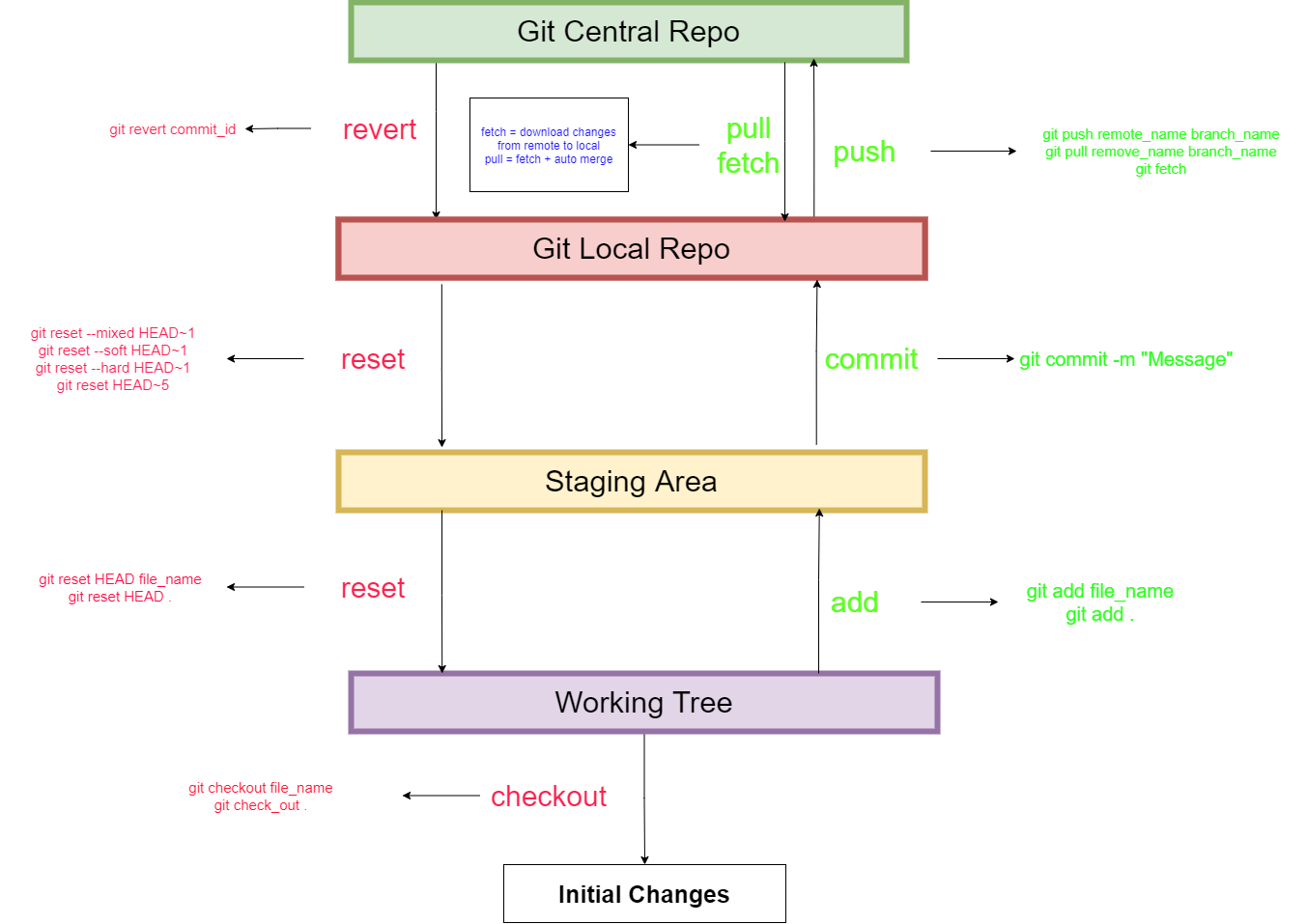
# Git basic structure



# Create a local repository:

$git init

# Create a remote repo from Local repo:

* Login to git hub and create a new repository.
* Copy repo https url.

In local repo git bash type:

$git remote add <remote-name> <remote-url>

$git remote add origin <https://github.com/TSSumanth/cheatsheets.git>

# Pull changes:

$git pull <remote-name> <branch-name>

$git pull origin master

# Push changes:

$git push <remote-name> <branch-name>

$git push origin master

# Clone a remote repo:

$git clone <https-repo-url>

# Undo changes in Working area

$git check\_out <filename\_1> <filename\_2> <filename\_3> .. <filename\_n>

$git check\_out .

$git check\_out \*

# Stage changes

$git add <filename1> <filename2> …<filename\_n>

$git add .

$git add \*

# Unstage changes

$git reset HEAD <File\_Name>

$git reset HEAD .

$git reset HEAD \*

# Commit changes

$git commit –m “message”

# Uncommit changes Local repo:

$git reset HEAD~1

$git reset HEAD~n => last n commits

$git reset --mixed HEAD~n => moves changes from local repo to working area

$git reset --soft HEAD~n => moves changes from local repo to staging area

$git reset --hard HEAD~n => removes changes completely from local repo, staging area, working area

# fetch changes from remote

$git fecth

# pull changes from remote

$git pull remote\_name branch\_name

$git pull origin master

# push changes to remote

$git push remote\_name branch\_name

$git push origin master

# Undo commits in remote

$git revert commit\_id

# get commit details

$git log

# get short commitid

$git log --oneline

# Move head to x commit

$git checkout commit\_id

# List all branchs

$git branch => local branches

$git branch –r => remote branches

# Checkout branch

$git checkout <Branch\_Name>

$git checkout master

# Create Branch

$git branch <branch\_name>

$git branch sprint\_1

To create and checkout branch in single command:

$git checkout –b sprint\_1

# Merge Operations

## fast-farward

when no extra commits are present in current branch.

$git checkout master

$git merge sprint\_1

$git log => new commit will not be created and master will be pointed to Sprint\_1 HEAD

## merge

when master has some commits and sprint\_1 has some commits.

$git checkout master

$git merge sprint\_1

$git log => a new commit is created

# Rebase

if branch sprint\_1 is created from master and in master an important fix is done which need to be present in Sprint\_1 the we use rebase

$git checkout sprint\_1

$git rebase master

$git log => a new commit will be created

# View differences

$git diff => to list all the changes in working tree

$git diff <commit\_id1> <commit\_id2>

$git diff <branch\_1> <branch\_2>

# cherry-pick commits

master -> sprint is created from master.

if a commit is made to sprint and the commit needs to be sent to master as well then we use cherry-pick.

$git checkout master

$git cherry-pick commit\_id

* resolve merge conflicts if any.

$git add .

$git cherry-pick –continue

:wq enter.

# Interactive rebase

used to remove/ alter one or more commits.  
Combine two commits to one -> squashing

remove a commit from history -> drop

change commit message -> reword

$git rebase –I HEAD~n => n is number of commits we will be alterning.