

# GIT Cheat Sheet

## Linux commands

To move to the previous directory	<code>cd ..</code>
This will add an empty file in current working directory.	<code>touch sairam.txt</code>
This will add a new file and insert the file content.	<code>echo Ranjith &gt; sairam.txt</code> <code>echo Name: Ranjith &gt; Rajagopal/Ranjith.txt</code>
This will add a content in the existing file name. If file not exists it will create a new file.	<code>echo Ranjith &gt;&gt; sairam.txt</code>
This will delete the file on the current working directory.	<code>rm sairam.txt</code>
This will rename the file from filename to filename1. If file not found, it will throw an error.	<code>mv sairam.txt ram.txt</code>
This will move the file from one path to another path with same or different name.	<code>mv sairam.txt newpath/sairam.txt</code> (OR) <code>mv sairam.txt newpath/ram.txt</code>
To display all the content in the directory	<code>ls</code>
To display all the content including hidden files in the directory	<code>ls -a</code>
To delete the file or folder	<code>rm -rf .git</code> <code>rm -rf sairam.txt</code> <code>rm -rf Rajagopal/Ranjith.txt</code>

## Git Global Config

Set the user name globally.	<code>git config --global user.name "Ranjith Rajagopal"</code>
Set the user email globally.	<code>git config --global user.email "ranji.r@gmail.com"</code>

## Initialize a repo

Initialize a new repo.	<code>git init</code>
------------------------	-----------------------

## Clone a repo

Clone the repo to local directory.	<code>git clone <a href="https://github.com/Demo.git">https://github.com/Demo.git</a></code>
Clone the repo to the destination path.	<code>git clone <a href="https://github.com/Demo.git">https://github.com/Demo.git</a> C:/temp/test/</code>

# View Status

Full status	<code>git status</code>
Shorthand status	<code>git status -s</code>

# Staging Files

Stages a single file	<code>git add sairam.txt</code>
Stages a multiple file	<code>git add sai.txt ram.txt</code>
Stages with a pattern	<code>git add *.txt</code>
This will stage only the modified and deleted files	<code>git add -u</code>
Stages all the file in the current working directory and child directory and it will ignore any files which start with "." e.g. .gitignore	<code>git add *</code>
Stages all the file in the current working directory and child directory also it will add any files which begins with "."e.g. .gitignore	<code>git add .</code>
Stages all files in the working directory of git	<code>git add -A</code>

# Undo the staging Change

When we add the changes behind the scenes git add a temporary head. This command will remove the temporary head. Note: This is not useful command	<code>git reset --soft</code>
This command will remove the temporary head and move the files to working directory. This command will mark the file as untracked or unmodified.	<code>git status --mixed</code>
This command will permanently remove the files both from staging and working area.	<code>git status --hard</code>
This command will undo the staging changes for one file. Meaning this command will change the file status from staging area to working directory	<code>git reset sairam.txt</code> OR <code>git reset HEAD -- sairam.txt</code>
To clear all the files on staging area	<code>git clean --f</code>

## View log

Full log status	<code>git log</code>
Shorthand log status	<code>git log --oneline</code>
Shorthand log status with last n commits	<code>git log --oneline -5</code>

## Commit the files

This command will commit the files in the current directory. Meaning it will push the file from staging area to git repo	<code>git commit -m "Initial Commit"</code>
This command will commit the files only from staging area. It will skip any files available on working directory	<code>git commit -am "Initial commit"</code>
To revert to the old commit. This command will keep the history and it will add the changes a new commit.	<code>git revert f6hg6t3y</code>
This command will delete all the commits in the git. DON'T USE THIS COMMAND.	<code>git reset --hard f6hg6t3y</code>
You can revert a range of commits using commit hashes	<code>git revert 0d1d7fc..a867b4a</code>
This will revert the last two commits.	<code>git revert HEAD~2..HEAD</code>

## Push

To push the changes to remote	<code>git push origin master</code>
To Push the changes to upstream	<code>git push upstream master</code>

## To view the detailed commit

Shows the given commit	<code>git show 921a2ff</code>
Shows the last commit	<code>git show HEAD</code>
Two steps before the last commit	<code>git show HEAD~2</code>
Shows the version of file.js stored in the last commit	<code>git show HEAD:file.js</code>

## Branch

To create new branch	<code>git branch branchname</code> OR <code>git branch newbranchname oldbranchname</code>
To show all branch	<code>git branch</code>
To checkout to new branch	<code>git checkout branchname</code>
To delete the branch	<code>Git checkout -d branchname</code>

## Merge

To merge the branch with another branch	<code>git merge branchname</code>
---	-----------------------------------

## Display remote details

This will display the git origin details	<code>git remote -v</code>
To display the git version installed	<code>git version</code>