

CONSULTADD INC

PRESENTS

# GIT COMMANDS IN NUTSHELL



Riyaz Ul Haque

Software Engineer | Corporate Trainer

LinkedIn: <https://www.linkedin.com/in/riyazulhaque/>



## CREATE

Clone an existing repository

```
$ git clone https://github.com/test/demorepo.git
```

Initialise an empty repository

```
$ git init
```

## LOCAL CHANGES

Latest changes on files

```
$ git diff
```

Add changes to the staging area of all files

```
$ git add .
```

Add changes to the staging area of a specific file

```
$ git add <file_name.extension>
```

Commit all changes in the tracked file

```
$ git commit -a
```

Commit the changes with commit message

```
$ git commit -m "your message here"
```

Status of changed files after commit

```
$ git status
```

## COMMIT HISTORY

History of all commits

```
$ git log
```

Show changes over a time for a specific file

```
$ git log -p <file_name>
```

Who changed what and when in specific file

```
$ git blame <file_name>
```

## BRANCHING & MERGING

Check all branch

```
$ git branch
```

Create a new branch

```
$ git branch <branch_name>
```

Create a new branch and Switch

```
$ git checkout -b <branch_name>
```

Switch between branches

```
$ git checkout <branch_name>
```

Commit all changes in the tracked file

```
$ git commit -a
```

Delete a local branch

```
$ git branch -d <branch_name>
```

Merge branch into Current HEAD

```
$ git merge <branch_name>
```

## UNDOING CHANGES

Undo all of the changes made in <commit>

```
$ git revert <commit>
```

Remove <file> from the staging area, but leave the WD unchanged.

```
$ git reset <file_name>
```

## CONFIGURATION

Define the author name to be used for all commits by the current user.

```
$ git config --global user.name <name>
```

Define the author email to be used for all commits by the current user.

```
$ git config --global user.email <email>
```

Open the global configuration file in a text editor for manual editing.

```
$ git config --global --edit
```

## UPDATE & PUBLISH

List all recently configured remotes

```
$ git remote -v
```

Add a new remote repository

```
$ git remote add <name_here> <url>
```

Fetch all changes from <remote> but do not integrate with HEAD

```
$ git fetch <remote>
```

Fetch all changes and merge with HEAD

```
$ git pull <remote> <branch>
```

Publish local changes on a remote

```
$ git push <remote> <branch>
```

Rebase your current HEAD onto <branch>

```
$ git rebase <branch>
```

Abort a rebase

```
$ git rebase --abort
```

Continue a rebase

```
$ git rebase --continue
```

## HOW GIT WORKS

- Create a "repository" (consultadd) with a git hosting tool (like GitHub, Bitbucket)
- Clone the repository to your local machineAdd a file to your local repo and "commit" (save) the changes with some message
- "Push" your latest changes to your master branch
- Make a change to your file with a git hosting tool and commit
- "Pull" the latest changes to your local machine
- Create a "branch" (testing), make a change, commit the change
- Open a "pull request" (propose changes to the master branch)
- "Merge" your branch to the master branch