**Github Command**

# To check Changes

**Git status**

#To add new project on github

1. **Make a repro on github and copy url**
2. **Git init**
3. **Git add .**
4. **Git commit –m “dgjklfg”**
5. **Git branch –M main**
6. **Git remote add origin url**
7. **Git push –u origin main**

Or

### …or create a new repository on the command line

**git init**

**git add .**

**git commit -m "first commit"**

**git branch -M main**

**git remote add origin https://github.com/DammarXdezo/hello.git**

**git push -u origin main**

**…or push an existing repository from the command line**

**git remote add origin https://github.com/DammarXdezo/hello.git**

**git branch -M main**

**git push -u origin main**

To Create new branch and work on it

Before work on branch, you should create a branch first

1. **Git branch feature\_branchName //create new branch**
2. **Git branch //check branch**
3. **Git checkout feature\_branchname //go to create branch**

**After create branch you should enter this command**

1. **Git add .**
2. **Git commit –m “dhjkfg”**
3. **Git push origin featch\_branchname**

// steps for work on branch only

If you wand to merge your branch with main branch. You should enter givn command after above comman.

1. **Git fetch**
2. **Git rebase –I origin/master**
3. **Git checkout master**
4. **Git pull**
5. **Git merge feature\_branchname**
6. **Git push origin**

To delete branch locally only

**Git branch –d feature\_branchNmae**

To delete branch Remotely

**Git push origin –delete branchName**

**Change directory**

Cd folderNmae

**Go to back**

Cd ...

**Add git project on your deveice/ pull new project from git**

//git clone paste\_link

**Add new project in git**

//git init

//git add .

//git commit –m “message”

//git branch –M main

//git remote add origin link\_paste

//git push –u origin main

**Create Branch, work on it & push, merge with main branch and pull push**

**Check changes**

// git status

**Check branch**

// git branch

**Check remote Branch**

//git checkout -r

**Create Branch**

// git branch feature\_BranchName Or

* //**git checkout -b branchName**

**Switch branch(local repo)**

// git checkout feature\_BranchName

// git checkout BranchName

**Switch branch(Remotely)**

//git checkout --track origin/my-branch-name

**Push branch work**

//git add .

// git commit –m “message”

// git push origin feature\_branchName

//git fetch

//Git rebase –I origin/master

**Pull From Main, Merge, and Push**

//git checkout master

//git pull

//git merge feature\_branchName

//git push origin

**To delete Branch locally**

//git branch –d feature\_branchName

**To delete branch remotely**

**After checkout main**

//git push origin - - delete nameOfBranch

**Git Clone**

Git clone link

**Git branch**

**Create bew branch** //git branch branchName

**Viewing Branch**//git branch or git branch –list

**Add a single file**: //git add filename

**Add everything at once**: //git add –A

**Git push**: //git push –set upstrim origin branchName

**Merge Branch:** git checkout dev

Git fetch

Git merge branchName

### Getting & Creating Projects

| **Command** | **Description** |
| --- | --- |
| git init | Initialize a local Git repository |
| git clone ssh://git@github.com/[username]/[repository-name].git | Create a local copy of a remote repository |

### Basic Snapshotting

| **Command** | **Description** |
| --- | --- |
| git status | Check status |
| git add [file-name.txt] | Add a file to the staging area |
| git add -A | Add all new and changed files to the staging area |
| git commit -m "[commit message]" | Commit changes |
| git rm -r [file-name.txt] | Remove a file (or folder) |

### Branching & Merging

| **Command** |  | **Description** |
| --- | --- | --- |
| git branch |  | List branches (the asterisk denotes the current branch) |
| git branch -a |  | List all branches (local and remote) |
| git branch [branch name] |  | Create a new branch |
| git branch -d [branch name] |  | Delete a branch |
| git push origin --delete [branch name] |  | Delete a remote branch |
| git checkout -b [branch name] |  | Create a new branch and switch to it |
| git checkout -b [branch name] origin/[branch name] |  | Clone a remote branch and switch to it |
| git branch -m [old branch name] [new branch name] |  | Rename a local branch |
| git checkout [branch name] |  | Switch to a branch |
| git checkout - |  | Switch to the branch last checked out |
| git checkout -- [file-name.txt] |  | Discard changes to a file |
| git merge [branch name] |  | Merge a branch into the active branch |
| git merge [source branch] [target branch] |  | Merge a branch into a target branch |
| git stash |  | Stash changes in a dirty working directory |
| git stash clear |  | Remove all stashed entries |

### Sharing & Updating Projects

| **Command** | **Description** |
| --- | --- |
| git push origin [branch name] | Push a branch to your remote repository |
| git push -u origin [branch name] | Push changes to remote repository (and remember the branch) |
| git push | Push changes to remote repository (remembered branch) |
| git push origin --delete [branch name] | Delete a remote branch |
| git pull | Update local repository to the newest commit |
| git pull origin [branch name] | Pull changes from remote repository |
| git remote add origin ssh://git@github.com/[username]/[repository-name].git | Add a remote repository |
| git remote set-url origin ssh://git@github.com/[username]/[repository-name].git | Set a repository's origin branch to SSH |

### Inspection & Comparison

| **Command** | **Description** |
| --- | --- |
| git log | View changes |
| git log --summary | View changes (detailed) |
| git log --oneline | View changes (briefly) |
| git diff [source branch] [target branch] | Preview changes before merging |

### git config

Usage: git config –global user.name “[name]”

Usage: git config –global user.email “[email address]”

### git reset

Usage: git reset [file] This command unstages the file, but it preserves the file contents.

Usage: git reset [commit] This command undoes all the commits after the specified commit and preserves the changes locally.

Usage: git reset –hard [commit]  This command discards all history and goes back to the specified commit.

### git rm

Usage: git rm [file]

This command deletes the file from your working directory and stages the deletion.

### git log

Usage: git log

This command is used to list the version history for the current branch.

Usage: git log –follow[file]

This command lists version history for a file, including the renaming of files also.

### git show

Usage: git show [commit]

This command shows the metadata and content changes of the specified commit.

### git tag

Usage: git tag [commitID]

This command is used to give tags to the specified commit.

### git branch

Usage: git branch

This command lists all the local branches in the current repository.

Usage: git branch [branch name]

This command creates a new branch.

Usage: git branch -d [branch name]

This command deletes the feature branch.

### git checkout

Usage: git checkout [branch name]

This command is used to switch from one branch to another.

Usage: git checkout -b [branch name]

This command creates a new branch and also switches to it.

### git merge

Usage: git merge [branch name]

This command merges the specified branch’s history into the current branch.

### git remote

Usage: git remote add [variable name] [Remote Server Link]

This command is used to connect your local repository to the remote server.

### git push

Usage: git push [variable name] master

This command sends the committed changes of master branch to your remote repository.

Usage: git push [variable name] [branch]

This command sends the branch commits to your remote repository.

Usage: git push –all [variable name]

This command pushes all branches to your remote repository.

Usage: git push [variable name] :[branch name]

This command deletes a branch on your remote repository.

### git pull

Usage: git pull [Repository Link]

This command fetches and merges changes on the remote server to your working directory.

### git stash

Usage: git stash save

This command temporarily stores all the modified tracked files.

Usage: git stash pop

This command restores the most recently stashed files.

Usage: git stash list

This command lists all stashed changesets.

Usage: git stash drop

This command discards the most recently stashed changeset.

Usage: git stash drop

This command discards the most recently stashed changeset.

**Tell Git who you are**

| **Description** | **Command** |
| --- | --- |
| Configure the author name. | git config --global user.name "<username>" |
| Configure the author email address. | git config --global user.email <email address> |

**Getting & Creating Projects**

| **Description** | **Command** |
| --- | --- |
| Initialize a local Git repository | git init |
| Create a local copy of a remote repository | git clone ssh://git@github.com/<username>/<repository-name>.git |

**Basic Snapshotting**

| **Description** | **Command** |
| --- | --- |
| Check status | git status |
| Add a file to the staging area | git add <file-name.txt> |
| Add all new and changed files to the staging area | git add -A or git add . |
| Commit changes | git commit -m "<commit message>" |
| Remove a file (or folder) | git rm -r <file-name.txt> |

**Inspection & Comparison**

| **Description** | **Command** |
| --- | --- |
| View changes | git log |
| View changes (detailed) | git log --summary |
| View changes in one line (briefly) | git log --oneline or git log --pretty=oneline or git log --pretty=short |

**Undo to previous file**

| **Description** | **Command** |
| --- | --- |
| List of all commit with commit id and commit message) | git log --oneline |
| Return to previous commit | git checkout<commit id> |
| Revert commit (undo one particular commit) | git revert <commit id> |
| Reset to previous commit (remove history of all commit after ) | git reset <commit id> |
| Stop a file being tracked | git rm --cached <file/folder> |
| Restore a file to a previous commit | git checkout <file/to/restore> |

**Branching & Merging**

| **Description** | **Command** |
| --- | --- |
| List branches (the asterisk denotes the current branch) | git branch |
| List all branches (local and remote) | git branch -a |
| Create a new branch | git branch <branch name> |
| Create a new branch and switch to it | git checkout -b <branch name> |
| Clone a remote branch and switch to it | git checkout -b <branch name> origin/<branch name> |
| Rename a local branch | git branch -m <old branch name> <new branch name> |
| Switch to a branch | git checkout <branch name> |
| Switch to the branch last checked out | git checkout - |
| Discard changes to a file | git checkout -- <file-name.txt> |
| Delete a branch | git branch -d <branch name> |
| Delete a remote branch | git push origin --delete <branch name> |
| Preview changes before merging | git diff <source branch> <target branch> |
| Merge a branch into the active branch | git merge <branch name> |
| Merge a branch into a target branch | git merge <source branch> <target branch> |
| Stash changes in a dirty working directory | git stash |
| Remove all stashed entries | git stash clear |

**Sharing & Updating Projects**

| **Description** | **Command** |
| --- | --- |
| Push a branch to your remote repository | git push origin <branch name> |
| Push changes to remote repository (and remember the branch) | git push -u origin <branch name> |
| Push changes to remote repository (remembered branch) | git push |
| Delete a remote branch | git push origin --delete <branch name> |
| Update local repository to the newest commit | git pull |
| Pull changes from remote repository | git pull origin <branch name> |
| Add a remote repository | git remote add origin ssh://git@github.com/<username>/<repository-name>.git |
| Set a repository's origin branch to SSH | git remote set-url origin ssh://git@github.com/<username>/<repository-name>.git |