GIT NOTES

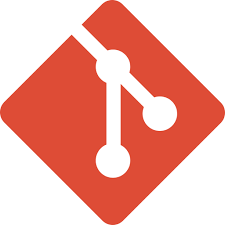


TABLE OF CONTENT

* Table of content -------------------------------------------------------- 2
* Table of Descriptions -------------------------------------------------- 3
* Git Uploading Process ------------------------------------------------- 3
* Git Commands ---------------------------------------------------------- 4
* Git Command-init ------------------------------------------------------ 4
* Git Command-status --------------------------------------------------- 4
* Git Command-add ------------------------------------------------------ 4
* Git Command-commit ------------------------------------------------- 4
* Git Command-config --------------------------------------------------- 5
* Git Command-branch -------------------------------------------------- 5
* Git Command-checkout ----------------------------------------------- 5
* Git Command-switch -------------------------------------------------- 6
* Git Command-marge -------------------------------------------------- 6
* Git Command-diff ------------------------------------------------------ 6
* Git Command-stash ---------------------------------------------------- 7
* Git Command-remote ------------------------------------------------- 7
* Git Command-push ---------------------------------------------------- 7
* Git Command-clone ---------------------------------------------------- 7
* Git Command-pull ------------------------------------------------------ 8
* Shapes Comparison ---------------------------------------------------- 8
* Git Steps for Join GitHub Project ----------------------------------- 9

|  |  |
| --- | --- |
| Word | Description |
| Repo | Repository, it is the directory in git with contains all data, which uploaded from stage |
| Git | Git, it is the software which developer deal with it using commands |
| GitHub | Github, it is the services which using git to show data using GUI |
| Working Directory | The current environment which has all data in the pc or laptop |
| Branch | Branch, it is a copy of the master branch |

GIT UPLAODING PROCESS

* Working Directory
  + git add
* Stage
  + git commit
* Repo
  + git push
* Github or GitLab or etc.

GIT COMMANDS

* git init
  + - To create new git container in your current directory
      * Create default branch named as ‘master’
  + git status
    - To print the status of the current branch, if have any additional files/folders
  + git add
    - git add . or git add \*
      * upload all files/folders from current directory to Stage
    - git add ‘file.txt’
      * upload only this file from current directory to Stage
  + git commit
    - git commit
      * Upload all in Stage to the repo
    - git commit -m “Commit Massage”
    - git commit -am “Commit Massage”
      * Upload all in Work Directory to the repo directly
  + git log
    - To print all commits with it data as commit-key, date, massage, etc.
  + git config – global
    - git config –global user.name “Ahmed”
      * create username for git
    - git config –global user.name
      * print the current username in git
    - git config –global user.email “[ahmed@gmail.com](mailto:ahmed@gmail.com)”
      * create email to git account
    - git config –global user. Email
      * print the current email of git
  + git branch
    - git branch
      * To print all branches names
    - git branch “new branch name”
      * to create new branch only, the current used branch not changes yet
    - git branch -d “branch name”
      * to remove branch
    - git branch -M “newname”
      * to rename the currnet branch
  + git checkout
    - git checkout “exists branch name”
      * To change the current used branch to the “branch name” branch
    - Git checkout -b “not exists branch name”
      * To change the current used branch to the “branch name” branch, not exists, will create new one
  + git switch
    - git switch ‘exists branch name’
      * To change the current used branch to the “branch name” branch
    - git switch -c ‘new branch name’
      * To change the current used branch to the “branch name” branch, not exists, will create new one
  + git marge “name of the other branch”
    - To marge both the current branch with the called branch, put all data in the current branch
    - Types of merging
      * With conflict, if same file with same name with same type have different content, then git will marge both data, but will rise conflict error
      * Without conflict, if both have not file with same name
  + git diff
    - git diff –staged
      * compare the same file which in stage and in repo
        + a: file.txt, file before commit
        + b: file.txt, file after commit
        + --- before commit
        + +++ after commit
        + - in lines, removed lines
        + + in lines, added or modified lines
    - git diff ‘commit-ID’ ‘another-commit-ID’
      * compare same file status at both commits
* git stash
  + - if you are in branch X and edit file.txt then need to leave this branch Y using ‘switch’ or ‘checkout’, git will rise error so you might add the changes or can use ‘stash’ to provide git from raising error and switch to another branch
    - git stash
      * will save the data, till make pop for it
    - git stash pop
      * to remove stash and return data in it place, even if you are in another branch
    - git stash list
      * print list of stashes which created by the user
* git remote
  + - to upload your git repo to github repo
    - create new github repo and copy the https link [FOLLOW](https://www.youtube.com/watch?v=f26KI43FK58&pp=ygUSY3JlYXRlIGdpdGh1YiByZXBv)
    - git remote add origin ‘copied https link’
      * connect between git ang github repos
    - git remote -v
      * print the branches of github repo
* git push
  + - git push origin ‘git branch’
      * to upload your git repo to github repo
* git clone
  + - git clone ‘https link’
    - clone the github repo into your work directory
* git pull
  + - pull the code from github repo into your work directory

|  |  |  |
| --- | --- | --- |
| Shape 1 | Shape 2 | Shape 3 |
| Data in github repo will clone to work directory | **Data in git repo and will connect with github repo** | **Update data from github to work directory** |
| Copy ‘https link’ | Create github repo, copy ‘https link’ | first connect git repo with github repo |
| *git clone ‘https link’* | *git remote origin ‘https link’* | *git pull* |

GITHUB WORKING AT OPEN PROJECTS OR WITH YOUR TEAM

* git clone ‘github https link’
  + - to get clone form the project in your work directory
* make your changes
  + - make your task in the project
* git add \*
  + - to add all files after make changes
* git commit -m “Massage”
  + - to commit project from stage to git repo
* git push origin ‘branch name’
  + - to push data from git repo to github repo
* owner of the project will receive your changes and merge it
* if the project have any changes form each team member and need to get the last version will use
  + - git pull