Kedari Gowri

GITHUB FUNDAMENTALS

Commands of GIT

**GIT COMMANDS**

Repository is the foundational building block in Github. Everything is built around repositories.

**DOWNLOAD GIT**

**TO CONFIGURE GIT:**

git config -- global user.name “GeeksGirls”

git config --global user.email “geeksgirls123@gmail.com”

git config --edit –global

**TO CONFIGURE EDITOR :**

git config core.editor “notepad++ -multiInst –nosession”

**BASIC COMMANDS :**

git config //configure tools

git init // initialize the local repo

git clone // clone a project from remote

git add // prepare a file (to staging area)

git commit // commit changes to local repo

**WORKING WITH GIT :**

Move to working directory:

cd “d:\code\pluralsight”

get init DemoApp //creates empty repo

cd DemoApp // moving to master branch

git status //to check status

create README.md file

git add README.md

git commit –m “first commit”

**TO CHECK WHETHER REMOTE CONFIGURED :**

git remote –v

git remote add origin https://github.com/geeksgirls/firstrepo.git

git push -u origin master

README files are written in MarkDown format. The extension will be .md

**SPECIAL FILES IN GITHUB : (.md)**

1. README
2. LICENSE
3. CONTRIBUTERS AND CONTRIBUTING
4. CHANGELOG
5. SUPPORT
6. CODE\_OF\_CONDUCT

**MARKDOWN FILE SYNTAX:**

Can be converted to HTML easily.

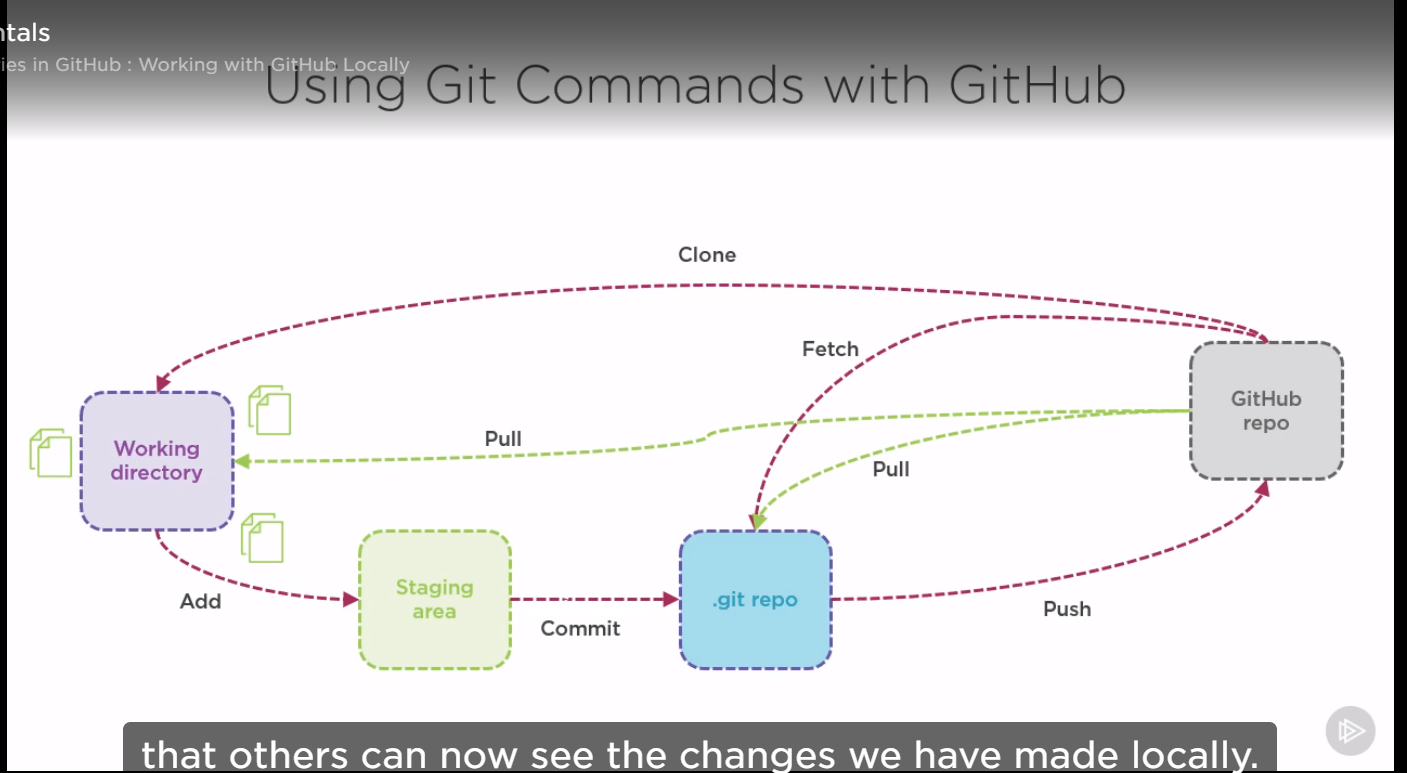
# This is an H!

## This is an H2

\*Red

\*Blue

\*green



Fetch command will bring the changes in remote repo to local repo. However it will not merge.

You have to perform git merge.

Git pull is a combination of fetch+merge.

And then we need to do git push to remote repo.

Git push origin master

**TO GENERATE SSH KEY :**

ssh–keygen –t rsa –b 4096 –C “geeksgirls123@gmail.com”

ssh –T [git@github.com](mailto:git@github.com) // to check the ssh configured properly or not

**IF FEW FILES ARE UPDATED IN REMOTE :**

To bring remote and local repo in sync

git fetch //download the changes made to repo

git pull // will perform both fetch and merge … if it didn’t cause any conflict in changes

**NOW IF BOTH REMOTE AND LOCAL HAS SOME CHANGES :**

notepad++ index.html // changing the content of html file

git add . // to move the files for staging

git commit –m “changes made”

Now if you trying to push the local repo , github will reject the changes. We need to fetch the changes made in remote repo first

git pull // to bring in changes from remote repo

this will open merge notepad … this will perform a merge commit

then perform git push

git push origin master

**ARCHIEVING REPOS :**

If a project is over , then we can move to archieval. This will make the issues as READ ONLY

**CREATING CLONE :**

git clone

**OVERVIEW OF BRANCHES :**

* Lightweight in GIT
* Creates branches faster
* Most recommended to work with several branches for application development
* Team work

It work with Snapshots

Branch is a pointer to the commit

Use Branch for :

Features

Bugs

Experiments

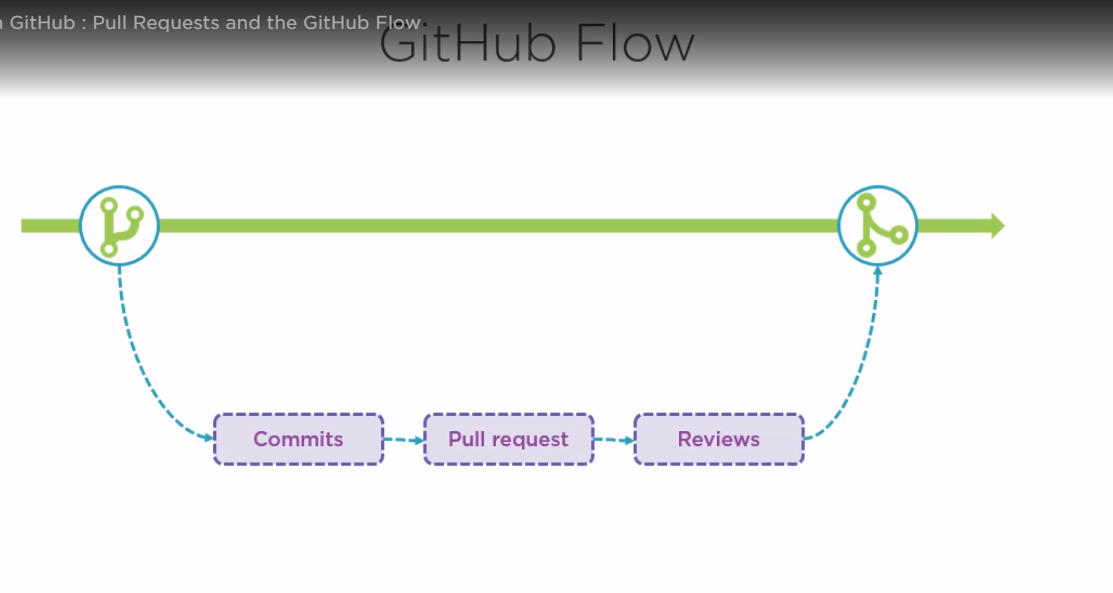
**COMMANDS FOR BRANCHES :**

git branch [branch-name] // create a new branch

git checkout [branch-name] //to move to a particular branch

git push –u [origin] [branch] // to push the branch to github and this creates tracking relationship

git checkout –b “sample-branch” // to create and checkout



**TAGS AND RELEASES**

A branch points to the snapshot in the history of ur project

Tag points to the important points in History

**Types of tags :**

1. Lightweight - acts as pointer to commit
2. Annotated – contains email,date and message

Commands

Git tag

Git –a tag // for annotated tags

Git log //gives a verbose output of the history

Git log --oneline --graph --decorated --all // gives the output in more readable form

Git tag –a v0.1 –m “version 0.1” a6b446e // annotated tag with commit id

Git push --tags // to push tags from local to github

Git tag –d “stable” //to delete a tag locally

Git push origin :v0.2 // to push the changes made to tags … deleted the tag

**RELEASES**

Based on tags

Can add release notes

Can add binary files related to the release

**WORKING WITH FORKS**

Fork is a copy of repository

If you make changes to forks it will not impact the original repo

It will have a link with the original repo. If there are any updates in original repo those changes

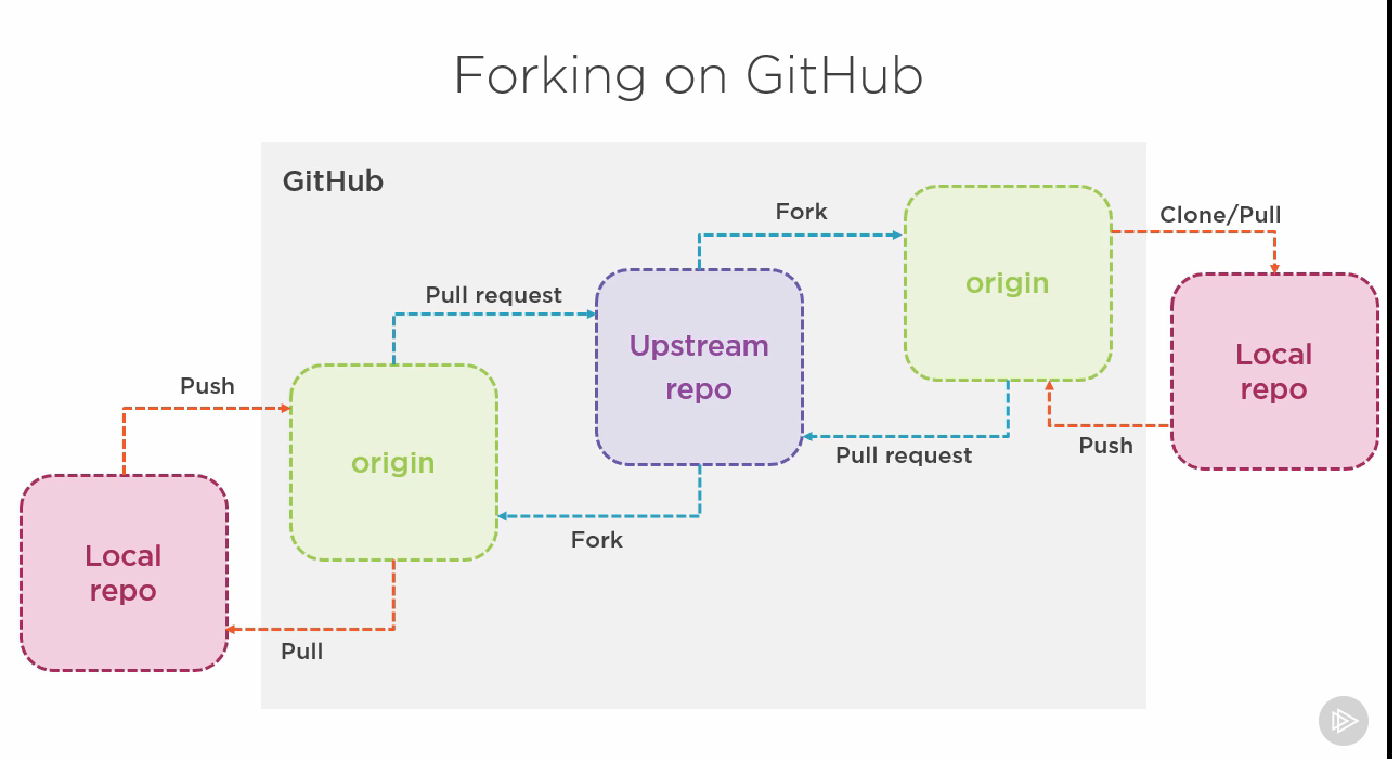
Can be merged again via pull requests

Available to do for anyone

* Once you create a fork that is your own repo now

**Difference between Fork and Branch??**

Branches work on same repo whereas Forks work on different repo



**STEPS:**

Fork a repo from other acct (github interface)

Clone to local machine

Git clone https or ssh address

Git remote –v

**PULL REQUESTS**

Update others about the changes made in the branch or fork

Discuss and review

Optionally merged to the main branch

**GIT GISTS**

Gist.github.com

To share snippets, piece of codes and notes

Gist is also a repo

So we can clone it to the git

Copy ssh link

Git clone “ssh link”

We can make changes to git locally

Git add .

Git commit –m “updated gist file”

Git push origin master

**GITHUB PAGES**

Static site hosting without server side code

Site is effectively a repo

Can create online or offline

**ENABLING NOTIFICATIONS FOR PROJECTS**

**Types:**

Participating notification – when directly involved

Review of pull request, open a pull req

Watching notification – If some changes in repo which u r watching

**MANAGING YOUR PROJECT THRU GITHUB**

1. Working with issues
2. Creating milestone
3. Working with project boards
4. Creating wiki

**Working with issues**

GitHub issues – all the feedback items are collected as issues

Bugs, enhancements, tasks and ideas

Issues :

Can be notified

Can be linked to project board and assign to a team member

Can be associated with pull requests

Combined towards a milestone

**Creating milestone**

Group issues together

Track progress

**Working with project boards**

Organization of work

Work on a specific feature

Checklists

Roadmaps

Project boards can contain **Issues and Pull Requests**

**SETTING UP WIKI**

Documentation is vital

Wikis can be created for your repo

Long living document on how to handle the project

Online or Offline (push)

**We can clone the wiki locally**

**Git clone** [**https://github.com/geeksgirls/Java.wiki.git**](https://github.com/geeksgirls/Java.wiki.git)

**WORKING WITH ORGANIZATION AND TEAM**

**Organization != company**

Shared accts to work on projects

Fine-grained access to manage access

Free or paid

**ORGANIZATION:**

**Can be created from scratch**

**Converting an existing acct**

**Roles within org – Owner and member**

**INTERACTING WITH GITHUB DESKTOP CLIENT**

Desktop client can be installed. Instead of using git cmd.