**DAY-3-AGENDA**

**Git commands**

**Branching**

**Merging**

**Git work flow**

**Forking**

**->Merge conflict**

**Git-lifecycle**

|  |  |  |  |
| --- | --- | --- | --- |
| **Working Directory**  **---------->** | **Staging Area**  **----------->** | **Commit**  **--------->** | **GITHUB** |

**Git commands**

1. **Creating repository**
2. **Modification**
3. **Parllel dev**
4. **Syncing repo.**

**Commands:**

**Working directory:**

**git init - initilizes the git repo, (.git) file is created(hidden)- helps in tracking**

**git status - tells about files and folders, tells if any file/folder is staged or not.(for our better insights)**

**Staging Area**

**git add . - to stage a file(to track a file)**

**git add <name1> : only that file will be staged**

**git commit -m “File updated” - commits(saving) the changes, -m(message flag)**

**git remote add origin “URL” - adds the file to github**

**git push origin master - pushes changes to master branch**

**Username:**

**Pass:**

**-> for copying someones repo**

**git clone <url> : url of his/her repo. It will give u a copy of that repo.**

**->update something**

**git pull origin master - to make changes in current repo.**

**Note:**

**git push origin <name\_branch> - pushes changes to that specific branch**

**General Commands:**

**git log : history- what all git commands you have used**

**Commit id - it is generated every time we perform any operation on our repo. - modification, updation, deletion**

**Unique everytime**

**git revert <commit\_id> : go back to previous version**

**Branching : for parllel dev. - no chnages will be done on other branches, saves time and money, more than one feature at a time could be worked upon..!**

**-> Last commit of master is the first commit of new branch\***

**git branch <name> : create a new branch of name given**

**git branch -D <name> : delete a branch**

**git checkout <name> : to go inside that branch**

**Merging:**

**After the dev. Or the work is done in branches we merge the branches to req. Branch or the master branch.**

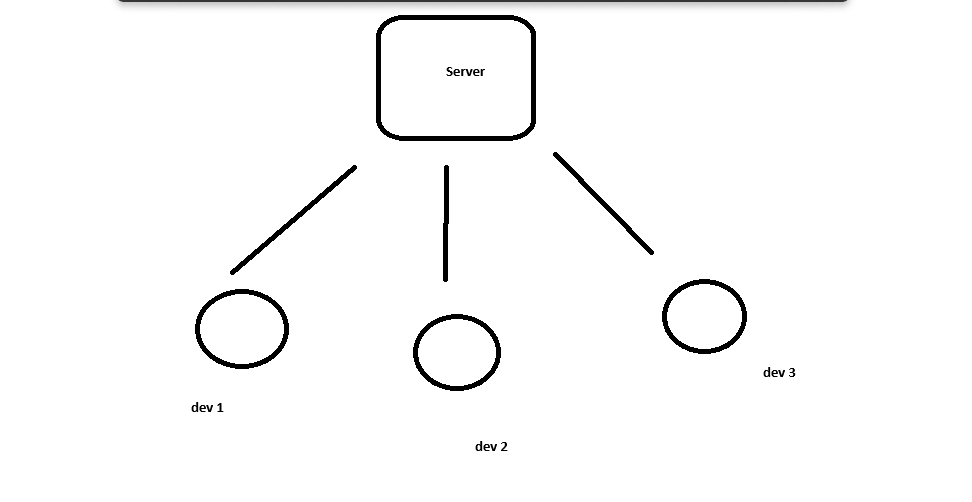
**git merge <source\_branch\_name>**

**Git workflow:**

**It’s a kind of a recommendation on how yo use git in work!**

**Few popular workflow:**

**Centralized:**

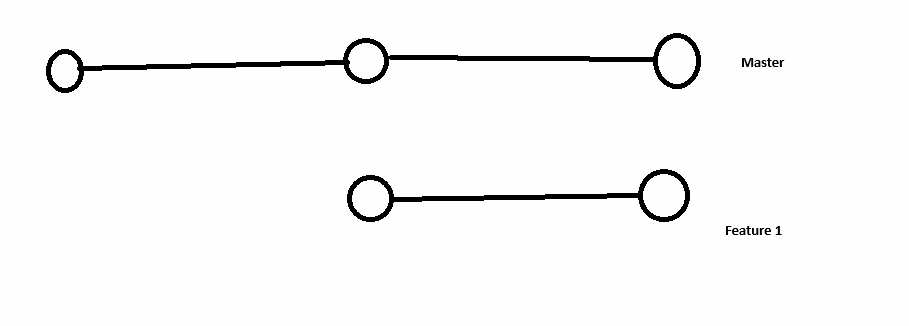


->**All developers once they are done with the work they commit changes**

**->No branching is done,**

**->good for small projects**

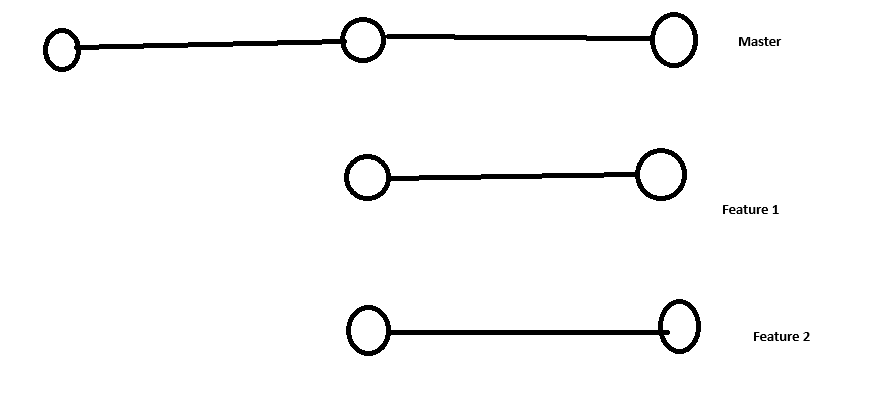
**Feature Branching:**



**->Different branches for diff. Features**

**-> once we have enough features we merge it to the master**

**Gitflow workflow**



**-> Master is not touched until and unless each and everything is ready**

**-> enough features to launch an update only then we merge**

**Forking:**

**->Fork is a copy of the repo.**

**->allows u to experiment to different repositories and make changes to them without making actual changes in the code**

**->suggestion**

**->Open-sourced contribution**