**GIT**



* **git pull origin master**
* **git checkout -b myBranchName**
* **--made changes (after taking latest pull from master in this branch as well)**
* **git status --** files will be in red color
* **git diff “filename” :** you can check what changes has been done
* **git add myfileName\_1**
* **git add myfileName\_2**
* **git status ----** files will be in green color
* **git commit -m "message"**
* **git push origin myBranchName(copy your changes in local, take the latest pull from master and then push) :** is used to upload local changes to a remote repository
* **Create a PR (Pull Request): It can be Approved or Declined**
* **reviewer will merge your changes if it approved into master branch**



**git push** is used to upload local changes to a remote repository.

**git merge** is used to integrate changes from one branch into another branch within your local repository and if there are conflicts, Git will ask you to resolve the conflicts before the merge can be completed.

**How to resolve merge conflict?**

A merge conflict occurs when two branches have made different changes to the same portion of a file, and Git is unable to automatically resolve the differences. When this happens, Git will mark the conflicting sections in the file and ask you to resolve the conflict manually.

**Here are the steps to resolve a merge conflict:**

* Identify the conflicting file: Git will usually inform you about the conflicting files when you run the git merge command. You can also use the command git status to see a list of conflicting files.
* Open the conflicting file: Open the conflicting file in a text editor to see the conflicting changes.
* Identify the conflicting changes: The conflicting changes will be marked with conflict markers, such as <<<<<<<, |||||||, and >>>>>>>. The text between <<<<<<< and ||||||| is the version of the file from the current branch, and the text between ||||||| and >>>>>>> is the version of the file from the branch being merged.
* Resolve the conflict: Decide which changes you want to keep, and which changes you want to discard. Remove the conflict markers and any unwanted changes, then save the file.
* Commit the resolved file: Once you have resolved the conflict, you need to stage and commit the resolved file to complete the merge. You can use the command git add <file> to stage the resolved file, and then use the command git commit to commit the changes.

**What is git cherrypick?**

Cherry picking in Git means to choose a commit from one branch and apply it onto another. It can be done if you eg. made a mistake and committed a change into wrong branch, but do not want to merge the whole branch. You can just eg. revert the commit and cherry-pick it on another branch.

1. checkout (switch to) target branch.
2. git cherry-pick <commit id>

**What is git rebase?**

In Git, the rebase command integrates changes from one branch into another. It is an alternative to the better known "merge" command.

**Graphical user interface, text, application

Description automatically generated**