|  |  |
| --- | --- |
| git fetch/ git pull  remotes  git remote add <name> <remote-url>  git remote -v | Pull is (fetch + merge)  upstream: original project  origin: your forked |
| Peer to Peer (P2P)  Porcelain/Plumbing  Fast Forward | -means each computer can act as a server/client  -decentralized communications model where each party has the same capabilities  High-level/Low-level  -if no additional work on main, Git can perform a FF merge  -like working on the main the entire time |
| Detached HEAD  git checkout <commit> | -when Head is not point to a branch but a commit  Checkout a commit -> make multiple commits-> to save them, create a branch on detached HEAD otherwise GC will delete them |
| Rebase | -the branch you are on, gets “detached” and put on top of the branch you are git rebase <branch>  -In reality, makes new commits with new hashes because commits are DB objects (immutable)  -GC will remove unreachable commits |
| Rebase vs Tradeoff  \*Never rebase shared commits!  Scenarios:  -clean up your local history before sharing branch  -pull changes from a branch into your branch without performing a merge | -Refactors history to look cleaner but a lie  Merge  -preserve history as exactly as it happened but can look messy  -if conflicts, merge commit would include fixes to the conflict |
| git push -f | -we can lose commits by GC  -history will be conflicting  -doesn’t solve conflicts, just puts the burden on other users |
| git push -u origin feature4  git ls-remote  git remote -v  git fetch origin feature4  git branch -a  git checkout –track origin/feature4 | -Push the feature4 branch to remote and set the origin as upstream branch  -u sets the origin as upstream flag  -List branches on remote  -List remotes and URLs  -Fetch the f4 branch from remote to local  -List both remote-tracking and local branches  -Set up a local branch to track remote branch |
| git merge-base ticket1 main  git rebase -i <use return commit above>  pick <commit>  squash <commit>  …  Then commit message  git rebase main | -get the original base of ticket1 branch from main  -start the rebase from the commit sha  PICK the first one  Squash the commits you want to combine into a single commit  -pull the current changes from main |
|  |  |
|  |  |
|  |  |
|  |  |