**Comparing Changes using git diff:**

Git diff command humy help krti hy k hum comparison kr skien different commits, different branches or different files k drmian mein. Or staging area of working directory k drmian mein.

It is all about showing changes.

1. Changes between the large commit and our working directory
2. Changes between staging area and working directory
3. Changes between two branches over a certain number of commits
4. Changes between two files over a certain number of commits.
5. Changes between what we have locally and what is on github.

Ye command git diff repository ko kuch ni krti.

* Git diff : ye likhne se wo sari changes khul k samny aa jiengi jo abhi tak stage ni hui mtlb abhi tk apne unhy git add ni kia. Abhi jo mny changes kiye hein is se pehle meri files kis halat me thi.

**Reading Git Diff Output:**

Watch Slides.

* **Git diff HEAD :** Ye command stage or unstage dono changes ko list krti hy HEAD tak jitney b commits kiye hoty hein waha tk. Anything new in the working directory since HEAD. Last commit jb kia tha meny tab meri file ya files kis halt me thi or ab kya hy.
* **Git diff –staged , --cached:** ye command humy bas wo changes show krti hy jo stage ho chuki hein.Kya difference hy mere last commit or mere staging area mein. Last commit k wqt apki files kis halat me thi or ab jo files stage krdi hein unme or last commit wali files me kya frk hy.

**Diffing Specific Files:**

* Git diff HEAD [FILE-NAME] [FILENAME IF MULTIPLE]
* Git diff [filename]
* Git diff –stage [filename]

**Diffing Branches:**

* Git diff branch1name..branch2name : order matters

**Comparing Commits:**

* Git diff commit1..commit2 : order matters

Extras:

* Git diff HEAD..HEAD~1 : compare changes between HEAD OR Us se prior commit jo k HEAD ka parent hoga.
* HEAD : jhn hum hein
* HEAD~1: HEAD ka parent mtlb HEAD se pichla commit