Git Basics - Summary Notes

Basic Git Workflow

- 1. Create folders and files:

mkdir folderone foldertwo

vi myfolderone/testfile

- 2. Git Commands Sequence:
- - git status : Shows the current status of files.
- - git add -A : Stages all changes (new, modified, deleted files).
- - git commit -m "message" : Saves the changes locally with a message.
- - git push : Sends the changes to the remote repository.

Tracking Changes

- - git log : Shows the commit history.
- - Each commit has a unique SHA code.
- - git show <short_sha> : Shows details of a specific commit.

Git Internals (How Git Stores Data)

- - Git stores data as objects: Commit, Tree, Blob.
- - Commands to explore objects:
- - git cat-file -t <sha> : Shows the type (commit/tree/blob).
- - git cat-file -p <sha> : Shows the content of the object.
- - git cat-file -s <sha> : Size of the object.
- - git cat-file -e <sha> : Checks if the object exists.

https://git-scm.com/docs/git-cat-file

Understanding HEAD

- - HEAD points to the latest commit on the current branch.
- - Stored in .git/refs/heads/<branch_name>.
- - When a new commit is made, HEAD moves forward.

Using Git Stash

- - git stash : Temporarily saves work without committing.

- - Scenario: Developer is interrupted and needs to save 90% progress.
- - Steps:

  git add -A

  git stash

- - Later:

  git stash apply : Restores saved changes.

- - Resolve conflicts if any, and then:

  git commit -m "Final work done"

Pending Topics

- 1. Stash ✅
- 2. Fast-forwarding

3. Cherry-pick

4. Rebase

5. Hooks

6. Merge Conflicts