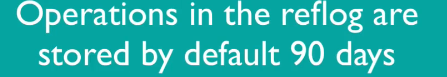
# Lec: 225 Modify Last Commit Using Amend Option.

1. To adjust info in just last commit as you made some errors in last commit. The previous one will be replaced by new one. Destructive Operation.
2. **Command**: git commit **--amend** -m “new msg”
   1. To amend author: --amend --author=”Yatan Kumar [Yatan.kumar@sc.com](mailto:Yatan.kumar@sc.com)”.

# Lec: 226 Cherry-picking commits

1. Pick one or more commits from one branch and apply onto another branch. Will create a new commit.
2. Scenarios:
   1. Instead of merging, just apply a specific commit.
   2. Take a commit from Detached head and apply to some other branch.
3. **Command**: cherry pick
   1. git cherry-pick <commit-id>
   2. git cherry-pick --no-commit <commit-id>: It will not create commit but stage them.
4. **Command**:
   1. **git status** -v : same as git diff --staged or --cached
   2. **git status** -v -v : Same as git diff + git diff --staged.

# Lec: 227 Reflog - log of all Git Operations

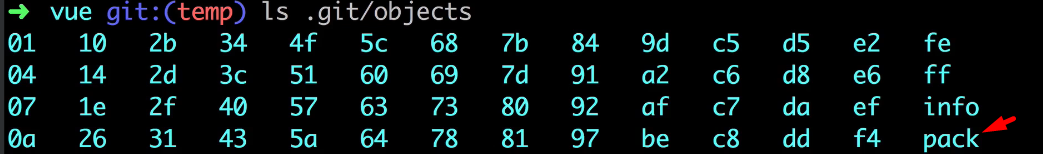
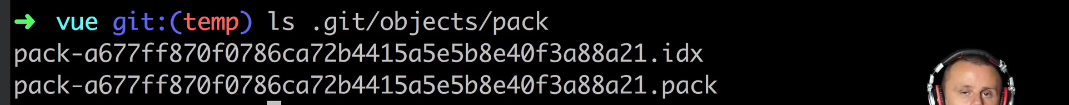
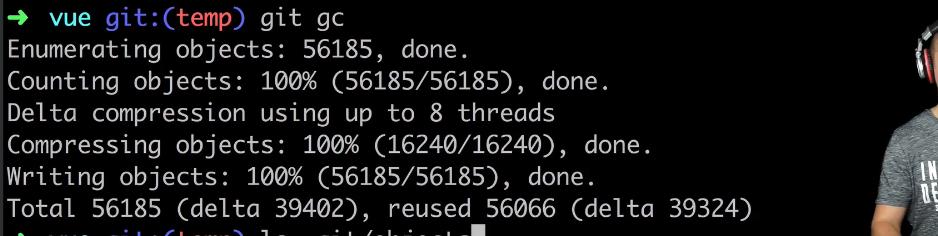
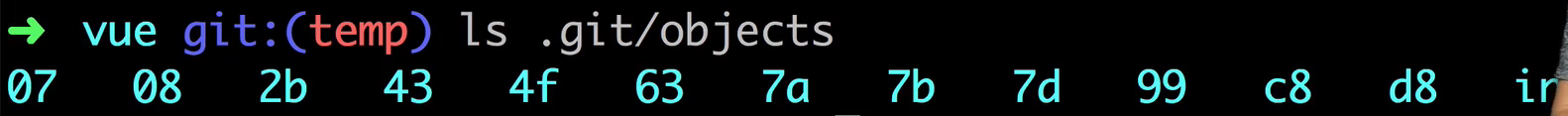
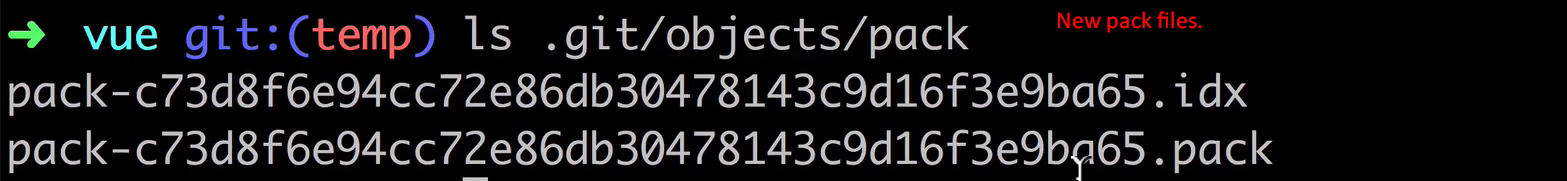
1. To list operations performed only locally.  
   
2. **Commands**:
   1. git reflog
   2. git reflog show <branch>
3. **Applications**:
   1. To get a commit.
   2. To restore a hard reset as we can have the commit from there.

# 228 - Stashing changes Using Terminal

1. **Command**:
   1. git stash

# 229 - Stashing Using Github Desktop

# 230 - Garbage Collection

1. **Application**:
   1. To clean unreachable Git objects (Automatic Operation)
   2. old references in reflog.
2. **Command**:
   1.   
        
        
        
        
      

# Lec: 231 Squashing of commits of Github