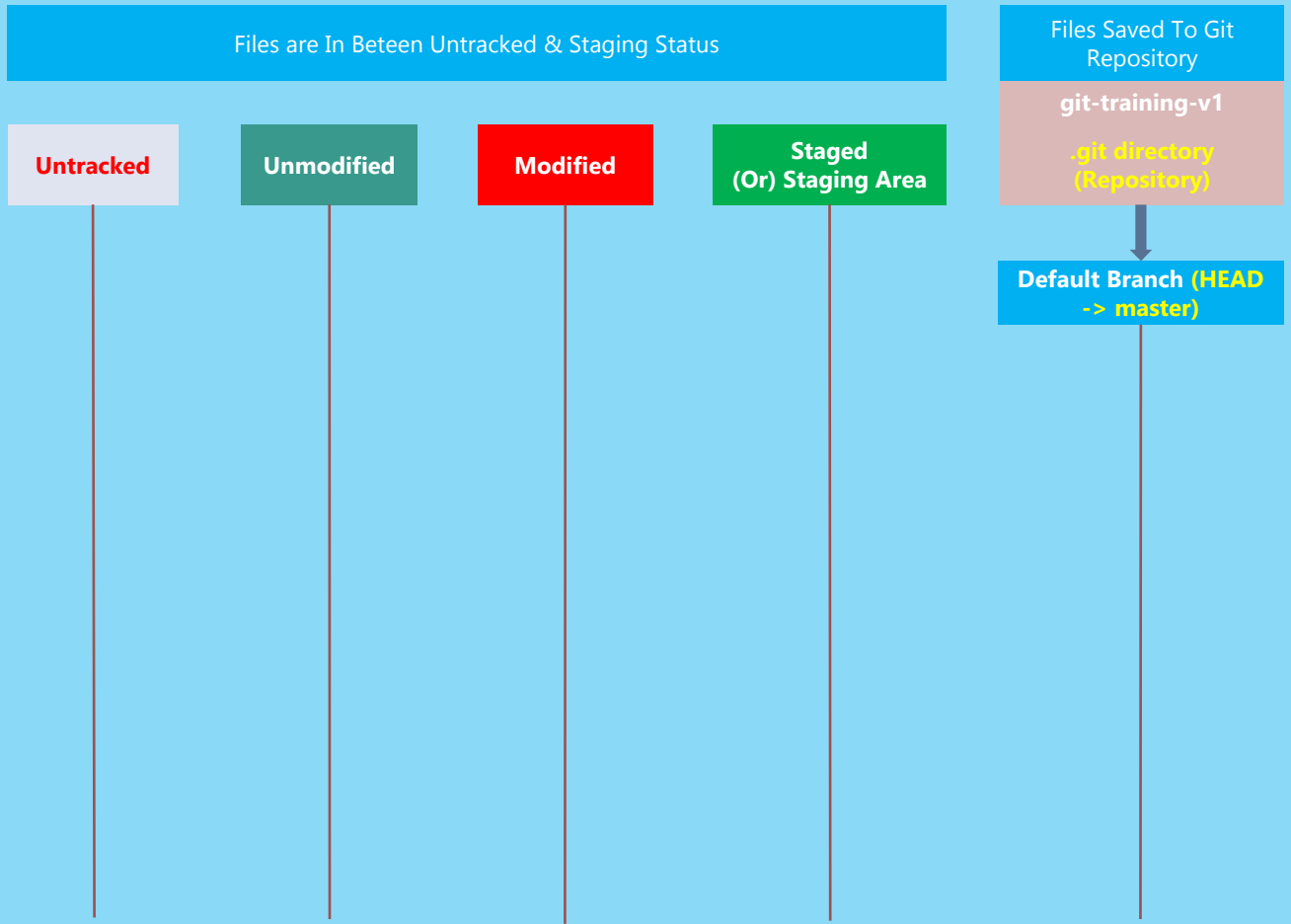
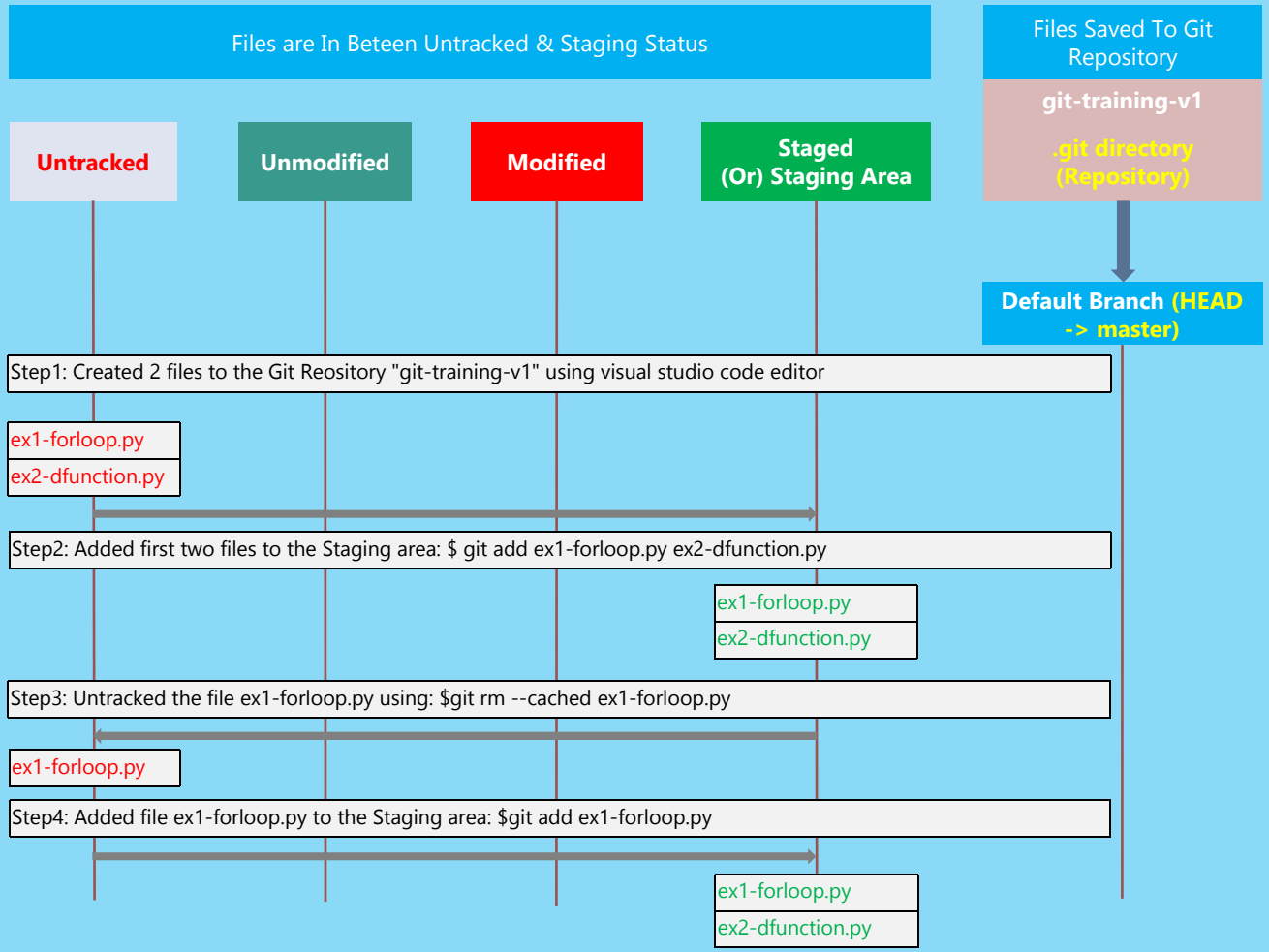
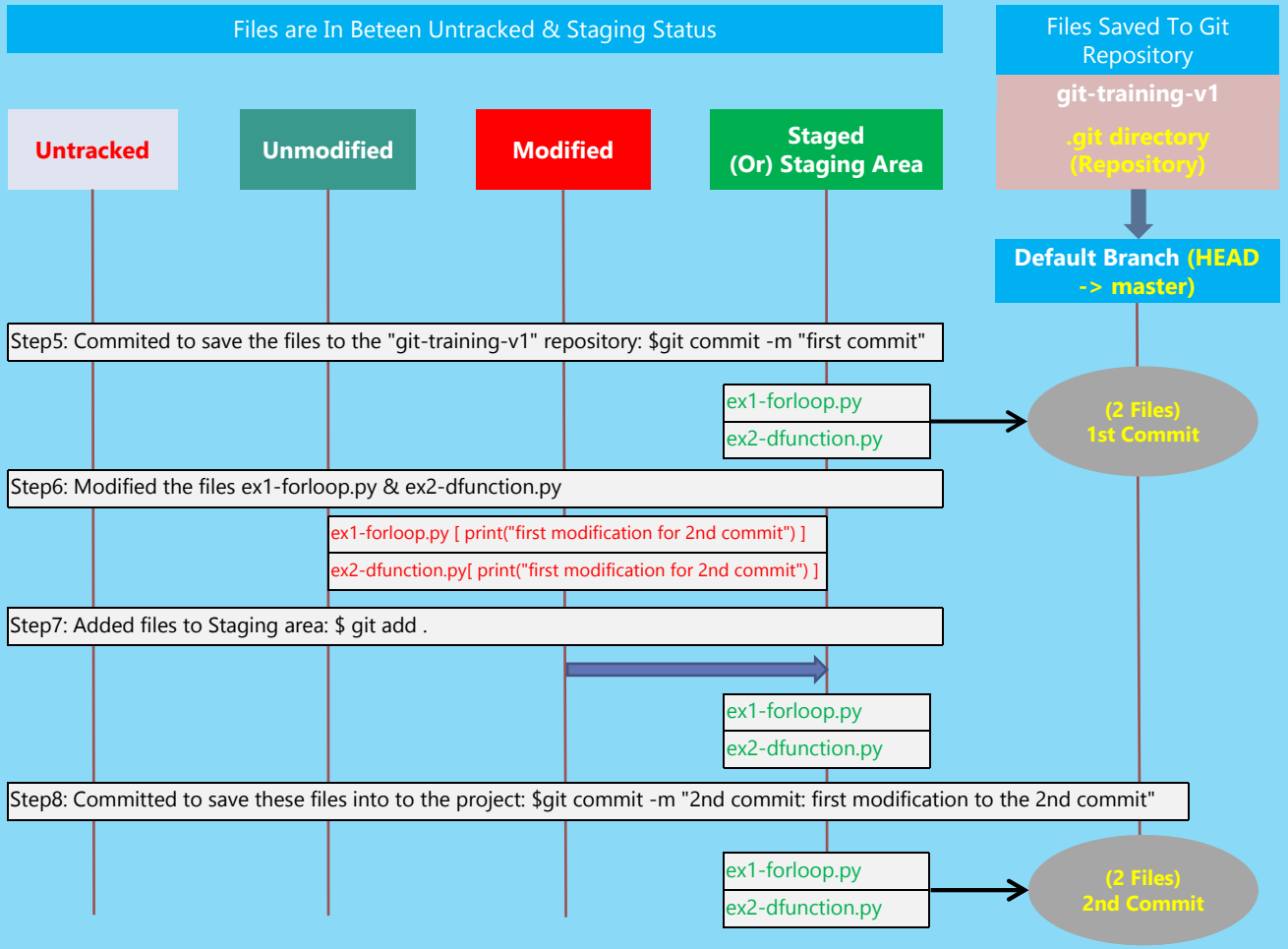
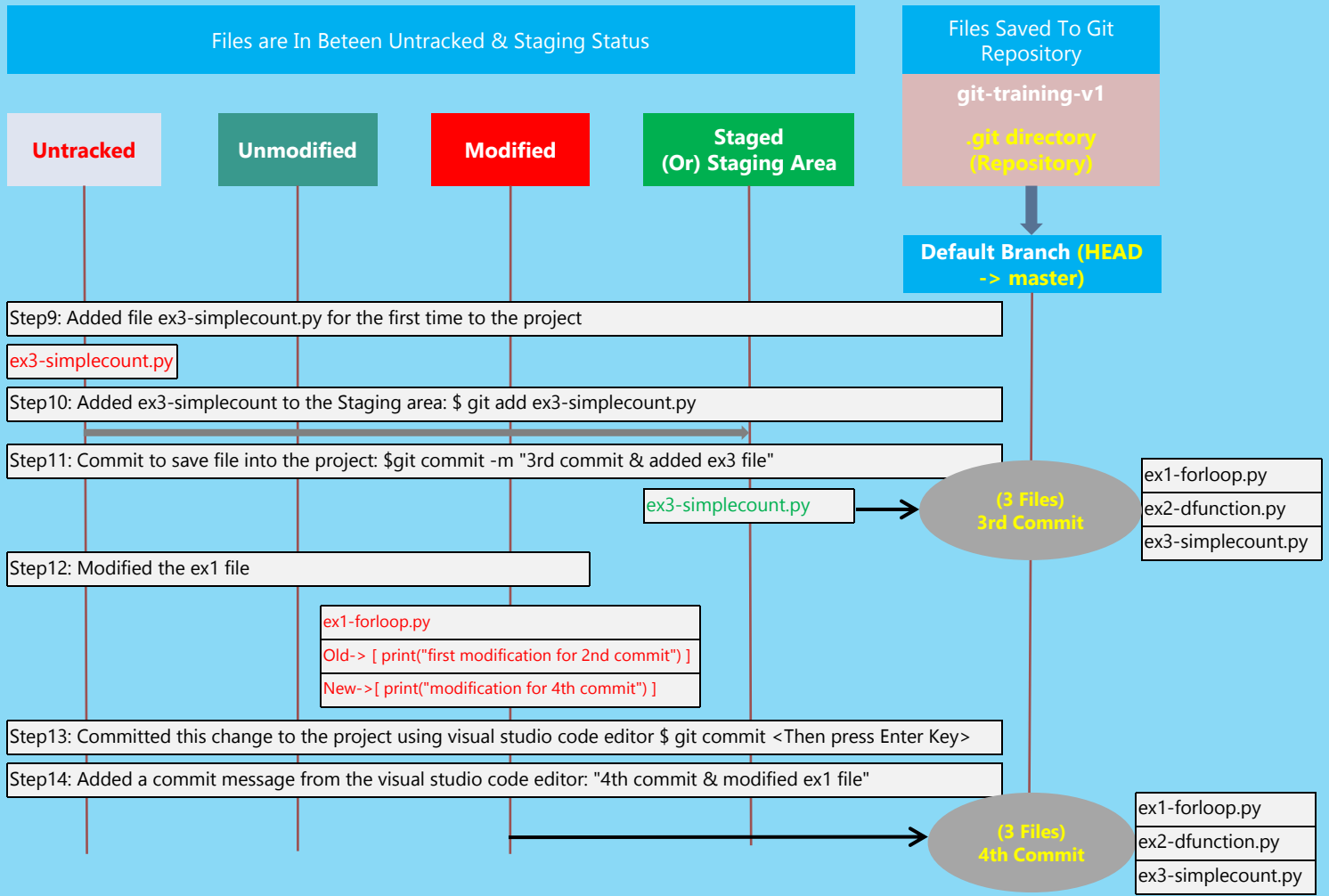
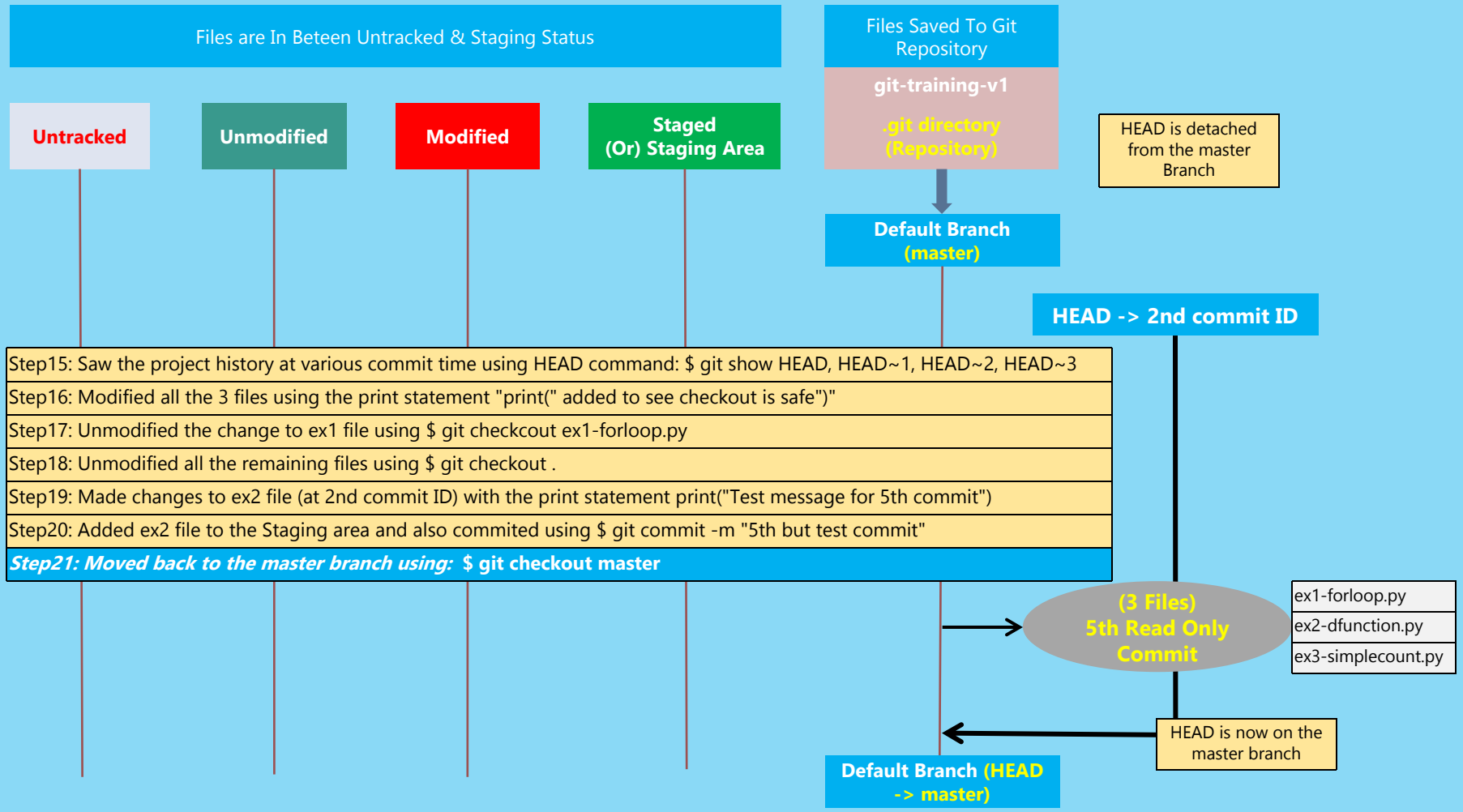
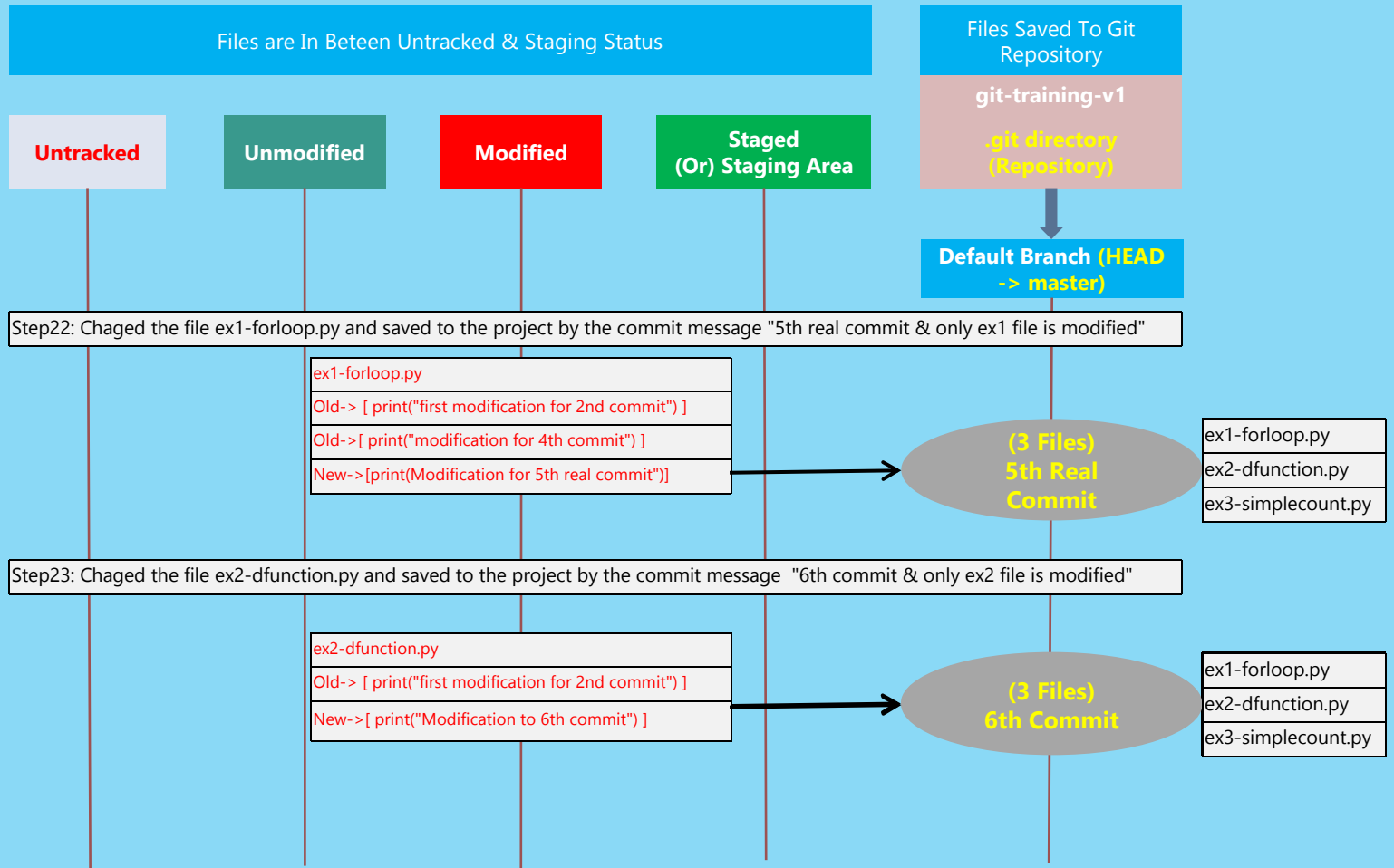

Files are In Beteen Untracked & Staging Status
Files Saved To Git Repository
Untracked
Unmodified
Modified
Staged (Or) Staging Area
git-training-v1
.git directory (Repository)
Default Branch (HEAD -> master)

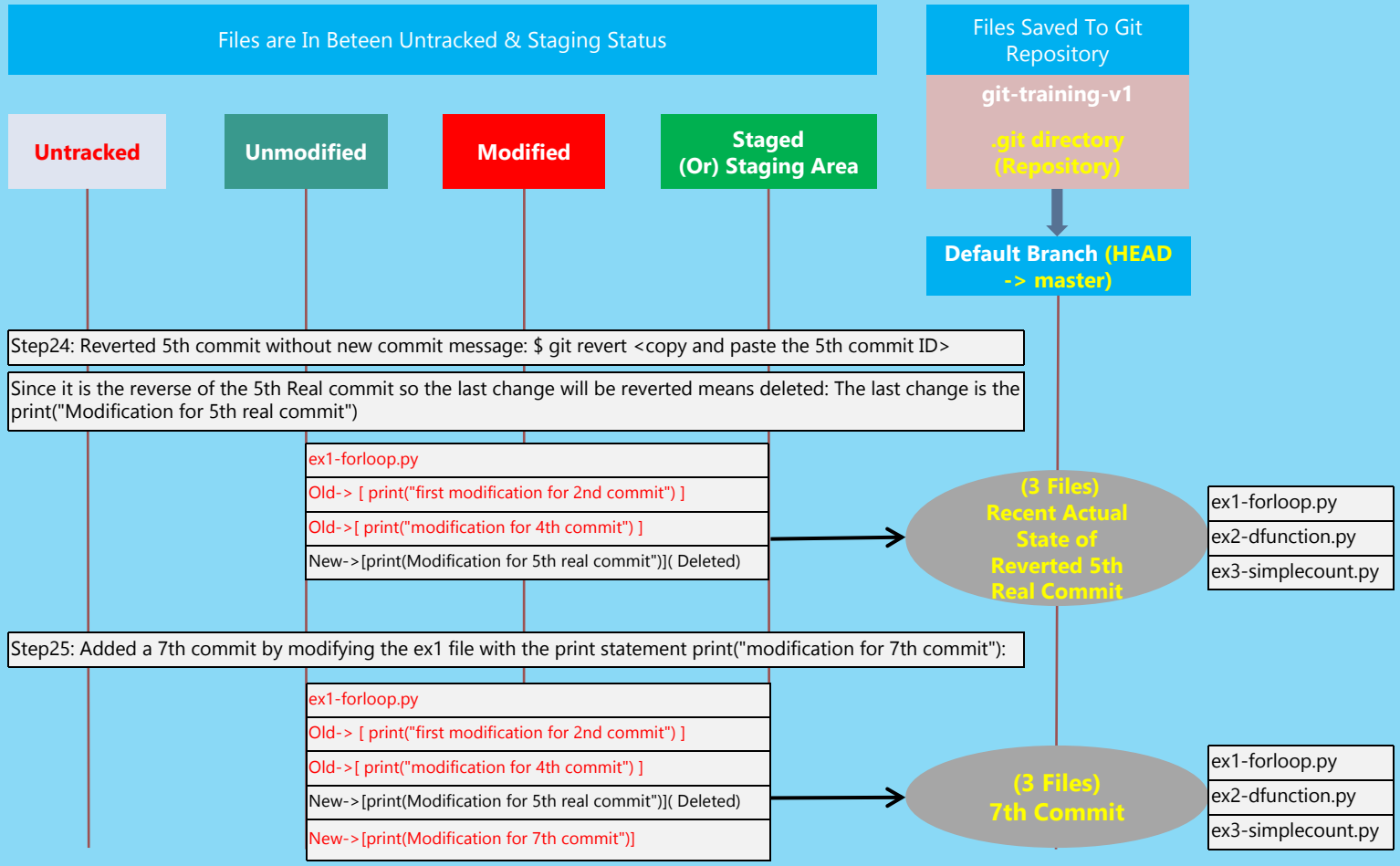

Files are In Beteen Untracked & Staging Status
Files Saved To Git Repository
git-training-v1
.git directory (Repository)
Untracked
Unmodified
Modified
Staged (Or) Staging Area
Default Branch (HEAD -> master)
Step1: Created 2 files to the Git Reository "git-training-v1" using visual studio code editor
ex1-forloop.py
ex2-dfunction.py
Step2: Added first two files to the Staging area: $ git add ex1-forloop.py ex2-dfunction.py
ex1-forloop.py
ex2-dfunction.py
Step3: Untracked the file ex1-forloop.py using: $git rm --cached ex1-forloop.py
ex1-forloop.py
Step4: Added file ex1-forloop.py to the Staging area: $git add ex1-forloop.py
ex1-forloop.py
ex2-dfunction.py

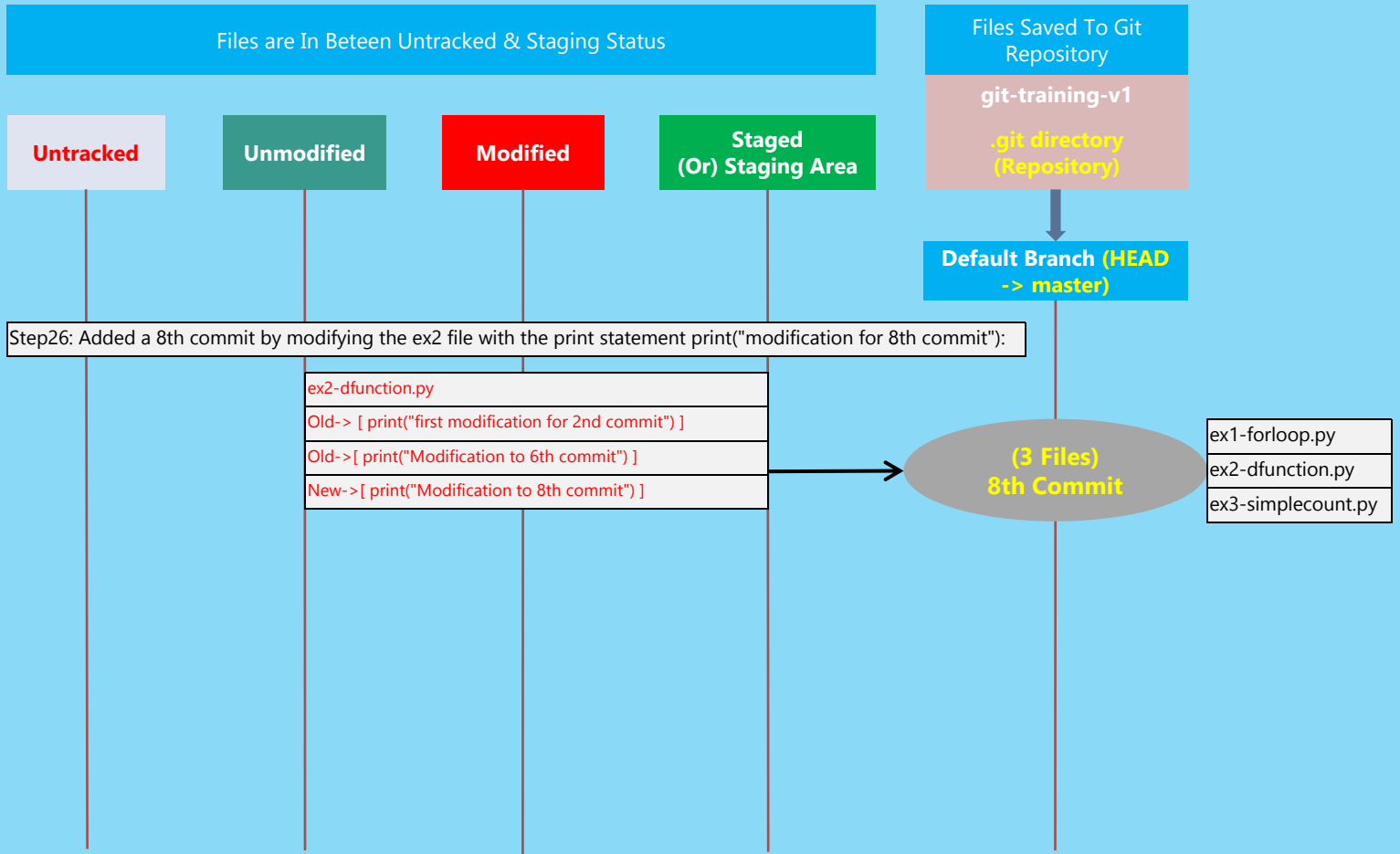

Files are In Beteen Untracked & Staging Status
Files Saved To Git Repository
git-training-v1
.git directory (Repository)
Untracked
Unmodified
Modified
Staged (Or) Staging Area
Default Branch (HEAD -> master)
Step5: Commited to save the files to the "git-training-v1" repository: $git commit -m "first commit"
ex1-forloop.py
ex2-dfunction.py
(2 Files) 1st Commit
Step6: Modified the files ex1-forloop.py & ex2-dfunction.py
ex1-forloop.py [print("first modification for 2nd commit")]
ex2-dfunction.py[print("first modification for 2nd commit")]
Step7: Added files to Staging area: $ git add .
ex1-forloop.py
ex2-dfunction.py
Step8: Committed to save these files into to the project: $git commit -m "2nd commit: first modification to the 2nd commit"
ex1-forloop.py
ex2-dfunction.py
(2 Files) 2nd Commit

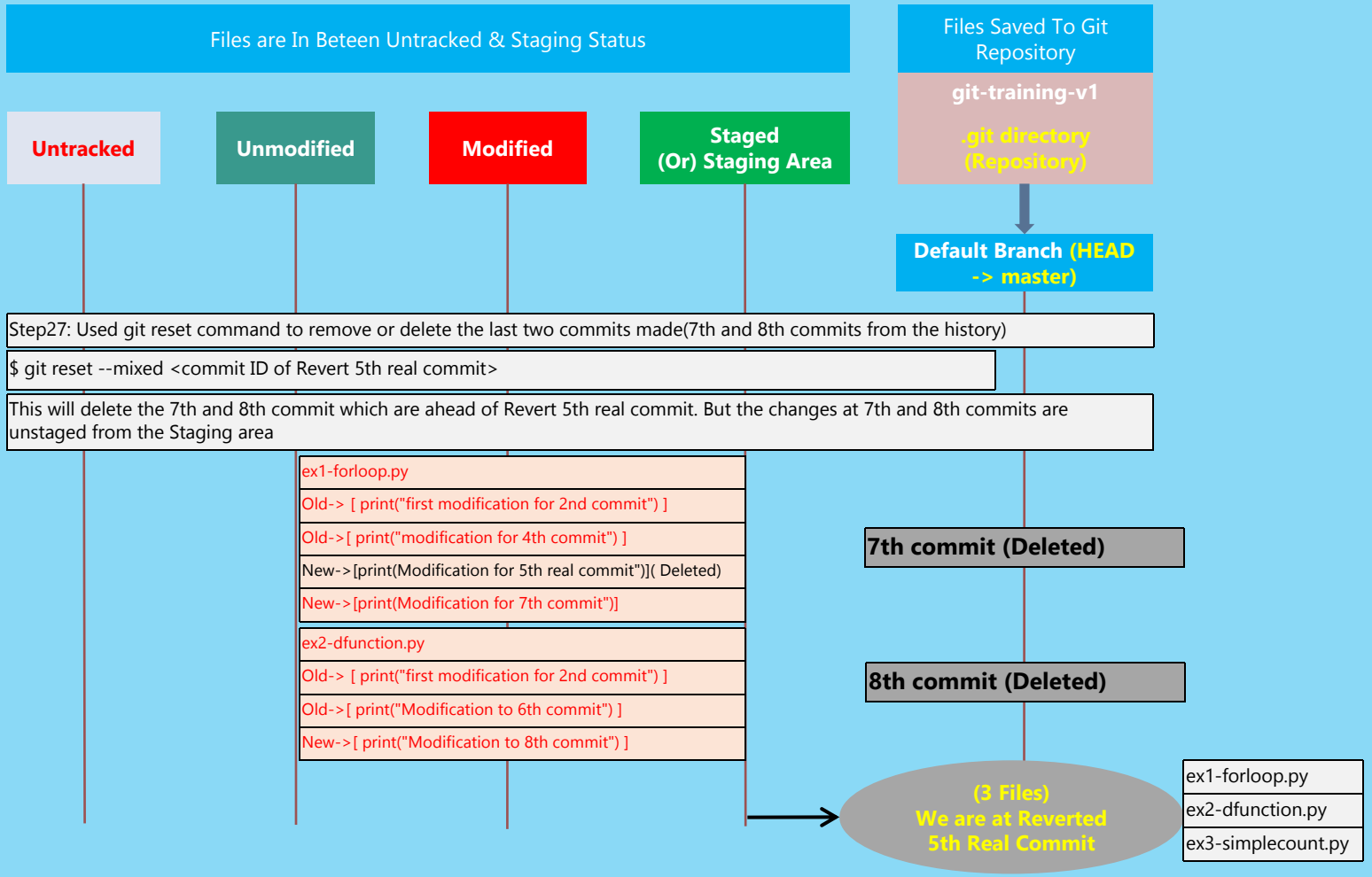

Files are In Beteen Untracked & Staging Status
Files Saved To Git Repository
git-training-v1
.git directory (Repository)
Untracked
Unmodified
Modified
Staged (Or) Staging Area
Default Branch (HEAD -> master)
Step9: Added file ex3-simplecount.py for the first time to the project
ex3-simplecount.py
Step10: Added ex3-simplecount to the Staging area: $ git add ex3-simplecount.py
Step11: Commit to save file into the project: $git commit -m "3rd commit & added ex3 file"
ex3-simplecount.py
(3 Files) 3rd Commit
ex1-forloop.py
ex2-dfunction.py
ex3-simplecount.py
Step12: Modified the ex1 file
ex1-forloop.py
Old-> [print("first modification for 2nd commit")]
New->[print("modification for 4th commit")]
Step13: Committed this change to the project using visual studio code editor $ git commit <Then press Enter Key>
Step14: Added a commit message from the visual studio code editor: "4th commit & modified ex1 file"
(3 Files) 4th Commit
ex1-forloop.py
ex2-dfunction.py
ex3-simplecount.py

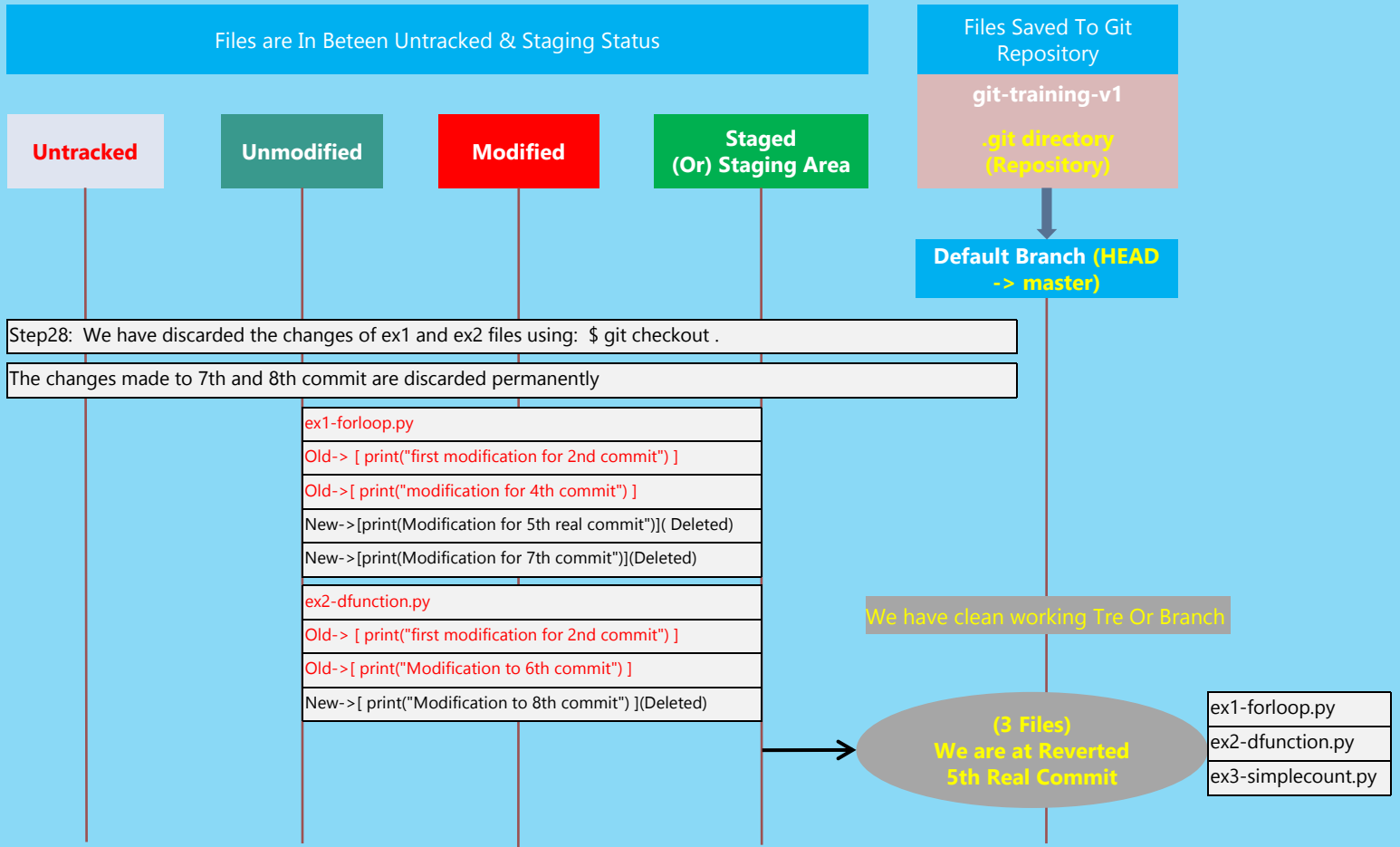

Files are In Beteen Untracked & Staging Status
Files Saved To Git Repository
git-training-v1
.git directory (Repository)
Untracked
Unmodified
Modified
Staged (Or) Staging Area
HEAD is detached from the master Branch
Default Branch (master)
HEAD -> 2nd commit ID
Step15: Saw the project history at various commit time using HEAD command: $ git show HEAD, HEAD~1, HEAD~2, HEAD~3
Step16: Modified all the 3 files using the print statement "print(" added to see checkout is safe")"
Step17: Unmodified the change to ex1 file using $ git checkcout ex1-forloop.py
Step18: Unmodified all the remaining files using $ git checkout .
Step19: Made changes to ex2 file (at 2nd commit ID) with the print statement print("Test message for 5th commit")
Step20: Added ex2 file to the Staging area and also commited using $ git commit -m "5th but test commit"
Step21: Moved back to the master branch using: $ git checkout master
(3 Files) 5th Read Only Commit
ex1-forloop.py
ex2-dfunction.py
ex3-simplecount.py
HEAD is now on the master branch
Default Branch (HEAD -> master)

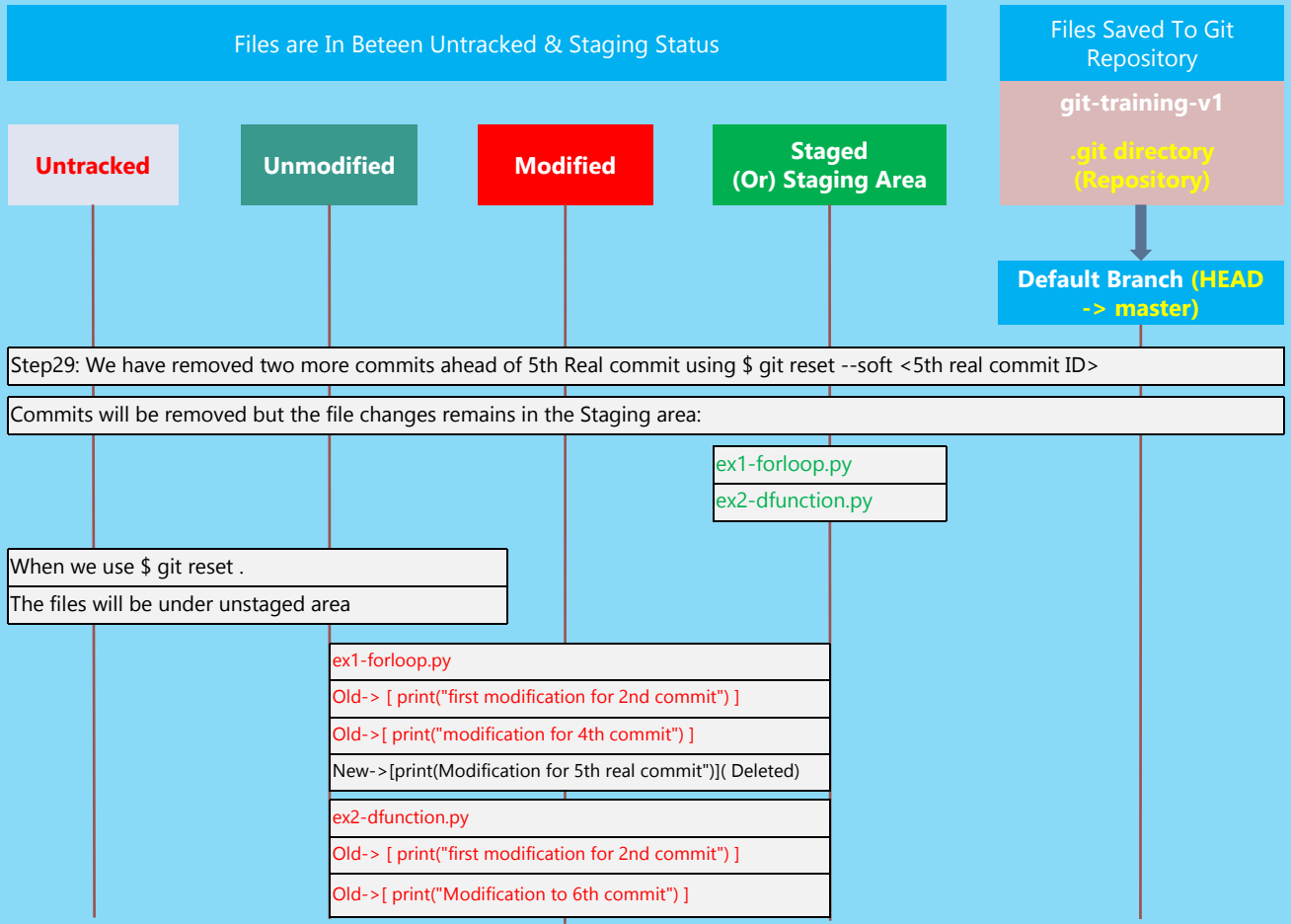

Files are In Beteen Untracked & Staging Status
Files Saved To Git Repository
git-training-v1
.git directory (Repository)
Untracked
Unmodified
Modified
Staged (Or) Staging Area
Default Branch (HEAD -> master)
Step22: Chaged the file ex1-forloop.py and saved to the project by the commit message "5th real commit & only ex1 file is modified"
ex1-forloop.py
Old-> [print("first modification for 2nd commit")]
Old->[print("modification for 4th commit")]
New->[print(Modification for 5th real commit")]
(3 Files) 5th Real Commit
ex1-forloop.py
ex2-dfunction.py
ex3-simplecount.py
Step23: Chaged the file ex2-dfunction.py and saved to the project by the commit message "6th commit & only ex2 file is modified"
ex2-dfunction.py
Old-> [print("first modification for 2nd commit")]
New->[print("Modification to 6th commit")]
(3 Files) 6th Commit
ex1-forloop.py
ex2-dfunction.py
ex3-simplecount.py

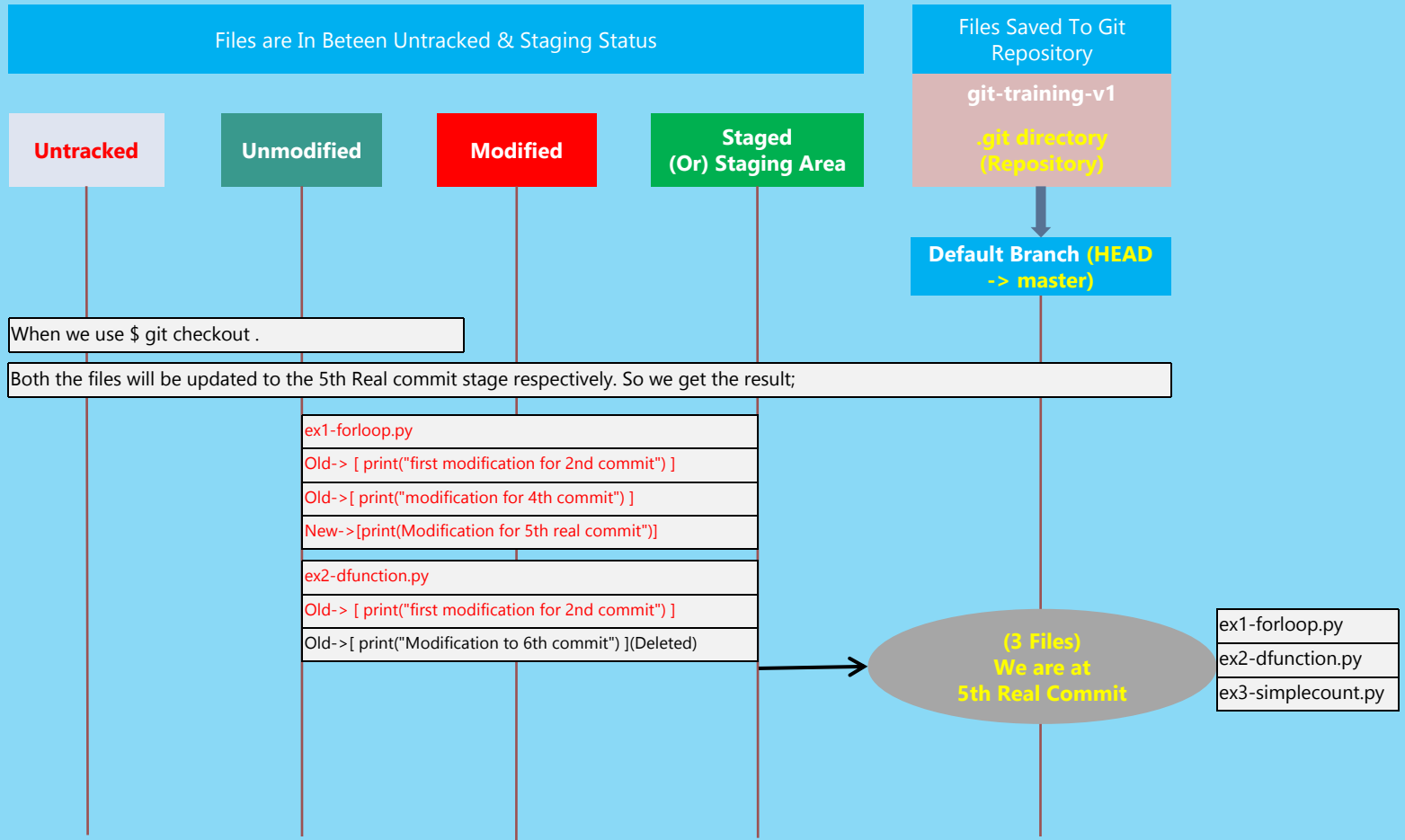

Files are In Beteen Untracked & Staging Status
Files Saved To Git Repository
git-training-v1
.git directory (Repository)
Untracked
Unmodified
Modified
Staged (Or) Staging Area
Default Branch (HEAD -> master)
Step24: Reverted 5th commit without new commit message: $ git revert <copy and paste the 5th commit ID>
Since it is the reverse of the 5th Real commit so the last change will be reverted means deleted: The last change is the print("Modification for 5th real commit")
ex1-forloop.py
Old-> [print("first modification for 2nd commit")]
Old->[print("modification for 4th commit")]
New->[print(Modification for 5th real commit")](Deleted)
(3 Files) Recent Actual State of Reverted 5th Real Commit
ex1-forloop.py
ex2-dfunction.py
ex3-simplecount.py
Step25: Added a 7th commit by modifying the ex1 file with the print statement print("modification for 7th commit"):
ex1-forloop.py
Old-> [print("first modification for 2nd commit")]
Old->[print("modification for 4th commit")]
New->[print(Modification for 5th real commit")](Deleted)
New->[print(Modification for 7th commit")]
(3 Files) 7th Commit
ex1-forloop.py
ex2-dfunction.py
ex3-simplecount.py

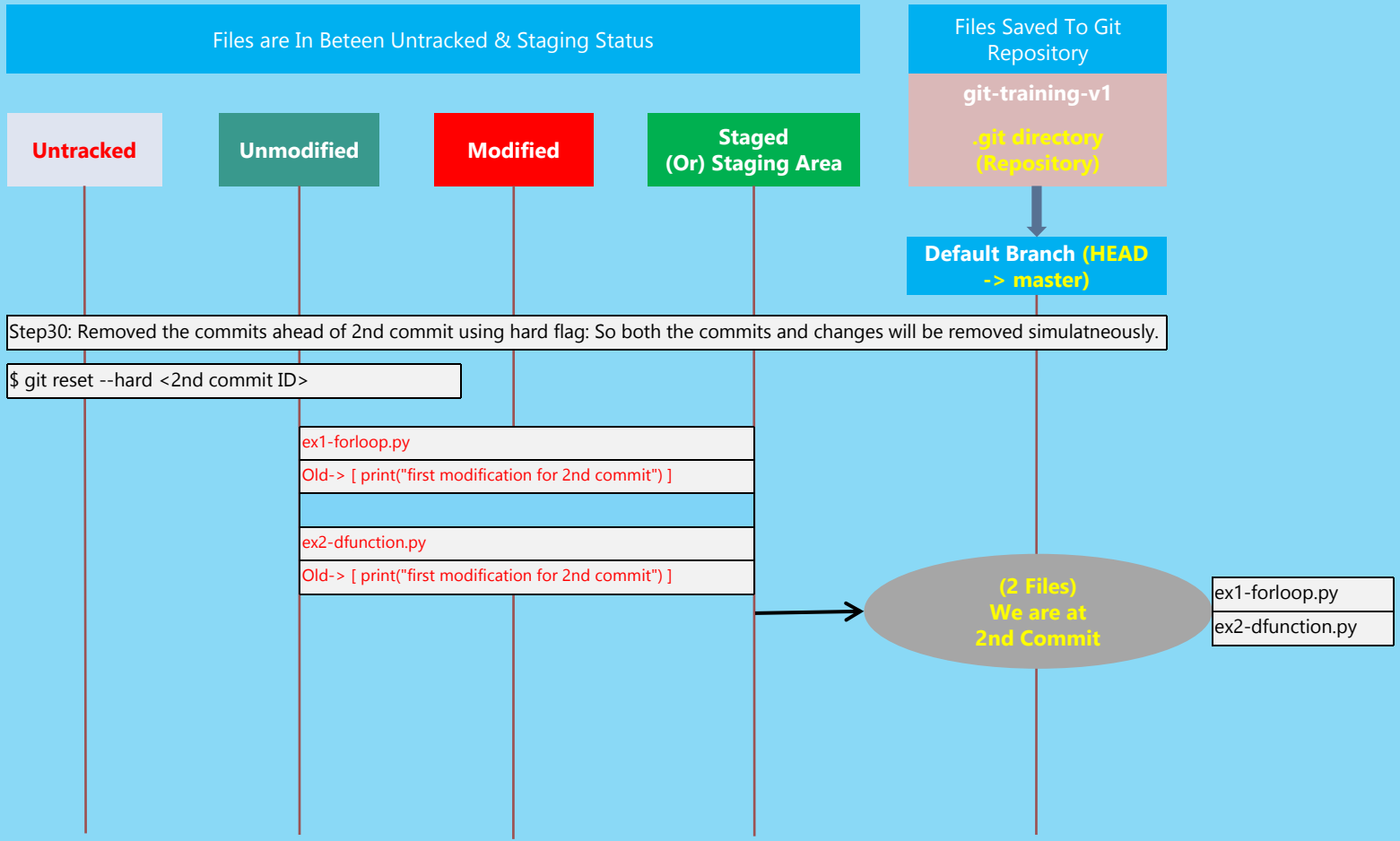

Files are In Beteen Untracked & Staging Status
Files Saved To Git Repository
git-training-v1
.git directory (Repository)
Untracked
Unmodified
Modified
Staged (Or) Staging Area
Default Branch (HEAD -> master)
Step26: Added a 8th commit by modifying the ex2 file with the print statement print("modification for 8th commit"):
ex2-dfunction.py
Old-> [print("first modification for 2nd commit")]
Old->[print("Modification to 6th commit")]
New->[print("Modification to 8th commit")]
(3 Files) 8th Commit
ex1-forloop.py
ex2-dfunction.py
ex3-simplecount.py


Files are In Beteen Untracked & Staging Status
Files Saved To Git Repository
git-training-v1
.git directory (Repository)
Untracked
Unmodified
Modified
Staged (Or) Staging Area
Default Branch (HEAD -> master)
Step27: Used git reset command to remove or delete the last two commits made(7th and 8th commits from the history)
$ git reset --mixed <commit ID of Revert 5th real commit>
This will delete the 7th and 8th commit which are ahead of Revert 5th real commit. But the changes at 7th and 8th commits are unstaged from the Staging area
ex1-forloop.py
Old-> [print("first modification for 2nd commit")]
Old->[print("modification for 4th commit")]
New->[print(Modification for 5th real commit")](Deleted)
New->[print(Modification for 7th commit")]
ex2-dfunction.py
Old-> [print("first modification for 2nd commit")]
Old->[print("Modification to 6th commit")]
New->[print("Modification to 8th commit")]
7th commit (Deleted)
8th commit (Deleted)
(3 Files)
We are at Reverted 5th Real Commit
ex1-forloop.py
ex2-dfunction.py
ex3-simplecount.py


Files are In Beteen Untracked & Staging Status
Files Saved To Git Repository
git-training-v1
.git directory (Repository)
Untracked
Unmodified
Modified
Staged (Or) Staging Area
Default Branch (HEAD -> master)
Step28: We have discarded the changes of ex1 and ex2 files using: $ git checkout .
The changes made to 7th and 8th commit are discarded permanently
ex1-forloop.py
Old-> [print("first modification for 2nd commit")]
Old->[print("modification for 4th commit")]
New->[print(Modification for 5th real commit")](Deleted)
New->[print(Modification for 7th commit")](Deleted)
ex2-dfunction.py
Old-> [print("first modification for 2nd commit")]
Old->[print("Modification to 6th commit")]
New->[print("Modification to 8th commit")](Deleted)
We have clean working Tre Or Branch
(3 Files)
We are at Reverted
5th Real Commit
ex1-forloop.py
ex2-dfunction.py
ex3-simplecount.py


Files are In Beteen Untracked & Staging Status
Files Saved To Git Repository
git-training-v1
.git directory (Repository)
Untracked
Unmodified
Modified
Staged (Or) Staging Area
Default Branch (HEAD -> master)
Step29: We have removed two more commits ahead of 5th Real commit using $ git reset --soft <5th real commit ID>
Commits will be removed but the file changes remains in the Staging area:
ex1-forloop.py
ex2-dfunction.py
When we use $ git reset .
The files will be under unstaged area
ex1-forloop.py
Old-> [print("first modification for 2nd commit")]
Old->[print("modification for 4th commit")]
New->[print(Modification for 5th real commit")](Deleted)
ex2-dfunction.py
Old-> [print("first modification for 2nd commit")]
Old->[print("Modification to 6th commit")]


Files are In Beteen Untracked & Staging Status
Files Saved To Git Repository
git-training-v1
.git directory (Repository)
Untracked
Unmodified
Modified
Staged (Or) Staging Area
Default Branch (HEAD -> master)
When we use $ git checkout .
Both the files will be updated to the 5th Real commit stage respectively. So we get the result;
ex1-forloop.py
Old-> [print("first modification for 2nd commit")]
Old->[print("modification for 4th commit")]
New->[print(Modification for 5th real commit")]
ex2-dfunction.py
Old-> [print("first modification for 2nd commit")]
Old->[print("Modification to 6th commit")](Deleted)
(3 Files) We are at 5th Real Commit
ex1-forloop.py
ex2-dfunction.py
ex3-simplecount.py


Files are In Beteen Untracked & Staging Status
Files Saved To Git Repository
git-training-v1
.git directory (Repository)
Untracked
Unmodified
Modified
Staged (Or) Staging Area
Default Branch (HEAD -> master)
Step30: Removed the commits ahead of 2nd commit using hard flag: So both the commits and changes will be removed simulatneously.
$ git reset --hard <2nd commit ID>
ex1-forloop.py
Old-> [print("first modification for 2nd commit")]
ex2-dfunction.py
Old-> [print("first modification for 2nd commit")]
(2 Files) We are at 2nd Commit
ex1-forloop.py
ex2-dfunction.py