

1. Set your username and email in git config

```
ayushchouhan@Practos-MacBook-Pro ~ % git config --list
credential.helper=osxkeychain
user.name=Ayush Chouhan
user.email=ayush9977@gmail.com
ayushchouhan@Practos-MacBook-Pro ~ %
```

2. Create a new branch named "feature-branch" and switch to it.

```
ayushchouhan@Practos-MacBook-Pro Practo Git Assignment % git checkout -b feature
Switched to a new branch 'feature'
ayushchouhan@Practos-MacBook-Pro Practo Git Assignment %
```

3. List all branches in the repository.

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git branch
* feature
  main
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

4. Delete the branch "feature-branch"

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
[ayushchouhan@Practos-MacBook-Pro GAssignment % git branch -d feature
Deleted branch feature (was ee68c6b).
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

5. How do you undo the last commit

```
ayushchouhan@Practos-MacBook-Pro GAssignment % git reset --soft HEAD^
fatal: ambiguous argument 'HEAD^': unknown revision or path not in the working tree.
Use '---' to separate paths from revisions, like this:
'git <command> [<revision>...] --- [<file>...]'
ayushchouhan@Practos-MacBook-Pro GAssignment % git reset --mixed HEAD^
fatal: ambiguous argument 'HEAD^': unknown revision or path not in the working tree.
Use '---' to separate paths from revisions, like this:
'git <command> [<revision>...] --- [<file>...]'
ayushchouhan@Practos-MacBook-Pro GAssignment % git reset --hard HEAD^
fatal: ambiguous argument 'HEAD^': unknown revision or path not in the working tree.
Use '---' to separate paths from revisions, like this:
'git <command> [<revision>...] --- [<file>...]'
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

6. Create a new branch names “conflict-branch”

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git branch conflict-branch  
ayushchouhan@Practos-MacBook-Pro GAssignment %]
```

7. Create a another branch named “feature1”

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git branch feature1  
fatal: a branch named 'feature1' already exists  
ayushchouhan@Practos-MacBook-Pro GAssignment %]
```

8. Make some changes in to feature1 branch

```
ayushchouhan@Practos-MacBook-Pro GAssignment % git status  
  
On branch feature1  
nothing to commit, working tree clean  
ayushchouhan@Practos-MacBook-Pro GAssignment % git status  
On branch feature1  
Changes not staged for commit:  
  (use "git add <file>..." to update what will be committed)  
  (use "git restore <file>..." to discard changes in working directory)  
    modified:   README.md  
  
no changes added to commit (use "git add" and/or "git commit -a")  
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

9. Merge “feature1” branch into main branch

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git checkout feature1
 M README.md
Switched to branch 'feature1'
[ayushchouhan@Practos-MacBook-Pro GAssignment % git status
On branch feature1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
[ayushchouhan@Practos-MacBook-Pro GAssignment % git add .
[ayushchouhan@Practos-MacBook-Pro GAssignment % git commit -m "merging"
 [feature1 796e6b8] merging
  1 file changed, 4 insertions(+)
[ayushchouhan@Practos-MacBook-Pro GAssignment % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
[ayushchouhan@Practos-MacBook-Pro GAssignment % git merge feature1
Updating ee68c6b..796e6b8
Fast-forward
 README.md | 4 +++
 1 file changed, 4 insertions(+)
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

10. Make changes in “conflict-branch”, in the same file and line that you had made changes in feature1

```
ayushchouhan@Practos-MacBook-Pro GAssignment % git checkout conflict-branch

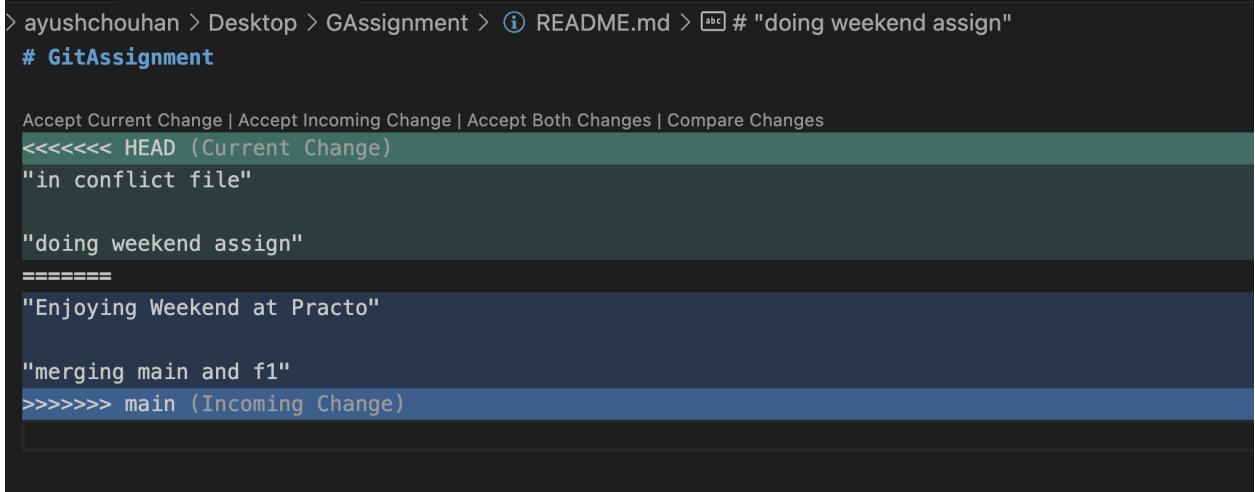
Switched to branch 'conflict-branch'
ayushchouhan@Practos-MacBook-Pro GAssignment % git status
On branch conflict-branch
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

11. Merge master into conflict-branch [Attach screenshot of terminal & file]

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git merge master  
  
merge: master - not something we can merge  
ayushchouhan@Practos-MacBook-Pro GAssignment % git merge origin/master  
  
merge: origin/master - not something we can merge  
[ayushchouhan@Practos-MacBook-Pro GAssignment % git merge main  
Auto-merging README.md  
CONFLICT (content): Merge conflict in README.md  
Automatic merge failed; fix conflicts and then commit the result.  
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

12. Resolve merge conflicts



```
> ayushchouhan > Desktop > GAssignment > ⓘ README.md > abc # "doing weekend assign"  
# GitAssignment  
  
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes  
<<<<< HEAD (Current Change)  
"in conflict file"  
  
"doing weekend assign"  
=====  
"Enjoying Weekend at Practo"  
  
"merging main and f1"  
>>>>> main (Incoming Change)
```

ayushchouhan@Practos-MacBook-Pro GAssignment % git merge conflict-branch

```
error: Merging is not possible because you have unmerged files.  
hint: Fix them up in the work tree, and then use 'git add/rm <file>'  
hint: as appropriate to mark resolution and make a commit.  
fatal: Exiting because of an unresolved conflict.  
[ayushchouhan@Practos-MacBook-Pro GAssignment % git add .  
[ayushchouhan@Practos-MacBook-Pro GAssignment % git commit -m "eleven"  
[conflict-branch 983548a] eleven  
[ayushchouhan@Practos-MacBook-Pro GAssignment % git merge main  
Already up to date.  
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

13. Add a remote named "origin" pointing to a GitHub repository.

```
ayushchouhan@Practos-MacBook-Pro GAssignment % git remote add origin <repository_url>  
zsh: parse error near `\'n'  
[ayushchouhan@Practos-MacBook-Pro GAssignment % git remote add origin https://github.com/Ac-ayush/GitAssignment  
error: remote origin already exists.  
ayushchouhan@Practos-MacBook-Pro GAssignment % git remote -v  
origin  https://github.com/Ac-ayush/GitAssignment.git (fetch)  
origin  https://github.com/Ac-ayush/GitAssignment.git (push)  
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

14. Fork a repository on GitHub and clone it to your local machine.

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git clone https://github.com/Ac-ayush/GitAssignmentSuraj  
Cloning into 'GitAssignmentSuraj'...  
remote: Enumerating objects: 7, done.  
remote: Counting objects: 100% (7/7), done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0  
Receiving objects: 100% (7/7), 2.62 MiB | 4.67 MiB/s, done.  
[ayushchouhan@Practos-MacBook-Pro GAssignment % cd GitAssignmentSuraj  
ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj %
```

15. Create a new branch on your fork, make changes, and open a pull request to the original repository.

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git remote add origin https://github.com/Ac-ayush/GitAssignment
error: remote origin already exists.
ayushchouhan@Practos-MacBook-Pro GAssignment % git remote -v
origin https://github.com/Ac-ayush/GitAssignment.git (fetch)
origin https://github.com/Ac-ayush/GitAssignment.git (push)
[ayushchouhan@Practos-MacBook-Pro GAssignment % git clone https://github.com/Ac-ayush/GitAssignmentSuraj
Cloning into 'GitAssignmentSuraj'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0
Receiving objects: 100% (7/7), 2.62 MiB | 4.67 MiB/s, done.
[ayushchouhan@Practos-MacBook-Pro GAssignment % cd GitAssignmentSuraj
[ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % git checkout -b GitAssignmentSuraj
Switched to a new branch 'GitAssignmentSuraj'
ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % git add .

[ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % git commit -m "Forked suraj file"
On branch GitAssignmentSuraj
nothing to commit, working tree clean
ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % git push origin feature-branch
error: src refspec feature-branch does not match any
error: failed to push some refs to 'https://github.com/Ac-ayush/GitAssignmentSuraj'
ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % git add .

[ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % git commit -m "forked suraj file"
On branch GitAssignmentSuraj
nothing to commit, working tree clean
[ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % git push origin feature1
error: src refspec feature1 does not match any
error: failed to push some refs to 'https://github.com/Ac-ayush/GitAssignmentSuraj'
[ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % git push origin GitAssignmentSurajj
error: src refspec GitAssignmentSurajj does not match any
error: failed to push some refs to 'https://github.com/Ac-ayush/GitAssignmentSuraj'
[ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % git push origin GitAssignmentSuraj
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'GitAssignmentSuraj' on GitHub by visiting:
remote:     https://github.com/Ac-ayush/GitAssignmentSuraj/pull/new/GitAssignmentSuraj
remote:
To https://github.com/Ac-ayush/GitAssignmentSuraj
 * [new branch]      GitAssignmentSurajj -> GitAssignmentSuraj
[ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % touch test.txt
[ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % git add .
[ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % git commit -m "s"
[GitAssignmentSuraj 843a081] s
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test.txt
[ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % git push origin GitAssignmentSuraj
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 252 bytes | 252.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Ac-ayush/GitAssignmentSuraj
 7255b90..843a081 GitAssignmentSurajj -> GitAssignmentSuraj
ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj %
```

16. Comment on a PR and suggest improvements

The screenshot shows a GitHub pull request page. At the top, it displays the repository name 'ValluruSuraj / GitAssignment'. Below the title 'S #2', there's a green button labeled 'Open' and a message stating 'Ac-ayush wants to merge 1 commit into ValluruSuraj:main from Ac-ayush:GitAssignmentSuraj'. The pull request has 2 commits, 0 checks, and 1 file changed. Two comments from 'Ac-ayush' are visible:

- The first comment says 'No description provided.' and is timestamped 'commented now'.
- The second comment says 'I have added a new file' and is timestamped 'commented now'.

At the bottom of the pull request page, there's a note: 'Add more commits by pushing to the GitAssignmentSuraj branch on Ac-ayush/GitAssignmentSuraj.' A green box highlights the message 'This branch has no conflicts with the base branch'.

17. Create a Git alias for the command `git log --oneline` named `gitlol`.

```
ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % git config --global alias.gitlol "log --oneline"
ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % git config --get alias.gitlol
log --oneline
ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj % log --oneline
log: unrecognized option `--oneline'
usage:
  log <command>

global options:
  -?, --help
  -q, --quiet
  -v, --verbose

commands:
  collect      gather system logs into a log archive
  config       view/change logging system settings
  erase        delete system logging data
  show         view/search system logs
  stream       watch live system logs
  stats        show system logging statistics

further help:
  log help <command>
  log help predicates
ayushchouhan@Practos-MacBook-Pro GitAssignmentSuraj %
```

18. Create a pre-commit hook

19. You have made local changes in your branch, but you need to switch to another branch urgently without committing. How would you handle this situation?

20. You accidentally deleted a file in your local repository. How do you restore it using Git?

```
first commit
[ayushchouhan@Practos-MacBook-Pro GAssignment % git checkout tmp.txt
Updated 1 path from the index
ayushchouhan@Practos-MacBook-Pro GAssignment % ]
```

21. You have committed changes to your branch but forgot to include a file. How do you add the file to the last commit without creating a new commit?

```
Last login: Sun Jan 21 16:37:19 on ttys000
ayushchouhan@Practos-MacBook-Pro ~ % cd Desktop/GAssignment
ayushchouhan@Practos-MacBook-Pro GAssignment % git add eleven
fatal: pathspec 'eleven' did not match any files
ayushchouhan@Practos-MacBook-Pro GAssignment % touch tmp.txt
ayushchouhan@Practos-MacBook-Pro GAssignment % git add tmp.txt
ayushchouhan@Practos-MacBook-Pro GAssignment % git commit --amend

hint: Waiting for your editor to close the file... error: There was a problem with the editor 'vi'.
Please supply the message using either -m or -F option.
ayushchouhan@Practos-MacBook-Pro GAssignment % git commit --amend -m "Updated commit message if needed"

[conflict-branch 53c9420] Updated commit message if needed
Date: Sun Jan 21 17:29:45 2024 +0530
ayushchouhan@Practos-MacBook-Pro GAssignment % ]
```

22. You want to discard all changes in your working directory and revert to the last commit.

What Git command would you use?

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git reset --hard HEAD

HEAD is now at 53c9420 Updated commit message if needed
ayushchouhan@Practos-MacBook-Pro GAssignment % ]
```

23. You need to view a specific commit's changes. What Git command can be used to show the changes introduced by a particular commit?

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git show 53c94200a1fb1b863e3d6cd85a821b6f9fd2199d
commit 53c94200a1fb1b863e3d6cd85a821b6f9fd2199d (HEAD -> conflict-branch)
Merge: c5f2657 796e6b8
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 17:29:45 2024 +0530

        Updated commit message if needed

diff --cc random
index 000000,000000..e69de29
new file mode 100644
--- /dev/null
+++ b/random
diff --cc tmp.txt
index 000000,000000..e69de29
new file mode 100644
--- /dev/null
+++ b/tmp.txt
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

24. You want to change a commit message, after you have already committed, how do you do so?

```
Last login: Sun Jan 21 18:15:38 on ttys000
[ayushchouhan@Practos-MacBook-Pro ~ % cd Desktop/GAssignment
ayushchouhan@Practos-MacBook-Pro GAssignment % git commit --amend

hint: Waiting for your editor to close the file... error: There was a problem with the editor 'vi'.
Please supply the message using either -m or -F option.
ayushchouhan@Practos-MacBook-Pro GAssignment % git commit --amend -m "New and improved commit message"

[conflict-branch 91d21e0] New and improved commit message
  Date: Sun Jan 21 17:29:45 2024 +0530
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

25. Your colleague has made changes in their branch, and you want to incorporate those changes into your branch without merging. How do you achieve this?

```

ayushchouhan@Practos-MacBook-Pro GAssignment % git fetch origin
ayushchouhan@Practos-MacBook-Pro GAssignment % git fetch origin

[ayushchouhan@Practos-MacBook-Pro GAssignment % git log origin/feature1
commit ee68c6b2a0d1ab21a46d659a6cb89d41b8303976 (origin/main, origin/feature1)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 16:38:00 2024 +0530

    first commit
[ayushchouhan@Practos-MacBook-Pro GAssignment % git checkout feature1
Switched to branch 'feature1'
[ayushchouhan@Practos-MacBook-Pro GAssignment % git log
commit 796e6b82c5b8c41291b505b5d86f7c05f9733cec (HEAD -> feature1, main)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 17:13:59 2024 +0530

    merging

commit ee68c6b2a0d1ab21a46d659a6cb89d41b8303976 (origin/main, origin/feature1)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 16:38:00 2024 +0530

    first commit
[ayushchouhan@Practos-MacBook-Pro GAssignment % git cherry-pick 796e6b82c5b8c41291b505b5d86f7c05f9733cec
On branch feature1
You are currently cherry-picking commit 796e6b8.
(all conflicts fixed: run "git cherry-pick --continue"
(use "git cherry-pick --skip" to skip this patch)
(use "git cherry-pick --abort" to cancel the cherry-pick operation)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .DS_Store
    GitAssignmentSuraj/

nothing added to commit but untracked files present (use "git add" to track)
The previous cherry-pick is now empty, possibly due to conflict resolution.
If you wish to commit it anyway, use:

    git commit --allow-empty

Otherwise, please use 'git cherry-pick --skip'
[ayushchouhan@Practos-MacBook-Pro GAssignment % git cherry-pick origin/feature1
Auto-merging README.md
CONFLICT (add/add): Merge conflict in README.md
error: could not apply ee68c6b... first commit
hint: After resolving the conflicts, mark them with
hint: "git add/rm <pathspec>", then run
hint: "git cherry-pick --continue".
hint: You can instead skip this commit with "git cherry-pick --skip".
hint: To abort and get back to the state before "git cherry-pick",
hint: run "git cherry-pick --abort".
ayushchouhan@Practos-MacBook-Pro GAssignment % git cherry-pick --continue

U     README.md
error: Committing is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>''
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
ayushchouhan@Practos-MacBook-Pro GAssignment %

```

26. You've made several commits on a branch, but you want to club them into a single commit before pushing to the remote repository. How would you do that?

```

[ayushchouhan@Practos-MacBook-Pro GAssignment % git push origin feature1 --force

Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 308 bytes | 308.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Ac-ayush/GitAssignment.git
  ee68c6b..796e6b8  feature1 -> feature1
ayushchouhan@Practos-MacBook-Pro GAssignment %

```

27. You accidentally staged a file that you don't want to commit. How do you unstage it?

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git reset tmp2.txt
Unstaged changes after reset:
M README.md
ayushchouhan@Practos-MacBook-Pro GAssignment % git reset

Unstaged changes after reset:
M README.md
ayushchouhan@Practos-MacBook-Pro GAssignment % git status

On branch feature1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .DS_Store
    GitAssignmentSuraj/
    tmp2.txt

no changes added to commit (use "git add" and/or "git commit -a")
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

28. You don't want to commit files that have .yml in the end, and also files inside folder config. How do you do that?

```
ayushchouhan@Practos-MacBook-Pro GAssignment % git add .gitignore
git commit -m "Add .gitignore to ignore .yml files and files inside config folder"

[feature1 fd4b6a1] Add .gitignore to ignore .yml files and files inside config folder
 1 file changed, 5 insertions(+)
  create mode 100644 .gitignore
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

29. You want to see a list of all the files changed in the last commit. What Git command would you use?

```
ayushchouhan@Practos-MacBook-Pro GAssignment % git diff --name-only HEAD^ HEAD

.gitignore
ayushchouhan@Practos-MacBook-Pro GAssignment % git diff --name-only @^1 @

.gitignore
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

30. You realize that your local branch is outdated, and you want to fetch the latest changes from the remote repository. How do you do this without merging?

```
ayushchouhan@Practos-MacBook-Pro GAssignment % git fetch origin
git reset --hard origin/feature1
HEAD is now at 796e6b8 merging
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

31. You accidentally deleted a branch. How do you recover it?

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git reflog
796e6b8 (HEAD -> feature1, origin/feature1, main) HEAD@{0}: reset: moving to origin/feature1
fd4b6a1 HEAD@{1}: commit: Add .gitignore to ignore .yml files and files inside config folder
796e6b8 (HEAD -> feature1, origin/feature1, main) HEAD@{2}: reset: moving to HEAD
796e6b8 (HEAD -> feature1, origin/feature1, main) HEAD@{3}: reset: moving to HEAD
796e6b8 (HEAD -> feature1, origin/feature1, main) HEAD@{4}: checkout: moving from conflict-branch to feature1
91d21e0 (conflict-branch) HEAD@{5}: commit (amend): New and improved commit message
53c9420 HEAD@{6}: reset: moving to HEAD
53c9420 HEAD@{7}: commit (amend): Updated commit message if needed
983548a HEAD@{8}: commit (merge): eleven
c5f2657 HEAD@{9}: commit: conflict
026722a HEAD@{10}: checkout: moving from conflict-branch to conflict-branch
026722a HEAD@{11}: commit: lala
ee68c6b (origin/main) HEAD@{12}: checkout: moving from main to conflict-branch
796e6b8 (HEAD -> feature1, origin/feature1, main) HEAD@{13}: merge feature1: Fast-forward
ee68c6b (origin/main) HEAD@{14}: checkout: moving from feature1 to main
796e6b8 (HEAD -> feature1, origin/feature1, main) HEAD@{15}: commit: merging
ee68c6b (origin/main) HEAD@{16}: checkout: moving from main to feature1
ee68c6b (origin/main) HEAD@{17}: checkout: moving from main to main
ee68c6b (origin/main) HEAD@{18}: checkout: moving from main to main
ee68c6b (origin/main) HEAD@{19}: checkout: moving from feature1 to main
ee68c6b (origin/main) HEAD@{20}: checkout: moving from main to feature1
ee68c6b (origin/main) HEAD@{21}: checkout: moving from feature to main
ee68c6b (origin/main) HEAD@{22}: checkout: moving from main to feature
ee68c6b (origin/main) HEAD@{23}: Branch: renamed refs/heads/main to refs/heads/main
ee68c6b (origin/main) HEAD@{25}: commit (initial): first commit
[ayushchouhan@Practos-MacBook-Pro GAssignment % git branch feature2 796e6b8
[ayushchouhan@Practos-MacBook-Pro GAssignment % git checkout feature2
Switched to branch 'feature2'
ayushchouhan@Practos-MacBook-Pro GAssignment % git log
commit 796e6b82c5b8c41291b505b5d86f7c05f9733cec (HEAD -> feature2, origin/feature1, main, feature1)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 17:13:59 2024 +0530

    merging

commit ee68c6b2a0d1ab21a46d659a6cb89d41b8303976 (origin/main)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 16:38:00 2024 +0530

    first commit
[ayushchouhan@Practos-MacBook-Pro GAssignment % git push origin feature2
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature2' on GitHub by visiting:
remote:     https://github.com/Ac-ayush/GitAssignment/pull/new/feature2
remote:
To https://github.com/Ac-ayush/GitAssignment.git
 * [new branch]      feature2 -> feature2
ayushchouhan@Practos-MacBook-Pro GAssignment % ]
```

32. You want to remove untracked files and directories from your working directory. What Git command would you use?

```
* [new branch]      featurez -> featurez
ayushchouhan@Practos-MacBook-Pro GAssignment % git clean -fd

Removing .DS_Store
Removing tmp2.txt
ayushchouhan@Practos-MacBook-Pro GAssignment % git clean -fd

[ayushchouhan@Practos-MacBook-Pro GAssignment % git status
On branch feature2
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    GitAssignmentSuraj/

nothing added to commit but untracked files present (use "git add" to track)
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

33. You have a commit from a feature branch that you want to apply to the main branch without merging the entire feature branch.

```
Removing .DS_Store
Removing tmp2.txt
ayushchouhan@Practos-MacBook-Pro GAssignment % git status
On branch feature2
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    GitAssignmentSuraj/

nothing added to commit but untracked files present (use "git add" to track)
ayushchouhan@Practos-MacBook-Pro GAssignment % git log
commit 796e6b82c5b8c41291b5b5d86f7c05f9733cec (HEAD -> feature2, origin/feature2, origin/feature1, main, feature1)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:  Sun Jan 21 17:13:59 2024 +0530

  first commit
ayushchouhan@Practos-MacBook-Pro GAssignment % git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
ayushchouhan@Practos-MacBook-Pro GAssignment % git cherry-pick 796e6b82c5b8c41291b5b5d86f7c05f9733cec
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

You are currently cherry-picking commit 796e6b8.
  (all conflicts fixed: run "git cherry-pick --continue")
  (use "git cherry-pick --skip" to skip this patch)
  (use "git cherry-pick --abort" to cancel the cherry-pick operation)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    GitAssignmentSuraj/

nothing added to commit but untracked files present (use "git add" to track)
The previous cherry-pick is now empty, possibly due to conflict resolution.
If you wish to commit it anyway, use:

  git commit --allow-empty

Otherwise, please use 'git cherry-pick --skip'
ayushchouhan@Practos-MacBook-Pro GAssignment % git cherry-pick --continue

On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

You are currently cherry-picking commit 796e6b8.
  (all conflicts fixed: run "git cherry-pick --continue")
  (use "git cherry-pick --skip" to skip this patch)
  (use "git cherry-pick --abort" to cancel the cherry-pick operation)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    GitAssignmentSuraj/

nothing added to commit but untracked files present (use "git add" to track)
The previous cherry-pick is now empty, possibly due to conflict resolution.
If you wish to commit it anyway, use:

  git commit --allow-empty

Otherwise, please use 'git cherry-pick --skip'
ayushchouhan@Practos-MacBook-Pro GAssignment % git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Ac-ayush/GitAssignment.git
  ee68c6b..796e6b8  main -> main
```

34. You mistakenly committed a change to the wrong branch and need to apply that commit to the correct branch.

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git log
commit 796e6b2c5b8c41291b505b5d86f7c05f9733cec (HEAD, origin/main, origin/feature2, origin/feature1, main, feature2, feature1)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 17:13:59 2024 +0530

merging

commit ee68c6b2a0d1ab21a46d659a6cb89d41b8303976
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 16:38:00 2024 +0530

first commit
[ayushchouhan@Practos-MacBook-Pro GAssignment % git log
commit 796e6b2c5b8c41291b505b5d86f7c05f9733cec (HEAD, origin/main, origin/feature2, origin/feature1, main, feature2, feature1)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 17:13:59 2024 +0530

merging

commit ee68c6b2a0d1ab21a46d659a6cb89d41b8303976
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 16:38:00 2024 +0530

first commit
[ayushchouhan@Practos-MacBook-Pro GAssignment % git checkout feature2
Switched to branch 'feature2'
[ayushchouhan@Practos-MacBook-Pro GAssignment % git cherry-pick ee68c6b2a0d1ab21a46d659a6cb89d41b8303976
Auto-merging README.md
CONFLICT (add/add): Merge conflict in README.md
error: could not apply ee68c6b... first commit
hint: After resolving the conflicts, mark them with
hint: "git add/rm <pathspec>", then run
hint: "git cherry-pick --continue".
hint: You can instead skip this commit with "git cherry-pick --skip".
hint: To abort and get back to the state before "git cherry-pick",
hint: run "git cherry-pick --abort".
ayushchouhan@Practos-MacBook-Pro GAssignment % git cherry-pick --continue

U     README.md
error: Committing is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
[ayushchouhan@Practos-MacBook-Pro GAssignment % git push origin correct-branch
error: src refspec correct-branch does not match any
error: failed to push some refs to 'https://github.com/Ac-ayush/GitAssignment.git'
[ayushchouhan@Practos-MacBook-Pro GAssignment % git push origin feature1
Everything up-to-date
ayushchouhan@Practos-MacBook-Pro GAssignment % ]
```

35. There is a series of commits on a feature branch, but you only want to cherry-pick a specific range of commits.

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git log feature1
commit 796e6b82c5b8c41291b505b5d86f7c0f973cec (HEAD -> feature2, origin/main, origin/feature2, origin/feature1, main, feature1)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 17:13:59 2024 +0530

merging

commit ee68c6b2a0d1ab21a46d659a6cb89d41b8303976
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 16:38:00 2024 +0530

    first commit
[ayushchouhan@Practos-MacBook-Pro GAssignment % git checkout feature2
README.md: needs merge
error: you need to resolve your current index first
[ayushchouhan@Practos-MacBook-Pro GAssignment % git cherry-pick ee68c6b2a0d1ab21a46d659a6cb89d41b8303976
error: Cherry-picking is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'.
hint: as appropriate to mark resolution and make a commit.
fatal: cherry-pick failed
ayushchouhan@Practos-MacBook-Pro GAssignment % git cherry-pick --continue

U     README.md
error: Committing is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'.
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
ayushchouhan@Practos-MacBook-Pro GAssignment % git push origin target-branch

error: src refspec target-branch does not match any
error: failed to push some refs to 'https://github.com/Ac-ayush/GitAssignment.git'
ayushchouhan@Practos-MacBook-Pro GAssignment % git log feature-branch

fatal: ambiguous argument 'feature-branch': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
[ayushchouhan@Practos-MacBook-Pro GAssignment % git log feature1
commit 796e6b82c5b8c41291b505b5d86f7c0f973cec (HEAD -> feature2, origin/main, origin/feature2, origin/feature1, main, feature1)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 17:13:59 2024 +0530

merging

commit ee68c6b2a0d1ab21a46d659a6cb89d41b8303976
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 16:38:00 2024 +0530

    first commit
[ayushchouhan@Practos-MacBook-Pro GAssignment % git checkout feature2
README.md: needs merge
error: you need to resolve your current index first
[ayushchouhan@Practos-MacBook-Pro GAssignment % git checkout feature1
README.md: needs merge
error: you need to resolve your current index first
[ayushchouhan@Practos-MacBook-Pro GAssignment % git checkout feature2
README.md: needs merge
error: you need to resolve your current index first
[ayushchouhan@Practos-MacBook-Pro GAssignment % git checkout feature2
M     README.md
Already on 'feature2'
warning: cancelling a cherry picking in progress
[ayushchouhan@Practos-MacBook-Pro GAssignment % git cherry-pick ee68c6b2a0d1ab21a46d659a6cb89d41b8303976
error: your local changes would be overwritten by cherry-pick.
hint: commit your changes or stash them to proceed.
fatal: cherry-pick failed
ayushchouhan@Practos-MacBook-Pro GAssignment % git cherry-pick --continue

error: no cherry-pick or revert in progress
fatal: cherry-pick failed
[ayushchouhan@Practos-MacBook-Pro GAssignment % git push origin feature2
Everything up-to-date
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

36. You want to clone a GitHub repository onto your local machine, but you only need a specific branch. How can you achieve this?

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git clone -b branch_name --single-branch https://github.com/ValluruSuraj/GitAssignment
Cloning into 'GitAssignment'...
warning: Could not find remote branch branch_name to clone.
fatal: Remote branch branch_name not found in upstream origin
[ayushchouhan@Practos-MacBook-Pro GAssignment % git clone -b main --single-branch https://github.com/ValluruSuraj/GitAssignment.git
Cloning into 'GitAssignment'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0
Receiving objects: 100% (7/7), 2.62 MB | 7.79 MiB/s, done.
[ayushchouhan@Practos-MacBook-Pro GAssignment % ]
```

37. You've made changes to your local repository and want to push them to your fork on GitHub. What Git commands would you use?

```
ayushchouhan@Practos-MacBook-Pro GAssignment % git add .
git commit -m "Happy Weekende"

warning: adding embedded git repository: GitAssignment
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> GitAssignment
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached GitAssignment
hint: See "git help submodule" for more information.
warning: adding embedded git repository: GitAssignmentSuraj
[feature2 258e44f] Happy Weekende
  4 files changed, 7 insertions(+), 5 deletions(-)
  create mode 100644 .gitignore
  create mode 160000 GitAssignment
  create mode 160000 GitAssignmentSuraj
ayushchouhan@Practos-MacBook-Pro GAssignment % git push origin feature1
Everything up-to-date
ayushchouhan@Practos-MacBook-Pro GAssignment % git push origin main

Everything up-to-date
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

38. You want to create a new branch both locally and on GitHub to work on a new feature.

What commands would you use?

```
Everything up-to-date
ayushchouhan@Practos-MacBook-Pro GAssignment % git checkout -b feature1
fatal: a branch named 'feature1' already exists
ayushchouhan@Practos-MacBook-Pro GAssignment % git push origin feature1
Everything up-to-date
ayushchouhan@Practos-MacBook-Pro GAssignment %
```

39. You want to see the commit history of a GitHub repository. How can you do this using Git commands?

```

ayushchouhan@Practos-MacBook-Pro GAssignment % git log

commit 258e44fea0d0b89605b15e94fc11060cffee9db5 (HEAD -> feature2)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 20:16:57 2024 +0530

    HAppy Weekende

commit 796e6b82c5b8c41291b505b5d86f7c05f9733cec (origin/main, origin/feature2, origin/feature1, main, feature1)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 17:13:59 2024 +0530

    merging

commit ee68c6b2a0d1ab21a46d659a6cb89d41b8303976
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 16:38:00 2024 +0530

    first commit
ayushchouhan@Practos-MacBook-Pro GAssignment % git log --oneline

258e44f (HEAD -> feature2) HAppy Weekende
796e6b8 (origin/main, origin/feature2, origin/feature1, main, feature1) merging
ee68c6b first commit
ayushchouhan@Practos-MacBook-Pro GAssignment % git fetch origin
|git log origin/feature1

commit 796e6b82c5b8c41291b505b5d86f7c05f9733cec (origin/main, origin/feature2, origin/feature1, main, feature1)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 17:13:59 2024 +0530

    merging

commit ee68c6b2a0d1ab21a46d659a6cb89d41b8303976
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 16:38:00 2024 +0530

    first commit
|ayushchouhan@Practos-MacBook-Pro GAssignment % git clone https://github.com/ValluruSuraj/GitAssignment
fatal: destination path 'GitAssignment' already exists and is not an empty directory.
ayushchouhan@Practos-MacBook-Pro GAssignment %

```

40. You've accidentally committed sensitive information and want to remove the commit from both your local and remote repositories on GitHub. What commands would you use?

```

ayushchouhan@Practos-MacBook-Pro GAssignment % git log

commit 258e44fea0d0b89605b15e94fc11060cffee9db5 (HEAD -> feature2)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 20:16:57 2024 +0530

    HAppy Weekende

commit 796e6b82c5b8c41291b505b5d86f7c05f9733cec (origin/main, origin/feature2, origin/feature1, main, feature1)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 17:13:59 2024 +0530

    merging

commit ee68c6b2a0d1ab21a46d659a6cb89d41b8303976
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 16:38:00 2024 +0530

    first commit
ayushchouhan@Practos-MacBook-Pro GAssignment % git log --oneline

258e44f (HEAD -> feature2) HAppy Weekende
796e6b8 (origin/main, origin/feature2, origin/feature1, main, feature1) merging
ee68c6b first commit
ayushchouhan@Practos-MacBook-Pro GAssignment % git fetch origin
git log origin/feature1

commit 796e6b82c5b8c41291b505b5d86f7c05f9733cec (origin/main, origin/feature2, origin/feature1, main, feature1)
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 17:13:59 2024 +0530

    merging

commit ee68c6b2a0d1ab21a46d659a6cb89d41b8303976
Author: Ayush Chouhan <ayush9977@gmail.com>
Date:   Sun Jan 21 16:38:00 2024 +0530

    first commit
[ayushchouhan@Practos-MacBook-Pro GAssignment % git clone https://github.com/ValluruSuraj/GitAssignment
fatal: destination path 'GitAssignment' already exists and is not an empty directory.
[ayushchouhan@Practos-MacBook-Pro GAssignment % git log
quote> git branch backup-branch

quote> git branch feature2
quote> git rebase -i ee68c6b2a0d1ab21a46d659a6cb89d41b8303976
quote> git push origin feature1 --force

quote> git push origin feature1 --force
quote>
quote>
```

41. You want to delete a remote branch on GitHub. What Git command would you use?

```

quote>
[ayushchouhan@Practos-MacBook-Pro GAssignment %
[ayushchouhan@Practos-MacBook-Pro GAssignment % git push origin --delete feature2
To https://github.com/Ac-ayush/GitAssignment.git
 - [deleted]          feature2
[ayushchouhan@Practos-MacBook-Pro GAssignment % git push origin --delete feature1
To https://github.com/Ac-ayush/GitAssignment.git
 - [deleted]          feature1
ayushchouhan@Practos-MacBook-Pro GAssignment % █
```

42. Create a git repository for all your assignments and upload them in it. Ask your peers to code review it, and you need to code review your peers assignments

```
[ayushchouhan@Practos-MacBook-Pro GAssignment % git clone https://github.com/Ac-ayush/Practo_Git_Assignment
Cloning into 'Practo_Git_Assignment'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
ayushchouhan@Practos-MacBook-Pro GAssignment % git add .
git commit -m "Initial commit with assignments"
git push origin main

warning: adding embedded git repository: Practo_Git_Assignment
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> Practo_Git_Assignment
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached Practo_Git_Assignment
hint:
hint: See "git help submodule" for more information.
[feature2 b513a4a] Initial commit with assignments
 1 file changed, 1 insertion(+)
 create mode 160000 Practo_Git_Assignment
Everything up-to-date
ayushchouhan@Practos-MacBook-Pro GAssignment % git add .
git commit -m "Address code review feedback"
git push origin main

On branch feature2
nothing to commit, working tree clean
Everything up-to-date
ayushchouhan@Practos-MacBook-Pro GAssignment % █
```