

Name: Vaishnavi Kahar

1. Set your username and email in git config  
git config --global user.name "vaish22"  
git config --global user.email "vaishnavi.kahar@practo.com"
2. Create a new branch named "feature-branch" and switch to it.  
git checkout -b feature-branch
3. List all branches in the repository.  
git branch
4. Delete the branch "feature-branch"  
git checkout master  
git branch -D feature-branch
5. How do you undo the last commit  
git revert HEAD

```
Desktop Downloads Movies Pictures
vaishnavihemantkahar@practos-MacBook-Pro ~ % cd Documents
vaishnavihemantkahar@practos-MacBook-Pro Documents % cd GitAssignment
vaishnavihemantkahar@practos-MacBook-Pro GitAssignment % git config --global user.name "vaish22"
vaishnavihemantkahar@practos-MacBook-Pro GitAssignment % git config --global user.email "vaishnavi.kahar@practo.com"
vaishnavihemantkahar@practos-MacBook-Pro GitAssignment % git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /Users/vaishnavihemantkahar/Documents/GitAssignment/.git/
vaishnavihemantkahar@practos-MacBook-Pro GitAssignment % git branch
vaishnavihemantkahar@practos-MacBook-Pro GitAssignment % git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    File 1.txt

vaishnavihemantkahar@practos-MacBook-Pro GitAssignment % git add .
vaishnavihemantkahar@practos-MacBook-Pro GitAssignment % git commit -m "initial commit"
[master (root-commit) 8d309b4] initial commit
1 file changed, 1 insertion(+)
create mode 100644 File 1.txt
vaishnavihemantkahar@practos-MacBook-Pro GitAssignment % git checkout -b feature-branch
Switched to a new branch 'feature-branch'
vaishnavihemantkahar@practos-MacBook-Pro GitAssignment % git branch
* feature-branch
master
vaishnavihemantkahar@practos-MacBook-Pro GitAssignment % git branch -d feature-branch
error: cannot delete branch 'feature-branch' used by worktree at '/Users/vaishnavihemantkahar/Documents/GitAssignment'
vaishnavihemantkahar@practos-MacBook-Pro GitAssignment % git log
commit 8d309b425c9f25ed5804ee956cae688cdc19b251 (HEAD -> feature-branch, master)
Author: vaish22 <vaishnavi.kahar@practo.com>
Date: Wed Jun 26 00:10:15 2024 +0530

    initial commit
vaishnavihemantkahar@practos-MacBook-Pro GitAssignment % git revert HEAD
hint: Waiting for your editor to close the file... [feature-branch d4aecd9] Revert "initial commit"
vaishnavihemantkahar@practos-MacBook-Pro GitAssignment %
```

6. Create a new branch names "conflict-branch"  
git checkout -b conflict-branch
7. Create a another branch named "feature1"  
git checkout -b feature1

8. Make some changes in to feature1 branch  
(made changes now committing them)  
git add .  
git commit -m "Made some changes in feature1"

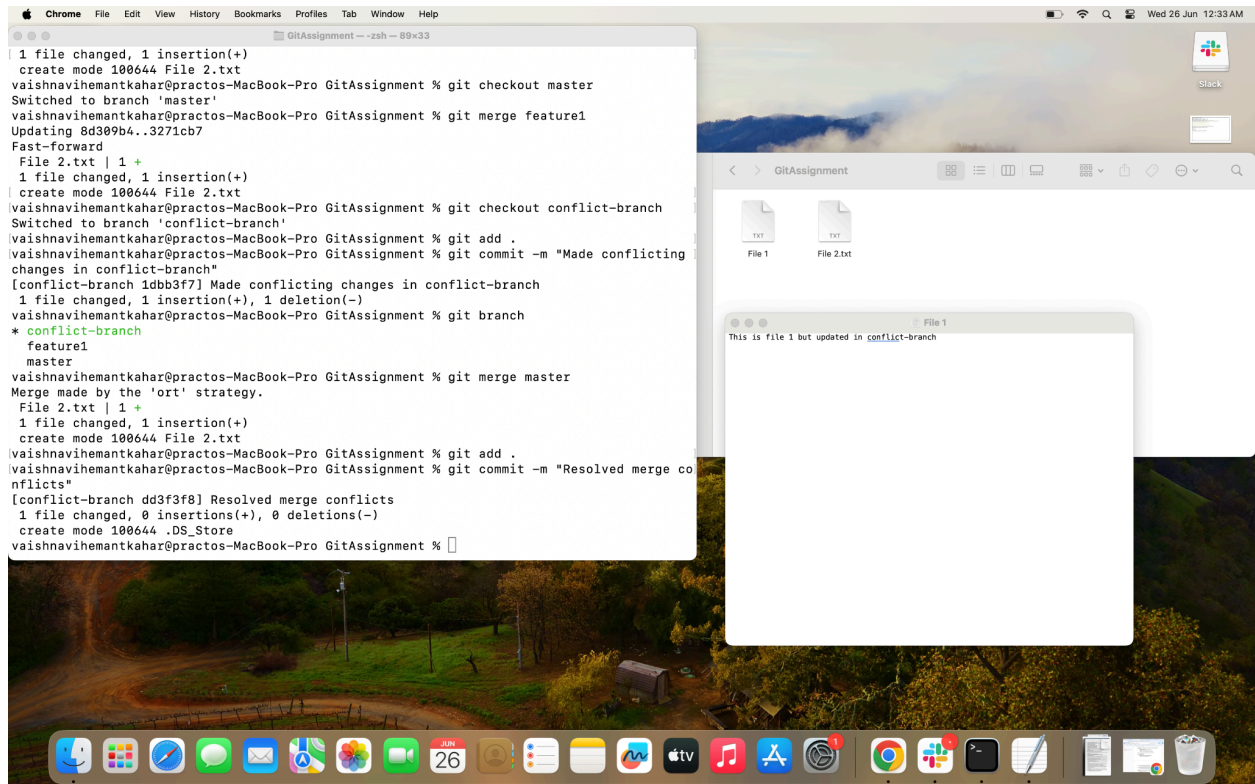
9. Merge "feature1" branch into main branch  
git checkout main  
git merge feature1

```
Untracked files:
| (use "git add <file>..." to include in what will be committed)
|   File 1.txt
|
nothing added to commit but untracked files present (use "git add" to track)
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % git add .
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % git commit -m "File 1 added"
[feature-branch 6d1dc9e] File 1 added
 1 file changed, 1 insertion(+)
| create mode 100644 File 1.txt
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % git branch
* feature-branch
  master
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % git branch -d feature-branch
error: cannot delete branch 'feature-branch' used by worktree at '/Users/vaishnavihemantkakar/Documents/GitAssignment'
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % git checkout master
Switched to branch 'master'
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % git branch -d feature-branch
error: the branch 'feature-branch' is not fully merged
hint: If you are sure you want to delete it, run 'git branch -D feature-branch'
hint: Disable this message with 'git config advice.forceDeleteBranch false'
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % git branch -D feature-branch
Deleted branch feature-branch (was 6d1dc9e).
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % git branch
* master
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % git checkout -b conflict-branch
Switched to a new branch 'conflict-branch'
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % git checkout -b feature1
Switched to a new branch 'feature1'
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % git status
On branch feature1
Untracked files:
| (use "git add <file>..." to include in what will be committed)
|   File 2.txt
|
nothing added to commit but untracked files present (use "git add" to track)
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % git add .
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % git commit -m "made some changes in feature1 branch"
[feature1 3271cb7] made some changes in feature1 branch
 1 file changed, 1 insertion(+)
| create mode 100644 File 2.txt
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % git checkout master
Switched to branch 'master'
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % git merge feature1
Updating 8d309b4..3271cb7
Fast-forward
 File 2.txt | 1 +
 1 file changed, 1 insertion(+)
| create mode 100644 File 2.txt
vaishnavihemantkakar@practos-MacBook-Pro GitAssignment % █
```

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

git checkout conflict-branch  
(made conflicting changes to the same file and line)  
git add .  
git commit -m "Made conflicting changes in conflict-branch"

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



## 12. Resolve merge conflicts

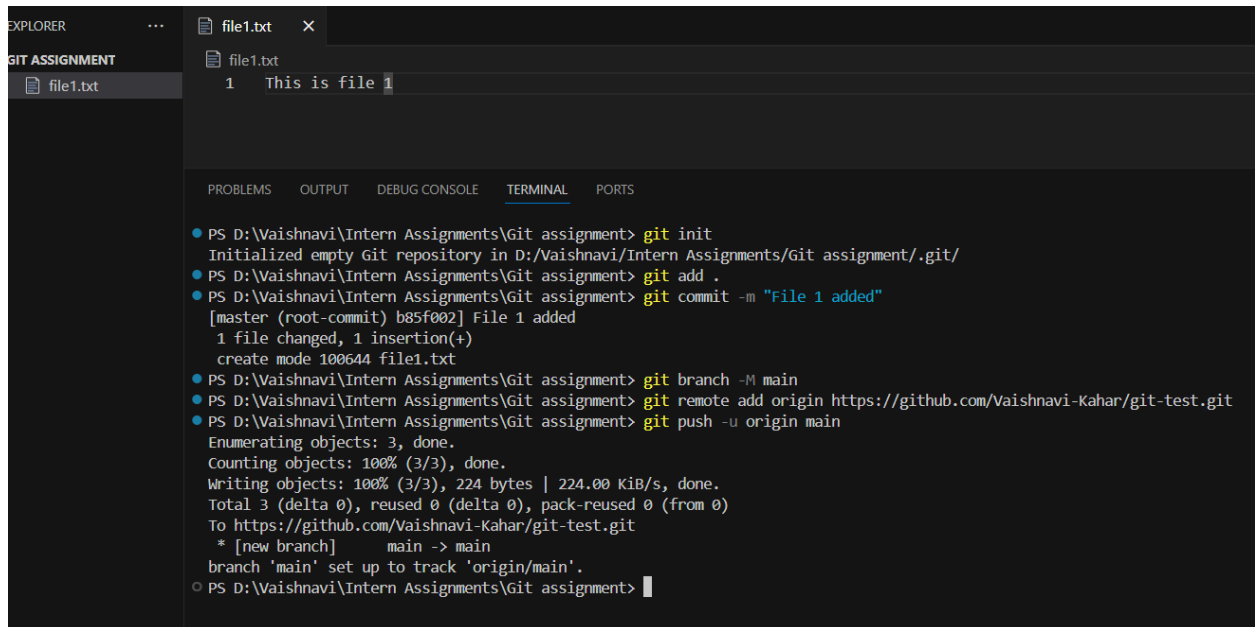
(Opened the conflicting files, resolved conflicts, then)

git add .

git commit -m "Resolved merge conflicts"

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

git remote add origin <url to github repo>



```
EXPLORER
...
GIT ASSIGNMENT
file1.txt
1 This is file 1

TERMINAL
PS D:\Vaishnavi\Intern Assignments\Git assignment> git init
Initialized empty Git repository in D:/Vaishnavi/Intern Assignments/Git assignment/.git/
PS D:\Vaishnavi\Intern Assignments\Git assignment> git add .
PS D:\Vaishnavi\Intern Assignments\Git assignment> git commit -m "File 1 added"
[master (root-commit) b85f002] File 1 added
1 file changed, 1 insertion(+)
create mode 100644 file1.txt
PS D:\Vaishnavi\Intern Assignments\Git assignment> git branch -M main
PS D:\Vaishnavi\Intern Assignments\Git assignment> git remote add origin https://github.com/Vaishnavi-Kahar/git-test.git
PS D:\Vaishnavi\Intern Assignments\Git assignment> git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 224 bytes | 224.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Vaishnavi-Kahar/git-test.git
* [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS D:\Vaishnavi\Intern Assignments\Git assignment>
```

14. Fork a repository on GitHub and clone it to your local machine.  
git clone <url to github repo>

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

git checkout -b new-branch

# Make changes, then:

git add .

git commit -m "File added"

git push origin new-branch

# Then open a pull request on GitHub from new-branch to the original repository

```
Dummy x
New > Dummy
1 dummy updated
2



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
zsh - New + - - - -

• vaishnavihemantkakar@practos-MacBook-Pro Cloning % git clone https://github.com/Vaishnavi-Kahar/New.git
Cloning into 'New'...
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.
• vaishnavihemantkakar@practos-MacBook-Pro Cloning % git branch
fatal: not a git repository (or any of the parent directories): .git
• vaishnavihemantkakar@practos-MacBook-Pro Cloning % cd New
• vaishnavihemantkakar@practos-MacBook-Pro New % git status
On branch main
Your branch is up to date with 'origin/main'.


nothing to commit, working tree clean
• vaishnavihemantkakar@practos-MacBook-Pro New % git branch
* main
• vaishnavihemantkakar@practos-MacBook-Pro New % git checkout -b new-branch
Switched to a new branch 'new-branch'
• vaishnavihemantkakar@practos-MacBook-Pro New % git add .
• vaishnavihemantkakar@practos-MacBook-Pro New % git commit -m "Dummy File changed"
[new-branch f6e2832] Dummy File changed
1 file changed, 1 insertion(+), 1 deletion(-)
• vaishnavihemantkakar@practos-MacBook-Pro New % git push origin new-branch
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 279 bytes | 279.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'new-branch' on GitHub by visiting:
remote:   https://github.com/Vaishnavi-Kahar/New/pull/new/new-branch
remote:
To https://github.com/Vaishnavi-Kahar/New.git
* [new branch]    new-branch -> new-branch
• vaishnavihemantkakar@practos-MacBook-Pro New %
```

16. Comment on a PR and suggest improvements


# Dummy File changed #1



 Open Vaishnavi-Kahar wants to merge 1 commit into [Bhargav1603:main](#) from [Vaishnavi-Kahar:new-branch](#) 

Conversation 0 Commits 1 Checks 0 Files changed 1


 **Vaishnavi-Kahar** commented now


No description provided.














  Dummy File changed f6e2832



Add more commits by pushing to the [new-branch](#) branch on [Vaishnavi-Kahar/New](#).


 **This branch has no conflicts with the base branch**  
Only those with [write access](#) to this repository can merge pull requests.

 **Add a comment**

Write Preview H B I           

Dummy file changed

 Markdown is supported  Paste, drop, or click to add files

 Close with comment **Comment**

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

18. Create a pre-commit hook  
`cd .git/hooks`  
`touch pre-commit`  
`chmod +x pre-commit`  
dummy scripts added in pre-commit

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?  
`git stash`  
`git checkout other-branch`  
# When you come back to your branch:

```
git checkout your-branch
git stash pop
```

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

```
git checkout -- path/to/your/file
```

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?

```
git add forgotten-file
git commit --amend
git push --force
```

22. You want to discard all changes in your working directory and revert to the last commit. What Git command would you use?

```
git reset --hard HEAD
```

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

```
git show <commit-hash>
```

```
● vaishnavihemantkakar@practos-MacBook-Pro New % git show f6e2832ac9b3001a0629751f4a2ba70e89f4f1be
commit f6e2832ac9b3001a0629751f4a2ba70e89f4f1be (HEAD -> new-branch, origin/new-branch)
Author: "vaish22" <vaishnavi.kakar@practo.com>
Date:   Wed Jun 26 12:38:10 2024 +0530
```

```
    Dummy File changed
```

```
diff --git a/Dummy b/Dummy
index 46b7291..120d3f5 100644
--- a/Dummy
+++ b/Dummy
@@ -1,2 +1,2 @@
-dummy
+dummy updated
```

```
○ vaishnavihemantkakar@practos-MacBook-Pro New %
```

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

```
git commit --amend -m "New commit message"
git push --force
```

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?

git checkout colleague-branch

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?

```
git rebase -i HEAD~n # n is the number of commits to combine
# Mark commits as "squash" or "fixup", then:
git push --force
```

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

```
git reset HEAD path/to/file
```

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?

```
echo "*.yml" >> .gitignore
echo "config/" >> .gitignore
git add .gitignore
git commit -m "Add .yml and config folder to .gitignore"
```

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

```
git diff --name-only HEAD~1 HEAD
```

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?

```
git fetch
```

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

```
git checkout -b deleted-branch-name <commit-hash>
```

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

```
git clean -fd
```

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



```
Git log feature-branch
git checkout main
git cherry-pick <commit-hash>
```

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

```
git checkout wrong-branch
git log # Get the commit hash of the wrong commit
git checkout correct-branch
git cherry-pick <commit-hash>
git checkout wrong-branch
git reset --hard HEAD~1 # Remove the wrong commit from the wrong branch
```

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

```
git cherry-pick <start-commit-hash>^..<end-commit-hash>
```

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

```
git clone --branch branch-name --single-branch
https://github.com/yourusername/repository.git
```

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?

```
git add .
git commit -m "Your commit message"
git push origin your-branch
```

38. You want to create a new branch both locally and on GitHub to work on a new feature. What commands would you use?

```
git checkout -b new-feature-branch
git push origin new-feature-branch
```

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

```
git log
```

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?

```
git log # Find the commit hash of the sensitive commit
git rebase -i <commit-hash>~1
# Mark the sensitive commit as "drop"
git push --force
```

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

```
git push origin --delete branch-name
```

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

```
git init
git remote add origin https://github.com/yourusername/assignments.git
git add .
git commit -m "Initial commit"
git push -u origin main
```

43. Create a pull request on any open source library on github, attach the pull request link to the readme file of this project's repository