

Git – Git Real Question Answers

Note that these aren't exactly challenges, but are closer to questions.

Introduction (Level 1)

01. How would you get help on the add command?

Answer: `git help add`

02. How would you configure your user name across all repositories on your machine?

Answer: `git config --global user.name "Your Name"`

03. How would you add all files in a css subdirectory in your repository?

Answer: `git add css/*`

Staging & Remotes (Level 2)

04. How would you see what has changed since your last commit?

Answer: `git diff`

05. How would you unstage the file fruit.html from the staging area?

Answer: `git reset --soft fruit.html`

06. How would you commit the people.html file, but skip the staging process?

Answer: `git commit -a -m "Commit Message"`

07. How would you amend the dog.html file to an existing commit?

Answer: `git add dog.html` then `git commit --amend -m "Amended"`

08. How would you undo the last commit, and put the files back into staging?

Answer: `git reset --soft HEAD^`

09. How would you discard changes to birds.html and people.html?

Answer: `git checkout -- birds.html people.html`

10. How would you remove the most recent commit and all its changes?

Answer: `git reset --hard HEAD^`

11. How would you add a remote named "origin" at git@example.com:example/petshop.git?

Answer: `git remote add origin git@example.com:example/petshop.git`

12. How would you push from local "master" branch to remote "origin" branch?

Answer: `git push -u origin master`

Cloning & Branching (Level 3)

13. How would you clone a repository from git@example.com:example/grocerystory.git?

Answer: `git clone git@example.com:example/grocerystory.git`

14. How would you get a list of your remotes with a verbose output?
Answer: `git remote -v`
15. How would you create a new branch called “stock”?
Answer: `git branch stock`
16. How would you switch to your newly created “stock” branch?
Answer: `git checkout stock`
17. How would you merge the “stock” branch into the “master” branch?
Answer: `git checkout master` then `git merge stock`
18. How would you create and checkout the “deli” branch in a single command?
Answer: `git checkout -b deli`

Collaboration Basics (Level 4)

19. How would you push your local commits to the remote, knowing your local is not up to date with the remote?
Answer: `git pull` then `git push`
You'll need to update your local repository with the state of the remote repository before you can push to the remote.
20. How would you resolve the following merge conflict to keep your changes, and disregard the other version?
- ```
<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="UTF-8">
 <title>Our Cat-alog</title>
 </head>
 <body>
 <nav>

 Cats
 Dogs

 </nav>
 </body>
</html>
```

### **Branching (Level 5)**

21. How would you push your local branch “fruits” to the origin?  
**Answer:** `git push origin fruits`
22. How would you retrieve the remote “vegetable” branch?  
**Answer:** `git fetch`
23. How would you get a list of remote branches?  
**Answer:** `git branch -r`

24. How would you delete the “fruits” branch on “origin”?  
**Answer:** `git push origin :fruits`
25. How would you check for stale branches that are tracking “origin”?  
**Answer:** `git remote show origin`
26. How do you clean up your local references to now-deleted remote branches?  
**Answer:** `git remote prune origin`
27. How would you list the tags in a repository?  
**Answer:** `git tag`
28. How would you create a tag for “v1.4.7”?  
**Answer:** `git tag -a "v1.4.7" -m "new tag"`
29. How would you push your tags to origin?  
**Answer:** `git push --tag`
30. How would you release the tagged “v1.8.9”?  
**Answer:** `git checkout "v1.8.9"`

## **Rebase Belong to Us (Level 6)**

31. How would you rebase the “bread” branch on “master”?  
**Answer:** `git checkout bread` then `git rebase master`  
You’ll need to checkout the branch you want to rebase before you can do it
32. After successfully rebasing the “bread” branch on “master”, how would you merge the “bread” branch into “master”?  
**Answer:** `git checkout master` then `git merge bread`  
Again, you’ll need to switch to master before you can merge into it
33. How would you retrieve changes from origin without merging them into local?  
**Answer:** `git fetch`
34. Now that you know the changes from origin/master, how would you move your local master commits after the commits from origin/master?  
**Answer:** `git rebase`

35. How would you resolve the conflict that occurred during a rebase to keep the version with cats and dogs?

```
<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="UTF-8">
 <title>Our Cat-a-log</title>
 </head>
 <body>
 <nav>

 Cats
 Dogs

 </nav>
 </body>
</html>
```

36. How would you mark the conflicts in “index.html” as resolved, and then continue the rebase that is in process?

**Answer:** `git add index.html` then `git rebase --continue`

## **History and Configuration (Level 7)**

37. How would you view the log with one commit per line?

**Answer:** `git log --oneline`

38. How would you see a summary of file changes?

**Answer:** `git diff`

39. How would you compare your “master” branch with your “forest” branch?

**Answer:** `git diff master forest`

40. How would you get a diff that includes the previous commit, as well as its parent?

**Answer:** `git diff HEAD~2`

41. How would you see the diff along with the log?

**Answer:** `git log -p`

42. How would you see who made what commits on the “index.html” file?

**Answer:** `git blame index.html`

43. What would you write in your .gitignore file to make git ignore all the files in the “logs” directory whose names end in .log?

**Answer:** `logs/*.log`

44. How would you set the user.email to admin@example.com for only the repo you’re working in?

**Answer:** `git config user.email admin@example.com`

45. How would you set the user.name to your name for all Git repos?

**Answer:** `git config --global user.name “Your Name”`

46. How would you alias “git commit” to “git beholdmycodeyoupeasants”?

**Answer:** `git config alias.beholdmycodeyoupeasants commit`