

Git – Git Real Questions

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?
02. How would you configure your user name across all repositories on your machine?
03. How would you add all files in a css subdirectory in your repository?

Staging & Remotes (Level 2)

04. How would you see what has changed since your last commit?
05. How would you unstage the file fruit.html from the staging area?
06. How would you commit the people.html file, but skip the staging process?
07. How would you amend the dog.html file to an existing commit?
08. How would you undo the last commit, and put the files back into staging?
09. How would you discard changes to birds.html and people.html?
10. How would you remove the most recent commit and all its changes?
11. How would you add a remote named "origin" at git@example.com:example/petshop.git?
12. How would you push from local "master" branch to remote "origin" branch?

Cloning & Branching (Level 3)

13. How would you clone a repository from git@example.com:example/grocerystory.git?
14. How would you get a list of your remotes with a verbose output?
15. How would you create a new branch called "stock"?
16. How would you switch to your newly created "stock" branch?
17. How would you merge the "stock" branch into the "master" branch?
18. How would you create and checkout the "deli" branch in a single command?

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?
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>
      <ul>
        <<<<<< HEAD
          <li><a href="cat.html">Cats</a></li>
          <li><a href="dog.html">Dogs</a></li>
        =====
          <li><a href="cat.html">Felines</a></li>
          <li><a href="dog.html">Canines</a></li>
        >>>>>> 6a487f9eb0e0a5110bdf2a45a4f5dbcc3d4eec53
      </ul>
    </nav>
  </body>
</html>
```

Branching (Level 5)

21. How would you push your local branch “fruits” to the origin?
22. How would you retrieve the remote “vegetable” branch?
23. How would you get a list of remote branches?
24. How would you delete the “fruits” branch on “origin”?
25. How would you check for stale branches that are tracking “origin”?
26. How do you clean up your local references to now-deleted remote branches?
27. How would you list the tags in a repository?
28. How would you create a tag for “v1.4.7”?
29. How would you push your tags to origin?
30. How would you release the tagged “v1.8.9”?

Rebase Belong to Us (Level 6)

31. How would you rebase the “breads” branch on “master”?

32. After successfully rebasing the “breads” branch on “master”, how would you merge the “breads” branch into “master”?
33. How would you retrieve changes from origin without merging them into local?
34. Now that you know the changes from origin/master, how would you move your local master commits after the commits from origin/master?
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-alog</title>
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="cat.html">Cats</a></li>
        <li><a href="dog.html">Dogs</a></li>
      </ul>
    </nav>
  </body>
</html>
```

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

History and Configuration (Level 7)

37. How would you view the log with one commit per line?
38. How would you see a summary of file changes?
39. How would you compare your “master” branch with you “forest” branch?
40. How would you get a diff that includes the previous commit, as well as its parent?
41. How would you see the diff along with the log?
42. How would you see who made what commits on the “index.html” file?
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?
44. How would you set the user.email to admin@example.com for only the repo you’re working in?

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

46. How would you alias "git commit" to "git beholdmycodeyoupeasants"?