

100+ GIT Interview Questions and Answers for developers

Table of Contents

GIT Interview Questions and Answers

1. How can we see n most recent commits in GIT?
2. How can we know if a branch is already merged into master in GIT?
3. What is the purpose of git stash drop?
4. What is the HEAD in GIT?
5. What is the most popular branching strategy in GIT?
6. What is SubGit?
7. What is the use of git instaweb?
8. What are git hooks?
9. What is GIT?
10. What is a repository in GIT?
11. What are the main benefits of GIT?
12. What are the disadvantages of GIT?
13. What are the main differences between GIT and SVN?
14. How will you start GIT for your project?
15. What is git clone in GIT?
16. How will you create a repository in GIT?
17. What are the different ways to start work in GIT?
18. GIT is written in which language?
19. What does 'git pull' command in GIT do internally?
20. What does 'git push' command in GIT do internally?
21. What is git stash?
22. What is the meaning of 'stage' in GIT?
23. What is the purpose of git config command?
24. How can we see the configuration settings of GIT installation?
25. How will you write a message with commit command in GIT?
26. What is stored inside a commit object in GIT?

27. How many heads can you create in a GIT repository?
28. Why do we create branches in GIT?
29. What are the different kinds of branches that can be created in GIT?
30. How will you create a new branch in GIT?
31. How will you add a new feature to the main branch?
32. What is a pull request in GIT?
33. What is merge conflict in GIT?
34. How can we resolve a merge conflict in GIT?
35. What command will you use to delete a branch?
36. What command will you use to delete a branch that has unmerged changes?
37. What is the alternative command to merging in GIT?
38. What is Rebasing in GIT?
39. What is the 'Golden Rule of Rebasing' in GIT?
40. Why do we use Interactive Rebasing in place of Auto Rebasing?
41. What is the command for Rebasing in Git?
42. What is the main difference between git clone and git remote?
43. What is GIT version control?
44. What GUI do you use for working on GIT?
45. What is the use of git diff command in GIT?
46. What is git rerere?
47. What are the three most popular version of git diff command?
48. What is the use of git status command?
49. What is the main difference between git diff and git status?
50. What is the use of git rm command in GIT?
51. What is the command to apply a stash?
52. Why do we use git log command?
53. Why do we need git add command in GIT?
54. Why do we use git reset command?
55. What does a commit object contain?

56. How can we convert git log messages to a different format?
57. What are the programming languages in which git hooks can be written?
58. What is a commit message in GIT?
59. How GIT protects the code in a repository?
60. How GIT provides flexibility in version control?
61. How can we change a commit message in GIT?
62. Why is it advisable to create an additional commit instead of amending an existing commit?
63. What is a bare repository in GIT?
64. How do we put a local repository on GitHub server?
65. How will you delete a branch in GIT?
66. How can we set up a Git repository to run code sanity checks and UAT tests just before a commit?
67. How can we revert a commit that was pushed earlier and is public now?
68. In GIT, how will you compress last n commits into a single commit?
69. How will you switch from one branch to a new branch in GIT?
70. How can we clean unwanted files from our working directory in GIT?
71. What is the purpose of git tag command?
72. What is cherry-pick in GIT?
73. What is shortlog in GIT?
74. How can you find the names of files that were changed in a specific commit?
75. How can we attach an automated script to run on the event of a new commit by push command?
76. What is the difference between pre-receive, update and post-receive hooks in GIT?
77. Do we have to store Scripts for GIT hooks within same repository?
78. How can we determine the commit that is the source of a bug in GIT?
79. How can we see differences between two commits in GIT?
80. What are the different ways to identify a commit in GIT?
81. When we run git branch , how does GIT know the SHA-1 of the last commit?
82. What are the different types of Tags you can create in GIT?
83. How can we rename a remote repository?
84. Some people use git checkout and some use git co for checkout. How is that possible?

85. How can we see the last commit on each of our branch in GIT?
86. Is origin a special branch in GIT?
87. How can we configure GIT to not ask for password every time?
88. What are the four major protocols used by GIT for data transfer?
89. What is GIT protocol?
90. How can we work on a project where we do not have push access?
91. What is git grep?
92. How can you reorder commits in GIT?
93. How will you split a commit into multiple commits?
94. What is filter-branch in GIT?
95. What are the three main trees maintained by GIT?
96. What are the three main steps of working GIT?
97. What are ours and theirs merge options in GIT?
98. How can we ignore merge conflicts due to Whitespace?
99. What is git blame?
100. What is a submodule in GIT?