**Top 20 Git Commands With Examples**

### 1)git config

Usage: git config –global user.name “[name]”

Usage: git config –global user.email “[email address]”

This command sets the author name and email address respectively to be used with your commits.

Git Config Command - Git Commands - Edureka

### 2)git init

Usage: git init [repository name]

This command is used to start a new repository.

GitInit Command - Git Commands - Edureka

### 3)git clone

Usage: git clone [url]

This command is used to obtain a repository from an existing URL.



### 4)git add

Usage: git add [file]

This command adds a file to the staging area.

Git Add Command - Git Commands - Edureka

Usage: git add \*

This command adds one or more to the staging area.

Git Add Command - Git Commands - Edureka

### 5)git commit

Usage: git commit -m “[ Type in the commit message]”

This command records or snapshots the file permanently in the version history.



Usage: git commit -a

This command commits any files you’ve added with the git add command and also commits any files you’ve changed since then.

Git Commit Command - Git Commands - Edureka

### 6)git diff

Usage: git diff

This command shows the file differences which are not yet staged.



Usage: git diff –staged

This command shows the differences between the files in the staging area and the latest version present.



Usage: git diff [first branch] [second branch]

This command shows the differences between the two branches mentioned.



### 7)git reset

Usage: git reset [file]

This command unstages the file, but it preserves the file contents.



Usage: git reset [commit]

This command undoes all the commits after the specified commit and preserves the changes locally.

Git Reset Command - Git Commands - Edureka

Usage: git reset –hard [commit]  This command discards all history and goes back to the specified commit.

Git Reset Command - Git Commands - Edureka

### 8)git status

Usage: git status

This command lists all the files that have to be committed.



### 9)git rm

Usage: git rm [file]

This command deletes the file from your working directory and stages the deletion.

Git Rm Command - Git Commands - Edureka

### 10)git log

Usage: git log

This command is used to list the version history for the current branch.



Usage: git log –follow[file]

This command lists version history for a file, including the renaming of files also.



### 11)git show

Usage: git show [commit]

This command shows the metadata and content changes of the specified commit.



### 12)git branch

Usage: git branch

This command lists all the local branches in the current repository.

Git Branch Command - Git Commands - Edureka

Usage: git branch [branch name]

This command creates a new branch.

Git Branch Command - Git Commands - Edureka

Usage: git branch -d [branch name]

This command deletes the feature branch.

Git Branch Command - Git Commands - Edureka

### 13)git checkout

Usage: git checkout [branch name]

This command is used to switch from one branch to another.

Git Checkout Command - Git Commands - Edureka

Usage: git checkout -b [branch name]

This command creates a new branch and also switches to it.

Git Checkout Command - Git Commands - Edureka

### 14)git merge

Usage: git merge [branch name]

This command merges the specified branch’s history into the current branch.

Git Merge Command - Git Commands - Edureka

### 15)git remote

Usage: git remote add [variable name] [Remote Server Link]

This command is used to connect your local repository to the remote server.

Git Remote Command - Git Commands - Edureka

### 16)git push

Usage: git push [variable name] master

This command sends the committed changes of master branch to your remote repository.



Usage: git push [variable name] [branch]

This command sends the branch commits to your remote repository.



Usage: git push –all [variable name]

This command pushes all branches to your remote repository.



Usage: git push [variable name] :[branch name]

This command deletes a branch on your remote repository.



### 17)git pull

Usage: git pull [Repository Link]

This command fetches and merges changes on the remote server to your working directory.



### 18)git stash

Usage: git stash save

This command temporarily stores all the modified tracked files.

Git Stash Command - Git Commands - Edureka

Usage: git stash pop

This command restores the most recently stashed files.



Usage: git stash list

This command lists all stashed change sets.

Git Stash Command - Git Commands - Edureka

Usage: git stash drop

This command discards the most recently stashed changeset.

Git Stash Command - Git Commands - Edureka

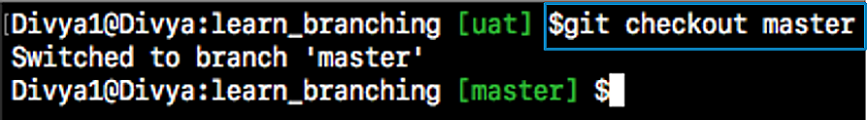
**Git Reflog – How to recover a deleted branch that was not merged**

### Step 1: List the branches that are merged into master

First, check out into the ‘**master**’ branch if you are on some other branch using the command:

|  |  |
| --- | --- |
| 1 | $git checkout master |

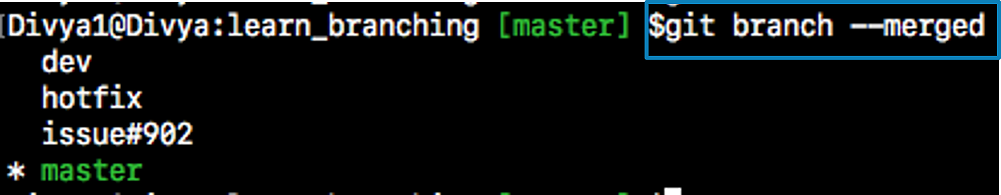
**Output**



Now, to get a list of merged branches, mention the following command:

|  |  |
| --- | --- |
| 1 | $git branch --merged |

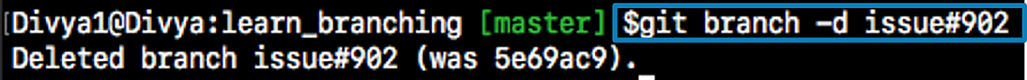
**Output:**



### Step 1.1: Then, delete the merged branch:

|  |  |
| --- | --- |
| 1 | $git branch -d issue#902 |

**Output:**

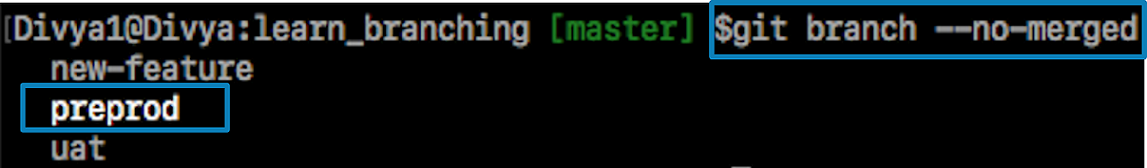


The branch ‘issue#902’ was successfully deleted as it is already merged into the ‘master’ branch.

### Step 2: Now, let us list the branches which are not-merged into master.

|  |  |
| --- | --- |
| 1 | $git branch --no-merged |

**Output**

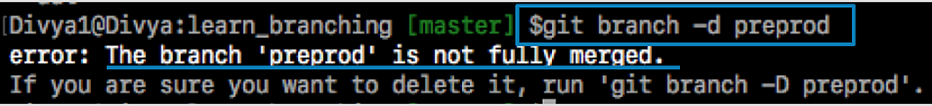
****

**Step 2.2: Finally, let us delete an un-merged branch with the following command:**

|  |  |
| --- | --- |
| 1 | $git branch -d prepod |

***If you try to delete one of the branches with un-finished work say “preprod” branch, git displays a warning message.***

**Output**



## ****Recover a deleted branch using Git Reflog****

### Step 1: History logs of all the references

Get a list of all the local recorded history logs for all the references (‘master’, ‘uat’ and ‘prepod’) in this repository.

|  |  |
| --- | --- |
| 1 | git reflog |

### Git Reflog - Git Reflog - Edureka

### Step 3: Recover

|  |  |
| --- | --- |
| 1 | git checkout -b preprod HEAD@{4} |

**Output**

