

If we want to check the configuration of the git hub, we can use the given below command.

```
git config --list
```

```
user.email=wright2khaliq@gmail.com  
user.name=wright2khaliq
```

If in case you have not configured please configure it by using the given below command

```
git config --global user.name "wright2khaliq"  
Git config --global user.email wright2khaliq@gmaail.com
```

GIT INIT COMMAND

- **git init:** Initializes a new Git repository in a directory.

"git init" is a command used to initialize a new Git repository. It creates a new subdirectory with the name of the repository and initializes the necessary files and directories for Git to start managing your project.

```
→ first-repo git init  
Initialized empty Git repository in /home/mohammed/git-demo/first-repo/.git/  
→ first-repo git:(master) touch khaliq
```

GIT COMMIT COMMAND

The "git commit" command is used to record changes made to the files in the Git repository. It creates a new commit object with a unique ID, and stores the changes made in the repository's history.

The "git commit" command is a crucial part of Git's version control system, as it allows you to keep track of the changes made to your files and collaborate with other developers.

```
→ first-repo git:(master) X git commit -m "commit done"
[master (root-commit) ie412f2] commit done
 1 file changed, 4 insertions(+)
 create mode 100644 khaliq
```

GIT ADD COMMAND

The git add command is used add changes in the current directory to the staging area

Git add . Used with the full stop it add all the file to staging area

```
no changes added to commit (use "git add" and/or "git commit -a")
→ first-repo git:(master) X git add .
First time committing? Y/n git commit -m "commit done"
```

GIT STATUS COMMAND

The "git status" command is used to show the current state of the Git repository. It provides information about the current branch, which files have been modified, which files are staged for commit, and which files are untracked.

```
no changes added to commit (use "git add" and/or "git commit -a")
→ first-repo git:(master) X git add .
First time committing? Y/n git commit -m "commit done"
```

GIT MERGE COMMAND

The "git merge" command is used to integrate changes from one branch into another branch. When you run the "git merge" command, Git takes the changes made in one branch (called the "source" branch) and applies them to another branch (called the "destination" branch).

```
→ branches01 git:(master) git merge feature
merge: feature - not something we can merge
```

Branch Rename

Assuming you have a Git remote named "origin" and you have a branch named "debug" locally, the command `git push origin debug:hero` will push the local "debug" branch to the remote "hero" branch on the "origin" remote.

This means that any changes you have made to the local "debug" branch will be uploaded to the "hero" branch on the remote "origin" repository. If the "hero" branch does not exist on the remote "origin" repository, it will be created.

It's worth noting that when pushing changes to a remote branch, it's important to make sure that you have the latest changes from the remote branch before you push your changes. This can be done by running `git pull` before `git push` or by using `git push --force` if you're sure that your changes won't conflict with changes made by others.

```
→ branches01 git:(debug) git push origin debug:hero
Username for 'https://github.com': wright2khaliq
Password for 'https://wright2khaliq@github.com':
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'hero' on GitHub by visiting:
remote:   https://github.com/wright2khaliq/branches-explanation/pull/new/hero
remote:
To https://github.com/wright2khaliq/branches-explanation.git
 * [new branch]      debug -> hero
```

Git push command

```
mkdir git-demo
```

```
→ ~ cd git-demo
```

```
→ git-demo pwd
/home/mohammed/git-demo
```

```
→ git-demo mkdir first-repo
```

```
→ git-demo cd first-repo

→ first-repo pwd
/home/mohammed/git-demo/first-repo

→ first-repo git init
Initialized empty Git repository in /home/mohammed/git-demo/first-repo/.git/

→ first-repo git:(master) touch khaliq

→ first-repo git:(master) ✘ vi khaliq

→ first-repo git:(master) ✘ git status
On branch master
```

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)
khaliq

nothing added to commit but untracked files present (use "git add" to track)

```
→ first-repo git:(master) ✘ git add .
```

```
→ first-repo git:(master) ✘ git status
On branch master
```

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)
new file: khaliq

```
→ first-repo git:(master) ✘ git commit -m "commit done"
[master (root-commit) 1e412f2] commit done
 1 file changed, 4 insertions(+)
 create mode 100644 khaliq
```

```
→ first-repo git:(master) git log
```

[3] + 5543 suspended git log

```
→ first-repo git:(master) touch abdul
→ first-repo git:(master) ✘ vi abdul
→ first-repo git:(master) ✘ vi khaliq
→ first-repo git:(master) ✘ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:  khaliq
```

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

no changes added to commit (use "git add" and/or "git commit -a")

```
→ first-repo git:(master) ✘ git add .
→ first-repo git:(master) ✘ git commit -m "commit2done"
[master b490342] commit2done
 2 files changed, 6 insertions(+)
 create mode 100644 abdul

→ first-repo git:(master) git remote add origin
https://github.com/wright2khaliq/sai-pawandemo.git
```

```
→ first-repo git:(master) git remote -v
origin  https://github.com/wright2khaliq/sai-pawandemo.git (fetch)
origin  https://github.com/wright2khaliq/sai-pawandemo.git (push)
```

```
→ first-repo git:(master) git push -u origin master
Username for 'https://github.com': wright2khaliq
Password for 'https://wright2khaliq@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (7/7), 587 bytes | 587.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0)
remote:
```

```
remote: Create a pull request for 'master' on GitHub by visiting:  
remote: https://github.com/wright2khaliq/sai-pawandemo/pull/new/master  
remote:  
To https://github.com/wright2khaliq/sai-pawandemo.git  
 * [new branch] master -> master  
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

→ first-repo git:(master)

```
~ ~ mkdir git-demo  
~ ~ cd git-demo  
git-demo pwd  
/home/mohammed/git-demo  
git-demo mkdir first-repo  
git-demo cd first-repo  
first-repo pwd  
/home/mohammed/git-demo/first-repo  
first-repo git init  
Initialized empty Git repository in /home/mohammed/git-demo/first-repo/.git/  
first-repo git:(master) touch khaliq  
first-repo git:(master) vi khaliq  
first-repo git:(master) git status  
On branch master  
No commits yet  
Untracked files:  
(use "git add <file>..." to include in what will be committed)  
khaliq  
nothing added to commit but untracked files present (use "git add" to track)  
first-repo git:(master) git add .  
first-repo git:(master) git status  
On branch master  
No commits yet  
Changes to be committed:  
(use "git rm --cached <file>..." to unstage)  
new file: khaliq  
first-repo git:(master) Xgit commit -m "commit done"  
[master (root-commit) 1e412f2] commit done  
1 file changed, 4 insertions(+)  
create mode 100644 khaliq  
first-repo git:(master) git log
```

```

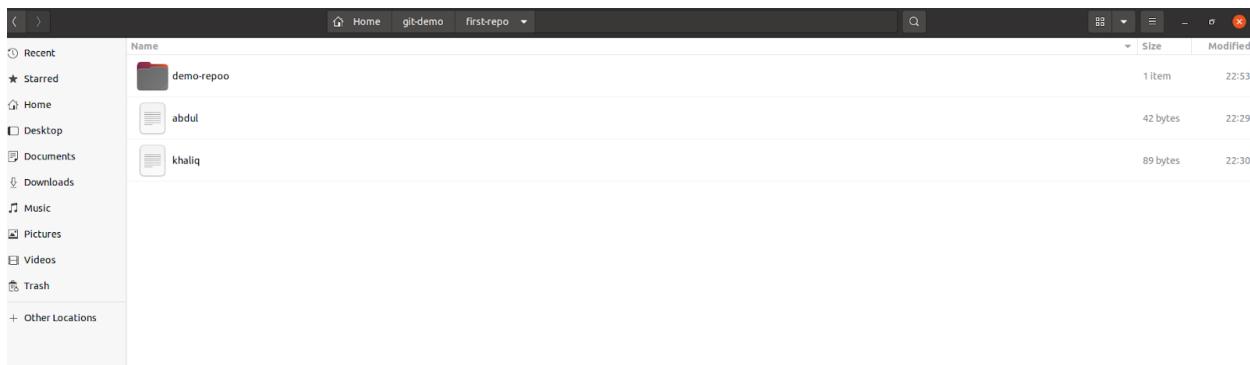
→ first-repo git:(master) git log
[3] + 5543 suspended git log
→ first-repo git:(master) touch abdul
→ first-repo git:(master) X vi abdul
→ first-repo git:(master) X vi khaliq
→ first-repo git:(master) X git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   khaliq

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

no changes added to commit (use "git add" and/or "git commit -a")
→ first-repo git:(master) X git add .
→ first-repo git:(master) X git commit -m "commit2done"
[master b490342] commit2done
  2 files changed, 6 insertions(+)
  create mode 100644 abdul
→ first-repo git:(master) git remote add origin https://github.com/wright2khaliq/sai-pawandemo.git
→ first-repo git:(master) git remote -v
origin https://github.com/wright2khaliq/sai-pawandemo.git (fetch)
origin https://github.com/wright2khaliq/sai-pawandemo.git (push)
→ first-repo git:(master) git push -u origin master
Username for 'https://github.com': wright2khaliq
Password for 'https://wright2khaliq@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (7/7), 587 bytes | 587.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/wright2khaliq/sai-pawandemo/pull/new/master

```

Now we will create folder manually, the same way mannaualy we will move it – home –git-demo-first -repo and will add the folder and will add it and commit it and push it



```
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/wright2khaliq/sai-pawandemo/pull/new/master
remote:
To https://github.com/wright2khaliq/sai-pawandemo.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
→ first-repo git:(master) ✘ git add .
→ first-repo git:(master) ✘ git commit -m "addfolder"
[master 142e0ef] addfolder
 1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 demo-repoo/jenkins.pdf
→ first-repo git:(master) ✘ git push -u origin master
Username for 'https://github.com': wright2khaliq
Password for 'https://wright2khaliq@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 1.39 MiB | 1.86 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/wright2khaliq/sai-pawandemo.git
 b490342..142e0ef  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
::: → first-repo git:(master) ✘
```

→ first-repo git:(master) ✘ git add .
→ first-repo git:(master) ✘ git commit -m "addfolder"
[master 142e0ef] addfolder
 1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 demo-repoo/jenkins.pdf

→ first-repo git:(master) git push -u origin master

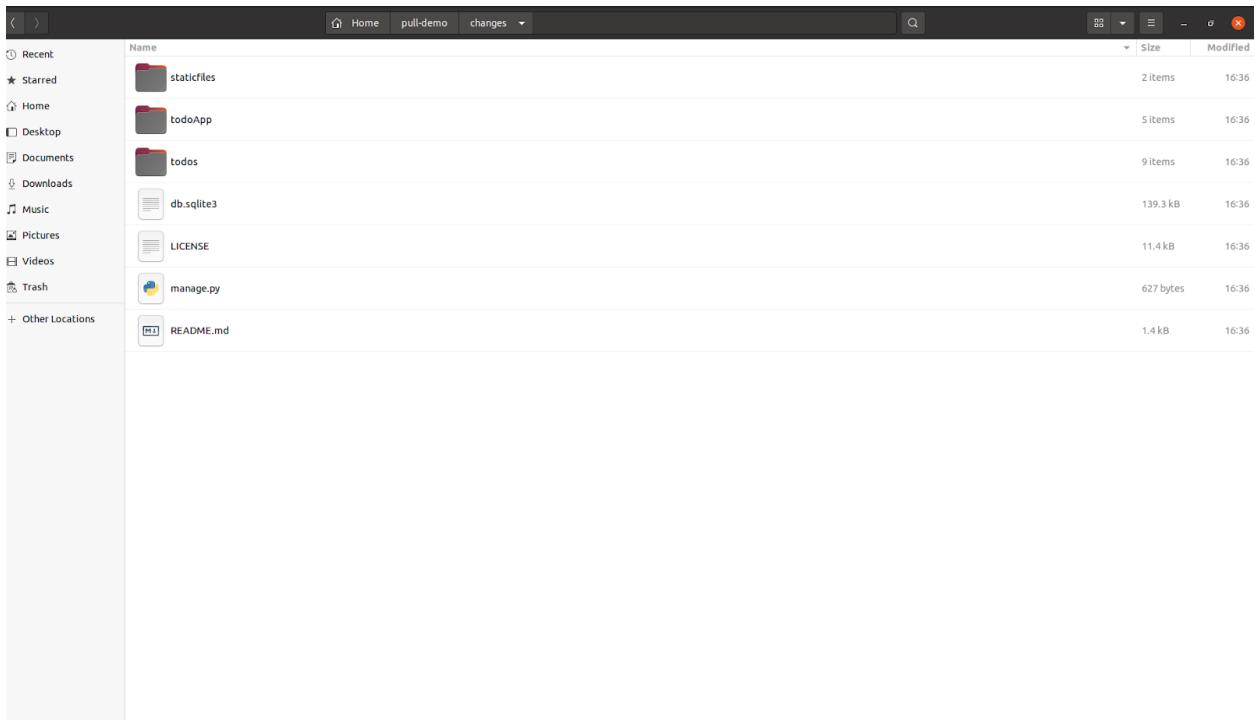
Username for 'https://github.com': wright2khaliq
Password for 'https://wright2khaliq@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 1.39 MiB | 1.86 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)

```
To https://github.com/wright2khaliq/sai-pawandemo.git  
b490342..142e0ef master -> master  
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

GIT PULL

The `git pull` command is used to fetch the latest changes from a remote repository and merge them into your current branch.

```
~ mkdir pull-demo  
~ cd pull-demo  
pull-demo mkdir changes  
pull-demo cd changes  
changes git init  
Initialized empty Git repository in /home/mohammed/pull-demo/changes/.git/  
changes git:(master) git pull https://github.com/shreys7/django-todo.git  
remote: Enumerating objects: 287, done.  
remote: Total 287 (delta 0), reused 0 (delta 0), pack-reused 287  
Receiving objects: 100% (287/287), 120.95 KiB | 333.00 KiB/s, done.  
Resolving deltas: 100% (152/152), done.  
From https://github.com/shreys7/django-todo  
 * branch HEAD      -> FETCH_HEAD  
changes git:(master) □
```



Git Branches

In Git, a branch is a lightweight movable pointer to a commit in the Git history graph. Branches allow you to diverge from the main line of development and make changes without affecting the main codebase until you are ready to merge your changes back in.

- 1) How to create a branch
- 2) How to merge that branch to remote repository

→ ~ mkdir branches

```
→ ~ mkdir branches01  
→ ~ cd branches01  
→ branches01 touch branch  
→ branches01 vi branch  
→ branches01 git init  
Initialized empty Git repository in /home/mohammed/branches01/.git/  
  
→ branches01 git:(master) ✘ git add .  
  
→ branches01 git:(master) ✘ git commit -m "hello"  
[master (root-commit) aa3f168] hello  
 1 file changed, 4 insertions(+)  
 create mode 100644 branch  
  
→ branches01 git:(master) git branch debug  
  
→ branches01 git:(master) git branch  
  
[8] + 19956 suspended git branch  
→ branches01 git:(master) git checkout debug  
Switched to branch 'debug'  
  
→ branches01 git:(debug) ls  
branch  
  
→ branches01 git:(debug) touch helloworldd  
  
→ branches01 git:(debug) ✘ vi helloworldd  
  
→ branches01 git:(debug) ✘ git add .  
  
→ branches01 git:(debug) ✘ git commit -m "firstcommit"  
[debug a328c17] firstcommit  
 1 file changed, 4 insertions(+)  
 create mode 100644 helloworldd  
  
→ branches01 git:(debug) git checkout master
```

```
Switched to branch 'master'
```

```
→ branches01 git:(master) git merge feature  
merge: feature - not something we can merge
```

```
→ branches01 git:(master) git merge debug
```

```
Updating aa3f168..a328c17
```

```
Fast-forward
```

```
helloworld | 4 ++++
```

```
1 file changed, 4 insertions(+)
```

```
create mode 100644 helloworld
```

```
→ branches01 git:(master) git checkout debug
```

```
Switched to branch 'debug'
```

```
→ branches01 git:(debug) git checkout master
```

```
Switched to branch 'master'
```

```
→ branches01 git:(master) git remote add origin
```

```
https://github.com/wright2khaliq/branches-explanation.git
```

```
→ branches01 git:(master) git remote
```

```
origin
```

```
→ branches01 git:(master) git remote -v
```

```
origin https://github.com/wright2khaliq/branches-explanation.git (fetch)
```

```
origin https://github.com/wright2khaliq/branches-explanation.git (push)
```

```
→ branches01 git:(master) git push -u origin master
```

```
Username for 'https://github.com': wright2khaliq
```

```
Password for 'https://wright2khaliq@github.com':
```

```
Enumerating objects: 6, done.
```

```
Counting objects: 100% (6/6), done.
```

```
Delta compression using up to 4 threads
```

```
Compressing objects: 100% (3/3), done.
```

```
Writing objects: 100% (6/6), 501 bytes | 501.00 KiB/s, done.
```

```
Total 6 (delta 0), reused 0 (delta 0)
```

```
remote:
```

```
remote: Create a pull request for 'master' on GitHub by visiting:
```

```
remote: https://github.com/wright2khaliq/branches-explanation/pull/new/master
```

```
remote:  
To https://github.com/wright2khaliq/branches-explanation.git  
 * [new branch] master -> master  
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

```
→ branches01 git:(master) git status  
On branch master  
Your branch is up to date with 'origin/master'.
```

```
nothing to commit, working tree clean
```

```
→ branches01 git:(master) git checkout debug  
Switched to branch 'debug'
```

```
→ branches01 git:(debug) git push origin debug  
Username for 'https://github.com': wright2khaliq  
Password for 'https://wright2khaliq@github.com':  
Total 0 (delta 0), reused 0 (delta 0)  
remote:  
remote: Create a pull request for 'debug' on GitHub by visiting:  
remote: https://github.com/wright2khaliq/branches-explanation/pull/new/debug  
remote:  
To https://github.com/wright2khaliq/branches-explanation.git  
 * [new branch] debug -> debug
```

```
→ branches01 git:(debug) git push origin debug  
Username for 'https://github.com': wright2khaliq  
Password for 'https://wright2khaliq@github.com':  
Everything up-to-date
```

```
→ branches01 git:(debug) git status  
On branch debug  
nothing to commit, working tree clean
```

```
→ branches01 git:(debug) git push origin debug:hero  
Username for 'https://github.com': wright2khaliq  
Password for 'https://wright2khaliq@github.com':  
Total 0 (delta 0), reused 0 (delta 0)
```

```
remote:  
remote: Create a pull request for 'hero' on GitHub by visiting:  
remote: https://github.com/wright2khaliq/branches-explanation/pull/new/hero  
remote:  
To https://github.com/wright2khaliq/branches-explanation.git  
 * [new branch] debug -> hero
```

→ branches01 git:(debug) git branch

[9] + 23066 suspended git branch

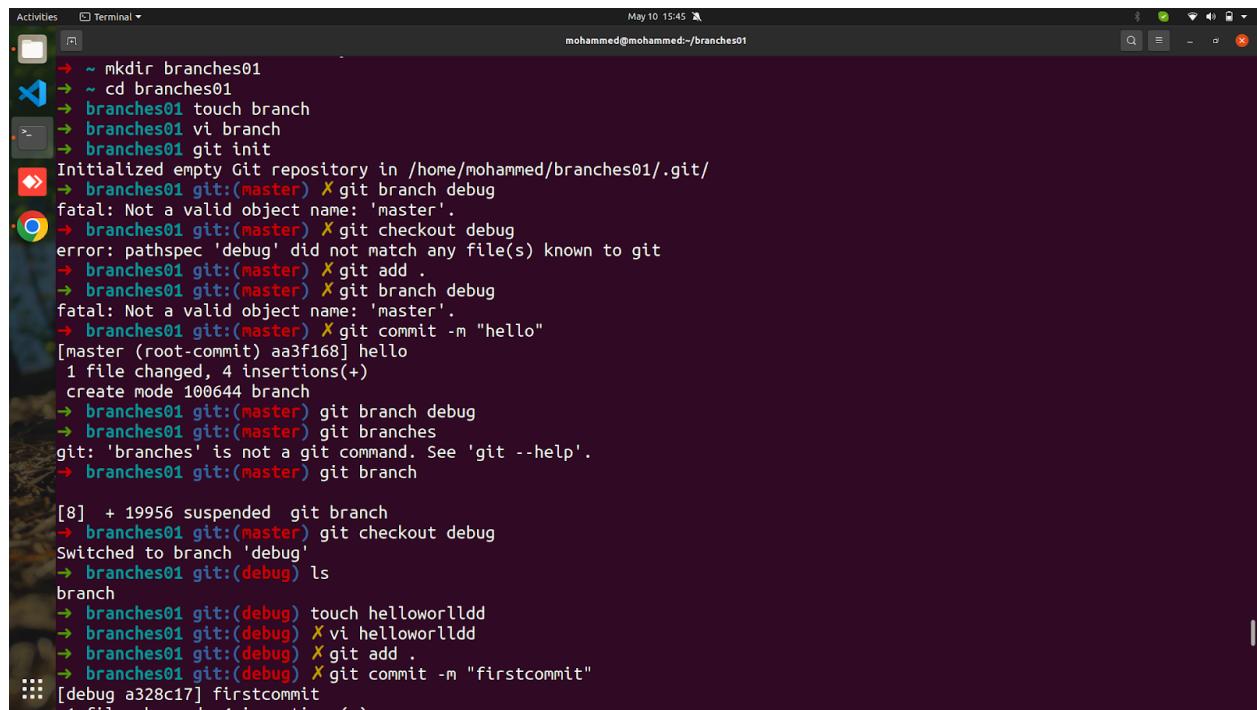
→ branches01 git:(debug) git branch -d debug

error: Cannot delete branch 'debug' checked out at '/home/mohammed/branches01'

→ branches01 git:(debug) git merge debug

Already up to date.

→ branches01 git:(debug)



The screenshot shows a terminal window on a Linux desktop environment. The title bar indicates it's an Activities Terminal window from May 10 15:45. The user is logged in as mohammed@mohammed:~/branches01. The terminal history shows the following commands:

```
~ mkdir branches01
~ cd branches01
branches01 touch branch
branches01 vi branch
branches01 git init
Initialized empty Git repository in /home/mohammed/branches01/.git/
branches01 git:(master) X git branch debug
fatal: Not a valid object name: 'master'.
branches01 git:(master) X git checkout debug
error: pathspec 'debug' did not match any file(s) known to git
branches01 git:(master) X git add .
branches01 git:(master) X git branch debug
fatal: Not a valid object name: 'master'.
branches01 git:(master) X git commit -m "hello"
[master (root-commit) aa3f168] hello
 1 file changed, 4 insertions(+)
create mode 100644 branch
branches01 git:(master) git branch debug
branches01 git:(master) git branches
git: 'branches' is not a git command. See 'git --help'.
branches01 git:(master) git branch

[8] + 19956 suspended git branch
branches01 git:(master) git checkout debug
Switched to branch 'debug'
branches01 git:(debug) ls
branch
branches01 git:(debug) touch helloworldd
branches01 git:(debug) X vi helloworldd
branches01 git:(debug) X git add .
branches01 git:(debug) X git commit -m "firstcommit"
[debug a328c17] firstcommit
```

```
Activities Terminal May 10 15:46
mohammed@mohammed:~/branches01
1 file changed, 4 insertions(+)
create mode 100644 helloworldd
→ branches01 git:(debug) git checkout master
Switched to branch 'master'
→ branches01 git:(master) git merge feature
merge: feature - not something we can merge
→ branches01 git:(master) git merge debug
Updating aa3f168..a328c17
Fast-forward
helloworldd | 4 +++
1 file changed, 4 insertions(+)
create mode 100644 helloworldd
→ branches01 git:(master) git checkout debug
Switched to branch 'debug'
→ branches01 git:(debug) git push origin debug
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
→ branches01 git:(debug) git checkout master
Switched to branch 'master'
→ branches01 git:(master) git remote add origin https://github.com/wright2khaliq/branches-explanation.git
→ branches01 git:(master) git remote
origin
→ branches01 git:(master) git remote -v
origin https://github.com/wright2khaliq/branches-explanation.git (fetch)
origin https://github.com/wright2khaliq/branches-explanation.git (push)
→ branches01 git:(master) git push -u origin master
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
```

```
Activities Terminal May 10 15:46
mohammed@mohammed:~/branches01
Please make sure you have the correct access rights
and the repository exists.
→ branches01 git:(master) git push -u origin masterhttps://github.com/wright2khaliq/branches-explanation.git
fatal: invalid refspec 'masterhttps://github.com/wright2khaliq/branches-explanation.git'
→ branches01 git:(master) git push -u origin master https://github.com/wright2khaliq/branches-explanation.git
fatal: invalid refspec 'https://github.com/wright2khaliq/branches-explanation.git'
→ branches01 git:(master) git push -u origin master
Username for 'https://github.com': wright2khaliq
Password for 'https://wright2khaliq@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 501 bytes | 501.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/wright2khaliq/branches-explanation/pull/new/master
remote:
To https://github.com/wright2khaliq/branches-explanation.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
→ branches01 git:(master) git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
→ branches01 git:(master) git checkout debug
Switched to branch 'debug'
→ branches01 git:(debug) git push origin debug
Username for 'https://github.com': wright2khaliq
Password for 'https://wright2khaliq@github.com':
```

```
Activities Terminal May 10 15:47 mohammed@mohammed:~/branches01
Password for 'https://wright2khaliq@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 501 bytes | 501.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/wright2khaliq/branches-explanation/pull/new/master
remote:
To https://github.com/wright2khaliq/branches-explanation.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
→ branches01 git:(master) git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
→ branches01 git:(master) git checkout debug
Switched to branch 'debug'
→ branches01 git:(debug) git push origin debug
Username for 'https://github.com': wright2khaliq
Password for 'https://wright2khaliq@github.com':
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'debug' on GitHub by visiting:
remote:     https://github.com/wright2khaliq/branches-explanation/pull/new/debug
remote:
To https://github.com/wright2khaliq/branches-explanation.git
 * [new branch]      debug -> debug
→ branches01 git:(debug) git push origin debug
```

```
Activities Terminal May 10 15:47 mohammed@mohammed:~/branches01
Password for 'https://wright2khaliq@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 501 bytes | 501.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote: https://github.com/wright2khaliq/branches-explanation/pull/new/master
remote:
To https://github.com/wright2khaliq/branches-explanation.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
-> branches01 git:(master) git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
-> branches01 git:(master) git checkout debug
Switched to branch 'debug'
-> branches01 git:(debug) git push origin debug
Username for 'https://github.com': wright2khaliq
Password for 'https://wright2khaliq@github.com':
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'debug' on GitHub by visiting:
remote: https://github.com/wright2khaliq/branches-explanation/pull/new/debug
remote:
To https://github.com/wright2khaliq/branches-explanation.git
 * [new branch]      debug -> debug
-> branches01 git:(debug) git push origin debug
```

GIT REBASE ;

In Git, `rebase` is a command used to apply a series of commits from one branch on top of another branch. Unlike `merge`, which creates a new merge commit, `rebase` rewrites the Git history by creating new commits with the same changes as the original commits but on top of a different branch.

It's worth noting that `rebase` is a powerful tool, but it can also be dangerous if not used carefully. Rewriting history can cause problems for other developers who are working on the same branch, so it's important to communicate with your team before using `rebase` and to use it only when it's appropriate.

```
-> branches01 git:(master) git rebase
Current branch master is up to date.
-> branches01 git:(master) vi hello120
-> branches01 git:(master) Xgit rebase
error: cannot rebase: You have unstaged changes.
error: Please commit or stash them.
-> branches01 git:(master) X
```

GIT RESET

**Without touching any commit objects you can write any thing in your working directory
(before commit)**

```
ERROR: Please commit or stash them.
→ branches01 git:(master) ✘ git reset
Unstaged changes after reset:
M      hello120
[::] → branches01 git:(master) ✘
```