craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git status

On branch main

nothing to commit, working tree clean

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git remote add origin "https://github.com/crajesh14/first\_repository.git"

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git push origin main

Enumerating objects: 6, done.

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

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (6/6), 481 bytes | 160.00 KiB/s, done.

Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)

remote: Resolving deltas: 100% (1/1), done.

To https://github.com/crajesh14/first\_repository.git

\* [new branch] main -> main

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git branch -b rajesh\_b1

error: unknown switch `b'

usage: git branch [<options>] [-r | -a] [--merged] [--no-merged]

or: git branch [<options>] [-f] [--recurse-submodules] <branch-name> [<start-point>]

or: git branch [<options>] [-l] [<pattern>...]

or: git branch [<options>] [-r] (-d | -D) <branch-name>...

or: git branch [<options>] (-m | -M) [<old-branch>] <new-branch>

or: git branch [<options>] (-c | -C) [<old-branch>] <new-branch>

or: git branch [<options>] [-r | -a] [--points-at]

or: git branch [<options>] [-r | -a] [--format]

Generic options

-v, --[no-]verbose show hash and subject, give twice for upstream branch

-q, --[no-]quiet suppress informational messages

-t, --[no-]track[=(direct|inherit)]

set branch tracking configuration

-u, --[no-]set-upstream-to <upstream>

change the upstream info

--[no-]unset-upstream unset the upstream info

--[no-]color[=<when>] use colored output

-r, --remotes act on remote-tracking branches

--contains <commit> print only branches that contain the commit

--no-contains <commit>

print only branches that don't contain the commit

--[no-]abbrev[=<n>] use <n> digits to display object names

Specific git-branch actions:

-a, --all list both remote-tracking and local branches

-d, --[no-]delete delete fully merged branch

-D delete branch (even if not merged)

-m, --[no-]move move/rename a branch and its reflog

-M move/rename a branch, even if target exists

--[no-]omit-empty do not output a newline after empty formatted refs

-c, --[no-]copy copy a branch and its reflog

-C copy a branch, even if target exists

-l, --[no-]list list branch names

--[no-]show-current show current branch name

--[no-]create-reflog create the branch's reflog

--[no-]edit-description

edit the description for the branch

-f, --[no-]force force creation, move/rename, deletion

--merged <commit> print only branches that are merged

--no-merged <commit> print only branches that are not merged

--[no-]column[=<style>]

list branches in columns

--[no-]sort <key> field name to sort on

--[no-]points-at <object>

print only branches of the object

-i, --[no-]ignore-case

sorting and filtering are case insensitive

--[no-]recurse-submodules

recurse through submodules

--[no-]format <format>

format to use for the output

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git checkout -b rajesh\_b1

Switched to a new branch 'rajesh\_b1'

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (rajesh\_b1)

$ git branch -b amit\_b1

error: unknown switch `b'

usage: git branch [<options>] [-r | -a] [--merged] [--no-merged]

or: git branch [<options>] [-f] [--recurse-submodules] <branch-name> [<start-point>]

or: git branch [<options>] [-l] [<pattern>...]

or: git branch [<options>] [-r] (-d | -D) <branch-name>...

or: git branch [<options>] (-m | -M) [<old-branch>] <new-branch>

or: git branch [<options>] (-c | -C) [<old-branch>] <new-branch>

or: git branch [<options>] [-r | -a] [--points-at]

or: git branch [<options>] [-r | -a] [--format]

Generic options

-v, --[no-]verbose show hash and subject, give twice for upstream branch

-q, --[no-]quiet suppress informational messages

-t, --[no-]track[=(direct|inherit)]

set branch tracking configuration

-u, --[no-]set-upstream-to <upstream>

change the upstream info

--[no-]unset-upstream unset the upstream info

--[no-]color[=<when>] use colored output

-r, --remotes act on remote-tracking branches

--contains <commit> print only branches that contain the commit

--no-contains <commit>

print only branches that don't contain the commit

--[no-]abbrev[=<n>] use <n> digits to display object names

Specific git-branch actions:

-a, --all list both remote-tracking and local branches

-d, --[no-]delete delete fully merged branch

-D delete branch (even if not merged)

-m, --[no-]move move/rename a branch and its reflog

-M move/rename a branch, even if target exists

--[no-]omit-empty do not output a newline after empty formatted refs

-c, --[no-]copy copy a branch and its reflog

-C copy a branch, even if target exists

-l, --[no-]list list branch names

--[no-]show-current show current branch name

--[no-]create-reflog create the branch's reflog

--[no-]edit-description

edit the description for the branch

-f, --[no-]force force creation, move/rename, deletion

--merged <commit> print only branches that are merged

--no-merged <commit> print only branches that are not merged

--[no-]column[=<style>]

list branches in columns

--[no-]sort <key> field name to sort on

--[no-]points-at <object>

print only branches of the object

-i, --[no-]ignore-case

sorting and filtering are case insensitive

--[no-]recurse-submodules

recurse through submodules

--[no-]format <format>

format to use for the output

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (rajesh\_b1)

$ git checkout -b amit\_b1

Switched to a new branch 'amit\_b1'

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (amit\_b1)

$ git checkout -b rajesh\_b1

fatal: a branch named 'rajesh\_b1' already exists

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (amit\_b1)

$ git add requirement.txt

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (amit\_b1)

$ git commit -m "first commit by b1"

[amit\_b1 679a1e7] first commit by b1

1 file changed, 1 insertion(+)

create mode 100644 requirement.txt

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (amit\_b1)

$ git push -u origin amit\_b1

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 8 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 305 bytes | 305.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

remote:

remote: Create a pull request for 'amit\_b1' on GitHub by visiting:

remote: https://github.com/crajesh14/first\_repository/pull/new/amit\_b1

remote:

To https://github.com/crajesh14/first\_repository.git

\* [new branch] amit\_b1 -> amit\_b1

branch 'amit\_b1' set up to track 'origin/amit\_b1'.

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (amit\_b1)

$ git tag -a v1.0 -m "version-1 created new branch b1"

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (amit\_b1)

$ git checkout -b raj\_b1

Switched to a new branch 'raj\_b1'

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$ git add .

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$ git commit -m "second commit by b1"

[raj\_b1 086fb31] second commit by b1

1 file changed, 1 insertion(+)

create mode 100644 design.txt

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$ git push -u origin raj\_b1

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 8 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 330 bytes | 10.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

remote:

remote: Create a pull request for 'raj\_b1' on GitHub by visiting:

remote: https://github.com/crajesh14/first\_repository/pull/new/raj\_b1

remote:

To https://github.com/crajesh14/first\_repository.git

\* [new branch] raj\_b1 -> raj\_b1

branch 'raj\_b1' set up to track 'origin/raj\_b1'.

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$ git tag -a v2.0 -m "version-2 created new branch b1"

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$ git checkout main

Switched to branch 'main'

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git pull origin main

From https://github.com/crajesh14/first\_repository

\* branch main -> FETCH\_HEAD

Already up to date.

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git tags

git: 'tags' is not a git command. See 'git --help'.

The most similar commands are

stage

tag

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git tag

v1.0

v2.0

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ cat content.txt

line1 --modified by main

line2 --modified by main

line3 --checking status after modification

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ cat requirement.txt

cat: requirement.txt: No such file or directory

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ cat design.txt

cat: design.txt: No such file or directory

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git checkout raj\_b1

Switched to branch 'raj\_b1'

Your branch is up to date with 'origin/raj\_b1'.

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$ cat requirement.txt

created by branch1--rajesh

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$ cat content.txt

line1 --modified by main

line2 --modified by main

line3 --checking status after modification

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$ cat design.txt

this is designed by me

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$ git checkout amit\_b1

Switched to branch 'amit\_b1'

Your branch is up to date with 'origin/amit\_b1'.

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (amit\_b1)

$ cat design.txt

cat: design.txt: No such file or directory

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (amit\_b1)

$ git checkout main

Switched to branch 'main'

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git merge raj-b1 -m "merging main and raj-b1"

merge: raj-b1 - not something we can merge

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ ^C

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git merge raj\_b1 -m "merging main and raj\_b1"

Updating df73eb3..086fb31

Fast-forward (no commit created; -m option ignored)

design.txt | 1 +

requirement.txt | 1 +

2 files changed, 2 insertions(+)

create mode 100644 design.txt

create mode 100644 requirement.txt

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ cat requirement.txt

created by branch1--rajesh

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ cat design.txt

this is designed by me

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ cat.content.txt

bash: cat.content.txt: command not found

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ cat content.txt

line1 --modified by main

line2 --modified by main

line3 --checking status after modification

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git tag -a v3 -m "merge the branch with main"

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git tag

v1.0

v2.0

v3

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git push origin --tags

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (5/5), done.

Writing objects: 100% (6/6), 673 bytes | 336.00 KiB/s, done.

Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)

remote: Resolving deltas: 100% (1/1), done.

To https://github.com/crajesh14/first\_repository.git

\* [new tag] v1.0 -> v1.0

\* [new tag] v2.0 -> v2.0

\* [new tag] v3 -> v3

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git push -u origin main

Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To https://github.com/crajesh14/first\_repository.git

df73eb3..086fb31 main -> main

branch 'main' set up to track 'origin/main'.

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ cat index.txt

cat: index.txt: No such file or directory

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git chekout raj\_b1

git: 'chekout' is not a git command. See 'git --help'.

The most similar command is

checkout

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (main)

$ git checkout raj\_b1

Switched to branch 'raj\_b1'

Your branch is up to date with 'origin/raj\_b1'.

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$ cat index.txt

cat: index.txt: No such file or directory

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$ git checkout amit\_b1

Switched to branch 'amit\_b1'

Your branch is up to date with 'origin/amit\_b1'.

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (amit\_b1)

$ cat index.txt

cat: index.txt: No such file or directory

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (amit\_b1)

$ git checkout raj\_b1

Switched to branch 'raj\_b1'

Your branch is up to date with 'origin/raj\_b1'.

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$ cat content.txt

line1 --modified by main

line2 --modified by main

line3 --checking status after modification

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$ cat requirement.txt

created by branch1--rajesh

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$ cat design.txt

this is designed by me

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$ cat index.txt

cat: index.txt: No such file or directory

craje@LAPTOP-8P76I5IN MINGW64 /d/git project (raj\_b1)

$