

## WEEK 2 Task 2

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SEE (master)
$ git clone https://github.com/Deekshitha-Y00/SquareSquash.git
Cloning into 'SquareSquash'...
fatal: unable to access 'https://github.com/Deekshitha-Y00/SquareSquash.git/': Failed
to connect to github.com port 443 after 19106 ms: could not connect to server

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SEE (master)
$ git clone https://github.com/Deekshitha-Y00/SquareSquash.git
Cloning into 'SquareSquash'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 24 (delta 1), reused 20 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (24/24), 108.36 KiB | 711.00 KiB/s, done.
Resolving deltas: 100% (1/1), done.

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SEE (master)
$ git init
Initialized empty Git repository in C:/Users/azeem/OneDrive/Desktop/SEE/.git/

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SEE (master)
$ git status
On branch master
```

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SEE (master)
$ git status
On branch master

No commits yet

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

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

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SEE (master)
$ git add .

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SEE (master)
$ git commit -m "first commit"
[master (root-commit) 07ccea5] first commit
9 files changed, 955 insertions(+)
create mode 100644 SquareSquash/.gitignore
create mode 100644 SquareSquash/LICENSE.txt
create mode 100644 SquareSquash/README.md
create mode 100644 SquareSquash/git - hub.png
create mode 100644 SquareSquash/pom.xml
create mode 100644 SquareSquash/src/main/java/com/squareup/squash/SquashBacktrace.jav
```

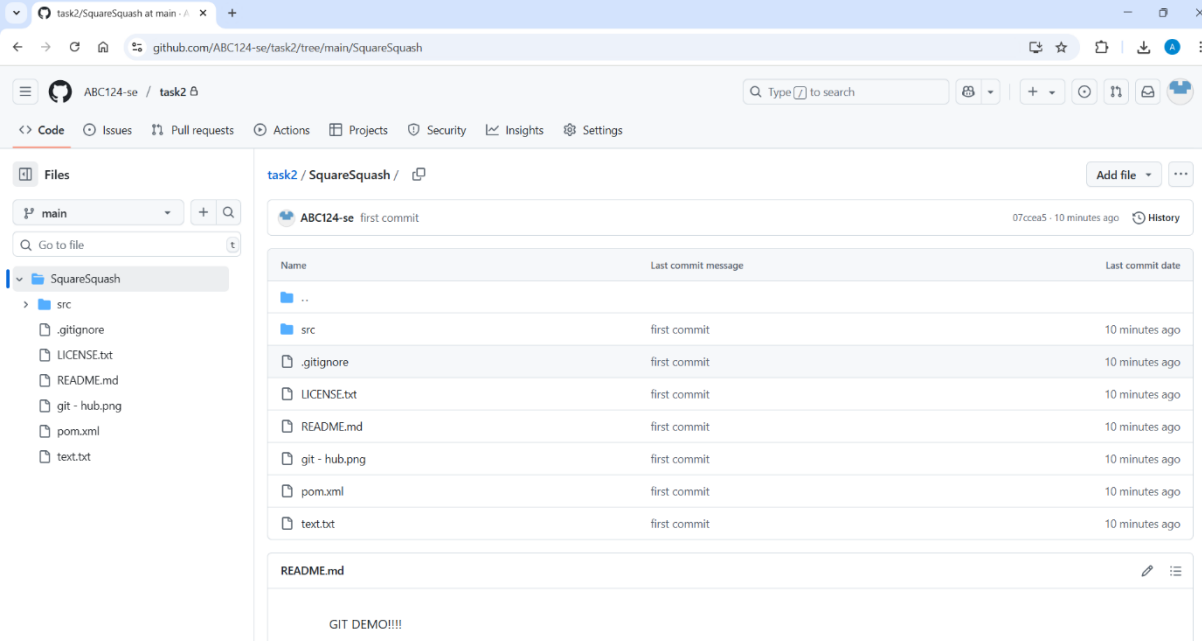
```
a
create mode 100644 SquareSquash/text.txt

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SEE (master)
$ git branch -M main

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SEE (main)
$ git remote add origin https://github.com/ABC124-se/task2.git

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SEE (main)
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 12 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (23/23), 107.58 KiB | 2.44 MiB/s, done.
Total 23 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/ABC124-se/task2.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SEE (main)
$
```



The screenshot shows the GitHub repository page for 'task2 / SquareSquash'. The page displays the commit history and the README content. The commit history table is as follows:

Name	Last commit message	Last commit date
..		
src	first commit	10 minutes ago
.gitignore	first commit	10 minutes ago
LICENSE.txt	first commit	10 minutes ago
README.md	first commit	10 minutes ago
git - hub.png	first commit	10 minutes ago
pom.xml	first commit	10 minutes ago
text.txt	first commit	10 minutes ago

The README.md content is:

```
GIT DEMO!!!!
```

## Scenario questions

1. You made changes to one file in the above project but haven't staged them yet. You realize they were a mistake. What Git command will you use to discard the local changes?

```

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ git status
On branch feature/login
Your branch is up to date with 'origin/feature/login'.

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:   text.txt

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

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ git restore text.txt

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ |

```

```

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ git status
On branch feature/login
Your branch is up to date with 'origin/feature/login'.

nothing to commit, working tree clean

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$

```

2. You accidentally ran `git add file1.txt` (note instead of `file.txt` consider one file from above project), but you're not ready to commit yet. How do you remove it from the staging area without losing the changes?

```

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ git restore --staged text.txt

```

3. You made a commit but typed the wrong commit message. You haven't pushed it yet. How do you fix it?

```

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ git commit --amend|

```



5. After cloning a repo, how can you set your name and email globally for all Git repositories?

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (main)
$ git config --global user.name "azeemafirdous04"
git config --global user.email "azeemafirdous14@gmail.com"

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (main)
$ git config --global --list
user.name=azeemafirdous04
user.email=azeemafirdous14@gmail.com
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
```

6. You've made some edits to your files but haven't staged them. How can you view the changes?

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (main)
$ git diff

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (main)
$ git diff
diff --git a/SquareSquash/text.txt b/SquareSquash/text.txt
index 0db79e7..5df338c 100644
--- a/SquareSquash/text.txt
+++ b/SquareSquash/text.txt
@@ -1,2 @@
 this is collaborative learning
+ddd
\ No newline at end of file
```

7. You're on the main branch but need to switch to feature/login. What command do you

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (main)
$ git switch -c feature/login
Switched to a new branch 'feature/login'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ git branch
* feature/login
main
```

8. You deleted the feature-ui branch by mistake. You haven't pushed the deletion. How

```

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ git reflog
c8adf22 (HEAD -> feature/login, main) HEAD@{0}: checkout: moving from main to feature/login
c8adf22 (HEAD -> feature/login, main) HEAD@{1}: commit (amend): first commit
d19fe5a (origin/main) HEAD@{2}: pull --rebase origin main (finish): returning to refs/heads/main
d19fe5a (origin/main) HEAD@{3}: pull --rebase origin main (start): checkout d19fe5ad33c7ac21017301fa5311f
e8d71424f4b
5c77811 HEAD@{4}: commit (amend): first commit
d19fe5a (origin/main) HEAD@{5}: Branch: renamed refs/heads/main to refs/heads/main
d19fe5a (origin/main) HEAD@{7}: commit (initial): first commit

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ git branch feature-ui 5c77811

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ git branch
  feature-ui
* feature/login
main

```

9. You've made some commits locally and now want to upload them to the remote repository. What do you run?

```

MINGW64/c/Users/azeem/OneDrive/Desktop/SE WEEK 2/SquareSquash
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ git push
fatal: The current branch feature/login has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin feature/login

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ git push --set-upstream origin feature/login
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 12 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (23/23), 107.59 KiB | 2.11 MiB/s, done.
Total 23 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature/login' on GitHub by visiting:
remote:   https://github.com/ABC124-se/SELAB2/pull/new/feature/login
remote:
To https://github.com/ABC124-se/SELAB2.git
 * [new branch]      feature/login -> feature/login
branch 'feature/login' set up to track 'origin/feature/login'.

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ git push
Everything up-to-date

```

10. How can you fetch the latest changes from the remote repository without merging them automatically?

```

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ git fetch origin

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ git diff origin/main
diff --git a/SquareSquash/text.txt b/SquareSquash/text.txt
index 0db79e7..5df338c 100644
--- a/SquareSquash/text.txt
+++ b/SquareSquash/text.txt
@@ -1,1,2 @@
 this is collaborative learning
+ddd
\ No newline at end of file

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/SE WEEK 2/SquareSquash (feature/login)
$ |

```

11. . You're on the main branch and want to start working on a new feature called search-filter. What command do you use?

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git checkout -b searchFilter
Switched to a new branch 'searchFilter'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (searchFilter)
$ |
```

12. You committed a file containing an API key and want to completely remove it from the repo's history. What should you do?

```
baman@karunya MINGW64 ~/OneDrive/Desktop/gitlab (main)
$ git init
Reinitialized existing Git repository in C:/Users/baman/OneDrive/Desktop/gitlab/.git/

baman@karunya MINGW64 ~/OneDrive/Desktop/gitlab (main)
$ echo "API_KEY=12345-FAKE-KEY" >secret.txt

baman@karunya MINGW64 ~/OneDrive/Desktop/gitlab (main)
$ git add secret.txt
warning: in the working copy of 'secret.txt', LF will be replaced by CRLF the next time Git touches it

baman@karunya MINGW64 ~/OneDrive/Desktop/gitlab (main)
$ git commit -m "Accidentally committed API KEY"
[main (root-commit) a64b287] Accidentally committed API KEY
 1 file changed, 1 insertion(+)
 create mode 100644 secret.txt

baman@karunya MINGW64 ~/OneDrive/Desktop/gitlab (main)
$ git filter-branch --force --index-filter \
> "git rm --cached --ignore-unmatch secret.txt" \
> --prune-empty --tag-name-filter cat -- --all
WARNING: git-filter-branch has a glut of gotchas generating mangled history
rewrites. Hit Ctrl-C before proceeding to abort, then use an
alternative filtering tool such as 'git filter-repo'
(https://github.com/newren/git-filter-repo/) instead. See the
filter-branch manual page for more details; to squelch this warning,
set FILTER_BRANCH_SQUELCH_WARNING=1.
Proceeding with filter-branch...

Rewrite a64b287e24239ba4993f606fa83f683a4cca5879 (1/1) (0 seconds passed, remaining 0 predicted)    rm
'secret.txt'

Ref 'refs/heads/main' was deleted
fatal: Not a valid object name HEAD

baman@karunya MINGW64 ~/OneDrive/Desktop/gitlab (main)
$
```



13. You want to see all the branches that exist both locally and on the remote. What

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git branch -a
feature
* main
rese
searchFilter
remotes/origin/HEAD -> origin/main
remotes/origin/feature
remotes/origin/main

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git branch -r
origin/HEAD -> origin/main
origin/feature
origin/main

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$
```

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git branch
feature
* main
rese
searchFilter
```

14. You're on main and want to merge the completed feature/signup branch into it. What command do you use?



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

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git checkout -b feature/signup
fatal: cannot lock ref 'refs/heads/feature/signup': 'refs/heads/feature' exists; cannot create 'refs/heads/feature/signup'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git checkout -b feature1/signup
Switched to a new branch 'feature1/signup'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ echo "<h2> Signup page </h2>">signup.html

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ git add signup.html
warning: in the working copy of 'OneDrive/Desktop/git_seema/signup.html', LF will be replaced by CRLF the next time Git touches it

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ git commit -m "Add signup page"
[feature1/signup 02f3e72] Add signup page
1 file changed, 1 insertion(+)
create mode 100644 OneDrive/Desktop/git_seema/signup.html

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ git checkout main
M       OneDrive/Desktop/git_seema/script.py
Switched to branch 'main'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git merge feature1/signup
Updating 4e7198f..02f3e72
Fast-forward
  OneDrive/Desktop/git_seema/signup.html | 1 +
  1 file changed, 1 insertion(+)
  create mode 100644 OneDrive/Desktop/git_seema/signup.html

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ |
```

16. You don't want Git to track changes to .log files or node\_modules/. How do you achieve this?

```
MINGW64~/OneDrive/Desktop/git_seema
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ ls -a
./ ../ .project README.md secret.txt see.txt

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ touch .gitignore

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git add .gitignore

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git commit -m "ignore log files"
[main ca00198] ignore log files
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 OneDrive/Desktop/git_seema/.gitignore

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ ls
README.md secret.txt see.txt

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$
```

17. You're investigating a bug and want to know who last changed line 25 in script.py. What command do you use?

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git add script1.py

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git commit -m "Added factorial and prime checker script"
[main 4e7198f] Added factorial and prime checker script
1 file changed, 35 insertions(+)

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git blame -L 25,25 script1.py
4e7198f6 (azeemafirdouss 2025-08-06 00:43:05 +0530 25)          return

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$
```

18. You're in the middle of working on a file but need to switch branches quickly. What do you do to save your work?

```

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ ls
1.txt README.md script.py script1.py secret.txt see.txt

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ echo "temp change">see.txt

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git status
warning: could not open directory 'AppData/Local/Application Data/': Permission denied
warning: could not open directory 'Application Data/': Permission denied
warning: could not open directory 'Cookies/': Permission denied
warning: could not open directory 'Local Settings/': Permission denied
warning: could not open directory 'My Documents/': Permission denied
warning: could not open directory 'NetHood/': Permission denied
warning: could not open directory 'PrintHood/': Permission denied
warning: could not open directory 'Recent/': Permission denied
warning: could not open directory 'SendTo/': Permission denied
warning: could not open directory 'Start Menu/': Permission denied
warning: could not open directory 'Templates/': Permission denied
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   script1.py

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:   script.py

Untracked files:
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git stash
warning: in the working copy of 'OneDrive/Desktop/git_seema/script.py'
, LF will be replaced by CRLF the next time Git touches it
Saved working directory and index state WIP on main: 55ed2ef accidentally committed API key

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ |

```

19. You previously ran git stash and now want to restore those changes. What command do you use?

```

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git stash pop
warning: could not open directory 'AppData/Local/Application Data/': Permission denied
warning: could not open directory 'Application Data/': Permission denied
warning: could not open directory 'Cookies/': Permission denied
warning: could not open directory 'Local Settings/': Permission denied
warning: could not open directory 'My Documents/': Permission denied
warning: could not open directory 'NetHood/': Permission denied
warning: could not open directory 'PrintHood/': Permission denied
warning: could not open directory 'Recent/': Permission denied
warning: could not open directory 'SendTo/': Permission denied
warning: could not open directory 'Start Menu/': Permission denied
warning: could not open directory 'Templates/': Permission denied
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   script1.py

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:   script.py

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git stash apply
No stash entries found.

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git stash list

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$

```

20. The feature/test branch is no longer needed. How do you delete it locally?

```

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git checkout -b test/test
Switched to a new branch 'test/test'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (test/test)
$ git branch -d test/test
error: cannot delete branch 'test/test' used by worktree at 'C:/Users/azeem'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (test/test)
$ git checkout main
M       OneDrive/Desktop/git_seema/script.py
A       OneDrive/Desktop/git_seema/script1.py
Switched to branch 'main'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git branch -d test/test
Deleted branch test/test (was 55ed2ef).

```

21. You just merged the feature-ui branch into main. You want to clean up your local branches. How do you safely delete feature-ui?

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git checkout -b feature-ui
Switched to a new branch 'feature-ui'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature-ui)
$ echo "this is new">> file1.txt

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature-ui)
$ git add file1.txt
warning: in the working copy of 'OneDrive/Desktop/git_seema/file1.txt', LF will be replaced by CRLF the next time Git touches it

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature-ui)
$ git commit -m "added a change in feature-ui"
[feature-ui e5abceb] added a change in feature-ui
1 file changed, 1 insertion(+)
 create mode 100644 OneDrive/Desktop/git_seema/file1.txt

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature-ui)
$ git checkout main
M       OneDrive/Desktop/git_seema/script.py
Switched to branch 'main'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git merge feature-ui
Updating 89e91d4..e5abceb
Fast-forward
 OneDrive/Desktop/git_seema/file1.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 OneDrive/Desktop/git_seema/file1.txt

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git branch -d feature-ui
Deleted branch feature-ui (was e5abceb).

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$
```

22. You created a branch `feature-experiment`, made some changes, but now realize the code is no longer needed. You want to delete it, even though it hasn't been merged. What command will you use?

```
Deleted branch test/test (was 99ca2ef).

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git checkout -b features-experiment
Switched to a new branch 'features-experiment'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (features-experiment)
$ echo "hii">new.txt

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (features-experiment)
$ git checkout main
M       OneDrive/Desktop/git_seema/script.py
A       OneDrive/Desktop/git_seema/script1.py
Switched to branch 'main'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git branch -d features-experiment
Deleted branch features-experiment (was 55ed2ef).

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$
```

23. You're currently on the `feature-login` branch and want to delete `feature-ui`. What must you ensure before running the delete command?

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git checkout -b feature-ui
Switched to a new branch 'feature-ui'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature-ui)
$ echo "this is new">> file1.txt

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature-ui)
$ git add file1.txt
warning: in the working copy of 'OneDrive/Desktop/git_seema/file1.txt', LF will
be replaced by CRLF the next time Git touches it

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature-ui)
$ git commit -m "added a change in feature-ui"
[feature-ui c3bf044] added a change in feature-ui
1 file changed, 2 insertions(+)
create mode 100644 OneDrive/Desktop/git_seema/file1.txt

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature-ui)
$ git checkout main
M       OneDrive/Desktop/git_seema/script.py
Switched to branch 'main'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git branch -d feature-ui
error: the branch 'feature-ui' is not fully merged
hint: If you are sure you want to delete it, run 'git branch -D feature-ui'
hint: Disable this message with "git config set advice.forceDeleteBranch false"

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git branch -D feature-ui
Deleted branch feature-ui (was c3bf044).

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ |
```



27°C



Search



24. You want to delete `bugfix-footer`, but you're unsure if it's been merged. What should you do before deleting it?

```
MINGW64~/OneDrive/Desktop/git_seema
Deleted branch feature-ui (was 55ed2ef).

azeeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git checkout -b bugfix-footer
Switched to a new branch 'bugfix-footer'

azeeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (bugfix-footer)
$ echo "fixing footer bug" >> footer.txt

azeeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (bugfix-footer)
$ git add footer.txt
warning: in the working copy of 'OneDrive/Desktop/git_seema/footer.txt', LF will be replaced by CRLF the next time git
touches it

azeeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (bugfix-footer)
$ git commit -m "footer.txt"
[bugfix-footer 89e91d4] footer.txt
2 files changed, 1 insertion(+)
create mode 100644 OneDrive/Desktop/git_seema/footer.txt
create mode 100644 OneDrive/Desktop/git_seema/script1.py

azeeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (bugfix-footer)
$ git checkout main
W       OneDrive/Desktop/git_seema/script.py
Switched to branch 'main'

azeeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git merge bugfix-footer
merge: bugfix-footer - not something we can merge

azeeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git branch
  bugfix-footer
* main
  rebase
  searchFilter

azeeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git merge bugfix-footer
Updating 55ed2ef..89e91d4
Fast-forward
 OneDrive/Desktop/git_seema/footer.txt | 1 +
 OneDrive/Desktop/git_seema/script1.py | 0
2 files changed, 1 insertion(+)
create mode 100644 OneDrive/Desktop/git_seema/footer.txt
create mode 100644 OneDrive/Desktop/git_seema/script1.py

azeeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
```

```
azeeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git merge bugfix-footer
Updating 55ed2ef..89e91d4
Fast-forward
 OneDrive/Desktop/git_seema/footer.txt | 1 +
 OneDrive/Desktop/git_seema/script1.py | 0
2 files changed, 1 insertion(+)
create mode 100644 OneDrive/Desktop/git_seema/footer.txt
create mode 100644 OneDrive/Desktop/git_seema/script1.py

azeeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git branch -d bugfix-footer
Deleted branch bugfix-footer (was 89e91d4).

azeeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ |
```

25. You finished working on `feature-a`, `feature-b`, and `feature-c`. All have been merged into `main`. How do you delete them in one command?

```
Deleted branch bugfix-footer (was 89e91d4).

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git checkout -b feature-a
Switched to a new branch 'feature-a'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature-a)
$ git checkout main
M       OneDrive/Desktop/git_seema/script.py
Switched to branch 'main'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git checkout -b feature-b
error: switch `b` requires a value

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git checkout -b feature-b
Switched to a new branch 'feature-b'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature-b)
$ git checkout main
M       OneDrive/Desktop/git_seema/script.py
Switched to branch 'main'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git checkout -b feature-c
Switched to a new branch 'feature-c'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature-c)
$ git checkout main
M       OneDrive/Desktop/git_seema/script.py
Switched to branch 'main'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git branch -d feature-a feature-b feature-c
Deleted branch feature-a (was 89e91d4).
Deleted branch feature-b (was 89e91d4).
Deleted branch feature-c (was 89e91d4).

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ |
```

1. Specify the git command when you're starting work on a project and need to clone a remote repository to your local machine.

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$ git clone https://github.com/ayusharma/birthday
Cloning into 'birthday'...
remote: Enumerating objects: 94, done.
remote: Total 94 (delta 0), reused 0 (delta 0), pack-reused 94 (from 1)
Receiving objects: 100% (94/94), 2.53 MiB | 194.00 KiB/s, done.
Resolving deltas: 100% (30/30), done.

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$
```

2. Specify the git command if you want to see the remotes that are connected to your local repository.

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$ git remote -v
origin https://github.com/azeemafirdouss/Test.git (fetch)
origin https://github.com/azeemafirdouss/Test.git (push)

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$
```

3. Specify the git command if you want to add a new remote repository to your local repository.

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$ git remote add origin https://github.com/azeemafirdouss/Test.git
```

4. Specify the git command to remove a remote repository from your local configuration:

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$ git remote remove origin
```

5. Specify the git command if you want to rename an existing remote:

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$ git remote rename Bday bd
```

6. Specify the git command to fetch updates from the remote repository but not merge them into your local branch.

```
azeem@LAPTOP-MUVD4INQ MINGW64
$ git fetch
```

7. Specify the git command to pull the latest changes from the remote repository and merge them into your local branch.

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$ git pull bd
Already up to date.
```

8. Specify the git command to push your local commits to a remote repository.

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$ git push bd
Everything up-to-date
```

9. Specify the git command to when pushing for the first time and want to set the remote branch you are pushing to.

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$ git push --set-upstream bd main
branch 'main' set up to track 'bd/main'.
Everything up-to-date
```

10. Specify the git command to change the URL of a remote (e.g., after changing the remote repository address).

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$ git remote set-url bd https://github.com/azeemafirdouss/Test.git
```

11. Specify the git command if you want to list all branches on the remote repository.

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$ git branch -r
bd/HEAD -> bd/main
bd/main
```

12. Specify the git command if a branch was deleted on the remote but still shows up locally.

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$ git remote prune bd
```

13. Specify the git command to fetch a specific branch from a remote.

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$ git fetch bd
```

14. Specify the git command if you want to see detailed information about a remote repository.

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$ git remote show bd
* remote bd
Fetch URL: https://github.com/azeemafirdouss/Test.git
Push URL: https://github.com/azeemafirdouss/Test.git
HEAD branch: main
Remote branch:
  main tracked
Local branch configured for 'git pull':
  main merges with remote main
Local ref configured for 'git push':
  main pushes to main (up to date)
```

15. Specify the git command if you want to rebase your local branch onto a remote branch (this can be useful to keep your history linear)

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$ git pull --rebase bd
Already up to date.
```