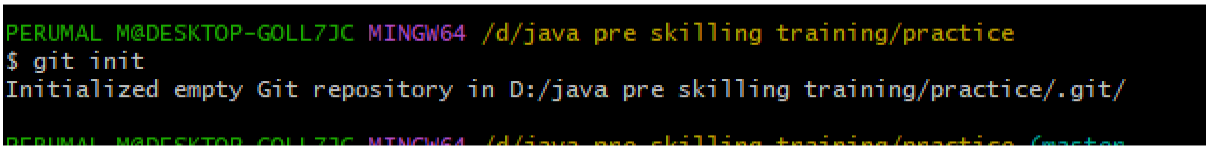
PERUMAL M

ASSIGNMENT-1

Initialize a New Git Repository

For initialize a new Git repository I use the ‘git init ‘command

Command: git init

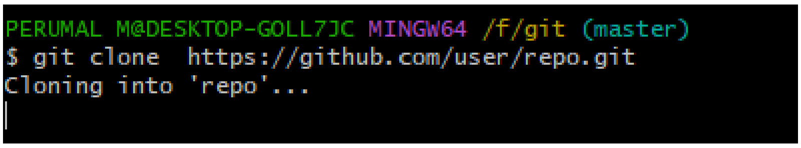


```
PERUMAL M@DESKTOP-GOLL7JC MINGW64 /d/java pre skilling training/practice
$ git init
Initialized empty Git repository in D:/java pre skilling training/practice/.git/

PERUMAL M@DESKTOP-GOLL7JC MINGW64 /d/java pre skilling training/practice (master
```

Clone a Repository

To clone a repository from GitHub using its URL and use the git clone command followed by the repository URL

Command: git clone https://github.com/user/repo.git

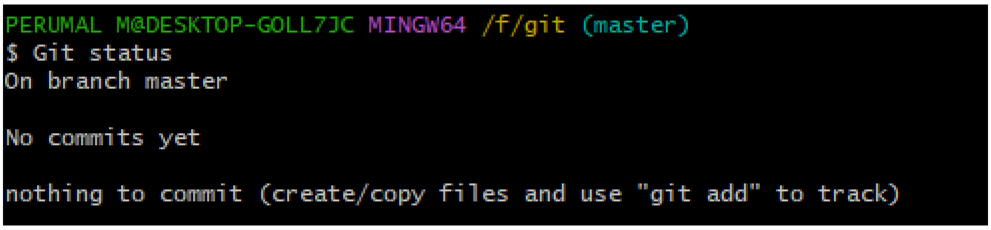


```
PERUMAL M@DESKTOP-GOLL7JC MINGW64 /f/git (master)
$ git clone  https://github.com/user/repo.git
Cloning into 'repo'...
```

Check the Status of the Repository

To check the status of the repository, use the git status command.

Command: git status

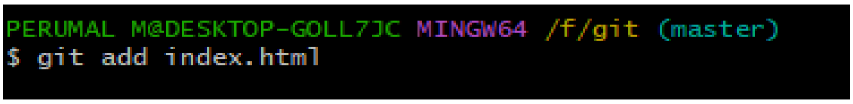


```
PERUMAL M@DESKTOP-GOLL7JC MINGW64 /f/git (master)
$ Git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

Add a File to the Staging Area

To add a file named index.html to the staging area, I use the git add command followed by the filename

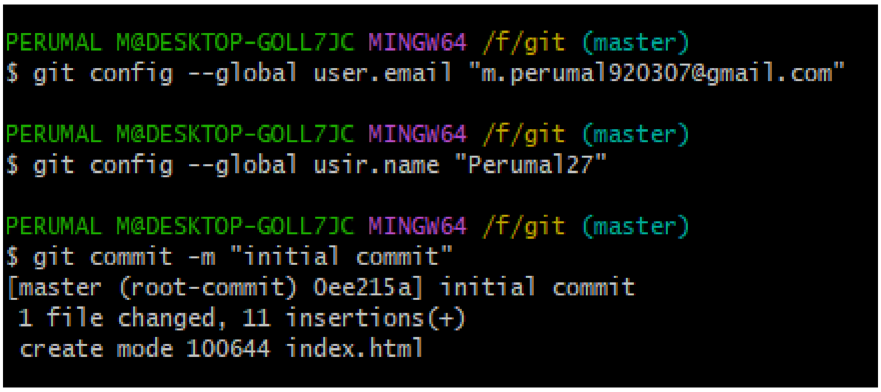
Command: git add file name



```
PERUMAL M@DESKTOP-GOLL7JC MINGW64 /f/git (master)
$ git add index.html
```

Commit Changes

Command: git commit -m "Initial commit”

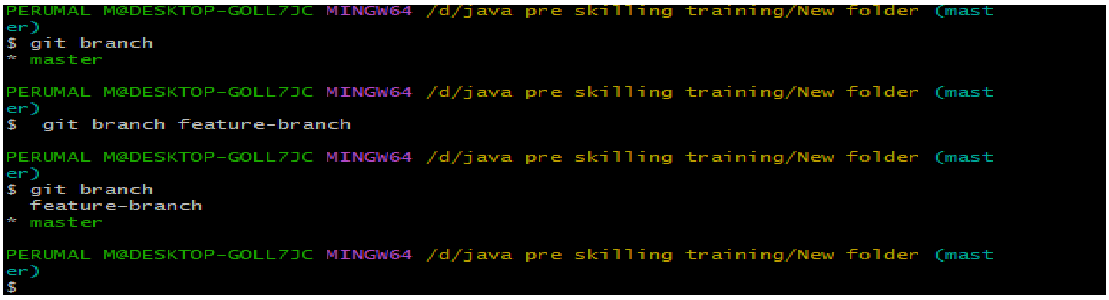


```
PERUMAL M@DESKTOP-GOLL7JC MINGW64 /f/git (master)
$ git config --global user.email "m.perumal920307@gmail.com"

PERUMAL M@DESKTOP-GOLL7JC MINGW64 /f/git (master)
$ git config --global usir.name "Perumal27"

PERUMAL M@DESKTOP-GOLL7JC MINGW64 /f/git (master)
$ git commit -m "initial commit"
[master (root-commit) 0ee215a] initial commit
 1 file changed, 11 insertions(+)
 create mode 100644 index.html
```

Create a New Branch

For create a new branch use the git branch command followed by the branch name (feature-branch in this case) to create the branch.

Command: git branch feature-branch

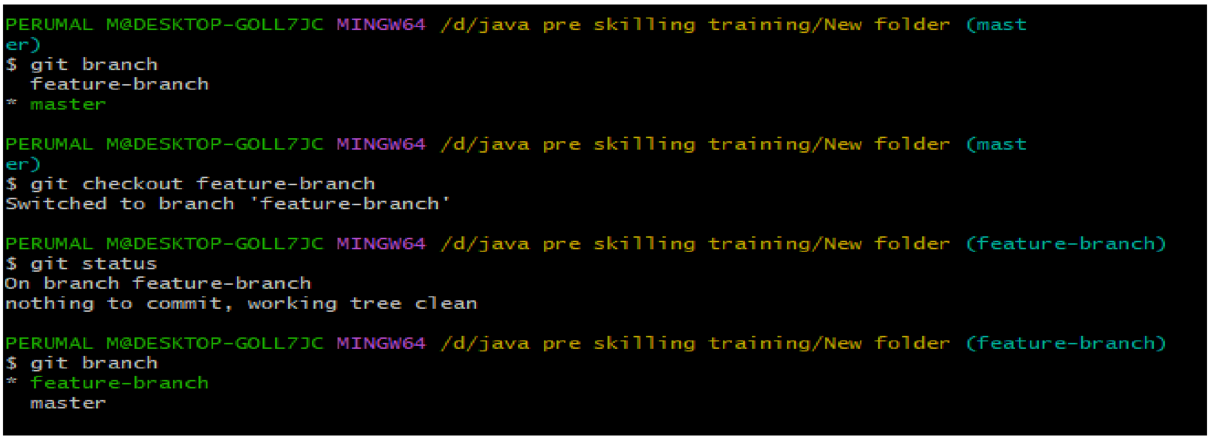


```
PERUMAL M@DESKTOP-GOLL7JC MINGW64 /d/java pre skilling training/New folder (mast
er)
$ git branch
* master

PERUMAL M@DESKTOP-GOLL7JC MINGW64 /d/java pre skilling training/New folder (mast
er)
$  git branch feature-branch

PERUMAL M@DESKTOP-GOLL7JC MINGW64 /d/java pre skilling training/New folder (mast
er)
$ git branch
  feature-branch
* master

PERUMAL M@DESKTOP-GOLL7JC MINGW64 /d/java pre skilling training/New folder (mast
er)
$
```

Switch to Another Branch

For switch the new branch use the command git checkout and followed by branch name

Command: git checkout feature-branch

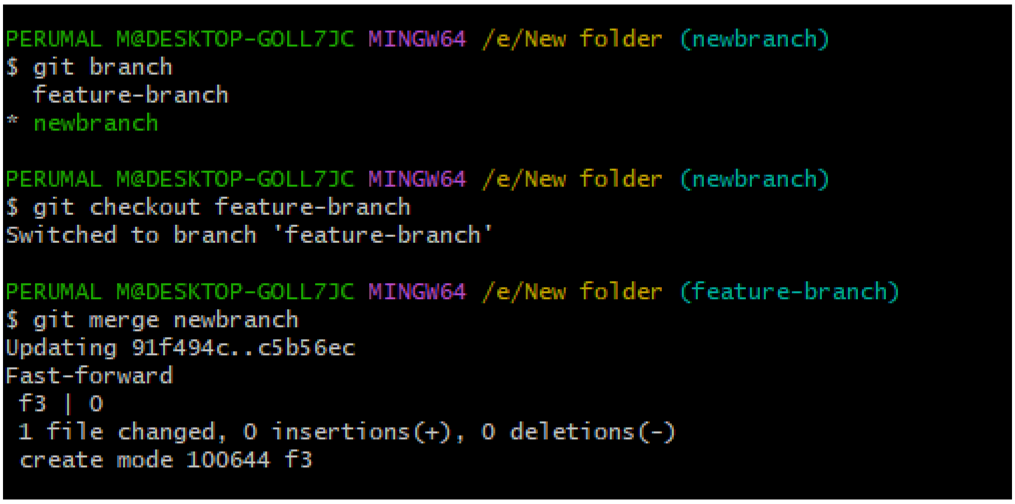


```
PERUMAL M@DESKTOP-GOLL7JC MINGW64 /d/java pre skilling training/New folder (mast
er)
$ git branch
  feature-branch
* master

PERUMAL M@DESKTOP-GOLL7JC MINGW64 /d/java pre skilling training/New folder (mast
er)
$ git checkout feature-branch
Switched to branch 'feature-branch'

PERUMAL M@DESKTOP-GOLL7JC MINGW64 /d/java pre skilling training/New folder (feature-branch)
$ git status
On branch feature-branch
nothing to commit, working tree clean

PERUMAL M@DESKTOP-GOLL7JC MINGW64 /d/java pre skilling training/New folder (feature-branch)
$ git branch
* feature-branch
  master
```

Merge Branches

For merge two branches use the command git merge 'branch name'

Command: git merge feature-branch

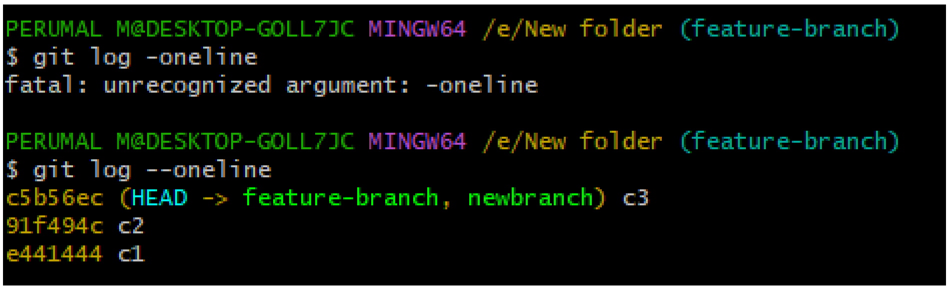


```
PERUMAL M@DESKTOP-GOLL7JC MINGW64 /e/New folder (newbranch)
$ git branch
  feature-branch
* newbranch

PERUMAL M@DESKTOP-GOLL7JC MINGW64 /e/New folder (newbranch)
$ git checkout feature-branch
Switched to branch 'feature-branch'

PERUMAL M@DESKTOP-GOLL7JC MINGW64 /e/New folder (feature-branch)
$ git merge newbranch
Updating 91f494c..c5b56ec
Fast-forward
 f3 | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 f3
```

View Commit History

For view commit history use the git log command

Command: git log or git log –oneline



```
PERUMAL M@DESKTOP-GOLL7JC MINGW64 /e/New folder (feature-branch)
$ git log -oneline
fatal: unrecognized argument: -oneline

PERUMAL M@DESKTOP-GOLL7JC MINGW64 /e/New folder (feature-branch)
$ git log --oneline
c5b56ec (HEAD -> feature-branch, newbranch) c3
91f494c c2
e441444 c1
```

Fetch Updates from Remote Repository

For fetch update from remote repository use command git fetch

Command : git fetch origin

-Origin is repository name

Push Changes to Remote Repository

To push changes to the remote repository named origin on the main branch

Command: git push origin main

Revert a Commit

To revert a commit use the command git revert commit id

Command: git revert a1b2c3d

Create and Apply a Stash

To create a stash command

Command: git stash -m ‘create ‘name’

To apply a stash command

Then go to command git stash list

Command: git stash apply stash@{0}

List All Branches

To list the all branches

Command: git branch

Delete a Branch

To delete the old branch name old-feature

Command: git branch -d old-feature

Check the Differences Between Two Commits

To check the difference between two commits with hashes a1b2c3d and d4e5f6g

Command: git diff a1b2c3d d4e5f6g

Create an Annotated Tag

To create annotated tag named v1.0 with the message "First release"

Command: git tag -a v1.0 -m “first release”

List Tags

To list the all tags

Command: git tag

Configure Git Username and Email

Command: git config –global user.name “perumal27”

git config –global user.email abcdfg@gmail.com

Remove a File from the Staging Area

Command: `git restore –staged file.txt`

View Changes in the Working Directory

Command: `git diff head`

Create a New Remote Repository

Command: `git remote add origin https://github.com/user/repo.git`

Rename a Branch

Command: `git branch -m “old branch name” “new branchname”`

Show the Commit History in a Compact Format

Command: `git log --oneline`

Amend the Last Commit

Command: `git commit --amend  -m  ”updated commit”`