sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (master)

$ git add .\*

sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (master)

$ git status

On branch master

No commits yet

Changes to be committed:

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

new file: a.txt.txt

new file: b.txt.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (master)

$ git config--global user.email"sirugumpu.pavansaikiran@gmail.com"

git: 'config--global' is not a git command. See 'git --help'.

sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (master)

$ git config --global user.email"sirugumpu.pavansaikiran@gmail.com"

sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (master)

$ git config global user.name "pavan"

error: key does not contain a section: global

sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (master)

$ git config --global user.name"pavan"

sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (master)

$ git commit-m

git: 'commit-m' is not a git command. See 'git --help'.

The most similar command is

commit-tree

sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (master)

$ git commit -m

error: switch `m' requires a value

sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (master)

$ git commit -m"this is my first commit"

[master (root-commit) b63a461] this is my first commit

2 files changed, 0 insertions(+), 0 deletions(-)

create mode 100644 a.txt.txt

create mode 100644 b.txt.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (master)

$ git status

On branch master

nothing to commit, working tree clean

sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (master)

$ git log

commit b63a461ed594d9bdaa21025e0ef324d2b79d43dc (HEAD -> master)

Author: pavansaikiran407 <sirugumpu.pavansaikiran@gmail.com>

Date: Wed Aug 14 23:14:16 2024 +0530

this is my first commit

sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (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: a.txt.txt

modified: b.txt.txt

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

sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (master)

$ git commit -a -m"both files are modified"

[master 1c107c8] both files are modified

2 files changed, 2 insertions(+)

sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (master)

$ git status

On branch master

nothing to commit, working tree clean

sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (master)

$ git log

commit 1c107c8ffa96caea571653390be065ec415857a6 (HEAD -> master)

Author: pavansaikiran407 <sirugumpu.pavansaikiran@gmail.com>

Date: Wed Aug 14 23:29:17 2024 +0530

both files are modified

commit b63a461ed594d9bdaa21025e0ef324d2b79d43dc

Author: pavansaikiran407 <sirugumpu.pavansaikiran@gmail.com>

Date: Wed Aug 14 23:14:16 2024 +0530

this is my first commit

sree@DESKTOP-2B28HN3 MINGW64 /c/myproject (master)

$

GIT DIFF COMMAND:

Req1: To see the difference in file content between working directory and staging area

Git Diff Command

Syntax:git diff name of the file

Git diff index.txt

$ git diff index.txt

**diff --git a/index.txt b/index.txt**

**index d29ee3f..f6cb171 100644**

**--- a/index.txt**

**+++ b/index.txt**

@@ -1 +1,2 @@

-animals

\ No newline at end of file

+animals

+birds

\ No newline at end of file

**a/index.txt -🡪 represent source(staging area)**

**b/index.txt -🡪represent destination(working directory)**

**d29ee3f--🡪hash of file content from source/staging**

**f6cb171 -🡪hash of file content from destination/workspace**

**100644-🡪 git file mode**

**100—represent type of file**

**644-file permissions**

100644-🡪 git file mode

100 -- represents type of the file

644 - File permissions rw-r-r

4-r

2 W

1-e

---a/index.txt 🡪Source file missing some lines (staging)

+++ b/index.txt 🡪New lines added in destination file (working dir)

if anyline prefixed with space means it is unchanged.

if anyline prefixed with + means it is added in destination copy.

if anyline prefixed with**-** means it is removed from destination copy

Req2: To see the difference in File Content between Working Directory and Last Commit.

\* Last commit refred using HEAD

git diff HEAD index.txt

Req3: To see the difference in File Content between staged Copy and Last Commit

git diff --staged HEAD index.txt

Req4: To see the difference in File Content between specific Commit and Working Directory Copy

git log --oneline

git diff 79b859c index.txt

Req5: To see the difference in file content between specific commit and staging area copy:

Git diff –staged 79b859c

Req6: To see the difference in file content between 2 specified commits

Git diff d7022f3 79b859c index.txt

Req7:To see the difference in content between 2 branches

Master

Test

Git diff master test 🡪it shows all the differences between master branch and test branch

Req8: To see the differences in content between local and remote repositry

Git diff master origin/master

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject

$ ls

index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject

$ git init

Initialized empty Git repository in C:/sampleproject/.git/

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git add .

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git status

On branch master

No commits yet

Changes to be committed:

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

new file: index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git status

On branch master

No commits yet

Changes to be committed:

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

new file: index.txt

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: index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git diff index.txt

**diff --git a/index.txt b/index.txt**

**index d29ee3f..f6cb171 100644**

**--- a/index.txt**

**+++ b/index.txt**

@@ -1 +1,2 @@

-animals

\ No newline at end of file

+animals

+birds

\ No newline at end of file

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ f^C

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git status

On branch master

No commits yet

Changes to be committed:

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

new file: index.txt

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: index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git diff index.txt

**diff --git a/index.txt b/index.txt**

**index d29ee3f..f6cb171 100644**

**--- a/index.txt**

**+++ b/index.txt**

@@ -1 +1,2 @@

-animals

\ No newline at end of file

+animals

+birds

\ No newline at end of file

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git commit -m"firstcommit"

[master (root-commit) 79b859c] firstcommit

1 file changed, 1 insertion(+)

create mode 100644 index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git diff head index.txt

**diff --git a/index.txt b/index.txt**

**index d29ee3f..50af968 100644**

**--- a/index.txt**

**+++ b/index.txt**

@@ -1 +1,3 @@

-animals

\ No newline at end of file

+animals

+birds

+fishes

\ No newline at end of file

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git add .

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git add .

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git diff --staged HEAD index.txt

**diff --git a/index.txt b/index.txt**

**index d29ee3f..50af968 100644**

**--- a/index.txt**

**+++ b/index.txt**

@@ -1 +1,3 @@

-animals

\ No newline at end of file

+animals

+birds

+fishes

\ No newline at end of file

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git commit -m""second commit"

>

> clear

> git status

> /

> clear

>

>

>

>

>

>

> ^C

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git commit -m"second commit"

[master d7022f3] second commit

1 file changed, 3 insertions(+), 1 deletion(-)

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git log--oneline

git: 'log--oneline' is not a git command. See 'git --help'.

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git log --oneline

d7022f3 (HEAD -> master) second commit

79b859c firstcommit

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git diff 79b859c index.txt

**diff --git a/index.txt b/index.txt**

**index d29ee3f..7c0f535 100644**

**--- a/index.txt**

**+++ b/index.txt**

@@ -1 +1,4 @@

-animals

\ No newline at end of file

+animals

+birds

+fishes

+trees

\ No newline at end of file

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git diff --staged 79b859c

**diff --git a/index.txt b/index.txt**

**index d29ee3f..50af968 100644**

**--- a/index.txt**

**+++ b/index.txt**

@@ -1 +1,3 @@

-animals

\ No newline at end of file

+animals

+birds

+fishes

\ No newline at end of file

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git log --oneline

d7022f3 (HEAD -> master) second commit

79b859c firstcommit

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git diff d7022f3 79b859c index.txt

**diff --git a/index.txt b/index.txt**

**index 50af968..d29ee3f 100644**

**--- a/index.txt**

**+++ b/index.txt**

@@ -1,3 +1 @@

-animals

-birds

-fishes

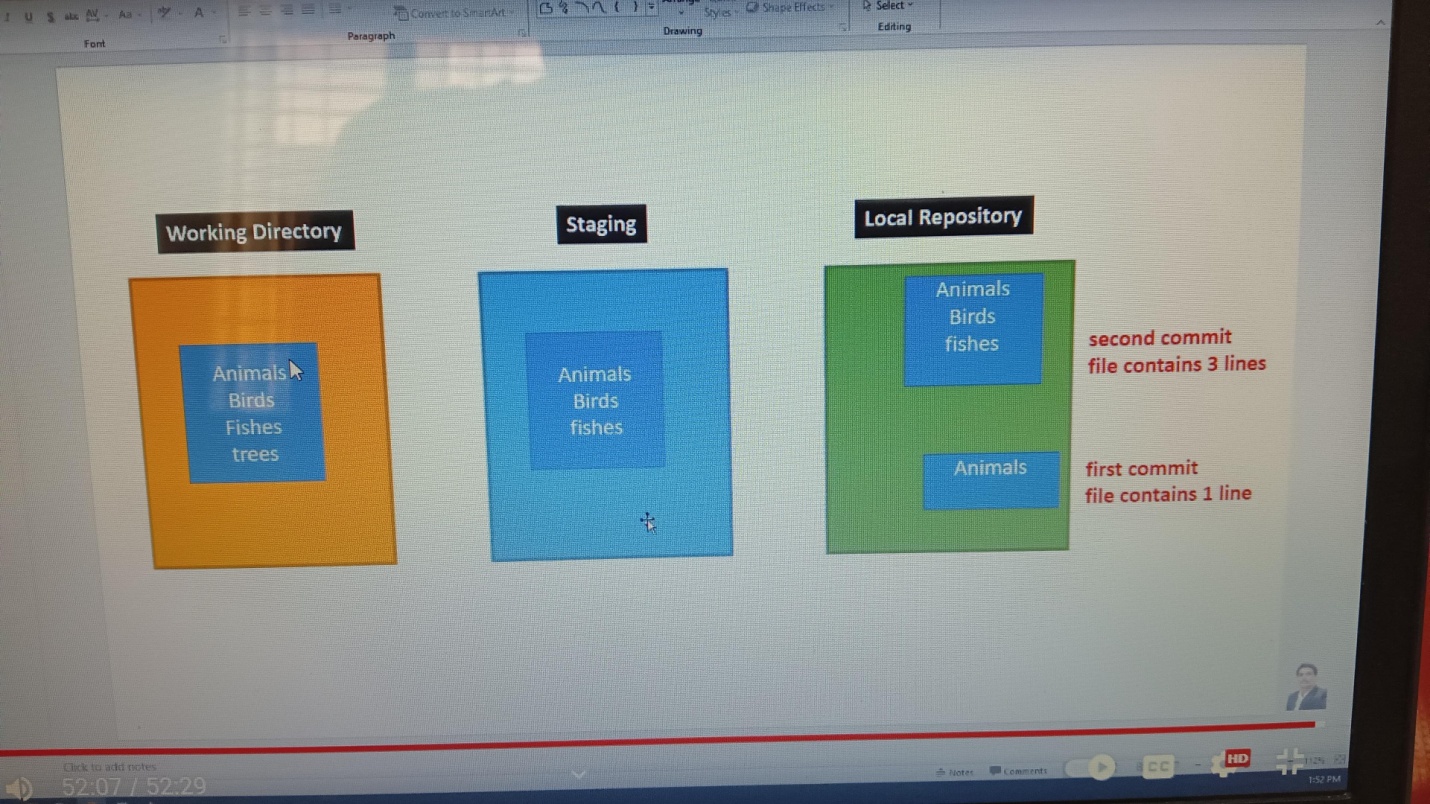
\ No newline at end of file

+animals

\ No newline at end of file

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$



------------------------------------------------------------------------------------------------------------------------------------------

GIT RM:

Req1:Remove files from both staging & working directory

Syntax:

Git rm filename:it removes files from both staging & working directory

Example:

Git rm file1.txt:

Git rm –r . 🡪 :it will removes all files from both staging & working directory

Req2:Remove files from only from staging

Syntax:

Git rm –cached filename

Example:

Git rm –cached file2.txt-----🡪 it will remove files only from staging

Req3:Remove files only from working directory

Syntax:

Rm filename -🡪 this command removes files from working directory only

Rm file3.txt: removes files only working directory

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git init

Reinitialized existing Git repository in C:/sampleproject/.git/

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git add file1.txt file2.txt file3.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: file1.txt

new file: file2.txt

new file: file3.txt

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: index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git commit -m"3 files commiting in first commiting in first round"

[master 0cd6781] 3 files commiting in first commiting in first round

3 files changed, 0 insertions(+), 0 deletions(-)

create mode 100644 file1.txt

create mode 100644 file2.txt

create mode 100644 file3.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (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: index.txt

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

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ ls

file1.txt file2.txt file3.txt index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git ls-files

file1.txt

file2.txt

file3.txt

index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git rm file1.txt

rm 'file1.txt'

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ ls

file2.txt file3.txt index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ ls

file2.txt file3.txt index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git ls -files

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

The most similar command is

lfs

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git ls-files

file2.txt

file3.txt

index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git rm --cached file2.txt

rm 'file2.txt'

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ ls

file2.txt file3.txt index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git ls-files

file3.txt

index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ ls

file2.txt file3.txt index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ git ls-files

file3.txt

index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ rm file3.txt

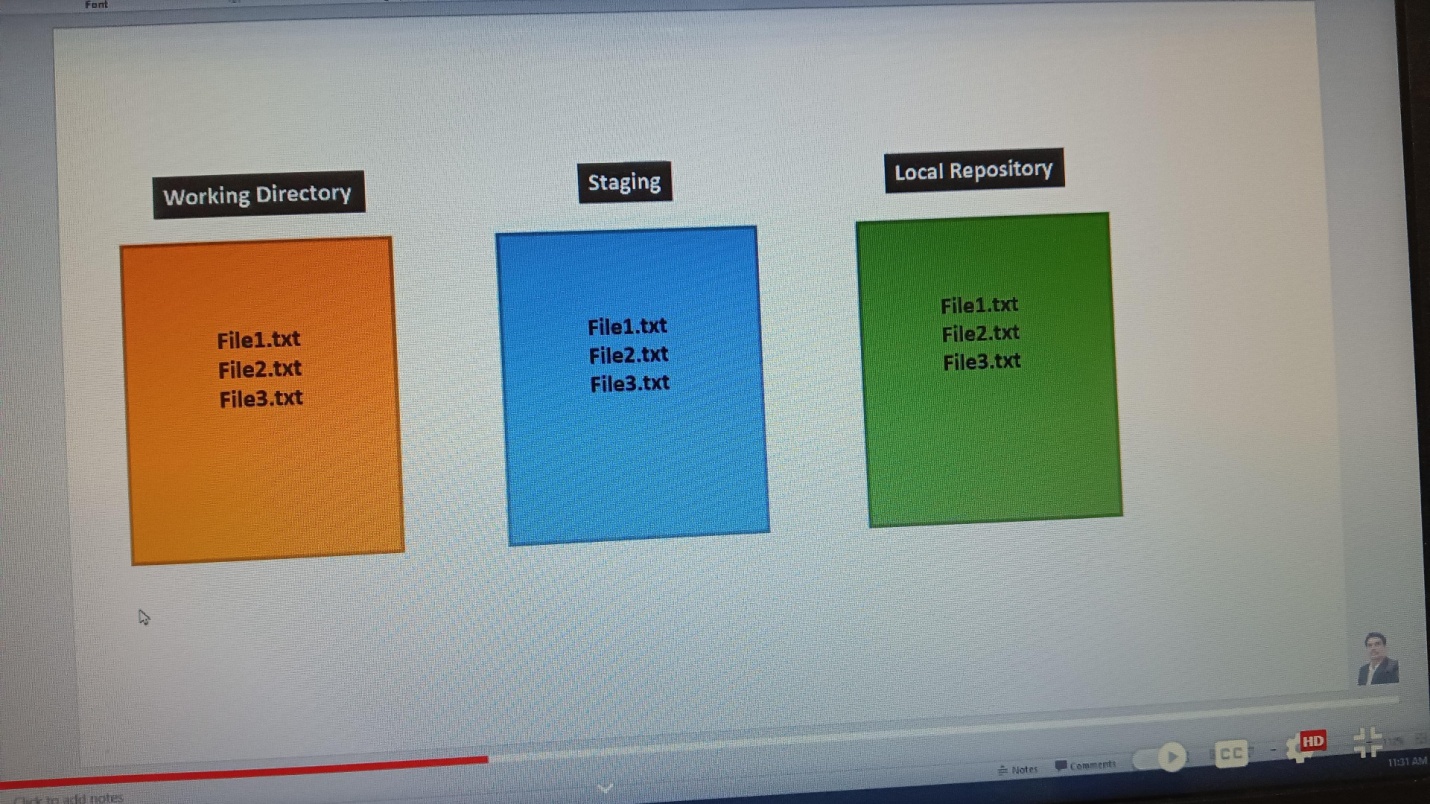
sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$ ls

file2.txt index.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/sampleproject (master)

$



GIT Checkout command

We can use checkout command to discard unstaged changes in tracked files of working directory

1.This command is only for working directory

2.To discard unstaged changes(The changes which are not added to staging area)

3.In the tracked files(the files which are already added to staging /commit)

Checkout command will do UNDO operation

Command :

Syntax:

cat filename: It shows content inside file

example :cat file1.txt

synatx:

git checkout – filename

example:

git checkout – file1.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/checkout

$ git init

Initialized empty Git repository in C:/checkout/.git/

sree@DESKTOP-2B28HN3 MINGW64 /c/checkout (master)

$ git add .

sree@DESKTOP-2B28HN3 MINGW64 /c/checkout (master)

$ git commit -m"first commit"

[master (root-commit) b414d4c] first commit

1 file changed, 2 insertions(+)

create mode 100644 file1.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/checkout (master)

$ ls

file1.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/checkout (master)

$ git -lsfiles

unknown option: -lsfiles

usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]

[--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]

[-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]

[--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]

[--config-env=<name>=<envvar>] <command> [<args>]

sree@DESKTOP-2B28HN3 MINGW64 /c/checkout (master)

$ cat file1.txt

this is first line

this is second line

sree@DESKTOP-2B28HN3 MINGW64 /c/checkout (master)

$ cat file1.txt

this is first line

this is second line

this is third line

this is fourth line

sree@DESKTOP-2B28HN3 MINGW64 /c/checkout (master)

$ git checkout -- file1.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/checkout (master)

$ cat file1.txt

this is first line

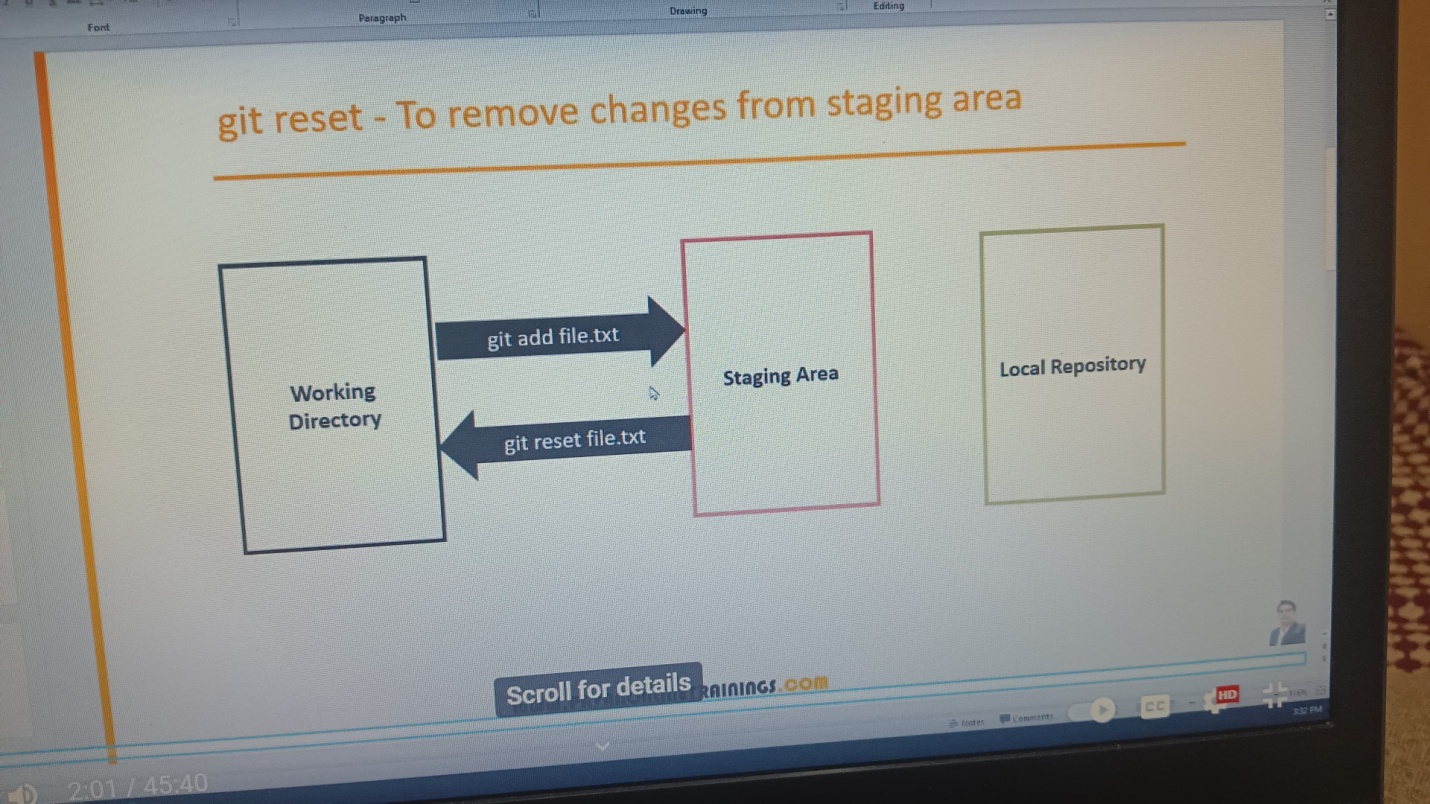
this is second line

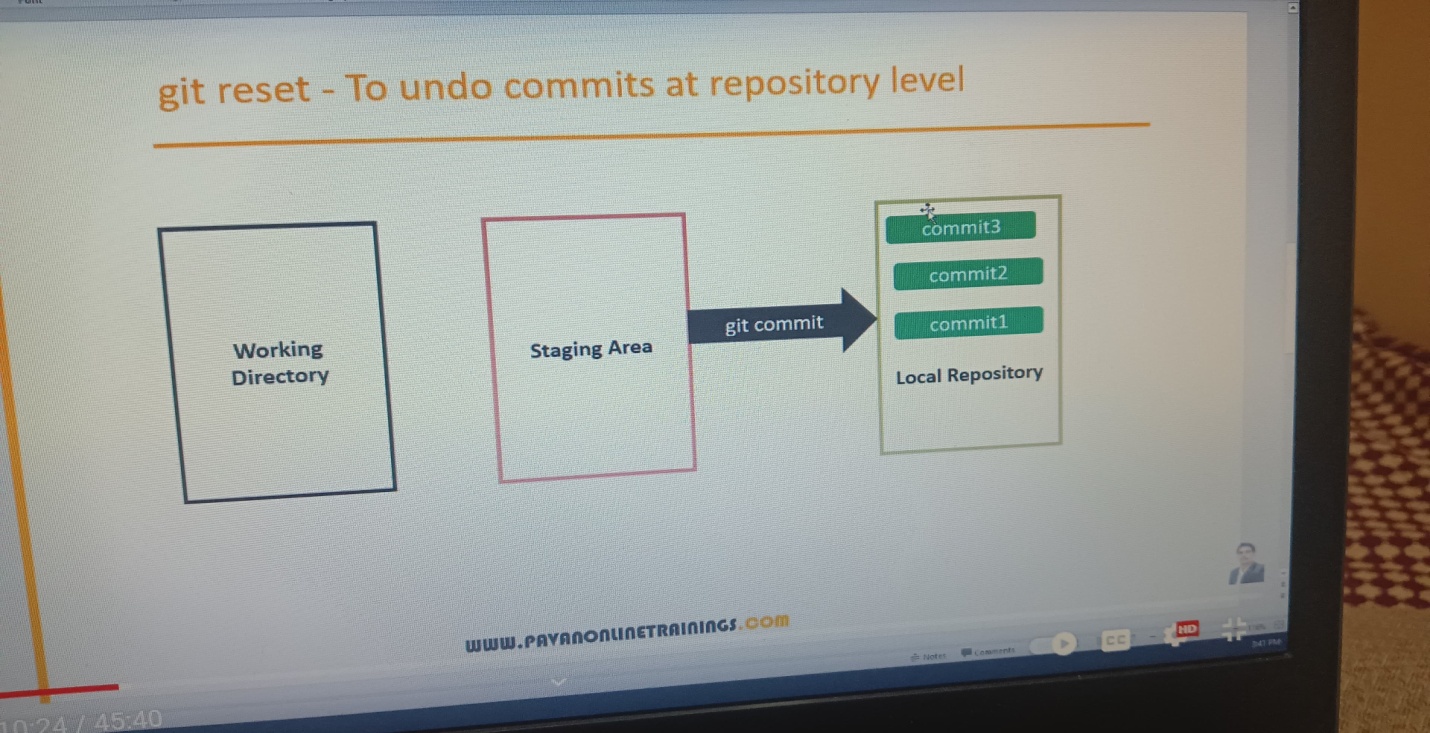
sree@DESKTOP-2B28HN3 MINGW64 /c/checkout (master)

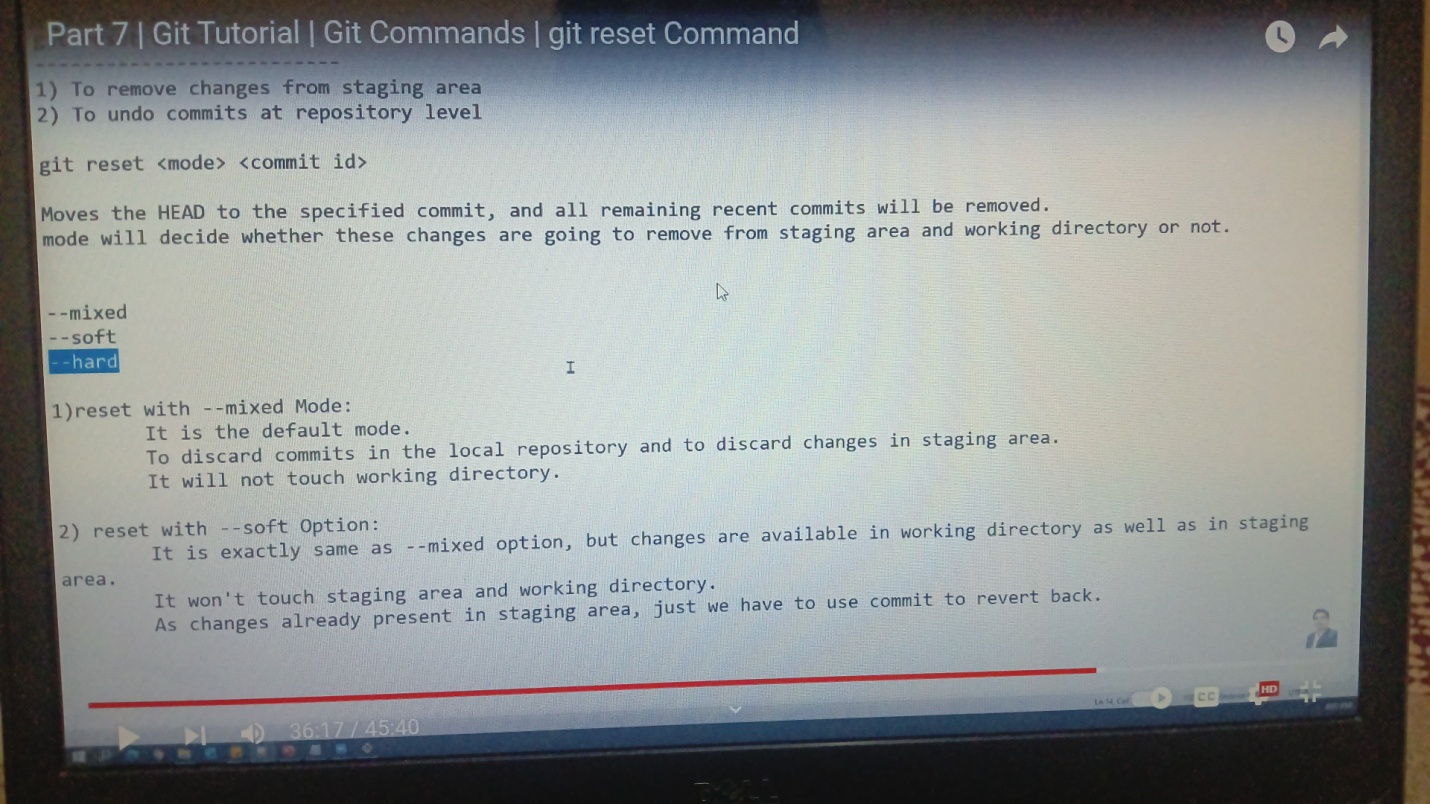
$ git checkout -- file2.txt

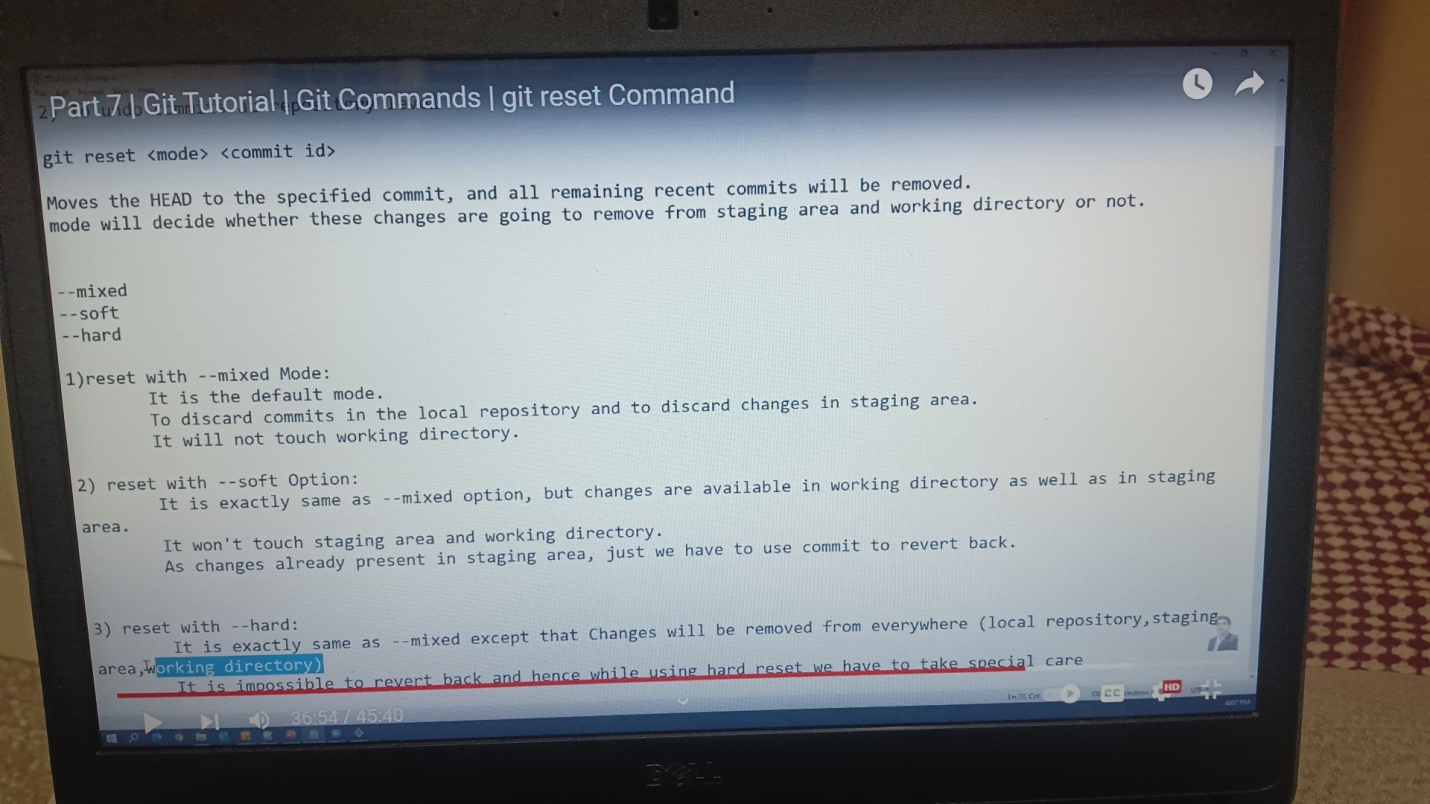
error: pathspec 'file2.txt' did not match any file(s) known to git

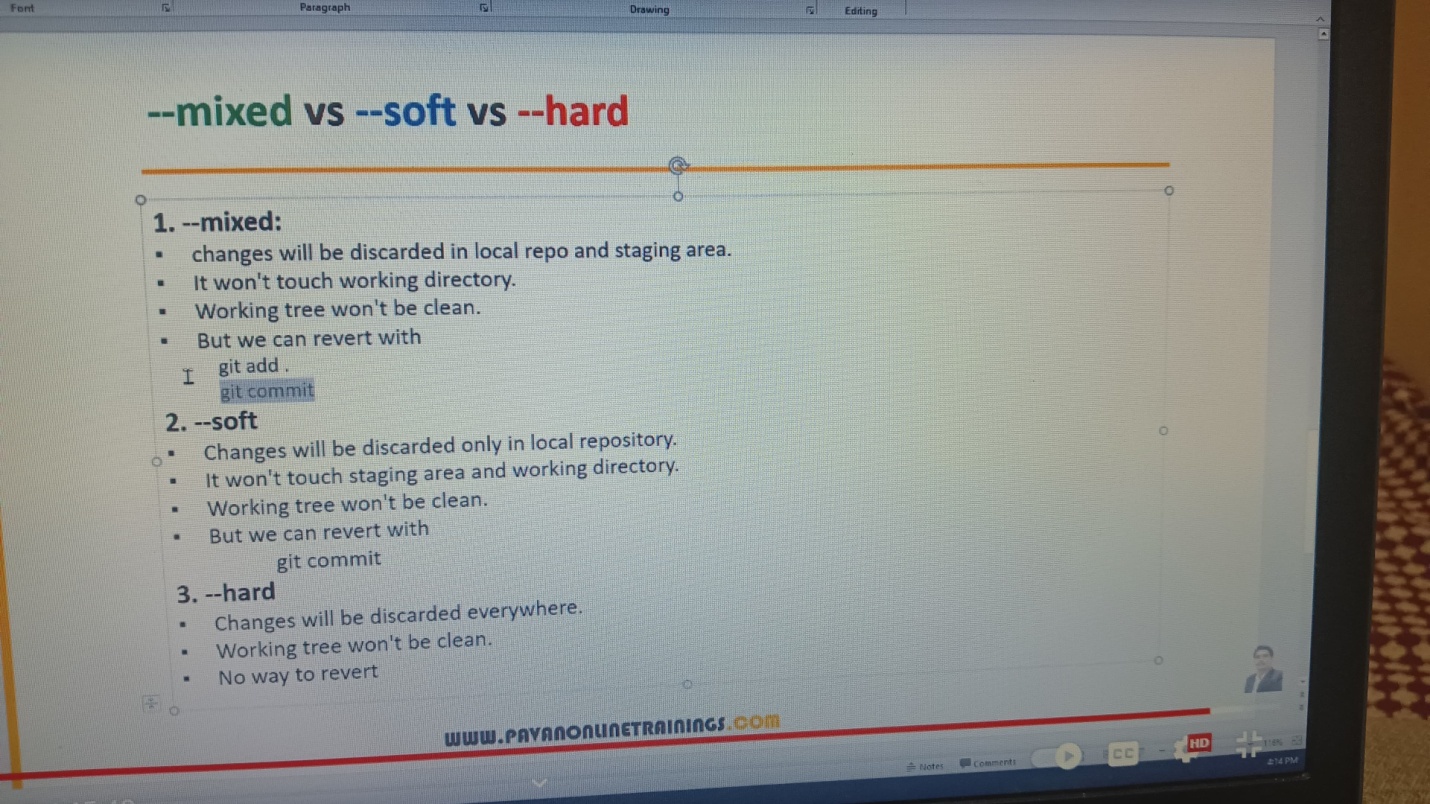
GIT RESET COMMAND:

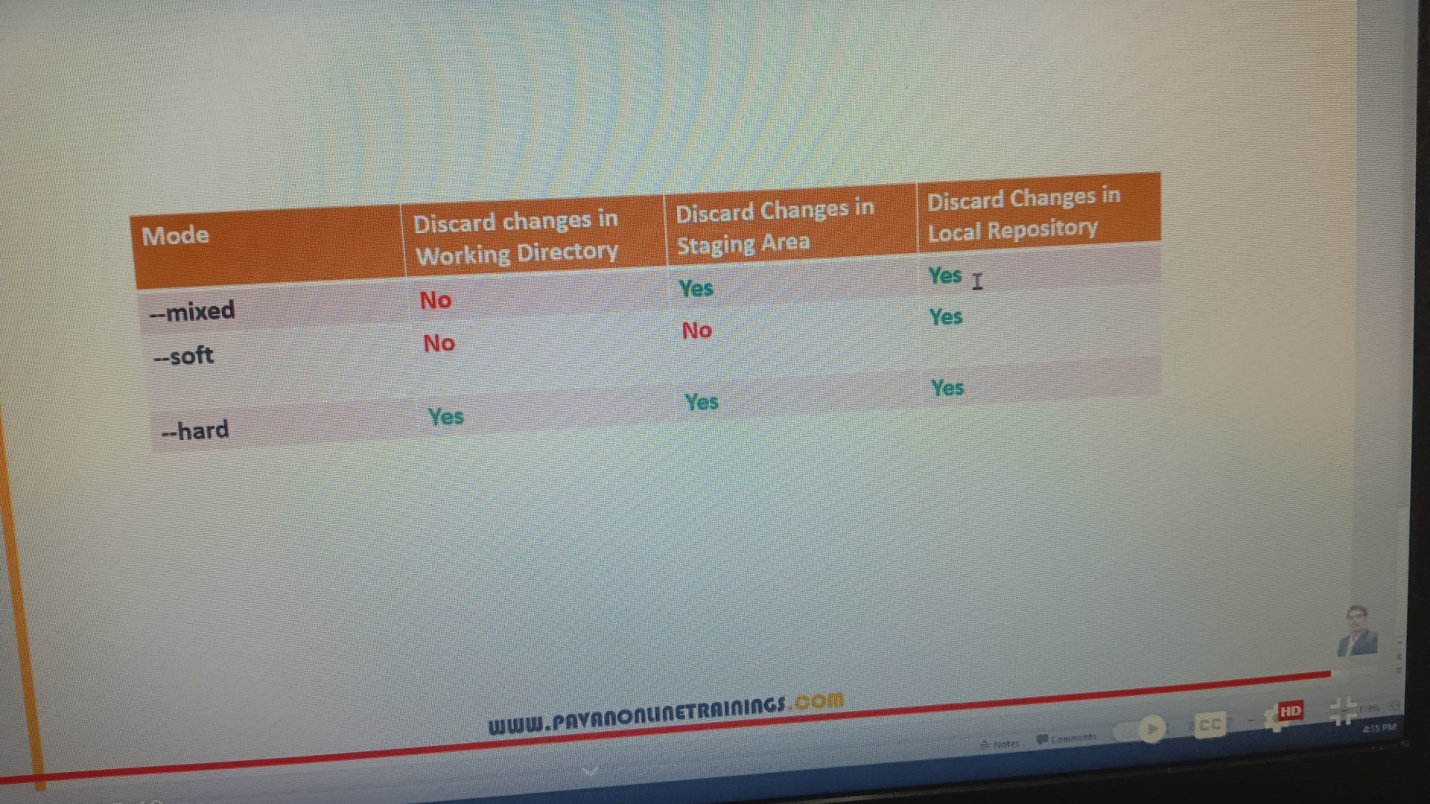












sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ ls

file1.txt file2.txt file3.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git add file1.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git commit -m"committed file1"

[master (root-commit) 40e956f] committed file1

1 file changed, 1 insertion(+)

create mode 100644 file1.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git add file2.txt;git commit -m"committed file2"

[master c694ced] committed file2

1 file changed, 1 insertion(+)

create mode 100644 file2.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git add file3.txt;git commit -m"committed file3"

[master e59029f] committed file3

1 file changed, 1 insertion(+)

create mode 100644 file3.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ ls

file1.txt file2.txt file3.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git ls-files

file1.txt

file2.txt

file3.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git log--oneline

git: 'log--oneline' is not a git command. See 'git --help'.

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git log --oneline

e59029f (HEAD -> master) committed file3

c694ced committed file2

40e956f committed file1

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git reset --mixed c694ced

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git log --oneline

c694ced (HEAD -> master) committed file2

40e956f committed file1

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ ls

file1.txt file2.txt file3.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git ls-files

file1.txt

file2.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git log--oneline

git: 'log--oneline' is not a git command. See 'git --help'.

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git log --oneline

c694ced (HEAD -> master) committed file2

40e956f committed file1

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ ls

file1.txt file2.txt file3.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git ls-files

file1.txt

file2.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git add file3.txt;

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git commit -m"adding file3 again"

[master 80fad55] adding file3 again

1 file changed, 1 insertion(+)

create mode 100644 file3.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ ls

file1.txt file2.txt file3.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git ls-files

file1.txt

file2.txt

file3.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git log--oneline

git: 'log--oneline' is not a git command. See 'git --help'.

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git log --oneline

80fad55 (HEAD -> master) adding file3 again

c694ced committed file2

40e956f committed file1

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git reset --soft c694ced

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git log --oneline

c694ced (HEAD -> master) committed file2

40e956f committed file1

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ ls

file1.txt file2.txt file3.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git ls-files

file1.txt

file2.txt

file3.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git commit -m"again commited file3"

[master 3e6752a] again commited file3

1 file changed, 1 insertion(+)

create mode 100644 file3.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git log --oneline

3e6752a (HEAD -> master) again commited file3

c694ced committed file2

40e956f committed file1

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git reset --hard 40e956f

HEAD is now at 40e956f committed file1

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ ls

file1.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git ls-files

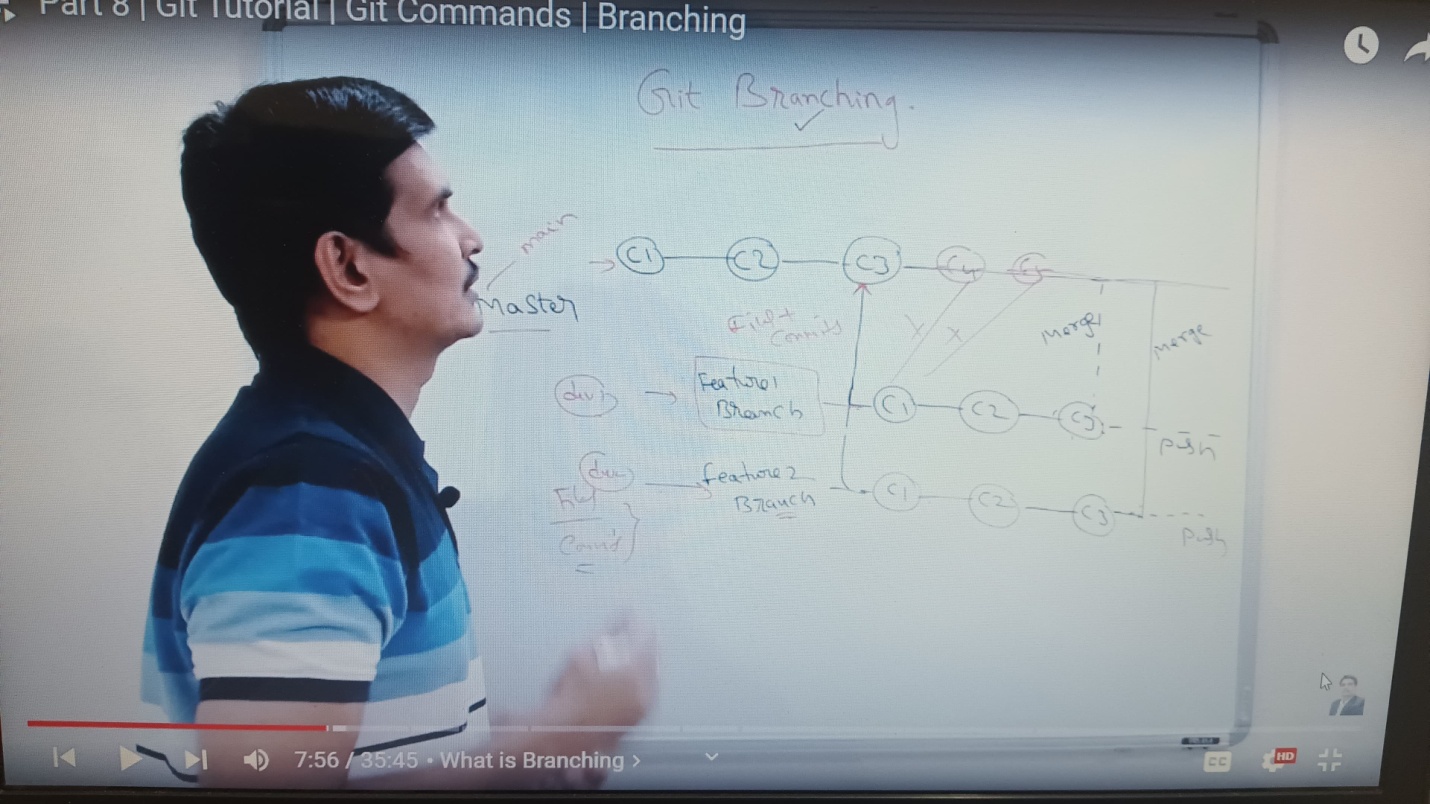
file1.txt

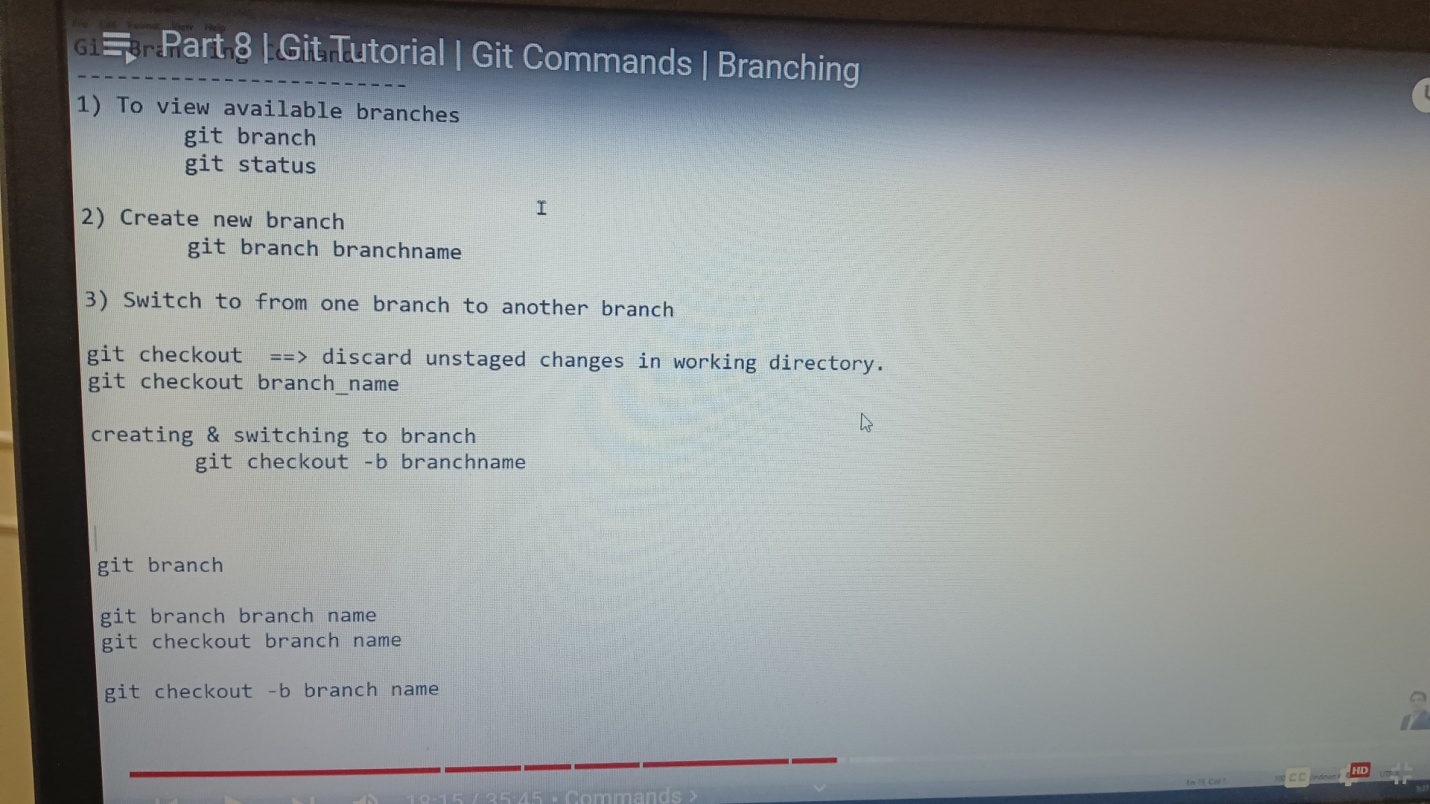
sree@DESKTOP-2B28HN3 MINGW64 /c/reset2 (master)

$ git log --oneline

40e956f (HEAD -> master) committed file1

GIT BRANCHING:





sree@DESKTOP-2B28HN3 MINGW64 /c/branch

$ git init

Initialized empty Git repository in C:/branch/.git/

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ touch a.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ ls

a.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git add a.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git commit -m"a.txt"

[master (root-commit) bd93f6e] a.txt

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 a.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git branch

\* master

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git status

On branch master

nothing to commit, working tree clean

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git branch br1

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git branch

br1

\* master

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git status

On branch master

nothing to commit, working tree clean

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git checkout br1

Switched to branch 'br1'

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

$ git branch

\* br1

master

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

$ git checkout -b br2

Switched to a new branch 'br2'

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br2)

$ git branch

br1

\* br2

master

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br2)

$ git checkout -b br1

fatal: a branch named 'br1' already exists

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br2)

$ git checkout master

Switched to branch 'master'

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git branch

br1

br2

\* master

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ ls

a.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ touch b.txt c.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ ls

a.txt b.txt c.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git add b.txt;git commit -m"b.txt"

[master 0230dc8] b.txt

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 b.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git add c.txt;git commit -m"c.txt"

[master 6e2d7b3] c.txt

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 c.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git ls-files

a.txt

b.txt

c.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git log --oneline

6e2d7b3 (HEAD -> master) c.txt

0230dc8 b.txt

bd93f6e (br2, br1) a.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git status

On branch master

nothing to commit, working tree clean

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ ls

a.txt b.txt c.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git checkout br1

Switched to branch 'br1'

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

$ ls

a.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

$ git log --oneline

bd93f6e (HEAD -> br1, br2) a.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

$ git branch

\* br1

br2

master

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

$ touch x.txt y.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

$ git add x.txt;git commit -m"x.txt"

[br1 6e1e5e4] x.txt

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 x.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

$ git add y.txt;git commit -m"y.txt"

[br1 eb3b4ff] y.txt

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 y.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

$ ls

a.txt x.txt y.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

$ git checkout master

Switched to branch 'master'

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ ls

a.txt b.txt c.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git checkout br1

Switched to branch 'br1'

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

$ ls

a.txt x.txt y.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

$ git checkout branch

error: pathspec 'branch' did not match any file(s) known to git

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

$ git checkout master

Switched to branch 'master'

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git branch

br1

br2

\* master

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git checkout br1

Switched to branch 'br1'

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

$ ls

a.txt x.txt y.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

$ git checkout master

Switched to branch 'master'

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ ls

a.txt b.txt c.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ touch d.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git add d.txt;git commit -m"d.txt"

[master 0da7584] d.txt

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 d.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ ls

a.txt b.txt c.txt d.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/branch (master)

$ git checkout br1

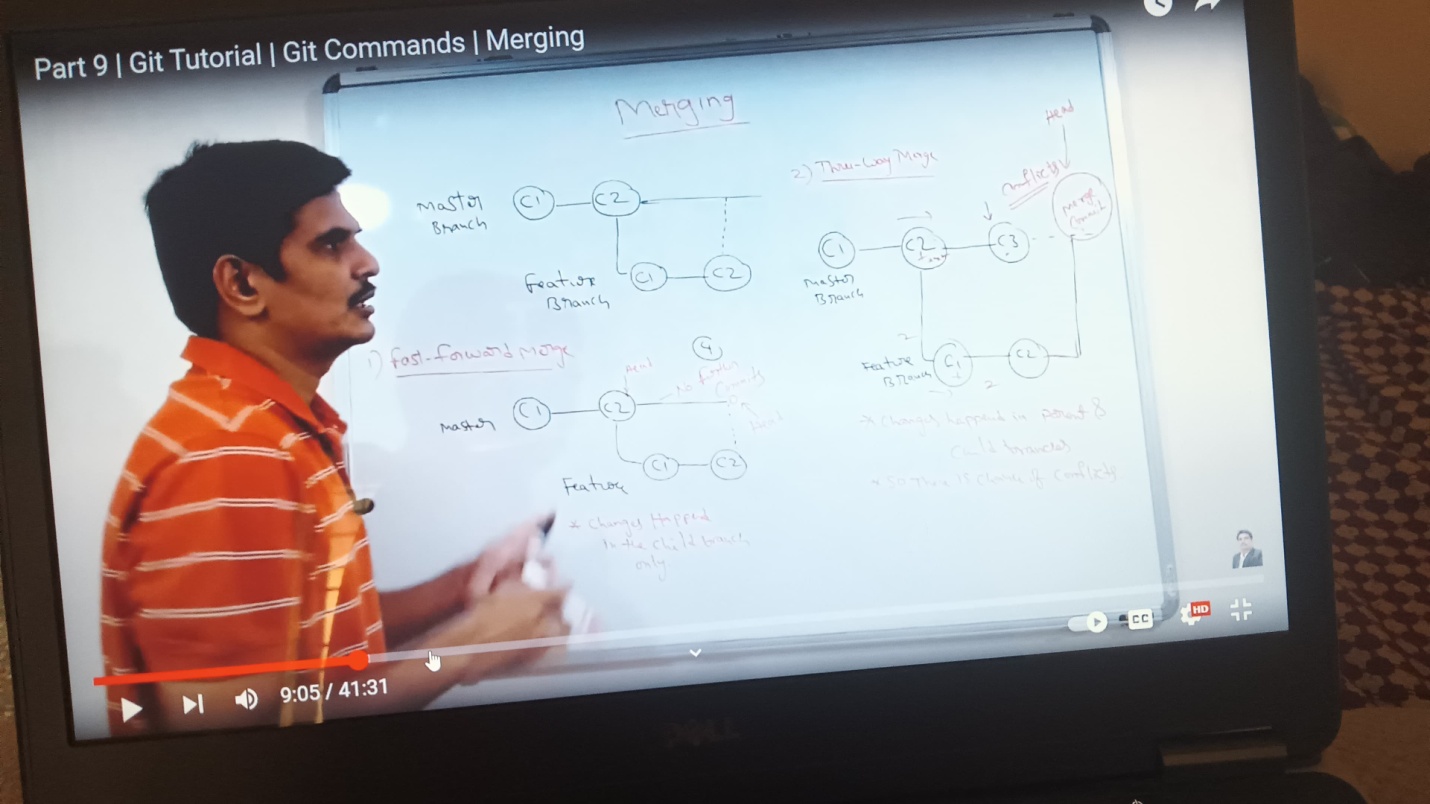
Switched to branch 'br1'

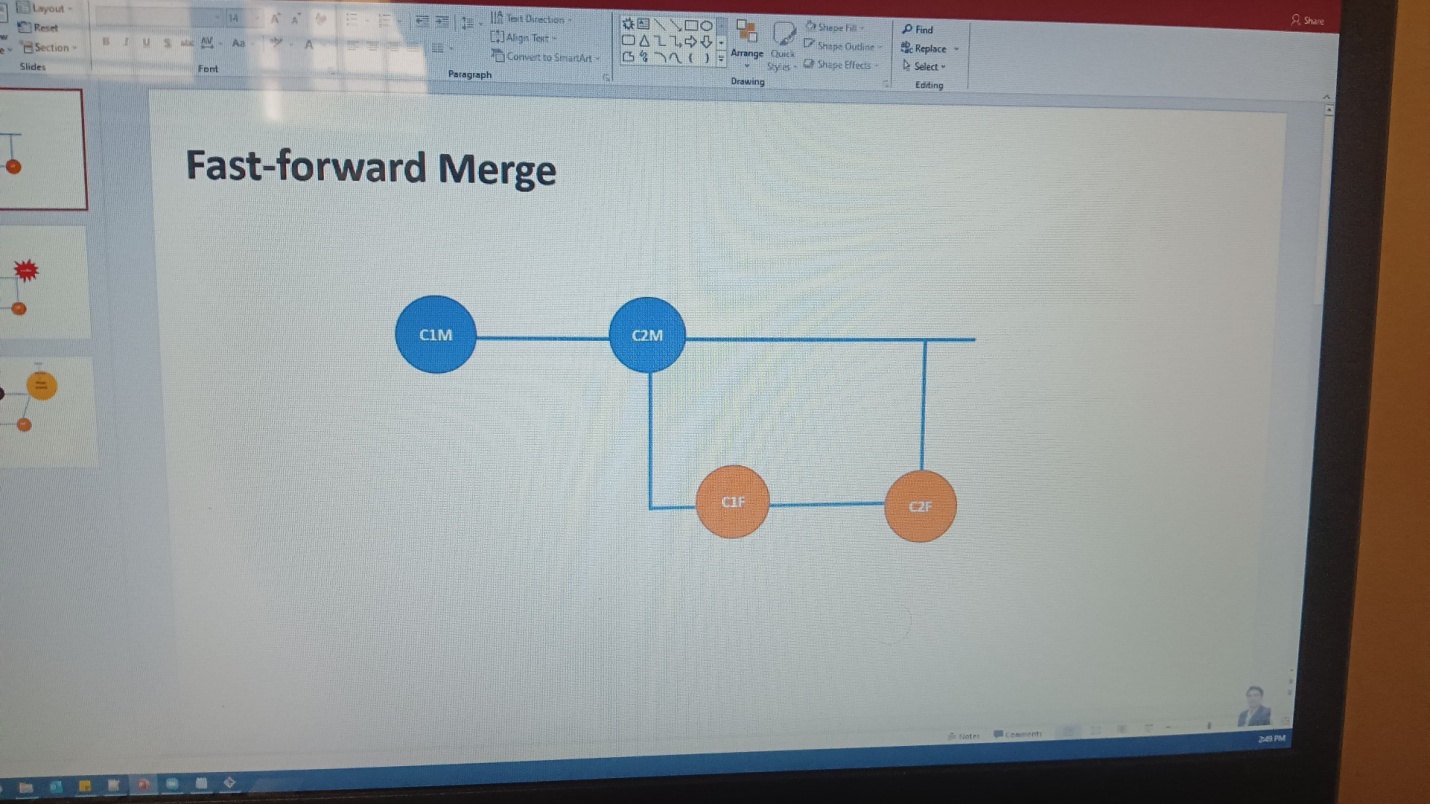
sree@DESKTOP-2B28HN3 MINGW64 /c/branch (br1)

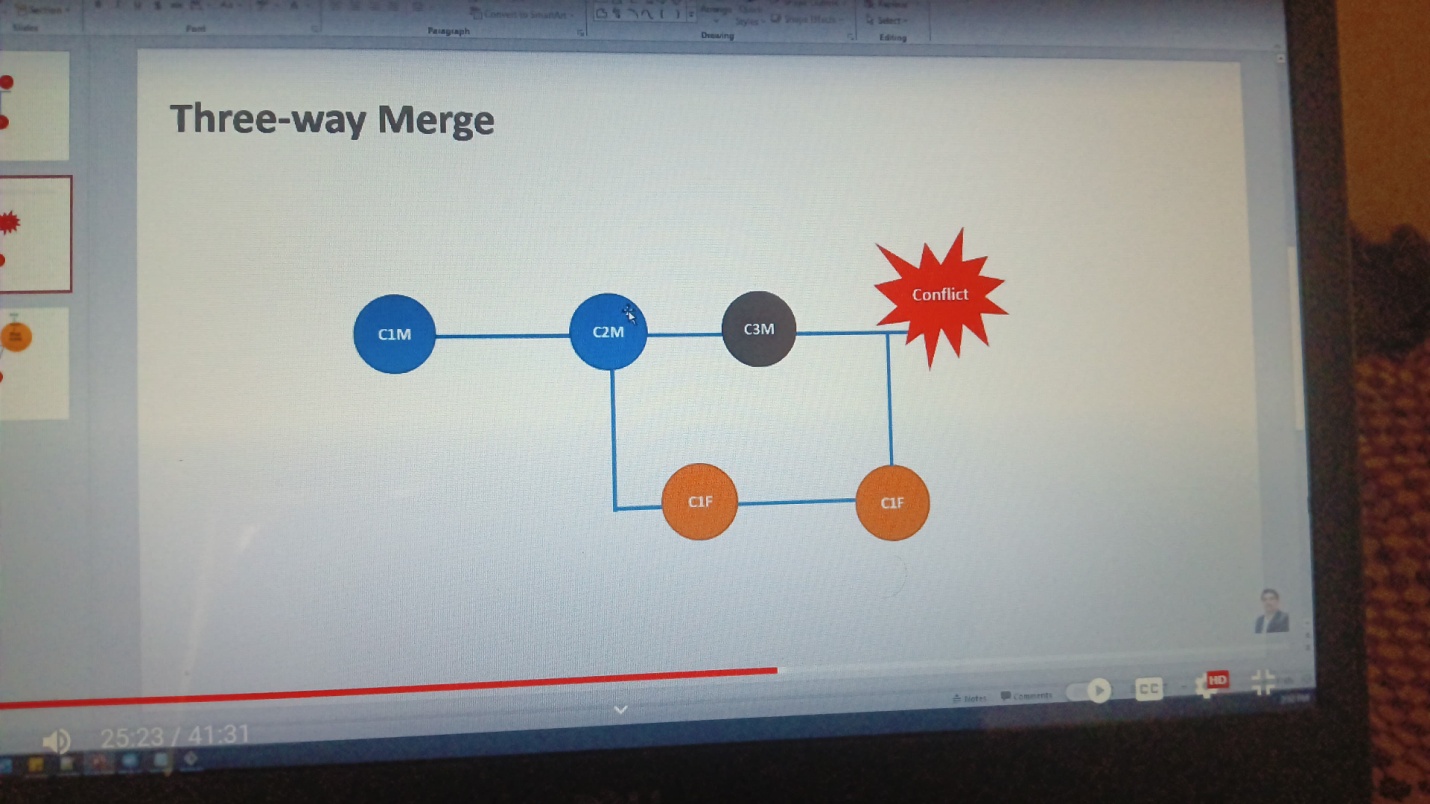
$ ls

a.txt x.txt y.txt

GIT MERGING:











sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge

$ git init

Initialized empty Git repository in C:/3waymerge/.git/

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ touch a.txt b.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ git add a.txt;git commit -m"C1M"

[master (root-commit) 7e2e3e6] C1M

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 a.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ git add b.txt;git commit -m"C2M"

[master 59fdfa9] C2M

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 b.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ git log --oneline

59fdfa9 (HEAD -> master) C2M

7e2e3e6 C1M

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ git branch master

fatal: a branch named 'master' already exists

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ git branch

\* master

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ git branch feature

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ git checkout feature

Switched to branch 'feature'

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (feature)

$ git branch

\* feature

master

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (feature)

$ touch x.txt y.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (feature)

$ ls

a.txt b.txt x.txt y.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (feature)

$ git add x.txt;git commit -m"C1F"

[feature b76c567] C1F

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 x.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (feature)

$ git add y.txt;git commit -m"C2F"

[feature 393a452] C2F

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 y.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (feature)

$ git log --oneline

393a452 (HEAD -> feature) C2F

b76c567 C1F

59fdfa9 (master) C2M

7e2e3e6 C1M

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (feature)

$ git checkout master

Switched to branch 'master'

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ git branch

feature

\* master

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ ls

a.txt b.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ git log--oneline

git: 'log--oneline' is not a git command. See 'git --help'.

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ git log --oneline

59fdfa9 (HEAD -> master) C2M

7e2e3e6 C1M

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ touch c.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ git add c.txt;git commit-m"C3M"

git: 'commit-mC3M' is not a git command. See 'git --help'.

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ git add c.txt;git commit -m"C3M"

[master f2e7ffb] C3M

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 c.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ ls

a.txt b.txt c.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ git branch

feature

\* master

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ git merge feature

Merge made by the 'ort' strategy.

x.txt | 0

y.txt | 0

2 files changed, 0 insertions(+), 0 deletions(-)

create mode 100644 x.txt

create mode 100644 y.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/3waymerge (master)

$ git log --oneline

db6b1fa (HEAD -> master) Merge branch 'feature'

f2e7ffb C3M

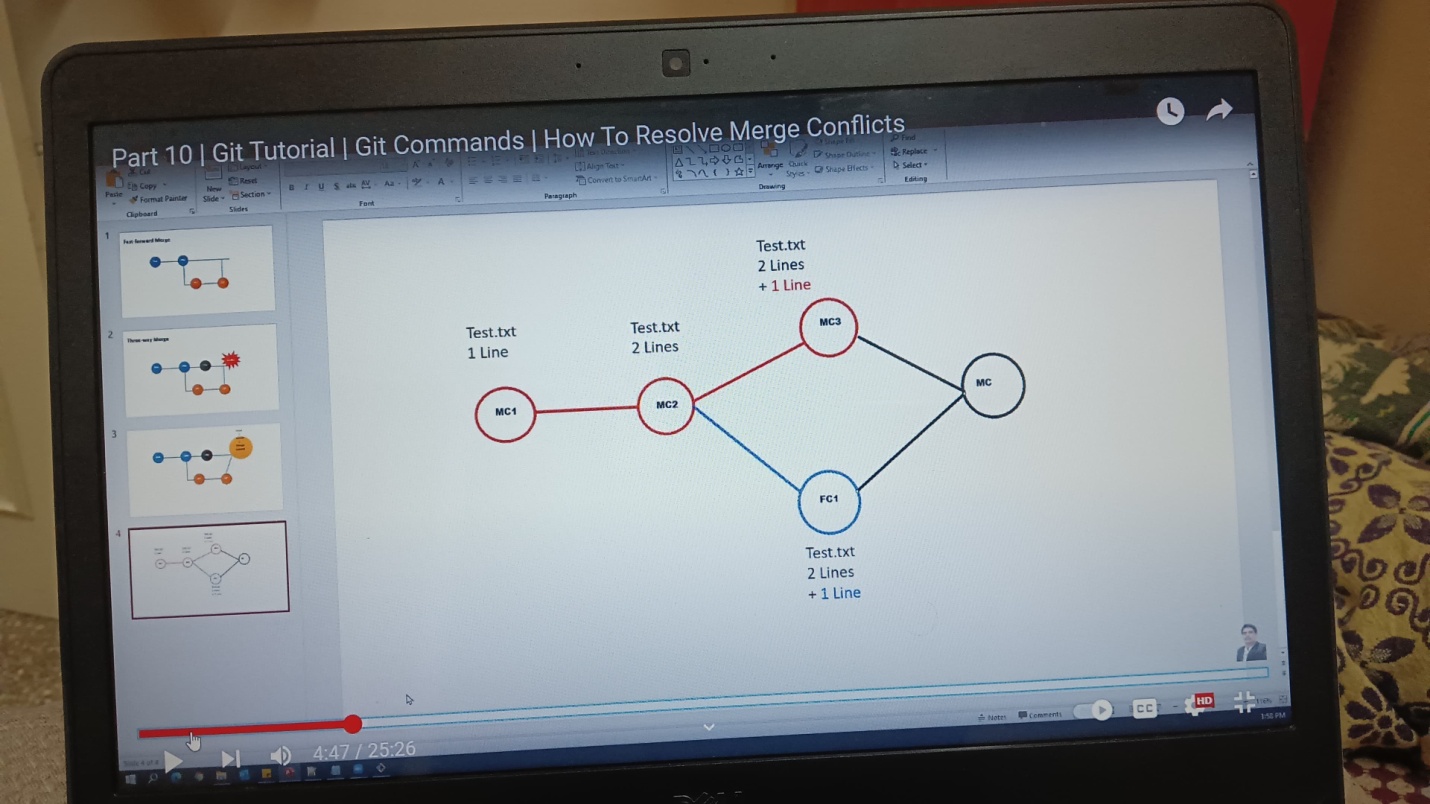
393a452 (feature) C2F

b76c567 C1F

59fdfa9 C2M

7e2e3e6 C1M

HOW TO RESOLVE MERGE CONFLICTS:



sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts

$ git init

Initialized empty Git repository in C:/mergeconflicts/.git/

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ ls

test.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ cat test.txt

this is my first line

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git add test.txt;git commit -m"Mc1"

[master (root-commit) 9580ee2] Mc1

1 file changed, 1 insertion(+)

create mode 100644 test.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ cat test.txt

this is my first line

this is my second line

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git add test.txt;git commit -m"Mc2"

[master f0d962f] Mc2

1 file changed, 1 insertion(+)

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git log --oneline

f0d962f (HEAD -> master) Mc2

9580ee2 Mc1

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git checkout branch

error: pathspec 'branch' did not match any file(s) known to git

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git branch feature

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git checkout feature

Switched to branch 'feature'

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (feature)

$ git log --oneline

f0d962f (HEAD -> feature, master) Mc2

9580ee2 Mc1

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (feature)

$ cat test.txt

this is my first line

this is my second line

added new line by feature

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (feature)

$ git add test.txt;git commit -m"FC1"

[feature 620853d] FC1

1 file changed, 2 insertions(+), 1 deletion(-)

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (feature)

$ git log --oneline

620853d (HEAD -> feature) FC1

f0d962f (master) Mc2

9580ee2 Mc1

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (feature)

$ git checkout master

Switched to branch 'master'

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ cat test.txt

this is my first line

this is my second line

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git add test.txt;git commit-m"MC3"

git: 'commit-mMC3' is not a git command. See 'git --help'.

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git add test.txt;git commit -m"MC3"

[master 6ff3ab9] MC3

1 file changed, 2 insertions(+), 1 deletion(-)

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git log --oneline

6ff3ab9 (HEAD -> master) MC3

f0d962f Mc2

9580ee2 Mc1

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git merge feature

Auto-merging test.txt

CONFLICT (content): Merge conflict in test.txt

Automatic merge failed; fix conflicts and then commit the result.

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master|MERGING)

$ cat test.txt

this is my first line

this is my second line

<<<<<<< HEAD

added new line by master

=======

added new line by feature

>>>>>>> feature

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master|MERGING)

$ cat test.txt

this is my first line

this is my second line

added new line by master

added new line by feature

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master|MERGING)

$ git add test.txt;git commit -m"MC"

[master 8b8b1a9] MC

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git log --oneline

8b8b1a9 (HEAD -> master) MC

6ff3ab9 MC3

620853d (feature) FC1

f0d962f Mc2

9580ee2 Mc1

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git log --oneline --graph

\* 8b8b1a9 (HEAD -> master) MC

|\

| \* 620853d (feature) FC1

\* | 6ff3ab9 MC3

|/

\* f0d962f Mc2

\* 9580ee2 Mc1

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git branch

feature

\* master

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git branch -d feature

Deleted branch feature (was 620853d).

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git branch

\* master

sree@DESKTOP-2B28HN3 MINGW64 /c/mergeconflicts (master)

$ git log --oneline

8b8b1a9 (HEAD -> master) MC

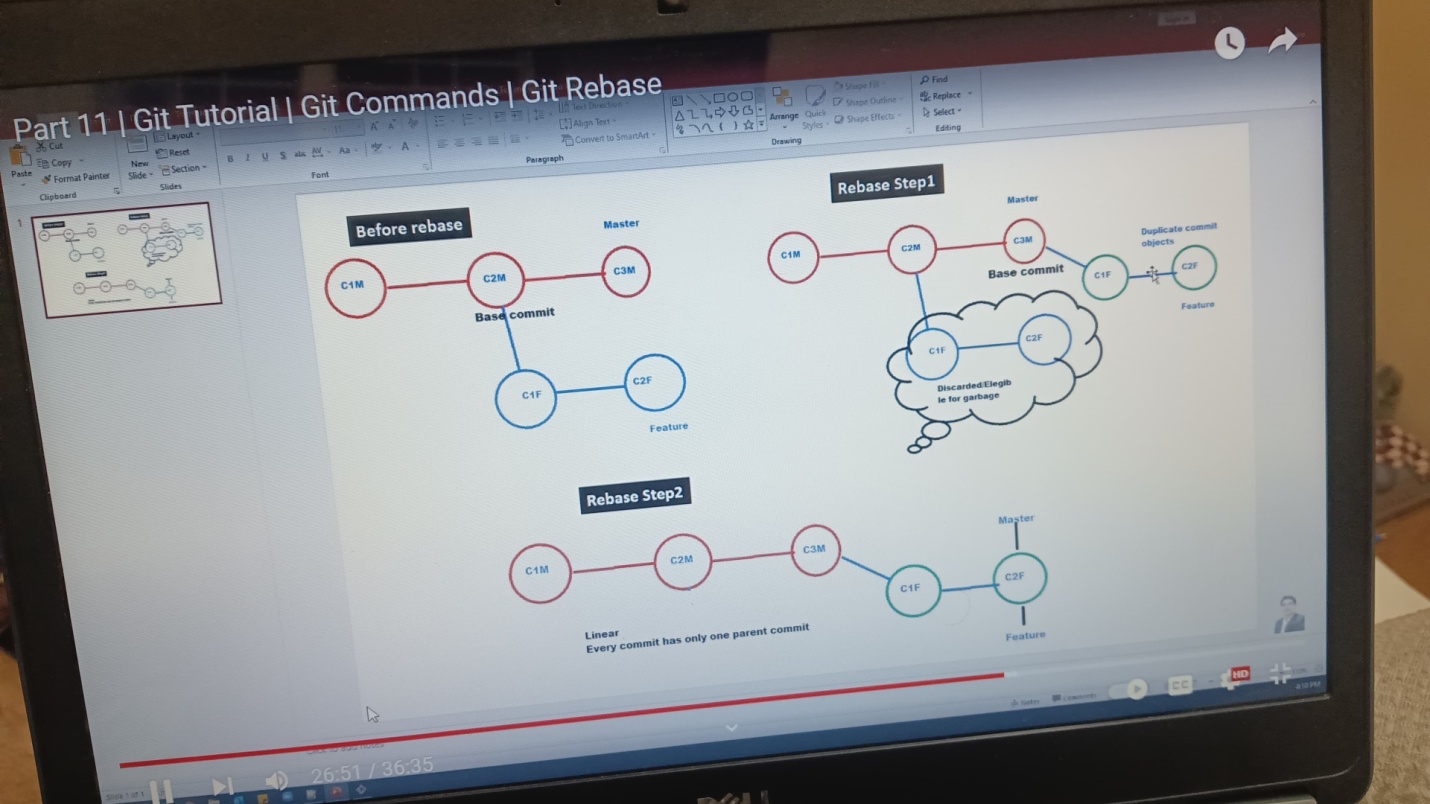
6ff3ab9 MC3

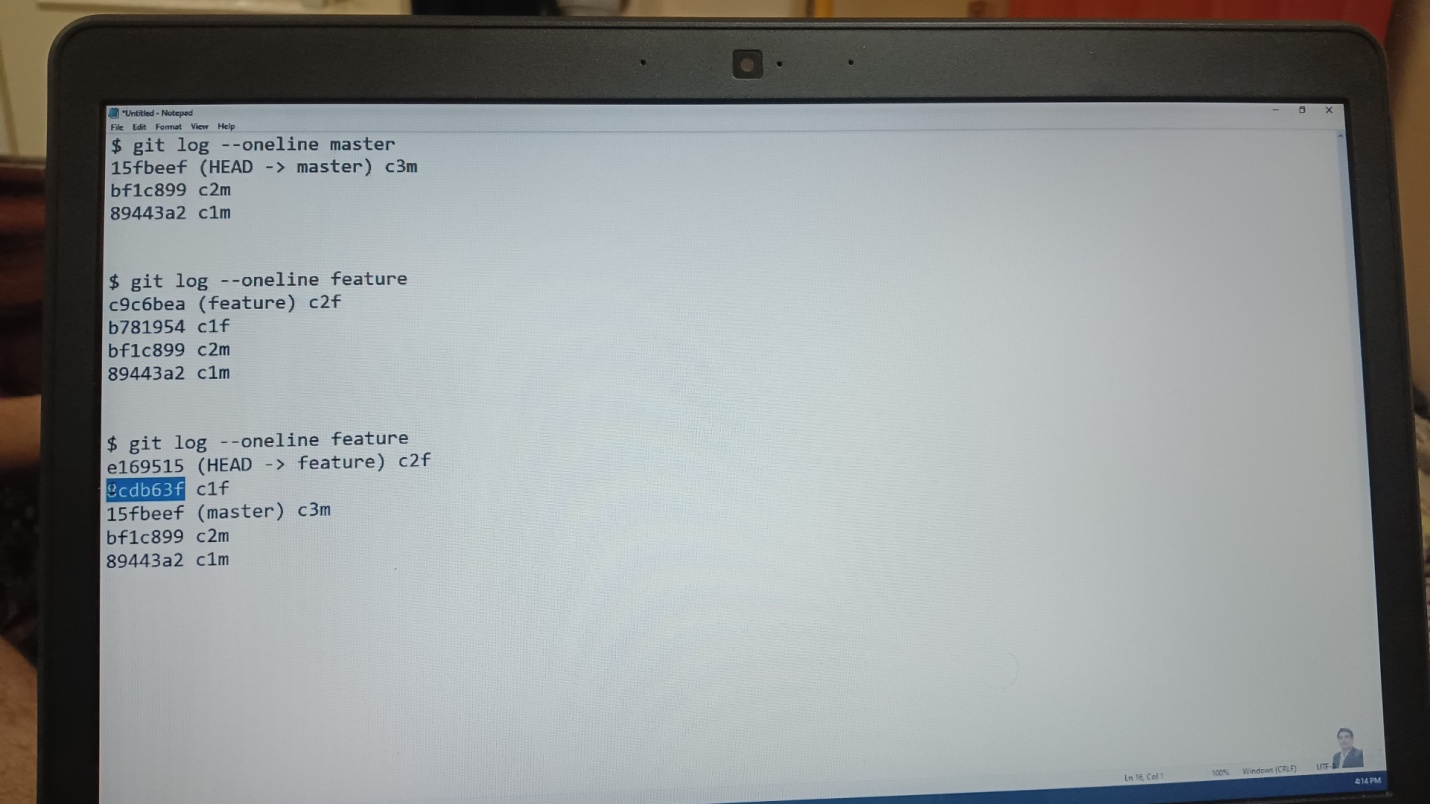
620853d FC1

f0d962f Mc2

9580ee2 Mc1

GIT REBASE:





sree@DESKTOP-2B28HN3 MINGW64 /c/rebase

$ git init

Initialized empty Git repository in C:/rebase/.git/

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ touch a.txt b.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git add a.txt;git commit -m"C1M"

[master (root-commit) 00b18ca] C1M

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 a.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git add b.txt;git commit -m"C2M"

[master d6081f7] C2M

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 b.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git log --oneline

d6081f7 (HEAD -> master) C2M

00b18ca C1M

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git branch

\* master

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git branch feature

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git checkout feature

Switched to branch 'feature'

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (feature)

$ ls

a.txt b.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (feature)

$ git branch

\* feature

master

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (feature)

$ echo x.txt y.txt

x.txt y.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (feature)

$ git add x.txt;git commit -m"C1F"

fatal: pathspec 'x.txt' did not match any files

On branch feature

nothing to commit, working tree clean

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (feature)

$ ls

a.txt b.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (feature)

$ touch x.txt y.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (feature)

$ git add x.txt;git commit -m"C1f"

[feature d81713c] C1f

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 x.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (feature)

$ git add y.txt;git commit -m"C2f"

[feature 5586bb6] C2f

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 y.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (feature)

$ git log --oneline

5586bb6 (HEAD -> feature) C2f

d81713c C1f

d6081f7 (master) C2M

00b18ca C1M

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (feature)

$ git branch

\* feature

master

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (feature)

$ git checkout master

Switched to branch 'master'

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git branch

feature

\* master

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ ls

a.txt b.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ touch x.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ ls

a.txt b.txt x.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git rm x.txt

fatal: pathspec 'x.txt' did not match any files

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ rm x.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ ls

a.txt b.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ touch c.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git add c.txt;git commit -m"C3M"

[master e441a9f] C3M

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 c.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git log --o0neline

fatal: unrecognized argument: --o0neline

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git log --oneline

e441a9f (HEAD -> master) C3M

d6081f7 C2M

00b18ca C1M

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git log --oneline master

e441a9f (HEAD -> master) C3M

d6081f7 C2M

00b18ca C1M

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git log --oneline feature

5586bb6 (feature) C2f

d81713c C1f

d6081f7 C2M

00b18ca C1M

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git branch

feature

\* master

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git checkout feature

Switched to branch 'feature'

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (feature)

$ git rebase master

Successfully rebased and updated refs/heads/feature.

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (feature)

$ git log --oneline master

e441a9f (master) C3M

d6081f7 C2M

00b18ca C1M

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (feature)

$ git log --oneline feature

e94b04d (HEAD -> feature) C2f

717aa5d C1f

e441a9f (master) C3M

d6081f7 C2M

00b18ca C1M

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (feature)

$ git checkout master

Switched to branch 'master'

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ ls

a.txt b.txt c.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git log --oneline

e441a9f (HEAD -> master) C3M

d6081f7 C2M

00b18ca C1M

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git merge feature

Updating e441a9f..e94b04d

Fast-forward

x.txt | 0

y.txt | 0

2 files changed, 0 insertions(+), 0 deletions(-)

create mode 100644 x.txt

create mode 100644 y.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git log --oneline graph

fatal: ambiguous argument 'graph': unknown revision or path not in the working tree.

Use '--' to separate paths from revisions, like this:

'git <command> [<revision>...] -- [<file>...]'

sree@DESKTOP-2B28HN3 MINGW64 /c/rebase (master)

$ git log --oneline --graph

\* e94b04d (HEAD -> master, feature) C2f

\* 717aa5d C1f

\* e441a9f C3M

\* d6081f7 C2M

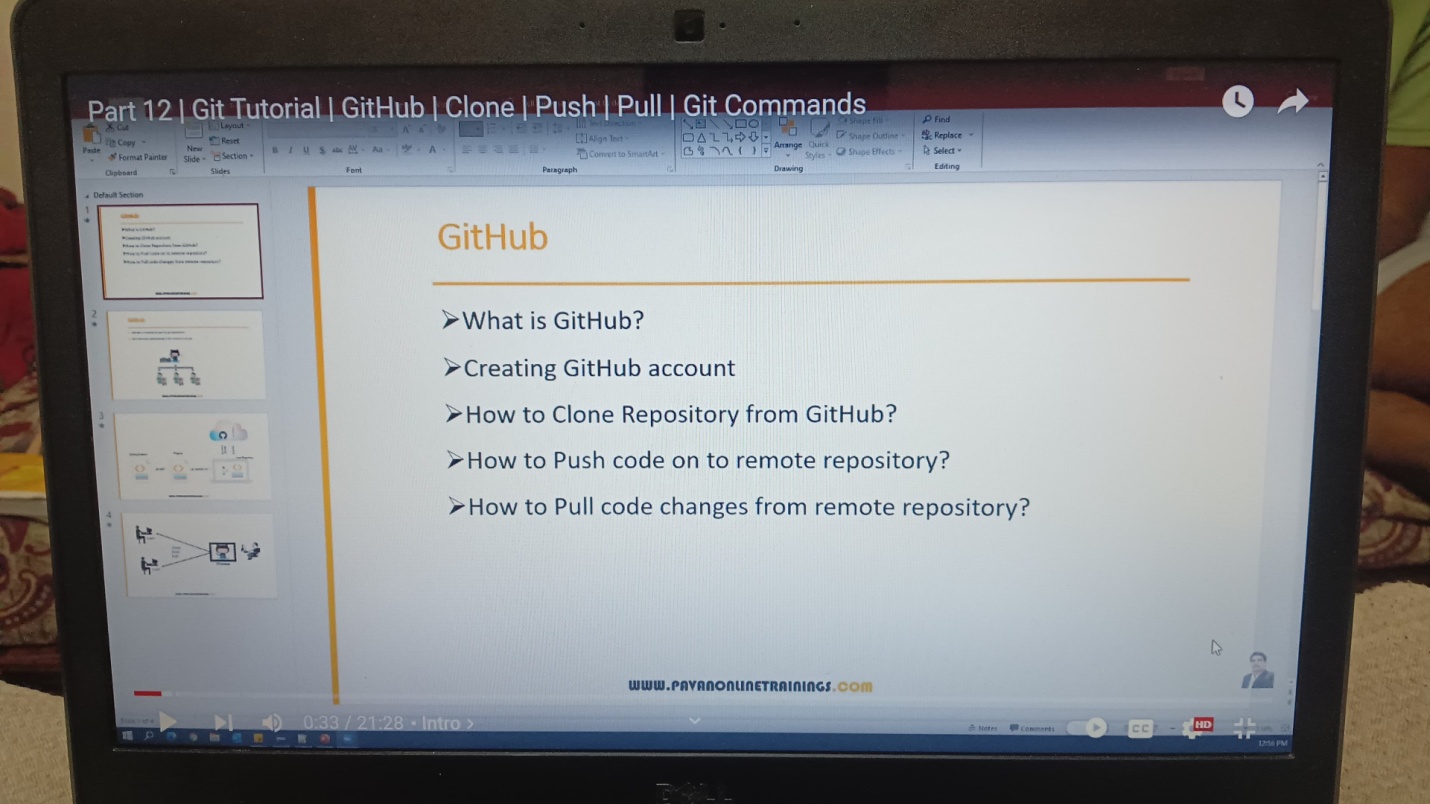
\* 00b18ca C1M

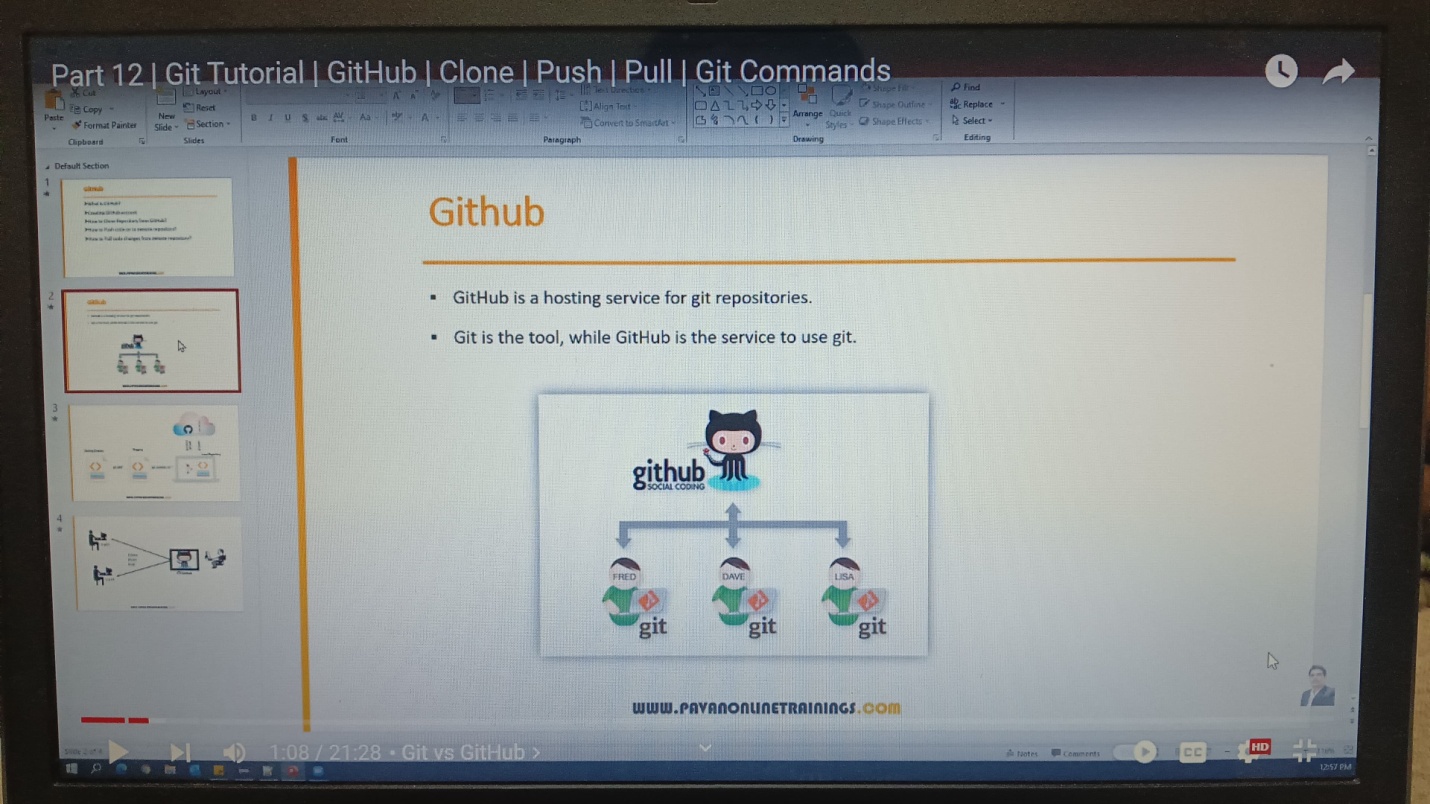
GIT CLONE/PUSH/PULL:

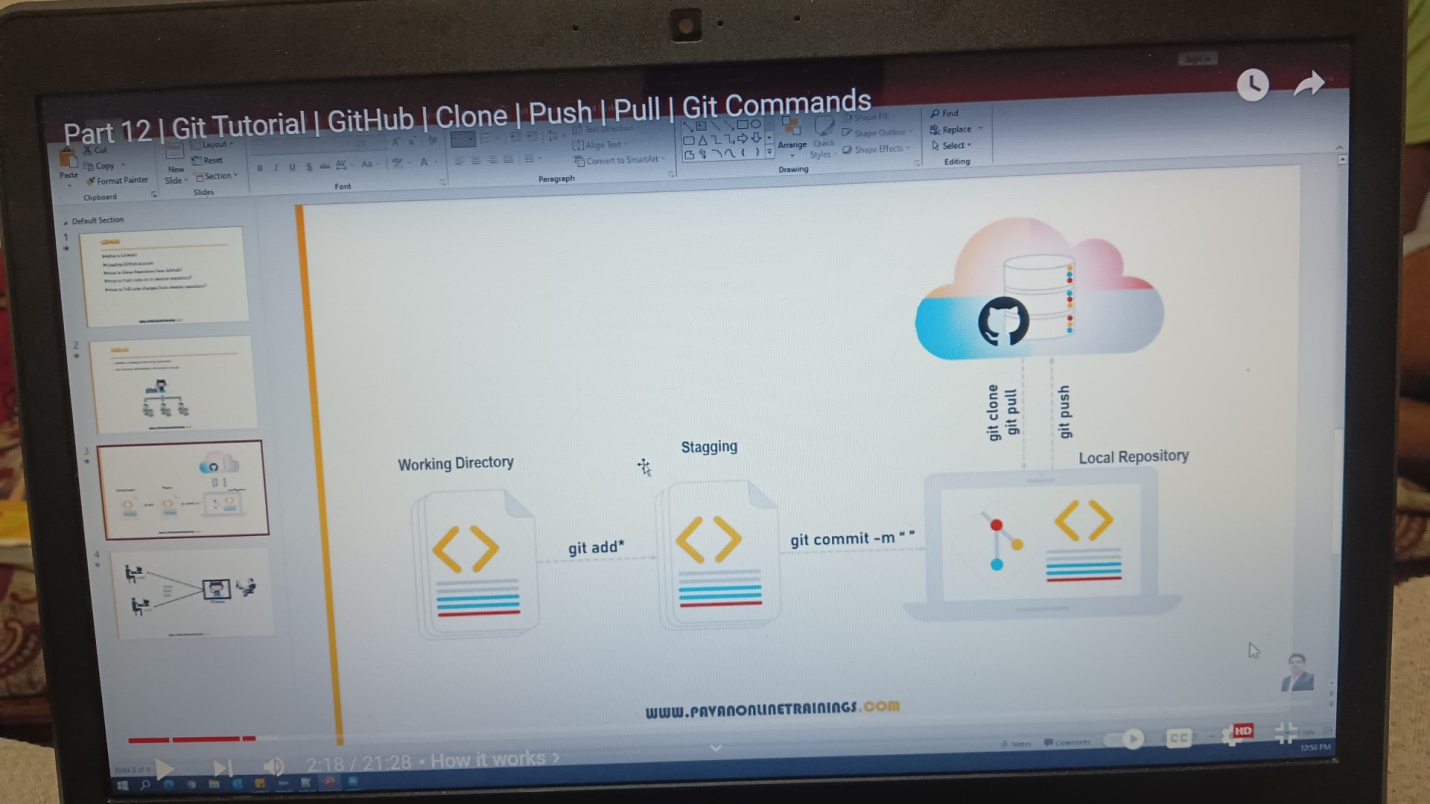
git pull origin main: This command fetches and merges changes from a remote repository into your current branch.

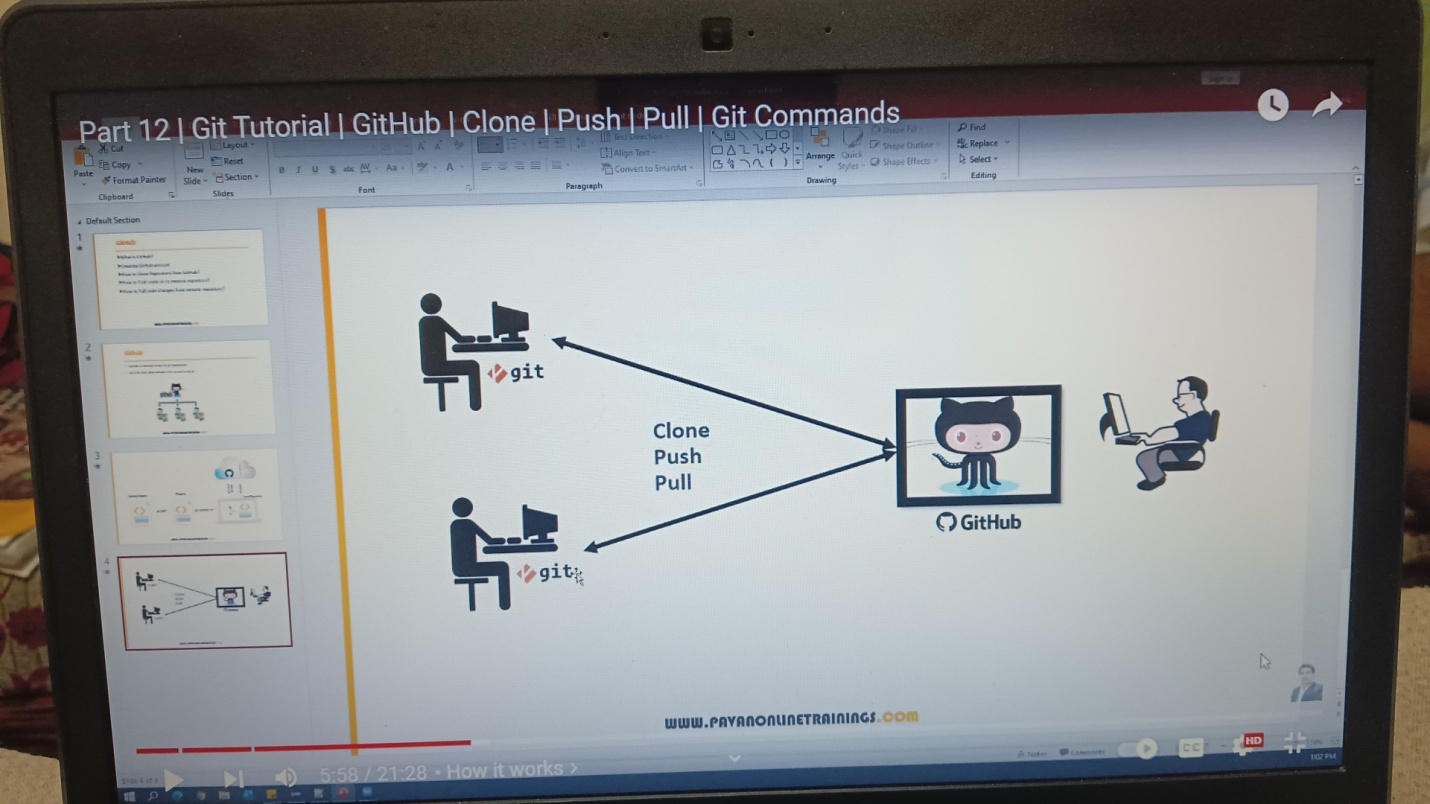
The git pull command is used to fetch and download content from a remote repository and immediately update the local repository to match that content.

GIT CLONE: The git clone command copies an existing Git repository









sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone

$ git clone 'https://github.com/Pavansaikiran407/gitrepo.git'

Cloning into 'gitrepo'...

remote: Enumerating objects: 3, done.

remote: Counting objects: 100% (3/3), done.

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

Receiving objects: 100% (3/3), done.

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone

$ ls

gitrepo/

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone

$ cd getrepo

bash: cd: getrepo: No such file or directory

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone

$ cd gitrepo

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone/gitrepo (main)

$ ls

myfile

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone/gitrepo (main)

$ cat myfile

my name is pavan

i am from banglore

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone/gitrepo (main)

$ git add .

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone/gitrepo (main)

$ git commit -m"updated file"

[main 4f88599] updated file

2 files changed, 2 insertions(+)

create mode 100644 myfile2.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone/gitrepo (main)

$ git log --oneline

4f88599 (HEAD -> main) updated file

ef132b0 (origin/main, origin/HEAD) Create myfile

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone/gitrepo (main)

$ git push origin main

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% (4/4), 357 bytes | 178.00 KiB/s, done.

Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To https://github.com/Pavansaikiran407/gitrepo.git

ef132b0..4f88599 main -> main

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone/gitrepo (main)

$ cat myfile2

cat: myfile2: No such file or directory

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone/gitrepo (main)

$ cat myfile2

cat: myfile2: No such file or directory

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone/gitrepo (main)

$ ls

myfile myfile2.txt

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone/gitrepo (main)

$ cat myfile2.txt

i am working as developer

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone/gitrepo (main)

$ git pull origin main

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

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

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

Unpacking objects: 100% (3/3), 957 bytes | 34.00 KiB/s, done.

From https://github.com/Pavansaikiran407/gitrepo

\* branch main -> FETCH\_HEAD

4f88599..b51933a main -> origin/main

Updating 4f88599..b51933a

Fast-forward

myfile2.txt | 8 +++++++-

1 file changed, 7 insertions(+), 1 deletion(-)

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone/gitrepo (main)

$ cat file2.txt

cat: file2.txt: No such file or directory

sree@DESKTOP-2B28HN3 MINGW64 /c/gitclone/gitrepo (main)

$ cat myfile2.txt

i am working as developer

i am working as developer

i am working as developer

i am working as developer

i am working as developer

i am working as developer

i am working as developer