Sec-0

hp@DESKTOP-VOF30A5 MINGW64 ~ (master)

$ mkdir <project\_dir>

bash: syntax error near unexpected token `newline'

hp@DESKTOP-VOF30A5 MINGW64 ~ (master)

$ mkdir proj\_dir

hp@DESKTOP-VOF30A5 MINGW64 ~ (master)

$ cd proj\_dir

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git init

Initialized empty Git repository in C:/Users/hp/proj\_dir/.git/

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git status

On branch master

No commits yet

Untracked files:

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

index.html

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git status

On branch master

No commits yet

Changes to be committed:

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

new file: index.html

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git commit

Aborting commit due to empty commit message.

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git commit -m "created index.html"

[master (root-commit) c66012c] created index.html

1 file changed, 7 insertions(+)

create mode 100644 index.html

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (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.html

Untracked files:

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

info.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ touch. gitignore

bash: touch.: command not found

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ touch .gitignore

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (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.html

Untracked files:

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

.gitignore

info.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git rm info.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (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.html

Untracked files:

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

.gitignore

info.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git rm \*.txt --cached

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (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.html

Untracked files:

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

.gitignore

info.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (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.html

Untracked files:

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

.gitignore

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (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.html

Untracked files:

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

.gitignore

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ touch .info\_1

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (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.html

Untracked files:

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

.gitignore

.info\_1

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (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.html

Untracked files:

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

.gitignore

.info\_1

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ touch .info\_1.txt

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (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.html

Untracked files:

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

.gitignore

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git commit -m "performed ignore op"

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.html

Untracked files:

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

.gitignore

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git commit

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.html

Untracked files:

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

.gitignore

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (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.html

Untracked files:

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

.gitignore

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git log

commit c66012cf7c8dec7ca4e5c6edf050444e619e5370 (HEAD -> master)

Author: vamshi71 <vamshigoli456@gmail.com>

Date: Thu Jul 29 18:13:34 2021 +0530

created index.html

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git checkout --index.html

error: unknown option `index.html'

usage: git checkout [<options>] <branch>

or: git checkout [<options>] [<branch>] -- <file>...

-b <branch> create and checkout a new branch

-B <branch> create/reset and checkout a branch

-l create reflog for new branch

--guess second guess 'git checkout <no-such-branch>' (default)

--overlay use overlay mode (default)

-q, --quiet suppress progress reporting

--recurse-submodules[=<checkout>]

control recursive updating of submodules

--progress force progress reporting

-m, --merge perform a 3-way merge with the new branch

--conflict <style> conflict style (merge or diff3)

-d, --detach detach HEAD at named commit

-t, --track set upstream info for new branch

-f, --force force checkout (throw away local modifications)

--orphan <new-branch>

new unparented branch

--overwrite-ignore update ignored files (default)

--ignore-other-worktrees

do not check if another worktree is holding the given ref

-2, --ours checkout our version for unmerged files

-3, --theirs checkout their version for unmerged files

-p, --patch select hunks interactively

--ignore-skip-worktree-bits

do not limit pathspecs to sparse entries only

--pathspec-from-file <file>

read pathspec from file

--pathspec-file-nul with --pathspec-from-file, pathspec elements are separated with NUL character

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git reset --hard

HEAD is now at c66012c created index.html

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (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.html

Untracked files:

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

.gitignore

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git add.

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

The most similar command is

add

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git status

On branch master

Changes to be committed:

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

new file: .gitignore

modified: index.html

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git status

On branch master

Changes to be committed:

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

new file: .gitignore

modified: index.html

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: .gitignore

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git rm .gitignore

error: the following file has staged content different from both the

file and the HEAD:

.gitignore

(use -f to force removal)

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git rm --cached .gitignore

error: the following file has staged content different from both the

file and the HEAD:

.gitignore

(use -f to force removal)

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git rm -f .gitignore

rm '.gitignore'

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git reset --hard

HEAD is now at c66012c created index.html

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git status

On branch master

Untracked files:

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

.info\_1.txt

info.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ touch .gitignore

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git status

On branch master

nothing to commit, working tree clean

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git status

On branch master

Untracked files:

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

.gitignore

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (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.html

Untracked files:

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

.gitignore

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (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.html

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

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git restore index.html

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git status

On branch master

nothing to commit, working tree clean

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git config --global alias.my-add add

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git status

On branch master

nothing to commit, working tree clean

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git my-add index.html

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git commit -m "last commit in stage 0"

On branch master

nothing to commit, working tree clean

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git reset --soft HEAD

hp@DESKTOP-VOF30A5 MINGW64 ~/proj\_dir (master)

$ git status

On branch master

nothing to commit, working tree clean

sec-1

hp@DESKTOP-VOF30A5 MINGW64 ~ (master)

$ mkdir assignment

hp@DESKTOP-VOF30A5 MINGW64 ~ (master)

$ cd assignment

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ init

bash: init: command not found

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git init

Initialized empty Git repository in C:/Users/hp/assignment/.git/

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ touch readme.txt

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git status

On branch master

No commits yet

Untracked files:

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

readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git commit -m "created readme.txt "

On branch master

Initial commit

Untracked files:

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

readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git status

On branch master

No commits yet

Changes to be committed:

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

new file: readme.txt

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git commit -m "created readme.txt "

[master (root-commit) a328671] created readme.txt

1 file changed, 5 insertions(+)

create mode 100644 readme.txt

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git branch html-assignment

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (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: readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git commit -m "assignments copied"

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: readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git checkout -b html-assignment

fatal: A branch named 'html-assignment' already exists.

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git checkout html-assignment

Switched to branch 'html-assignment'

M readme.txt

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (html-assignment)

$ git status

On branch html-assignment

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: readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (html-assignment)

$ git status

On branch html-assignment

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: readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (html-assignment)

$ git commit -m "html assignments copied"

On branch html-assignment

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: readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (html-assignment)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (html-assignment)

$ git commit -m "html assignments copied"

[html-assignment 3f4a58c] html assignments copied

1 file changed, 2 insertions(+)

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (html-assignment)

$ git checkout master

Switched to branch 'master'

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git commit -m " minor changes "

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: readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git commit -m " minor changes "

[master 848aa28] minor changes

1 file changed, 1 insertion(+), 1 deletion(-)

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git checkout html assignment

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

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git checkout html-assignment

Switched to branch 'html-assignment'

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (html-assignment)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (html-assignment)

$ git commit -m " again changes in html assign"

[html-assignment 820e309] again changes in html assign

1 file changed, 1 insertion(+), 1 deletion(-)

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (html-assignment)

$ git checkout master

Switched to branch 'master'

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git status

On branch master

nothing to commit, working tree clean

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git merge html-assignments Auto-merging readme.txt

merge: html-assignments - not something we can merge

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git status

On branch master

nothing to commit, working tree clean

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git merge html-assignments

merge: html-assignments - not something we can merge

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git branch -d html-assignments

error: branch 'html-assignments' not found.

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git commit -m " again changes in html "

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: readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git status

On branch master

Changes to be committed:

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

modified: readme.txt

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git merge html-assignments

merge: html-assignments - not something we can merge

sec-2

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git branch css-assignment

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git checkout css-assignment

Switched to branch 'css-assignment'

M readme.txt

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git status

On branch css-assignment

Changes to be committed:

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

modified: readme.txt

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git add \*.css

fatal: pathspec '\*.css' did not match any files

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git add css-assignment

fatal: pathspec 'css-assignment' did not match any files

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git commit -m "adding 2 css assignment"

[css-assignment c06b7ff] adding 2 css assignment

1 file changed, 1 insertion(+)

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git status

On branch css-assignment

nothing to commit, working tree clean

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git add readme.txt

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git commit -m "changes made to readme.txt"

On branch css-assignment

nothing to commit, working tree clean

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git checkout master

error: Your local changes to the following files would be overwritten by checkout:

readme.txt

Please commit your changes or stash them before you switch branches.

Aborting

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git commit

On branch css-assignment

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: readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git commit

Aborting commit due to empty commit message.

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git commit

[css-assignment 1ed645c] changes are done :wq!

1 file changed, 1 insertion(+)

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git checkout master

Switched to branch 'master'

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git status

On branch master

nothing to commit, working tree clean

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git add readme.txt

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git commit -m "changes made to readme.txt again"

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: readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git commit -m "changes made to readme.txt again"

[master 814e387] changes made to readme.txt again

1 file changed, 1 insertion(+)

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git checkout css-assignment

Switched to branch 'css-assignment'

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git status

On branch css-assignment

nothing to commit, working tree clean

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git add \*.css

fatal: pathspec '\*.css' did not match any files

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git add css-assignment

fatal: pathspec 'css-assignment' did not match any files

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git status

On branch css-assignment

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: readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git commit

[css-assignment b970af6] changes made to the css-assign

1 file changed, 18 insertions(+), 3 deletions(-)

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git status

On branch css-assignment

nothing to commit, working tree clean

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (css-assignment)

$ git checkout master

Switched to branch 'master'

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git merge css-assignment Auto-merging readme.txt

merge: Auto-merging - not something we can merge

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git merge css-assignment

Auto-merging readme.txt

Merge made by the 'recursive' strategy.

readme.txt | 19 ++++++++++++++++++-

1 file changed, 18 insertions(+), 1 deletion(-)

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git add readme.txt

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git commit -m "changes made to readme.txt again"

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: readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git add readme.txt

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git commit -m "changes made to readme.txt again"

[master b6f37e3] changes made to readme.txt again

1 file changed, 1 insertion(+), 4 deletions(-)

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git merge css-assignment

Already up to date.

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git branch -d css-assignment

Deleted branch css-assignment (was b970af6).

Sec-3

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git branch js-assignment

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git checkout js-assignment

Switched to branch 'js-assignment'

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (js-assignment)

$ git status

On branch js-assignment

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: readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (js-assignment)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (js-assignment)

$ git commit -m "copied js-assignment"

[js-assignment efdce67] copied js-assignment

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (js-assignment)

$ git status

On branch js-assignment

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: readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (js-assignment)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (js-assignment)

$ git commit -m "changes-1 made to js-assignment"

[js-assignment 9ded439] changes-1 made to js-assignment

1 file changed, 1 insertion(+), 1 deletion(-)

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (js-assignment)

$ git checkout master

Switched to branch 'master'

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git commit -m "changes-2 made to js-assignment"

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: readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git commit -m "changes-2 made to js-assignment"

[master 45c34e4] changes-2 made to js-assignment

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git checkout js-assignment

Switched to branch 'js-assignment'

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (js-assignment)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (js-assignment)

$ git commit -m "changes-3 made to js-assignment"

[js-assignment de0938d] changes-3 made to js-assignment

1 file changed, 1 insertion(+), 1 deletion(-)

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (js-assignment)

$ git checkout master

Switched to branch 'master'

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git merge js-assignment

Auto-merging readme.txt

CONFLICT (content): Merge conflict in readme.txt

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

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master|MERGING)

$ git merge js-assignment

error: Merging is not possible because you have unmerged files.

hint: Fix them up in the work tree, and then use 'git add/rm <file>'

hint: as appropriate to mark resolution and make a commit.

fatal: Exiting because of an unresolved conflict.

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master|MERGING)

$ git merge js-assignment

fatal: You have not concluded your merge (MERGE\_HEAD exists).

Please, commit your changes before you merge.

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master|MERGING)

$ git add .

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master|MERGING)

$ git add readme.txt

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master|MERGING)

$ git commit -m "edited"

[master dad7937] edited

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

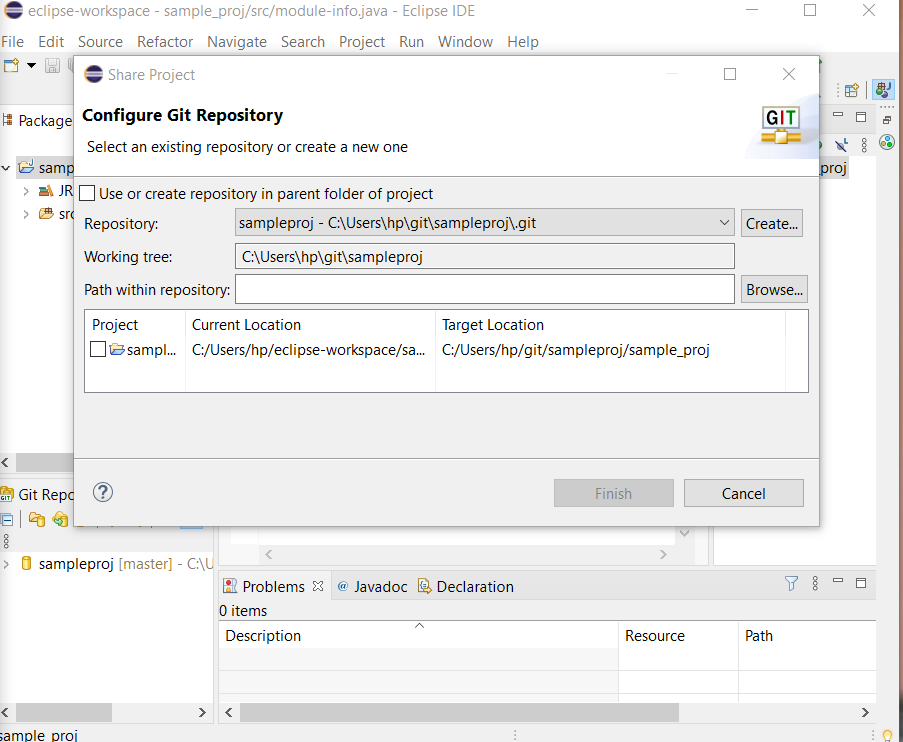
$ git merge js-assignment

Already up to date.

hp@DESKTOP-VOF30A5 MINGW64 ~/assignment (master)

$ git branch -d js-assignment

Deleted branch js-assignment (was de0938d).

**Sec-4**

