1. **Create a directory ‘project\_dir’ & cd to ‘project\_dir’.**

Command**-** mkdir project\_dir

Command- cd /c/Users/HOME/Desktop/project\_dir

1. **Initialize git version database. (git init)**

Command- git init

Output: Initialized empty Git repository in C:/Users/HOME/Desktop/project\_dir/.git/

1. **Create a new file index.html.**

Command**-** touch index.html

1. **Check the git status. You should find index.html as untracked file.**

Command- git status

Output-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)

1. **Stage the index.html file.**

Command- git add index.html

1. **Commit index.html**

Command- git commit -m "index.html"

Output- [master (root-commit) 3e8dbaa] index.html

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

create mode 100644 index.html

1. **Make few changes in index.html & create a new file info.txt file**.

Command- touch info.txt

1. **Check git status. You should find index.html & info.txt as untracked files.**

Command- git status

Output- 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

1. **Configure GIT to ignore all txt files.**

Command- touch .gitignore

1. **Again check the git status. You should find only index.html as untracked file.**

Command- git status

Output- 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")

1. **State & commit index.html**

Command- git add index.html

Command- git commit -m "again commit index.html"

Output- [master 4167698] again commit index.html

1 file changed, 1 insertion(+)

1. **Log all your comments so far.**

Command- git log

Output-

commit 416769870b71444f709afd58501a54b4baf1dcd7 (HEAD -> master)

Author: pooji <pylapooji@gmail.com>

Date: Wed Sep 29 19:27:11 2021 +0530

again commit index.html

commit 10626c8320776b49bce77ecbcec5b46b0b675d1b

Author: pooji <pylapooji@gmail.com>

Date: Wed Sep 29 19:21:38 2021 +0530

index.html

1. Make some changes in index.html.
2. **Revert the change made in the previous step using git command.**

Command- git restore index.html

git status

On branch master

nothing to commit, working tree clean

1. Again change index.html.
2. **Stage index.html**

Command- git add index.html

1. **Revert back the last stage.**

Command- git restore --staged index.html

Command- 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")

1. Rename ‘add’ command to ‘my-add’.

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

1. Using my\_add command Stage index.html again & commit the changes.

Command- git my-add index.html

Command- git commit -m "committing the changes"

[master 39357f4] committing the changes

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

1. Revert the last commit.

Command- git revert HEAD

[master 03c1473] Revert "committing the changes"

1 file changed, 1 insertion(+), 2 deletions(-)

*GIT Branching*

Objective: Commit HTML, CSS & JavaScript assignments into GIT.

SECTION-1 (HTML assignments) - Steps to follow:

1. **First take a backup of your assignments & projects. This is required because due to incorrect GIT operation you may lose your files.**

Command- git config --global user.name "pooji"

Command- git remote add origin https://github.com/poojipyla/my-first-project.git

1. **Create an empty directory ‘Assignments’ & cd to ‘Assignments’.**

Command- mkdir assignments

Command- cd assignments

1. **Create a file README.txt inside ‘Assignments’ & write few lines about the contents of ‘Assignments’ folder.**

Command-touch README.txt

Command- Notepad README.txt

1. **Commit README.txt file.**

Command- git commit -m "commit txt file"

[main (root-commit) 8839235] commit txt file

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

create mode 100644 Desktop/assignments/README.txt

1. **Now create a new branch ‘html-assignments’.**

Command- git branch html-assignments

1. **Switch to ‘html-assignments’ branch.**

Command- git checkout html-assignments

Switched to branch 'html-assignments'

1. **Copy all HTML assignments inside ‘Assignments’ folder.**

Command- touch assign1.html

Command- touch assign2.html

1. **Commit HTML assignments into ‘html-assignments’ branch.**

Command- git add assig1.html assign2.html

Command- git commit -m "commit this files"

[html-assignments 0867639] commit this files

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

create mode 100644 assig1.html

create mode 100644 assign2.html

1. **Make minor changes into few files belonging to ‘html-assignments’ branch.**

Output-

<!DOCTYPE html>

<html>

<head>

<body>

<h1>this is a assign1 html file</h1>

</body>

</head>

</html>

1. **Commit those changed files.**

Command- git add assign1.html

Command- git commit -m "commit assign1 changes"

[html-assignments a90f9f2] commit assign1 changes

1 file changed, 8 insertions(+)

1. **Switch to master branch.**

Command- git checkout master

Switched to branch 'master'

1. **Make minor changes into README.txt file & commit those changes into master.**
2. **Again switch to ‘html-assignments’ branch.**

Command- git checkout html-assignments

Switched to branch 'html-assignments'

1. **Make minor changes into few files belonging to ‘html-assignments’ branch.**

Output-

<!DOCTYPE html>

<html>

<h1>poojitha</h1>

</html>

1. **Commit those changes.**

Command- git add assign2.html

Command- git commit -m "commit changes"

[html-assignments ba72f3e] commit changes

1 file changed, 4 insertions(+)

1. **Switch to master.**

Command- git checkout master

Switched to branch 'master'

1. **Merge ‘html-assignments’ branch into master. Confirm all html assignments are shown in master.**

Command- git merge html-assignments

Merge made by the 'recursive' strategy.

Desktop/assignments/README.txt | 4 ++++

Desktop/assignments/assign1.html | 8 ++++++++

Desktop/assignments/assign2.html | 4 ++++

assig1.html | 0

assign2.html | 0

5 files changed, 16 insertions(+)

create mode 100644 Desktop/assignments/assign1.html

create mode 100644 Desktop/assignments/assign2.html

create mode 100644 assig1.html

create mode 100644 assign2.html-

1. **Finally delete the ‘html-assignments’ branch.**

Command- git branch -d html-assignments

Deleted branch html-assignments (was ba72f3e).

**SECTION-2** - **(CSS assignments) Steps to follow:**

1. **Create a new branch ‘css-assignments’.**

Command- git branch css-assignments

1. **Switch to ‘css-assignments’ branch.**

Command- git checkout css-assignments

Switched to branch 'css-assignments'

1. **Copy all CSS assignments inside ‘Assignments’ folder.**

Command- touch sample1.css

Command- touch sample2.css

1. **Commit CSS assignments into ‘css-assignments’ branch.**

Command- git add sample1.css sample2.css

Command- git commit -m "commit these files"

[css-assignments 93cc946] commit these files

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

create mode 100644 Desktop/assignments/sample1.css

create mode 100644 Desktop/assignments/sample2.css

1. **Make minor changes into README.txt file on line 1 belonging to ‘css-assignments’ branch.**
2. **Commit those changed files.**

Command- git add README.txt

Command- git commit -m "changes made in README.txt"

[css-assignments 5521f1b] changes made in README.txt

1 file changed, 2 insertions(+), 4 deletions(-)

1. **Switch to master branch.**

Command- git checkout master

Switched to branch 'master'

1. **Make minor changes into README.txt file on line 3 & commit those changes into master.**

Command- Notepad README.txt

Command- git add README.txt

Command- git commit -m "updated README.txt file"

[master 698ac02] updated README.txt file

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

1. **Again switch to ‘css-assignments’ branch.**

Command- git checkout css-assignments

Switched to branch 'css-assignments'

1. **Make minor changes into few files belonging to ‘css-assignments’ branch.**

Command- Notepad sample1.css

1. **Commit those changes.**

Command- git add sample1.css

Command- git commit -m "commit css file"

[css-assignments d1f89d4] commit css file

1 file changed, 4 insertions(+)

1. **Switch to master.**

Command- git checkout master

Switched to branch 'master'

1. **Merge ‘css-assignments’ branch into master. Confirm all css assignments are shown in master.**

Command- git merge css-assignments

1. **Finally delete the ‘css-assignments’ branch.**

Command- git branch -d css-assignments

Deleted branch css-assignments (was d1f89d4).

**SECTION-3 - (JavaScript assignments) Steps to follow:**

1. **Create a new branch ‘js-assignments’.**

Command**-** git branch js-assignment

1. **Switch to ‘js-assignments’ branch.**

Command**-** git checkout js-assignment

Switched to branch 'js-assignment'

1. **Copy all JavaScript assignments inside ‘Assignments’ folder.**

Command**-** touch js1

Command**-** touch js2

1. **Commit JavaScript assignments into ‘js-assignments’ branch.**

Command**-** git add js1 js2

Command**-** git commit -m "commit js files"

[js-assignment aa2a511] commit js files

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

create mode 100644 Desktop/assignments/js1

create mode 100644 Desktop/assignments/js2

1. **Make minor changes into README.txt file on line 1 belonging to ‘js-assignments’ branch.**

Command**-** Notepad README.txt

1. **Commit those changed files.**

**Command-** git add README.txt

**Command-** git commit -m "update the txt file"

[js-assignment e2ecf39] update the txt file

1 file changed, 2 insertions(+)

1. **Switch to master branch.**

**Command-** git checkout master

Switched to branch 'master'

1. **Make minor changes into README.txt file on line 1 & commit those changes into master.**

**Command-** Notepad README.txt

**Command-** git add README.txt

**Command-** git commit -m "commit these changes"

[master eb0f5a1] commit these changes

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

1. **Again switch to ‘js-assignments’ branch.**

**Command-** git checkout js-assignment

Switched to branch 'js-assignment'

1. **Make minor changes into few files belonging to ‘js-assignments’ branch.**

**Command-** Notepad js1

1. **Commit those changes.**

**Command-** git add js1

**Command-** git commit -m "commit changes"

[js-assignment 22181a2] commit changes

1 file changed, 7 insertions(+)

1. **Switch to master.**

**Command-** git checkout master

Switched to branch 'master'

1. **Merge ‘js-assignments’ branch into master. Confirm all JavaScript assignments are shown in master.**

**Command-** git merge js-assignment

Auto-merging Desktop/assignments/README.txt

Merge made by the 'recursive' strategy.

Desktop/assignments/README.txt | 2 ++

Desktop/assignments/js1 | 7 +++++++

Desktop/assignments/js2 | 0

3 files changed, 9 insertions(+)

create mode 100644 Desktop/assignments/js1

create mode 100644 Desktop/assignments/js2

1. **Finally delete the ‘js-assignments’ branch.**

**Command-** git branch -d js-assignment

Deleted branch js-assignment (was 22181a2).

*GIT Remoting*

Objective: Pushing source code into GITHUB & collaborate team members.

SECTION-3 (Pushing assignments to remote repository) - Steps to follow:

1. **Create a github account if you do not have already.**

https://github.com

1. **Login on into github account.**

**Userid**

**Password**

1. **Create new public repository ‘freshersbatch-oct16’.**

**New repository- ‘fresherbatch-oct16’**

1. **Commit & push any sample file to this repository under ‘Assignments’ directory.**

git config --global user.username poojipyla

git remote add origin https://github.com/poojipyla/fresherbatch-oct16.git

git push origin master

SECTION-4 (Pushing source code to remote repository using Eclipse GIT plugin) - Steps to follow:

1. **One developer from project team will create eclipse projects ‘SampleProj’ & add sample source code files. Then commit all files through eclipse GIT plugin.**

**Command-**

1. **Collaborate other team members with your github account so that they can also modify the committed files.**
2. **Other developers from same team will checkout all files from remote repository. This might get conflicts since certain files fail to merge. In such case, merge it manually.**
3. **Commit & push the ‘SampleProj’ project.**