Install GIT & make sure it is added into PATH.

Section 0 -Use GIT as local VCS. Steps to follow:

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

vraizada@LIN24009041 MINGW64 ~

$ mkdir project\_dir

vraizada@LIN24009041 MINGW64 ~

$ cd project\_dir

1. Initialize git version database. (git init)

vraizada@LIN24009041 MINGW64 ~/project\_dir

$ git init

Initialized empty Git repository in C:/Users/vraizada/project\_dir/.git/

1. Create a new file index.html.

C:\Users\vraizada\project\_dir

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

vraizada@LIN24009041 MINGW64 ~/project\_dir (master)

$ git status

On branch master

No commits yet

Untracked files:

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

index.txt

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

1. Stage the index.html file.

vraizada@LIN24009041 MINGW64 ~/project\_dir (master)

$ git add index.txt

vraizada@LIN24009041 MINGW64 ~/project\_dir (master)

$ git status

On branch master

No commits yet

Changes to be committed:

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

new file: index.txt

1. Commit index.html

vraizada@LIN24009041 MINGW64 ~/project\_dir (master)

$ git commit -m "commiting a text file"

[master (root-commit) 3d34c0f] commiting a text file

1 file changed, 1 insertion(+)

create mode 100644 index.txt

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

C:\Users\vraizada\project\_dir

Made Changes to index file as well as added Info.txt

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

vraizada@LIN24009041 MINGW64 ~/project\_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.txt

Untracked files:

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

Info.txt

demo.html

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

1. Configure GIT to ignore all HTML files.

vraizada@LIN24009041 MINGW64 ~/project\_dir (master)

$ touch .gitignore

Adding .Html file in gitignore

vraizada@LIN24009041 MINGW64 ~/project\_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.txt

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")

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

vraizada@LIN24009041 MINGW64 ~/project\_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.txt

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")

1. State & commit index.txt

vraizada@LIN24009041 MINGW64 ~/project\_dir (master)

$ git add index.txt

vraizada@LIN24009041 MINGW64 ~/project\_dir (master)

$ git commit -m "commiting the changed file"

[master b626d99] commiting the changed file

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

1. Log all your comments so far.

vraizada@LIN24009041 MINGW64 ~/project\_dir (master)

$ git log

commit b626d99e51a82526aba483af67563cc7a37aec61 (HEAD -> master)

Author: Vedant-Raizada <vedantraizada65@gmail.com>

Date: Thu Jan 6 14:55:13 2022 +0530

commiting the changed file

commit 3d34c0f5c4df13ab0eadfd4ac59895161e1239e5

Author: Vedant-Raizada <vedantraizada65@gmail.com>

Date: Thu Jan 6 14:27:53 2022 +0530

commiting a text file

1. Make some changes in index.txt

C:\Users\vraizada\project\_dir

Made Changes to the file

1. Revert the change made in the previous step using git command.

vraizada@LIN24009041 MINGW64 ~/project\_dir (master)

$ git stash

Saved working directory and index state WIP on master: b626d99 commiting the changed file

1. Again change index.txt

C:\Users\vraizada\project\_dir

Made changes

1. Stage index.txt

vraizada@LIN24009041 MINGW64 ~/project\_dir (master)

$ git add index.txt

1. Revert back the last stage.

vraizada@LIN24009041 MINGW64 ~/project\_dir (master|REVERTING)

$ git stash

index.txt: needs merge

1. Rename ‘add’ command to ‘my-add’.

vraizada@LIN24009041 MINGW64 ~/project\_dir (master|REVERTING)

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

1. Using my\_add command Stage info.txt again & commit the changes.

vraizada@LIN24009041 MINGW64 ~/project\_dir (master|REVERTING)

$ git my-add info.txt

vraizada@LIN24009041 MINGW64 ~/project\_dir (master|REVERTING)

$ git commit -m "new commit"

[master e949bf5] new commit

1 file changed, 4 insertions(+)

1. Revert the last commit.

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

vraizada@LIN24009041 MINGW64 ~

$ mkdir Assignments

vraizada@LIN24009041 MINGW64 ~

$ cd Assignments

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

C:\Users\vraizada\Assignments

Created a file

1. Commit README.txt file.

vraizada@LIN24009041 MINGW64 ~/Assignments

$ git init

Initialized empty Git repository in C:/Users/vraizada/Assignments/.git/

vraizada@LIN24009041 MINGW64 ~/Assignments (master)

$ git add README.txt

vraizada@LIN24009041 MINGW64 ~/Assignments (master)

$ git commit -m "commiting text file"

[master (root-commit) 246133a] commiting text file

1 file changed, 1 insertion(+)

create mode 100644 README.txt

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

vraizada@LIN24009041 MINGW64 ~/Assignments (master)

$ git branch html-assignments

1. Switch to ‘html-assignments’ branch.

vraizada@LIN24009041 MINGW64 ~/Assignments (master)

$ git checkout html-assignments

Switched to branch 'html-assignments'

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

C:\Users\vraizada\Assignments

Copied to folder

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

vraizada@LIN24009041 MINGW64 ~/Assignments (html-assignments)

$ git add test1.html test2.html

vraizada@LIN24009041 MINGW64 ~/Assignments (html-assignments)

$ git commit -m "new changes in branch"

[html-assignments 4ff0e84] new changes in branch

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

create mode 100644 test1.html

create mode 100644 test2.html

1. Make minor changes into few files belonging to ‘html-assignments’ branch.
2. Commit those changed files.
3. Switch to master branch.

vraizada@LIN24009041 MINGW64 ~/Assignments (html-assignments)

$ git checkout master

Switched to branch 'master'

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

Made changes to README

vraizada@LIN24009041 MINGW64 ~/Assignments (master)

$ git add README.txt

vraizada@LIN24009041 MINGW64 ~/Assignments (master)

$ git commit -m "change in text file "

[master 52b8844] change in text file

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

1. Again switch to ‘html-assignments’ branch.

vraizada@LIN24009041 MINGW64 ~/Assignments (master)

$ git checkout html-assignments

Switched to branch 'html-assignments'

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

C:\Users\vraizada\Assignments

Added another file

1. Commit those changes.

vraizada@LIN24009041 MINGW64 ~/Assignments (html-assignments)

$ git add test3.html

vraizada@LIN24009041 MINGW64 ~/Assignments (html-assignments)

$ git commit -m "added another file"

[html-assignments e172c3d] added another file

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

create mode 100644 test3.html

1. Switch to master.

vraizada@LIN24009041 MINGW64 ~/Assignments (html-assignments)

$ git checkout master

Switched to branch 'master'

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

vraizada@LIN24009041 MINGW64 ~/Assignments (master)

$ git merge html-assignments

Merge made by the 'ort' strategy.

test1.html | 0

test2.html | 0

test3.html | 0

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

create mode 100644 test1.html

create mode 100644 test2.html

create mode 100644 test3.html

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

vraizada@LIN24009041 MINGW64 ~/Assignments (master)

$ git branch -d html-assignments

Deleted branch html-assignments (was e172c3d).

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

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

vraizada@LIN24009041 MINGW64 ~/Assignments (main)

$ git branch css-assignments

1. Switch to ‘css-assignments’ branch.

vraizada@LIN24009041 MINGW64 ~/Assignments (main)

$ git checkout css-assignments

Switched to branch 'css-assignments'

1. Copy all CSS assignments inside ‘Assignments’ folder.
2. Commit CSS assignments into ‘css-assignments’ branch.

vraizada@LIN24009041 MINGW64 ~/Assignments (css-assignments)

$ git add DEMO.css DEMO2.css

vraizada@LIN24009041 MINGW64 ~/Assignments (css-assignments)

$ git commit -m "adding css"

[css-assignments fdad541] adding css

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

create mode 100644 DEMO.css

create mode 100644 DEMO2.css

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

vraizada@LIN24009041 MINGW64 ~/Assignments (css-assignments)

$ git add README.txt

vraizada@LIN24009041 MINGW64 ~/Assignments (css-assignments)

$ git commit -m "for css"

[css-assignments 7a7cb37] for css

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

1. Switch to master branch.

vraizada@LIN24009041 MINGW64 ~/Assignments (css-assignments)

$ git checkout main

Switched to branch 'main'

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

vraizada@LIN24009041 MINGW64 ~/Assignments (main)

$ git add README.txt

vraizada@LIN24009041 MINGW64 ~/Assignments (main)

$ git commit -m "change in line 3"

[main 7693913] change in line 3

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

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

vraizada@LIN24009041 MINGW64 ~/Assignments (main)

$ git checkout css-assignments

Switched to branch 'css-assignments'

1. Make minor changes into few files belonging to ‘css-assignments’ branch.
2. Commit those changes.
3. Switch to master.

vraizada@LIN24009041 MINGW64 ~/Assignments (css-assignments)

$ git checkout main

Switched to branch 'main'

1. Merge ‘css-assignments’ branch into master. Confirm all css assignments are shown in master.
2. Finally delete the ‘css-assignments’ branch.

vraizada@LIN24009041 MINGW64 ~/Assignments (main)

$ git branch -d css-assignments

Deleted branch css-assignments (was 7a7cb37).

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

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

vraizada@LIN24009041 MINGW64 ~/Assignments (main)

$ git branch js-assignments

1. Switch to ‘js-assignments’ branch.

vraizada@LIN24009041 MINGW64 ~/Assignments (main)

$ git checkout js-assignments

Switched to branch 'js-assignments'

1. Copy all JavaScript assignments inside ‘Assignments’ folder.
2. Commit JavaScript assignments into ‘js-assignments’ branch.

vraizada@LIN24009041 MINGW64 ~/Assignments (js-assignments)

$ git add java.js java2.js

vraizada@LIN24009041 MINGW64 ~/Assignments (js-assignments)

$ git commit -m "adding js"

[js-assignments 810b41e] adding js

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

create mode 100644 java.js

create mode 100644 java2.js

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

vraizada@LIN24009041 MINGW64 ~/Assignments (js-assignments)

$ touch READMEjava.txt

vraizada@LIN24009041 MINGW64 ~/Assignments (js-assignments)

$ git add READMEjava.txt

vraizada@LIN24009041 MINGW64 ~/Assignments (js-assignments)

$ git commit -m "java text file"

[js-assignments 9fad167] java text file

1 file changed, 1 insertion(+)

create mode 100644 READMEjava.txt

1. Switch to master branch.

vraizada@LIN24009041 MINGW64 ~/Assignments (js-assignments)

$ git checkout main

Switched to branch 'main'

Your branch is ahead of 'origin/main' by 4 commits.

(use "git push" to publish your local commits)

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

vraizada@LIN24009041 MINGW64 ~/Assignments (main)

$ git add README.txt

vraizada@LIN24009041 MINGW64 ~/Assignments (main)

$ git commit -m "README"

1. [main 9cda998] README

1 file changed, 2 insertions(+)

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

vraizada@LIN24009041 MINGW64 ~/Assignments (main)

$ git checkout js-assignments

Switched to branch 'js-assignments'

1. Make minor changes into few files belonging to ‘js-assignments’ branch.
2. Commit those changes.
3. Switch to master.

vraizada@LIN24009041 MINGW64 ~/Assignments (js-assignments)

$ git checkout main

Switched to branch 'main'

Your branch is ahead of 'origin/main' by 5 commits.

(use "git push" to publish your local commits)

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

vraizada@LIN24009041 MINGW64 ~/Assignments (main)

$ git merge js-assignments

Merge made by the 'ort' strategy.

READMEjava.txt | 1 +

java.js | 0

java2.js | 0

3 files changed, 1 insertion(+)

create mode 100644 READMEjava.txt

create mode 100644 java.js

create mode 100644 java2.js

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

vraizada@LIN24009041 MINGW64 ~/Assignments (main)

$ git branch -d js-assignments

Deleted branch js-assignments (was 9fad167).

*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.
2. Login on into github account.
3. Create new public repository ‘freshersbatch-oct16’.
4. Commit & push any sample file to this repository under ‘Assignments’ directory.

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.
2. Collaborate other team members with your github account so that they can also modify the committed files.
3. 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.
4. Commit & push the ‘SampleProj’ project.