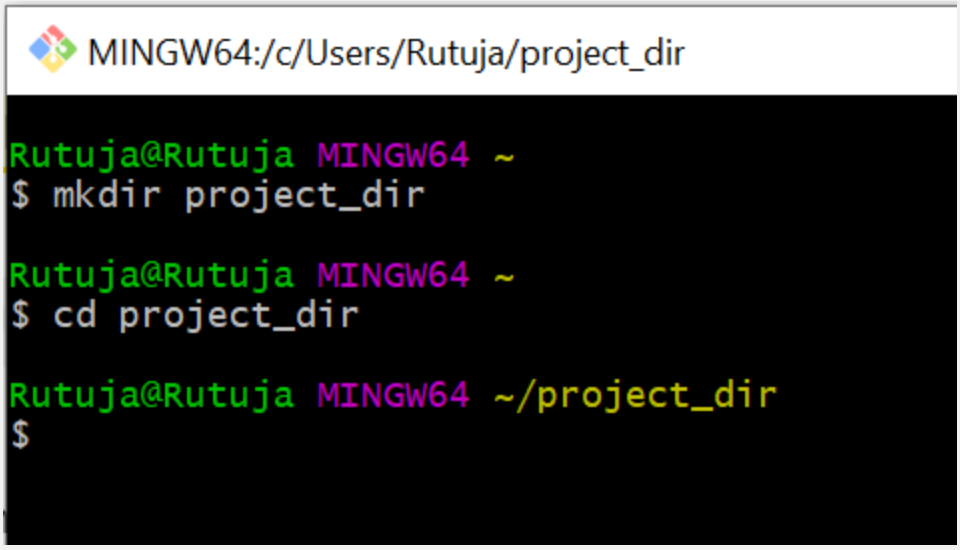
Git Assignment

Install GIT & make sure it is added into PATH.

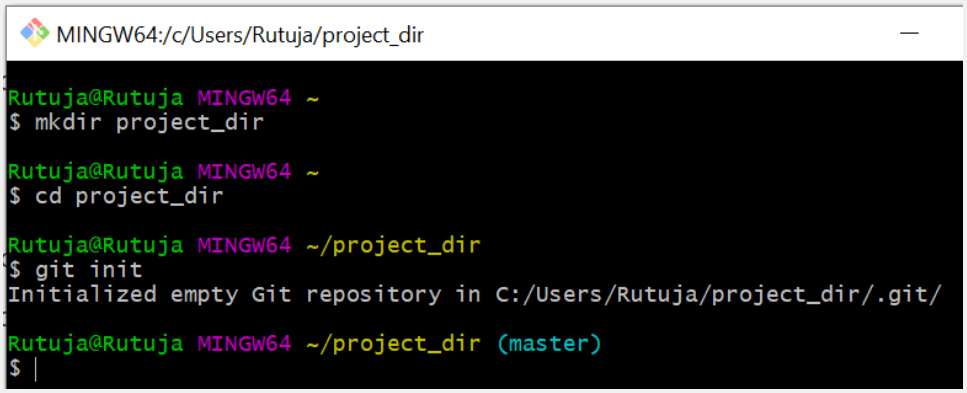
▼ Section 0 - Use GIT as local VCS.

Steps to follow:

1. Create a directory ‘project_dir’ & cd to ‘project_dir’.



2. Initialize git version database. (git init)



3. Create a new file index.html.

```
Rutuja@Rutuja MINGW64 ~/project_dir (master)
$ touch index.html
```

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

```
Rutuja@Rutuja 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.html

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

Rutuja@Rutuja MINGW64 ~/project_dir (master)
$
```

5. Stage the index.html file.

```
Rutuja@Rutuja MINGW64 ~/project_dir (master)
$ git add index.html
```

6. Commit index.html

```
Rutuja@Rutuja MINGW64 ~/project_dir (master)
$ git commit -m "Added index.html"
[master (root-commit) c075e6c] Added index.html
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html
```

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

```
Rutuja@Rutuja MINGW64 ~/project_dir (master)
$ touch info.txt
```

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

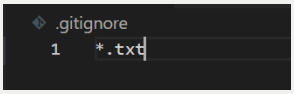
```
Rutuja@Rutuja MINGW64 ~/project_dir (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

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")
```

9. Configure GIT to ignore all txt files.

```
Rutuja@Rutuja MINGW64 ~/project_dir (master)
$ touch .gitignore
```



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

```
Rutuja@Rutuja MINGW64 ~/project_dir (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .gitignore

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
```

11. State & commit index.html

```
Rutuja@Rutuja MINGW64 ~/project_dir (master)
$ git add index.html

Rutuja@Rutuja MINGW64 ~/project_dir (master)
$ git commit -m "Modified index.html"
[master 6a1bdbd] Modified index.html
 2 files changed, 13 insertions(+)
 create mode 100644 .gitignore
```

12. Log all your comments so far.

```
Rutuja@Rutuja MINGW64 ~/project_dir (master)
$ git log
commit 6a1bdbdc46ab207e0ba418312baa3d531c8d5007 (HEAD -> master)
Author: rutu14 <vidyapunjabi7@gmail.com>
Date:   Wed Oct 20 19:32:37 2021 +0530

    Modified index.html

commit c075e6cade91b7516e6faac34bd05ba4f207635b
Author: rutu14 <vidyapunjabi7@gmail.com>
Date:   Wed Oct 20 19:24:10 2021 +0530

    Added index.html
```

13. Make some changes in index.html.
14. Revert the change made in the previous step using git command.

    git revert
15. Again change index.html.
16. Stage index.html
17. Revert back the last stage.
18. Rename ‘add’ command to ‘my-add’.

    git config alias.my-add add
19. Using my_add command Stage index.html again & commit the changes.
20. 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’.

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

```
rlabba@LIN25007681 MINGW64 ~
$ mkdir Assignments

rlabba@LIN25007681 MINGW64 ~
$ cd Assignments

rlabba@LIN25007681 MINGW64 ~/Assignments
$ touch README.txt
```

4. Commit README.txt file.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git add README.txt

rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git commit -m "Added a readme.txt"
[master (root-commit) 29beb96] Added a readme.txt
 1 file changed, 1 insertion(+)
 create mode 100644 README.txt
```

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

```
rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git branch html-assignments
```

6. Switch to ‘html-assignments’ branch.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git checkout html-assignments
Switched to branch 'html-assignments'
```

7. Copy all HTML assignments inside ‘Assignments’ folder.
8. Commit HTML assignments into ‘html-assignments’ branch.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (html-assignments)
$ git add index.html

rlabba@LIN25007681 MINGW64 ~/Assignments (html-assignments)
$ git commit -m "Added index.html"
[html-assignments a7bb4bf] Added index.html
 1 file changed, 12 insertions(+)
 create mode 100644 index.html

rlabba@LIN25007681 MINGW64 ~/Assignments (html-assignments)
$ git status
On branch html-assignments
nothing to commit, working tree clean
```

9. Make minor changes into few files belonging to ‘html-assignments’ branch.
10. Commit those changed files.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (html-assignments)
$ git commit -a -m "Made some changes in index file"
[html-assignments d3faabf] Made some changes in index file
 1 file changed, 1 insertion(+)
```

11. Switch to master branch.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (html-assignments)
$ git checkout master
Switched to branch 'master'
```

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

```
rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git commit -a -m "Made some changes in readme file"
[master db48326] Made some changes in readme file
 1 file changed, 2 insertions(+), 1 deletion(-)
```

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

```
rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git checkout html-assignments
Switched to branch 'html-assignments'
```

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

15. Commit those changes.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (html-assignments)
$ git commit -a -m "Made new changes in index file"
[html-assignments 2df9ba9] Made new changes in index file
 1 file changed, 1 insertion(+)
```

16. Switch to master.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (html-assignments)
$ git checkout master
Switched to branch 'master'
```

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

```
rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git merge html-assignments
Merge made by the 'recursive' strategy.
 index.html | 14 ++++++++++++++
 1 file changed, 14 insertions(+)
 create mode 100644 index.html
```

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

```
rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git branch -d html-assignments
Deleted branch html-assignments (was 2df9ba9).
```

▼ Section 2 - CSS Assignments ✔️

Steps to follow:

1. Create a new branch ‘css-assignments’.
2. Switch to ‘css-assignments’ branch.

```
rlabba@LIN25007681 MINGW64 ~
$ cd Assignments

rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git checkout -b css-assignments
Switched to a new branch 'css-assignments'
```

3. Copy all CSS assignments inside ‘Assignments’ folder.
4. Commit CSS assignments into ‘css-assignments’ branch.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (css-assignments)
$ git add style.css

rlabba@LIN25007681 MINGW64 ~/Assignments (css-assignments)
$ git commit -m "Added style file"
[css-assignments 8e99381] Added style file
 1 file changed, 5 insertions(+)
 create mode 100644 style.css
```

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

```
rlabba@LIN25007681 MINGW64 ~/Assignments (css-assignments)
$ git commit -a -m"Change line 1 of readme file"
[css-assignments 51d2ad1] Change line 1 of readme file
 1 file changed, 1 insertion(+), 1 deletion(-)
```

7. Switch to master branch.
8. Make minor changes into README.txt file on line 3 & commit those changes into
   master.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git commit -a -m "Added new line in master branch readme file"
[master 14dca6a] Added new line in master branch readme file
 1 file changed, 2 insertions(+), 1 deletion(-)
```

9. Again switch to ‘css-assignments’ branch.
10. Make minor changes into few files belonging to ‘css-assignments’ branch.
11. Commit those changes.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (css-assignments)
$ git commit -a -m"Added font-weight"
[css-assignments e16b952] Added font-weight
 1 file changed, 1 insertion(+)
```

12. Switch to master.
13. Merge ‘css-assignments’ branch into master. Confirm all css assignments are shown in
    master.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git merge css-assignments
Already up to date.
```

14. Finally delete the ‘css-assignments’ branch.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git branch -d css-assignments
Deleted branch css-assignments (was e16b952).
```

▼ Section 3 - JavaScript Assignments ✅

Steps to follow:

1. Create a new branch ‘js-assignments’.
2. Switch to ‘js-assignments’ branch.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git checkout -b js-assignments
Switched to a new branch 'js-assignments'
```

3. Copy all JavaScript assignments inside ‘Assignments’ folder.

4. Commit JavaScript assignments into ‘js-assignments’ branch.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (js-assignments)
$ git add app.js

rlabba@LIN25007681 MINGW64 ~/Assignments (js-assignments)
$ git commit -m "Addeda app js file"
[js-assignments 9cd4801] Addeda app js file
 1 file changed, 3 insertions(+)
 create mode 100644 app.js
```

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

```
rlabba@LIN25007681 MINGW64 ~/Assignments (js-assignments)
$ git commit -a -m "Changed line1 in readme"
[js-assignments daa3860] Changed line1 in readme
 1 file changed, 1 insertion(+), 1 deletion(-)
```

7. Switch to master branch.
8. Make minor changes into README.txt file on line 1 & commit those changes into
   master.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git commit -a -m "Added new phrase in readme"
[master a079614] Added new phrase in readme
 1 file changed, 1 insertion(+), 1 deletion(-)
```

9. Again switch to ‘js-assignments’ branch.
10. Make minor changes into few files belonging to ‘js-assignments’ branch.
11. Commit those changes.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (js-assignments)
$ git commit -a -m "Change value of w"
[js-assignments 96fa58b] Change value of w
 1 file changed, 2 insertions(+), 2 deletions(-)
```

12. Switch to master.
13. Merge ‘js-assignments’ branch into master. Confirm all JavaScript assignments are
    shown in master.

```
rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git merge js-assignments
Auto-merging README.txt
CONFLICT (content): Merge conflict in README.txt
Automatic merge failed; fix conflicts and then commit the result.

rlabba@LIN25007681 MINGW64 ~/Assignments (master|MERGING)
$ git commit -a -m "Merge conflict solved"
[master 90c3ed2] Merge conflict solved
```

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

```
rlabba@LIN25007681 MINGW64 ~/Assignments (master)
$ git branch -d js-assignments
Deleted branch js-assignments (was 96fa58b).
```

GIT Remoting

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

▼ Section 3 - Pushing assignments to remote repository

Steps to follow:

39. Create a github account if you do not have already. ✔
40. Login on into github account.✔
41. Create new public repository ‘freshersbatch-oct16’.✔
42. 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. ✔️