

Install GIT & make sure it is added into PATH.

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.
4. Check the git status. You should find index.html as untracked file.
5. Stage the index.html file.
6. Commit index.html
7. Make few changes in index.html & create a new file info.txt file.
8. Check git status. You should find index.html & info.txt as untracked files.
9. Configure GIT to ignore all txt files.
10. Again check the git status. You should find only index.html as untracked file.
11. Stage & commit index.html
12. Log all your comments so far.
13. Make some changes in index.html.
14. Revert the change made in the previous step using git command.
15. Again change index.html.
16. Stage index.html
17. Revert back the last stage.
18. Rename 'add' command to 'my-add'.
19. Using my_add command Stage index.html again & commit the changes.
20. Revert the last commit.

MINGW64:/c/Users/dell2/project_dir

```
dell2@DESKTOP-HDF1ABG MINGW64 ~  
$ mkdir project_dir
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~  
$ cd project_dir
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir  
$ git init  
Initialized empty Git repository in C:/Users/dell2/project_dir/.git/
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)  
$ touch index.html
```

```
dell2@DESKTOP-HDF1ABG 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)

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)  
$ git add index.html
```

```
dell2@DESKTOP-HDF1ABG 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.html
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)  
$ git commit -m "committing index file"  
[master (root-commit) 72df518] committing index file  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 index.html
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)  
$ git status  
On branch master  
nothing to commit, working tree clean
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)  
$ touch info.text
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)
```



MINGW64:/c/Users/dell2/project_dir

```
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    info.text
```

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

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    info.text
```

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

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)
$ vi .gitignore
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    info.text
```

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

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)
$ git commit -m "committing updated index file"
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    info.text
```

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

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    info.text
```

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



MINGW64:/c/Users/dell2/project_dir

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

.gitignore
info.txt

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

dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)
\$ touch info.txt

dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)
\$ git status
On branch master
Untracked files:
(use "git add <file>..." to include in what will be committed)

.gitignore
info.txt
info.txt

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

dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)
\$ git log
commit 72df518b2b6fc749483e040cd40bf9a995ecb3e7 (HEAD -> master)
Author: SiddhantSalgia <siddhant.salgia@gmail.com>
Date: Mon Feb 7 21:52:26 2022 +0530

committing index file

dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)
\$ git add index.html

dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)
\$ git reset

dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)
\$ alias my_add='git add'

dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)
\$ git status
On branch master
Untracked files:
(use "git add <file>..." to include in what will be committed)

.gitignore
info.txt
info.txt

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

dell2@DESKTOP-HDF1ABG MINGW64 ~/project_dir (master)
\$ |

GIT Branching

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

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

21. First take a backup of your assignments & projects. This is required because due to incorrect GIT operation you may lose your files.
22. Create an empty directory 'Assignments' & cd to 'Assignments'.
23. Create a file README.txt inside 'Assignments' & write few lines about the contents of 'Assignments' folder.
24. Commit README.txt file.
25. Now create a new branch 'html-assignments'.
26. Switch to 'html-assignments' branch.
27. Copy all HTML assignments inside 'Assignments' folder.
28. Commit HTML assignments into 'html-assignments' branch.

29. Make minor changes into few files belonging to 'html-assignments' branch.
30. Commit those changed files.
31. Switch to master branch.
32. Make minor changes into README.txt file & commit those changes into master.
33. Again switch to 'html-assignments' branch.
34. Make minor changes into few files belonging to 'html-assignments' branch.
35. Commit those changes.
36. Switch to master.
37. Merge 'html-assignments' branch into master. Confirm all html assignments are shown in master.
38. Finally delete the 'html-assignments' branch.

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

1. Create a new branch 'css-assignments'.
2. Switch to 'css-assignments' branch.
3. Copy all CSS assignments inside 'Assignments' folder.
4. Commit CSS assignments into 'css-assignments' branch.
5. Make minor changes into README.txt file on line 1 belonging to 'css-assignments' branch.
6. Commit those changed files.
7. Switch to master branch.
8. Make minor changes into README.txt file on line 3 & commit those changes into master.
9. Again switch to 'css-assignments' branch.
10. Make minor changes into few files belonging to 'css-assignments' branch.
11. Commit those changes.
12. Switch to master.
13. Merge 'css-assignments' branch into master. Confirm all css assignments are shown in master.
14. Finally delete the 'css-assignments' branch.

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

1. Create a new branch 'js-assignments'.
2. Switch to 'js-assignments' branch.
3. Copy all JavaScript assignments inside 'Assignments' folder.
4. Commit JavaScript assignments into 'js-assignments' branch.
5. Make minor changes into README.txt file on line 1 belonging to 'js-assignments' branch.
6. Commit those changed files.
7. Switch to master branch.

8. Make minor changes into README.txt file on line 1 & commit those changes into master.
9. Again switch to 'js-assignments' branch.
10. Make minor changes into few files belonging to 'js-assignments' branch.
11. Commit those changes.
12. Switch to master.
13. Merge 'js-assignments' branch into master. Confirm all JavaScript assignments are shown in master.
14. Finally delete the 'js-assignments' branch.

MINGW64:/c/Users/dell2/Assignments

```
dell2@DESKTOP-HDF1ABG MINGW64 ~  
$ mkdir Assignments
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~  
$ cd Assignments
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments  
$ git init  
Initialized empty Git repository in C:/Users/dell2/Assignments/.git/
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)  
$ touch README.txt
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)  
$ git add  
Nothing specified, nothing added.  
hint: Maybe you wanted to say 'git add .'?  
hint: Turn this message off by running  
hint: "git config advice.addEmptyPaths false"
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)  
$ git add .
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)  
$ git status  
On branch master
```

No commits yet

Changes to be committed:
 (use "git rm --cached <file>..." to unstage)
 new file: README.txt

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)  
$ git commit -m "Committing README.txt"  
[master (root-commit) 9590e73] Committing README.txt  
 1 file changed, 0 insertions(+), 0 deletions(-)  
 create mode 100644 README.txt
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)  
$ git branch html-assignments
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)  
$ git checkout html-assignments  
Switched to branch 'html-assignments'
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)  
$ git status  
On branch html-assignments  
nothing to commit, working tree clean
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)  
$ git add sample1.html  
fatal: pathspec 'sample1.html' did not match any files
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)
```


MINGW64:/c/Users/dell2/Assignments

```
$ touch sample1.html
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)
```

```
$ git add sample1.html
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)
```

```
$ git status
```

```
On branch html-assignments
```

```
Changes to be committed:
```

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

```
new file:   sample1.html
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)
```

```
$ git commit -m "Committing html file"
```

```
[html-assignments 46d5632] Committing html file
```

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

```
create mode 100644 sample1.html
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)
```

```
$ git status
```

```
On branch html-assignments
```

```
nothing to commit, working tree clean
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)
```

```
$ vi sample1.html
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)
```

```
$ git status
```

```
On branch html-assignments
```

```
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:   sample1.html
```

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

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)
```

```
$ git add sample1.html
```

```
warning: LF will be replaced by CRLF in sample1.html.
```

```
The file will have its original line endings in your working directory
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)
```

```
$ git commit -m "committing after change in html file"
```

```
[html-assignments 3f0acec] committing after change in html file
```

```
1 file changed, 2 insertions(+)
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)
```

```
$ git checkout master
```

```
Switched to branch 'master'
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
```

```
$ git status
```

```
On branch master
```

```
nothing to commit, working tree clean
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
```

```
$ vi README.txt
```

MINGW64:/c/Users/dell2/Assignments

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)

\$ git add README.txt

warning: LF will be replaced by CRLF in README.txt.

The file will have its original line endings in your working directory

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)

\$ git commit -m "committing README file after change"

[master cede854] committing README file after change

1 file changed, 1 insertion(+)

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)

\$ git checkout html-assignments

Switched to branch 'html-assignments'

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)

\$ vi sample1.html

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)

\$ git add sample1.html

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)

\$ git status

On branch html-assignments

Changes to be committed:

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

modified: sample1.html

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)

\$ git checkout master

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

sample1.html

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

Aborting

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)

\$ git commit -m "committing again after change"

[html-assignments 12204d6] committing again after change

1 file changed, 2 insertions(+)

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (html-assignments)

\$ git checkout master

Switched to branch 'master'

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)

\$ git merge html-assignments

Merge made by the 'ort' strategy.

sample1.html | 4 ++++

1 file changed, 4 insertions(+)

create mode 100644 sample1.html

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)

\$ git branch -d html-assignments

Deleted branch html-assignments (was 12204d6).

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)

\$

MINGW64:/c/Users/dell2/Assignments

```
de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
$ git branch css-assignments

de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
$ git checkout css-assignments
Switched to branch 'css-assignments'

de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (css-assignments)
$ git status
On branch css-assignments
nothing to commit, working tree clean

de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (css-assignments)
$ touch sample1.css

de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (css-assignments)
$ git status
On branch css-assignments
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        sample1.css

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

de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (css-assignments)
$ git add sample1.css

de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (css-assignments)
$ git commit -m "committing css file"
[css-assignments 5302301] committing css file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 sample1.css

de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (css-assignments)
$ vi README.txt

de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (css-assignments)
$ git add README.txt

de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (css-assignments)
$ git status
On branch css-assignments
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.txt

de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (css-assignments)
$ git commit -m "committing README.txt"
[css-assignments dd420cd] committing README.txt
1 file changed, 2 insertions(+), 1 deletion(-)

de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (css-assignments)
$ git checkout master
Switched to branch 'master'

de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
```


MINGW64:/c/Users/dell2/Assignments

```
[css-assignments 5302301] committing css file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 sample1.css
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (css-assignments)
$ vi README.txt
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (css-assignments)
$ git add README.txt
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (css-assignments)
$ git status
On branch css-assignments
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.txt
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (css-assignments)
$ git commit -m "committing README.txt"
[css-assignments dd420cd] committing README.txt
1 file changed, 2 insertions(+), 1 deletion(-)
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (css-assignments)
$ git checkout master
Switched to branch 'master'
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
$ git add README.txt
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
$ git commit -m "committing master README.txt"
On branch master
nothing to commit, working tree clean
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
$ git checkout css-assignment
error: pathspec 'css-assignment' did not match any file(s) known to git
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
$ git add sample.css
fatal: pathspec 'sample.css' did not match any files
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
$ git add sample1.css
fatal: pathspec 'sample1.css' did not match any files
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
$ git branch -d css-assignments
error: The branch 'css-assignments' is not fully merged.
If you are sure you want to delete it, run 'git branch -D css-assignments'.
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
$ git branch -D css-assignments
Deleted branch css-assignments (was dd420cd).
```

```
dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
$ |
```

MINGW64:/c/Users/dell2/Assignments

```
de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
$ git branch js-assignments
```

```
de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
$ git checkout js-assignments
Switched to branch 'js-assignments'
```

```
de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (js-assignments)
$ touch index.js
```

```
de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (js-assignments)
$ vi vi index.js
2 files to edit
```

```
de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (js-assignments)
$ git add index.js
```

```
de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (js-assignments)
$ git commit -m "committing js file"
[js-assignments 7501c00] committing js file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.js
```

```
de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (js-assignments)
$ git add README.txt
```

```
de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (js-assignments)
$ git commit -m "committing README.txt file"
On branch js-assignments
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  vi
```

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

```
de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (js-assignments)
$ git checkout master
Switched to branch 'master'
```

```
de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
$ git add README.txt
```

```
de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
$ git commit -m "committing README.txt file from master branch"
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  vi
```

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

```
de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)
$ git checkout js-assignments
Switched to branch 'js-assignments'
```

```
de112@DESKTOP-HDF1ABG MINGW64 ~/Assignments (js-assignments)
$ git commit -m "commit change"
```

MINGW64:/c/Users/dell2/Assignments

On branch master

Untracked files:

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

vi

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

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)

\$ git checkout js-assignments

Switched to branch 'js-assignments'

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (js-assignments)

\$ git commit -m "commit change"

On branch js-assignments

Untracked files:

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

vi

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

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (js-assignments)

\$ git checkout master

Switched to branch 'master'

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)

\$ git merge js-assignments

Updating d7fbf07..7501c00

Fast-forward

index.js | 0

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

create mode 100644 index.js

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)

\$ git status

On branch master

Untracked files:

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

vi

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

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)

\$ git add .

warning: LF will be replaced by CRLF in vi.

The file will have its original line endings in your working directory

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)

\$ git commit -m "final done"

[master 9ee2b5a] final done

1 file changed, 3 insertions(+)

create mode 100644 vi

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)

\$ git branch -d js-assignments

Deleted branch js-assignments (was 7501c00).

dell2@DESKTOP-HDF1ABG MINGW64 ~/Assignments (master)

\$ |