

Section-0

1. Create a directory 'project_dir' & cd to 'project_dir'.

```
MINGW64:/c/Users/sarva/project_dir
sarva@LAPTOP-TN91TFIR MINGW64 ~
$ mkdir project_dir
sarva@LAPTOP-TN91TFIR MINGW64 ~
$ cd project_dir
```


 project_dir	26-10-2021 16:45	File folder	(Empty folder)
---	------------------	-------------	----------------

2. Initialize git version database.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/project_dir
$ git init
Initialized empty Git repository in C:/Users/sarva/project_dir/.git/
```

3. Create a new file index.html.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/project_dir (master)
$ touch index.html
```

 index	26-10-2021 16:45	HTML File	0 KB
---	------------------	-----------	------

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

```
sarva@LAPTOP-TN91TFIR 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)
```

5. Stage the index.html file.

BEFORE STAGING WE NEED TO ADD GLOBAL EMAIL

git config --global user.email sarvade34@gmail.com

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/project_dir (master)
$ git add .
sarva@LAPTOP-TN91TFIR 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
```

6. Commit index.html

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/project_dir (master)
$ git commit -m 'first html file'
[master (root-commit) d59afc8] first html file
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

After changing some content in html file

```
sarva@LAPTOP-TN91TFIR 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
```

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

touch info.txt

Name	Date modified	Type	Size
 index	26-10-2021 17:51	HTML File	1 KB
 info	26-10-2021 17:52	Text Document	0 KB

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

```
sarva@LAPTOP-TN91TFIR 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.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/project_dir (master)
$ touch gitignore
```

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

```
sarva@LAPTOP-TN91TFIR 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)
 .gitignore

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

11. State & commit index.html

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/project_dir (master)
$ git add index.html

sarva@LAPTOP-TN91TFIR MINGW64 ~/project_dir (master)
$ git commit -m "modified html"
[master 783b30c] committed index with <head>
1 file changed, 5 insertions(+)
```

12. Log all your comments so far.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/project_dir (master)
$ git log
commit 02eb2b704b97dd2dd8d50031b384c386cf6bd5e8 (HEAD -> master)
Author: sarvade nithin <sarvade34@gmail.com>
Date: Tue Oct 26 18:33:38 2021 +0530

    modified html

commit d59afc8ad188b1a67b0b700a61357ac087a47174
Author: sarvade nithin <sarvade34@gmail.com>
Date: Tue Oct 26 17:47:32 2021 +0530

    first html file
```

13. Make some changes in index.html.

Changes has been done

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

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/project_dir (master)
$ git checkout --
```

15. Again some changes in index.html

16. Stage index.html

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/project_dir (master)
$ git add index.html

sarva@LAPTOP-TN91TFIR MINGW64 ~/project_dir (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

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

17. Revert back the last stage

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/project_dir (master)
$ git restore --staged index.html

sarva@LAPTOP-TN91TFIR 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)
`.gitignore`

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

18. Rename 'add' command to 'my-add'

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/project_dir (master)
$ git config --global alias.my-add add
```

19. Using my_add command Stage index.html again & commit the changes

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/project_dir (master)
$ git my-add index.html
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/project_dir (master)
```

```
$ git commit -m "committed index.html acc to steps"
[master 30832b3] committed index.html acc to steps
1 file changed, 3 insertions(+)
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/project_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)

20. Revert the last commit.

Git branching

SECTION-1:

22.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~  
$ mkdir Assignments
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~  
$ cd Assignments
```

23.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments  
$ touch README.txt
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments  
$ git init  
Initialized empty Git repository in C:/Users/sarva/Assignments/.git/
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)  
$ git add .
```

24.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)  
$ git commit -m "committed readme"  
[master (root-commit) 6216998] committed readme  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 README.txt
```

25.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)  
$ git branch html-assignments
```

26.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)  
$ git checkout html-assignments  
Switched to branch 'html-assignments'
```

27.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (html-assignments)  
$ git status  
On branch html-assignments  
nothing to commit, working tree clean
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (html-assignments)  
$ git checkout -b html-assignment  
Switched to a new branch 'html-assignment'
```

28.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (html-assignment)  
$ git checkout master -- README.txt
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (html-assignment)  
$ git status  
On branch html-assignment  
nothing to commit, working tree clean
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (html-assignment)  
$ git commit -m 'adding README.txt to a new branch'  
On branch html-assignment  
nothing to commit, working tree clean
```

29.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (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")

```

30.
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (html-assignment)
$ git commit -m 'changes in README.txt'
>
[html-assignments 6Y5698] changes in README.txt 1 file changed, 1 insertion(+)

31 to 38
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)
$ git checkout master
Already on 'master'
M       README.txt

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.txt

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)
$ git add README.txt

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)
$ git commit -m 'again modified README.txt in master branch'
[master 969e504] again modified README.txt in master branch
 1 file changed, 1 insertion(+)

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)
$ git checkout html-assignments
Switched to branch 'html-assignments'

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (html-assignments)
$ git status
On branch html-assignments
nothing to commit, working tree clean

sarva@LAPTOP-TN91TFIR 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:   README.txt

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

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (html-assignments)
$ git add README.txt

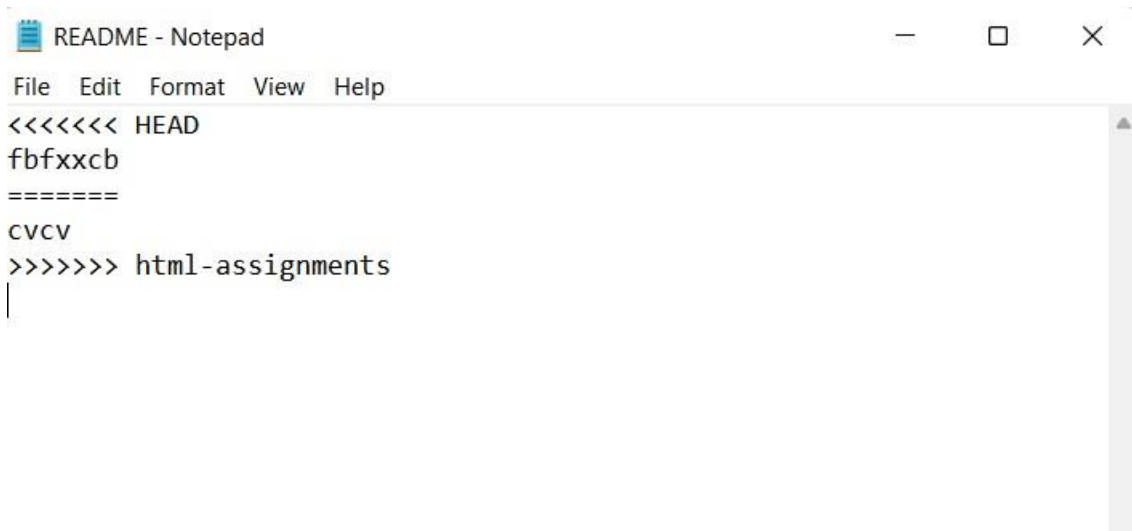
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (html-assignments)
$ git status
On branch html-assignments
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.txt

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (html-assignments)
$ git commit -m 'made changes in second branch'
[html-assignments 1fccdf4] made changes in second branch
 1 file changed, 1 insertion(+)

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (html-assignments)
$ git checkout master
Switched to branch 'master'

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)
$ git merge html-assignments
Auto-merging README.txt
CONFLICT (content): Merge conflict in README.txt
Automatic merge failed; fix conflicts and then commit the result.

```



```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (master|MERGING)
$ git branch -D html-assignments
Deleted branch html-assignments (was 1fccdf4).
```

SECTION -2

(CSS assignment)

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.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (css-assignments)
$ git checkout -b css-assignments
fatal: A branch named 'css-assignments' already exists.
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (css-assignments)
$ git checkout master README.txt
Updated 1 path from 3b52140
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (css-assignments)
$ git status
On branch css-assignments
nothing to commit, working tree clean
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (css-assignments)
$ git status
On branch css-assignments
nothing to commit, working tree clean
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (css-assignments)
$ git status
On branch css-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:   README.txt
```

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

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (css-assignments)
$ git add README.txt
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (css-assignments)
$ git status
```


On branch css-assignments

Changes to be committed:

(use "git restore --staged <file>..." to unstage)
modified: README.txt

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (css-assignments)

\$ git commit -m '1st change in css branch'

[css-assignments 998f177] 1st change in css branch
1 file changed, 1 insertion(+), 1 deletion(-)

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (css-assignments)

\$ git checkout master

bash: git: command not found

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (css-assignments)

\$ git checkout master

Switched to branch 'master'

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)

\$ git status

On branch master

nothing to commit, working tree clean

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (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")

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)

\$ git add README.txt

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)

\$ git status

On branch master

Changes to be committed:

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

modified: README.txt

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)

\$ git commit -m '2nd change in master branch'

[master c50a723] 2nd change in master branch
1 file changed, 1 insertion(+), 1 deletion(-)

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)

\$ git checkout css-assignments

Switched to branch 'css-assignments'

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (css-assignments)

\$ git status

On branch css-assignments

nothing to commit, working tree clean

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (css-assignments)

\$ git status

On branch css-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: README.txt

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

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (css-assignments)

\$ git add README.txt

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (css-assignments)

\$ git commit -m 'committing readme from css branch'

[css-assignments f621c8c] committing readme from css branch
1 file changed, 1 insertion(+), 1 deletion(-)

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (css-assignments)
$ git status
On branch css-assignments
nothing to commit, working tree clean

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (css-assignments)
$ git branch
css-assignments
* css-assignments
css-assinments
html-assignment
master

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (css-assignments)
$ git diff master css-assignments
diff --git a/README.txt b/README.txt
index d9e7c6f..e440e5c 100644
--- a/README.txt
+++ b/README.txt
@@ -1, +1 @@
-2nd change
+3
\ No newline at end of file

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (css-assignments)
$ git merge css-assignments
Already up to date.

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (css-assignments)
$ git checkout master
Switched to branch 'master'

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (master)
$ git branch -D css-assignments
Deleted branch css-assignments (was f621c8c).
```

SECTION-3

(JavaScript assignment)

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.

1.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)
$ git branch js-assignment

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)
$ git branch
js-assignment
* master
```

2.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)
$ git checkout js-assignments
error: pathspec 'js-assignments' did not match any file(s) known to git

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (master)
$ git checkout js-assignment
Switched to branch 'js-assignment'
```

3.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (js-assignment)
$ touch f1.txt

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (js-assignment)
$ touch f2.txt

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assignments (js-assignment)
$ git add f1.txt
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (js-assignment)
$ git add f2.txt
```

4.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (js-assignment)
$ git commit -m "committed js"
[js-assignment 82a94c3] committed js
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 f1.txt
create mode 100644 f2.txt
```

5.

Done some changes in README.txt that belongs to js-assignment branch

6.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (js-assignment)
$ git add README.txt
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (js-assignment)
$ git commit -m "changes in readme"
[js-assignment f898bfc] changes in readme
1 file changed, 1 deletion(-)
delete mode 100644 README.txt
```

7.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (js-assignment)
$ git checkout master
Switched to branch 'master'
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (master)
$ git branch
js-assignment
* master
```

8.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (master)
$ git add README.txt
```

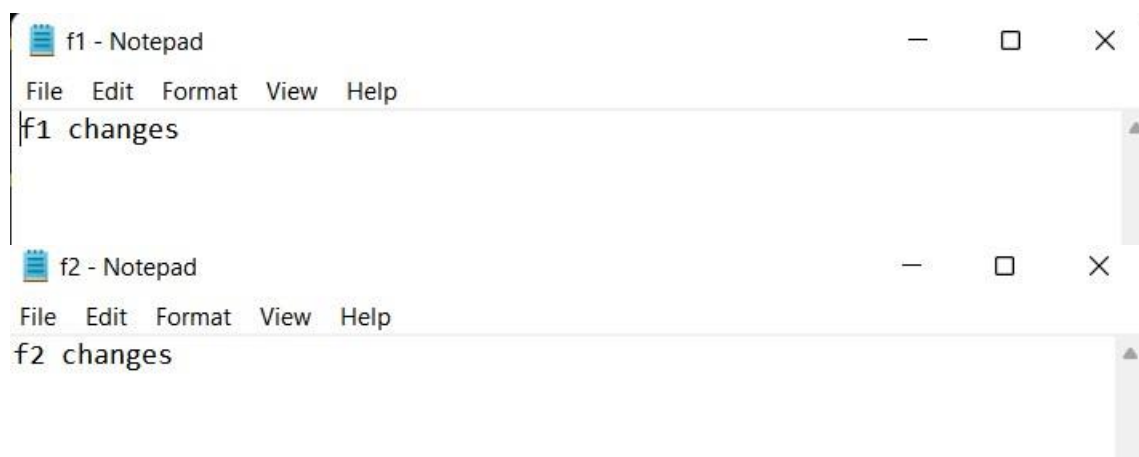
```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (master)
$ git commit -m "committed readme"
[master 189bf99] committed readme
1 file changed, 2 insertions(+), 1 deletion(-)
```

9.

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (master)
$ git checkout js-assignment
Switched to branch 'js-assignment'
```

```
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (js-assignment)
$ git branch
* js-assignment
master
```

10.



11.
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (js-assignment)
\$ git add f1.txt

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (js-assignment)
\$ git add f2.txt

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (js-assignment)
\$ git commit -m "f1 and f2 committed"
[js-assignment 1d4fa0c] f1 and f2 committed
2 files changed, 2 insertions(+)

12.
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (js-assignment)
\$ git checkout master
Switched to branch 'master'

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (master)
\$ git branch
js-assignment

13.
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (master)
\$ git merge js-assignment
Already up to date.

sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (master)
\$ git ls-files
README.txt
f1.txt
f2.txt
index.html

14.
sarva@LAPTOP-TN91TFIR MINGW64 ~/Assingnments (master)
\$ git branch -d js-assignment
Deleted branch js-assignment (was 1d4fa0c).