

Assignment 1:

Section 0:

- 1) Create a directory "project_dir" and cd to "project_dir":

```
cc@cc-PC MINGW32 ~ (master)
$ mkdir project_dir

cc@cc-PC MINGW32 ~ (master)
$ cd project_dir
```

- 2) Initialize git version database. (git init):

```
cc@cc-PC MINGW32 ~/project_dir (master)
$ git init
Initialized empty Git repository in C:/Users/cc/project_dir/.git/
```

- 3) Create a new file index.html:

```
cc@cc-PC MINGW32 ~/project_dir (master)
$ touch index.html
```

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

```
cc@cc-PC MINGW32 ~/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:

```
cc@cc-PC MINGW32 ~/project_dir (master)
$ git add index.html
```

6) commit index.html:

```
cc@cc-PC MINGW32 ~/project_dir (master)
$ git commit -m "index file created"
[master (root-commit) f00a8fc] index file created
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.html
```

7) Make few changes in index.html and create a new file info.txt:

```
cc@cc-PC MINGW32 ~/project_dir (master)
$ touch info.txt
```

8) Check git status. You should find index.html and info.txt as untracked file:

```
cc@cc-PC MINGW32 ~/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)
        info.txt

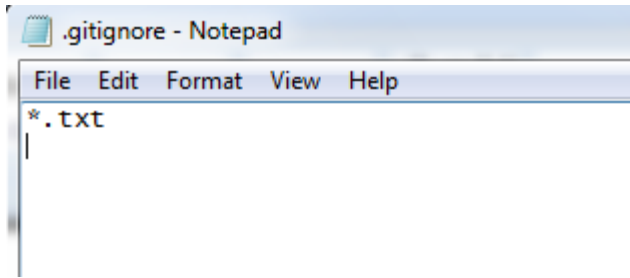
cc@cc-PC MINGW32 ~/project_dir (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

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

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

9) Configure git to ignore all txt files:

```
cc@cc-PC MINGW32 ~/project_dir (master)
$ touch .gitignore
```



10) Again check the git status. You should find only index.html as untracked files:

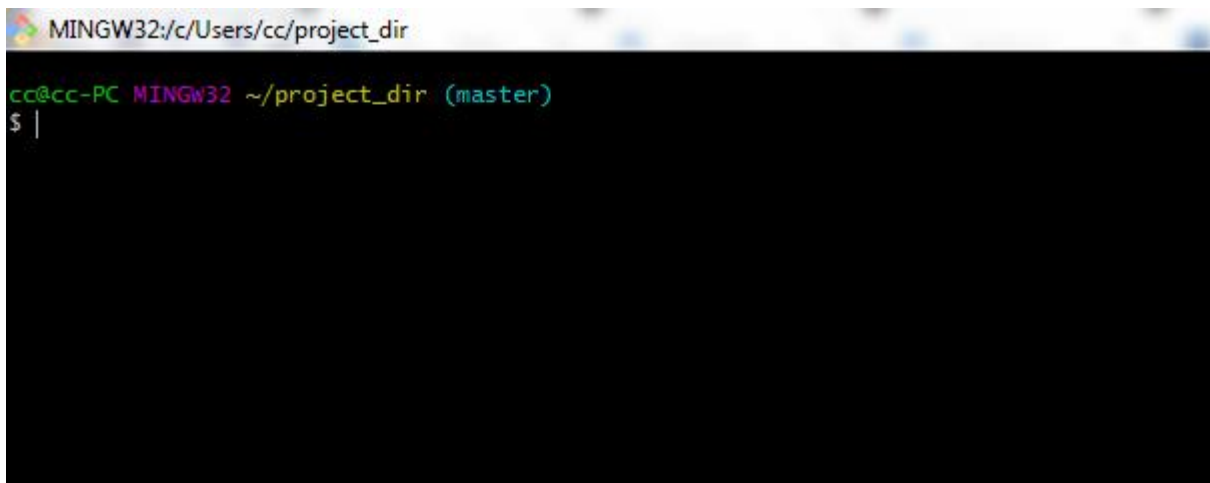
```
cc@cc-PC MINGW32 ~/project_dir (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

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

11) Stage and commit index.html:

```
cc@cc-PC MINGW32 ~/project_dir (master)
$ git commit -m index.html
[master bd47b73] index.html
1 file changed, 15 insertions(+)
```

12) Log all your comments so far:



13) Make some changes in index.html

```
cc@cc-PC MINGW32 ~/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
```

14) Revert the changes in the previous step using git command:

```
cc@cc-PC MINGW32 ~/project_dir (master)
$ git revert head
Removing .gitignore
[master f2e762f] Revert "index.html" reverted to previous stage This reverts commit d54aa15f265eff52a458c7f6a4fe646a4b6489ad.
2 files changed, 1 insertion(+), 3 deletions(-)
delete mode 100644 .gitignore
```

15) Again change index.html

```
cc@cc-PC MINGW32 ~/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
```

16) Stage index.html

```
cc@cc-PC MINGW32 ~/project_dir (master)
$ git add index.html
```

17) Revert back the last stage:

```
cc@cc-PC MINGW32 ~/project_dir (master)
$ git revert head
[master d6d876f] Revert "index.html" This reverts commit 3fef6c98322f1cc2c0778af2aa1eb492244f5bc6.
1 file changed, 1 insertion(+), 3 deletions(-)
```

18) Rename "add" command to "my-add"

```
cc@cc-PC MINGW32 ~/project_dir (master)
$ git config --global alias.my-add add
```

19) Using my-add command stage index.html again and commit changes

```
cc@cc-PC MINGW32 ~/project_dir (master)
$ git my-add index.html
```

20) Revert last commit:

```
cc@cc-PC MINGW32 ~/project_dir (master)
$ git revert head
[master 2fb1681] Revert "Revert "index.html"" Reverted to last commit This reverts commit
d6d876faaef458a24fe43729b429a4527f56bb8e.
1 file changed, 3 insertions(+), 1 deletion(-)
```

Section 1:

1. Create an empty directory "Assignments" and cd to "Assignments" :

```
cc@cc-PC MINGW32 ~ (master)
$ mkdir Assignments

cc@cc-PC MINGW32 ~ (master)
$ cd Assignments
```

2. Create a file README.txt inside "Assignments" & write few lines about contents of "Assignments" folder:

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git add README.txt

cc@cc-PC MINGW32 ~/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
```

3. Commit README.txt

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git commit -m README.txt
[master (root-commit) 9c0ccd9] README.txt
1 file changed, 1 insertion(+)
 create mode 100644 README.txt
```

4. Now create a new branch "html-assignments":

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git branch html-assignments
```

5. Switch to "html-assignments" branch:

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git checkout html-assignments
Switched to branch 'html-assignments'
```

6. Copy all HTML assignments inside "html-assignments":

```
cc@cc-PC MINGW32 ~/Assignments (html-assignments)
$ git add *.html

cc@cc-PC MINGW32 ~/Assignments (html-assignments)
$ git status
On branch html-assignments
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   page1.html
    modified:   simple.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    page1_css.css
    page1_js.js
    simple_css.css
    simple_js.js
```

7. Commit HTML assignments into "html-assignments":

```
cc@cc-PC MINGW32 ~/Assignments (html-assignments)
$ git commit -m "All html files"
[html-assignments e287e0a] All html files
 2 files changed, 23 insertions(+), 2 deletions(-)
 create mode 100644 page1.html
```

8. Make minor changes into few files belonging to "html-assignments" branch:

```
cc@cc-PC MINGW32 ~/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:   page1.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    page1_css.css
    page1_js.js
    simple_css.css
    simple_js.js

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

9. Commit those changed files:

```
cc@cc-PC MINGW32 ~/Assignments (html-assignments)
$ git add page1.html

cc@cc-PC MINGW32 ~/Assignments (html-assignments)
$ git commit -m "Modified html files"
[html-assignments 920479c] Modified html files
1 file changed, 1 insertion(+), 1 deletion(-)
```

10. Switch to master branch:

```
cc@cc-PC MINGW32 ~/Assignments (html-assignments)
$ git checkout master
Switched to branch 'master'
```

11. Make minor changes into README.txt files and commit those changes into master:

```
cc@cc-PC MINGW32 ~/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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        page1_css.css
        page1_js.js
        simple_css.css
        simple_js.js

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

cc@cc-PC MINGW32 ~/Assignments (master)
$ git add README.txt

cc@cc-PC MINGW32 ~/Assignments (master)
$ git commit -m README>txt
```

12. Again switch to "html-assignments":

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git checkout html-assignments
Switched to branch 'html-assignments'
```

13. Make minor changes to few files belonging to "html-assignments" branch:

```
cc@cc-PC MINGW32 ~/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:   page1.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        page1_css.css
        page1_js.js
        simple_css.css
        simple_js.js
        txt

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

14. Commit those changes:

```
cc@cc-PC MINGW32 ~/Assignments (html-assignments)
$ git add page1.html

cc@cc-PC MINGW32 ~/Assignments (html-assignments)
$ git commit -m "Final html update"
[html-assignments d60d31f] Final html update
1 file changed, 1 insertion(+), 1 deletion(-)
```

15. Switch to master:

```
cc@cc-PC MINGW32 ~/Assignments (html-assignments)
$ git checkout master
Switched to branch 'master'

cc@cc-PC MINGW32 ~/Assignments (master)
$ |
```

16. Merge "html-assignments" branch into master. Confirm all html assignments in master:

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git merge html-assignments
Merge made by the 'recursive' strategy.
 page1.html | 15 ++++++++
 simple.html | 10 ++++++--
 2 files changed, 23 insertions(+), 2 deletions(-)
 create mode 100644 page1.html

cc@cc-PC MINGW32 ~/Assignments (master)
$ git log --oneline
f45bcce (HEAD -> master) Merge branch 'html-assignments' Merged html-assignments
branch
d60d31f (html-assignments) Final html update
b22640d README
920479c Modified html files
e287e0a All html files
d4f5aee TML Assignments
9c0ccd9 README.txt
```



```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git branch --merged
html-assignments
*= master
```

17. Finally delete the “html-assignments” branch:

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git branch -d html-assignments
Deleted branch html-assignments (was d60d31f).
```

Section 2: CSS Assignments

1. Create a new branch “css-assignments”:

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git branch css-assignments
```

2. Switch to “css-assignments” branch:

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git checkout css-assignments
Switched to branch 'css-assignments'

cc@cc-PC MINGW32 ~/Assignments (css-assignments)
$ |
```

3. Copy all CSS assignments inside “Assignments” folder:
4. Commit CSS assignments into “css-assignments” branch:

```
cc@cc-PC MINGW32 ~/Assignments (css-assignments)
$ git add *.css

cc@cc-PC MINGW32 ~/Assignments (css-assignments)
$ git status
On branch css-assignments
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   page1_css.css
    new file:   simple_css.css

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    page1_js.js
    simple_js.js
    txt

cc@cc-PC MINGW32 ~/Assignments (css-assignments)
$ git commit -m "Added css files"
[css-assignments 82ef6ae] Added css files
2 files changed, 70 insertions(+)
create mode 100644 page1_css.css
create mode 100644 simple_css.css
```

5. Make minor changes into README.txt file on line 1 belonging to css-assignments branch:

```
cc@cc-PC MINGW32 ~/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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        page1_js.js
        simple_js.js
        txt

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

6. Commit those changed files:

```
cc@cc-PC MINGW32 ~/Assignments (css-assignments)
$ git add README.txt

cc@cc-PC MINGW32 ~/Assignments (css-assignments)
$ git commit -m "css-README"
[css-assignments 16cad79] css-README
1 file changed, 3 insertions(+), 1 deletion(-)
```

7. Switch to master branch:

```
cc@cc-PC MINGW32 ~/Assignments (css-assignments)
$ git checkout master
Switched to branch 'master'
```

8. Make minor changes into README.txt files on line 1 and commit those changes into master:

```
cc@cc-PC MINGW32 ~/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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        page1_js.js
        simple_js.js
        txt

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

cc@cc-PC MINGW32 ~/Assignments (master)
$ git add README.txt

cc@cc-PC MINGW32 ~/Assignments (master)
$ git commit -m "Modified readme"
[master c7cece4] Modified readme
1 file changed, 4 insertions(+), 1 deletion(-)
```

9. Again switch to css-assignments branch:

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git checkout css-assignments
Switched to branch 'css-assignments'

cc@cc-PC MINGW32 ~/Assignments (css-assignments)
$ |
```

10. Make minor changes into few files belonging to css-assignments branch”:

```
cc@cc-PC MINGW32 ~/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:   page1_css.css

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        page1_js.js
        simple_js.js
        txt

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

11. Commit those changes:

```
cc@cc-PC MINGW32 ~/Assignments (css-assignments)
$ git add *.css

cc@cc-PC MINGW32 ~/Assignments (css-assignments)
$ git commit -m "Complete CSS assignment"
[css-assignments 9a5a07d] Complete CSS assignment
1 file changed, 1 insertion(+), 1 deletion(-)
```

12. Switch to master:

```
cc@cc-PC MINGW32 ~/Assignments (css-assignments)
$ git checkout master
Switched to branch 'master'

cc@cc-PC MINGW32 ~/Assignments (master)
$ |
```

13. Merge css-assignments branch into master. Confirm all css assignments are shown in master:

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git merge css-assignments
Auto-merging README.txt
CONFLICT (content): Merge conflict in README.txt
Automatic merge failed; fix conflicts and then commit the result.

cc@cc-PC MINGW32 ~/Assignments (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
  new file:   page1_css.css
  new file:   simple_css.css

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   README.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    page1_js.js
    simple_js.js
    txt

cc@cc-PC MINGW32 ~/Assignments (master|MERGING)
$ vi README.txt

cc@cc-PC MINGW32 ~/Assignments (master|MERGING)
$ git add README.txt

cc@cc-PC MINGW32 ~/Assignments (master|MERGING)
$ git commit -m "Conflits cleared"
[master 31b1abd] Conflits cleared

cc@cc-PC MINGW32 ~/Assignments (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    page1_js.js
    simple_js.js
    txt

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

cc@cc-PC MINGW32 ~/Assignments (master)
$ git branch --merged
  css-assignments
* master
```

14. Finally delete css-assignments:

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git branch -d css-assignments
Deleted branch css-assignments (was 9a5a07d).
```

Section 3: JavaScript Assignments:

1. Create a new branch "js-assignments":

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git branch js-assignments
```

2. Switch to "js-assignments" branch

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git checkout js-assignments
Switched to branch 'js-assignments'

cc@cc-PC MINGW32 ~/Assignments (js-assignments)
$ |
```

3. Copy all JS assignments inside "Assignments" folder:

```
cc@cc-PC MINGW32 ~/Assignments (js-assignments)
$ git add *.js

cc@cc-PC MINGW32 ~/Assignments (js-assignments)
$ git status
On branch js-assignments
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   page1_js.js
        new file:   simple_js.js

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

4. Commit JS assignments into "js-assignments" branch:

```
cc@cc-PC MINGW32 ~/Assignments (js-assignments)
$ git commit -m "Js assignment"
[js-assignments bff2292] Js assignment
2 files changed, 240 insertions(+)
 create mode 100644 page1_js.js
 create mode 100644 simple_js.js
```

5. Make minor changes into README.txt file on line 1 belonging to js-assignments branch:

```
cc@cc-PC MINGW32 ~/Assignments (js-assignments)
$ git status
On branch js-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

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

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

6. Commit those changed files

```
cc@cc-PC MINGW32 ~/Assignments (js-assignments)
$ git add README.txt

cc@cc-PC MINGW32 ~/Assignments (js-assignments)
$ git commit -m "README JS"
[js-assignments 736cfa9] README JS
1 file changed, 2 insertions(+), 3 deletions(-)
```

7. Switch to master branch

```
cc@cc-PC MINGW32 ~/Assignments (js-assignments)
$ git checkout master
Switched to branch 'master'

cc@cc-PC MINGW32 ~/Assignments (master)
$ |
```

8. Make minor changes into README.txt files on line 1 and commit those changes into master:

```
cc@cc-PC MINGW32 ~/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

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

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

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git add README.txt

cc@cc-PC MINGW32 ~/Assignments (master)
$ git commit -m "Modified README js"
[master 6137c2c] Modified README js
1 file changed, 4 insertions(+), 3 deletions(-)
```

9. Again switch to js-assignments branch:

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git checkout js-assignments
Switched to branch 'js-assignments'

cc@cc-PC MINGW32 ~/Assignments (js-assignments)
$ |
```

10. Make minor changes into few files belonging to js-assignments branch”:

```
cc@cc-PC MINGW32 ~/Assignments (js-assignments)
$ git status
On branch js-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:   page1_js.js

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

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

11. Commit those changes:

```
cc@cc-PC MINGW32 ~/Assignments (js-assignments)
$ git add *.js

cc@cc-PC MINGW32 ~/Assignments (js-assignments)
$ git commit -m "Final JS assignment"
[js-assignments 3b0035c] Final JS assignment
1 file changed, 1 insertion(+), 1 deletion(-)
```

12. Switch to master:

```
cc@cc-PC MINGW32 ~/Assignments (js-assignments)
$ git checkout master
Switched to branch 'master'

cc@cc-PC MINGW32 ~/Assignments (master)
$ |
```

13. Merge js-assignments branch into master. Confirm all js assignments are shown in master:

```
cc@cc-PC MINGW32 ~/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.
```

```
cc@cc-PC MINGW32 ~/Assignments (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)
```

Changes to be committed:

```
new file:   page1_js.js
new file:   simple_js.js
```

Unmerged paths:

```
(use "git add <file>..." to mark resolution)
  both modified:   README.txt
```

Untracked files:

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

```
cc@cc-PC MINGW32 ~/Assignments (master|MERGING)
$ vi README.txt
```

```
cc@cc-PC MINGW32 ~/Assignments (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)
```

Changes to be committed:

```
new file:   page1_js.js
new file:   simple_js.js
```

Unmerged paths:

```
(use "git add <file>..." to mark resolution)
  both modified:   README.txt
```

Untracked files:

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

```
cc@cc-PC MINGW32 ~/Assignments (master|MERGING)
$ git add README.txt
```

```
cc@cc-PC MINGW32 ~/Assignments (master|MERGING)
$ git commit -m "Final README file"
[master 526b1df] Final README file
```



```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git branch --merged
  js-assignments
* master
```

14. Finally delete js-assignments:

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git branch -d js-assignments
Deleted branch js-assignments (was 3b0035c).
```

Section 4: Pushing assignments to remote repository:

1. Create a github account if you do not have already:
2. Login into that account:
3. Create new public repository "Freshersbranc-july20":
4. Commit and push any sample file to this repository:










```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git config --global user.name "Omeshwari"

cc@cc-PC MINGW32 ~/Assignments (master)
$ git config --global user.email "omeshwari.attarde@gmail.com"
> "

cc@cc-PC MINGW32 ~/Assignments (master)
$ git remote add origin https://github.com/Omeshwari/Freshersbranc-july20.git
```

```
cc@cc-PC MINGW32 ~/Assignments (master)
$ git push -u origin master

Enumerating objects: 53, done.
Counting objects: 100% (53/53), done.
Delta compression using up to 4 threads
Compressing objects: 100% (52/52), done.
Writing objects: 100% (53/53), 8.32 KiB | 266.00 KiB/s, done.
Total 53 (delta 21), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (21/21), done.
To https://github.com/Omeshwari/Freshersbranc-july20.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

 Omeshwari Final README file 526b1df 8 minutes ago  17 commits		
	README.txt	Final README file 8 minutes ago
	page1.html	Final html update 1 hour ago
	page1_css.css	Complete CSS assignment 34 minutes ago
	page1_js.js	Final JS assignment 11 minutes ago
	simple.html	All html files 1 hour ago
	simple_css.css	Added css files 42 minutes ago
	simple_js.js	Js assignment 18 minutes ago

Section 5: Pushing source code to remote repository using eclipse GIT plug-in:

1. One developer from the project team will create a eclipse project sample-project and add sample 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 te 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 and push the sample-project project: