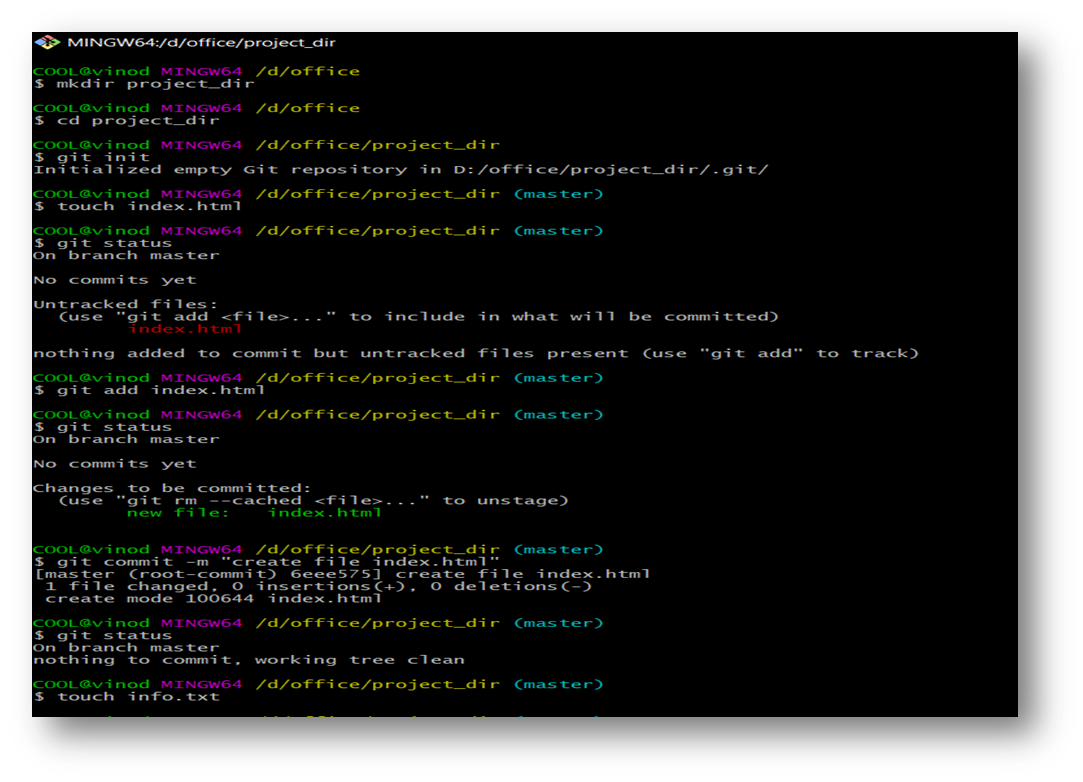
#### Section 0 -Use GIT as local VCS. Steps to follow:

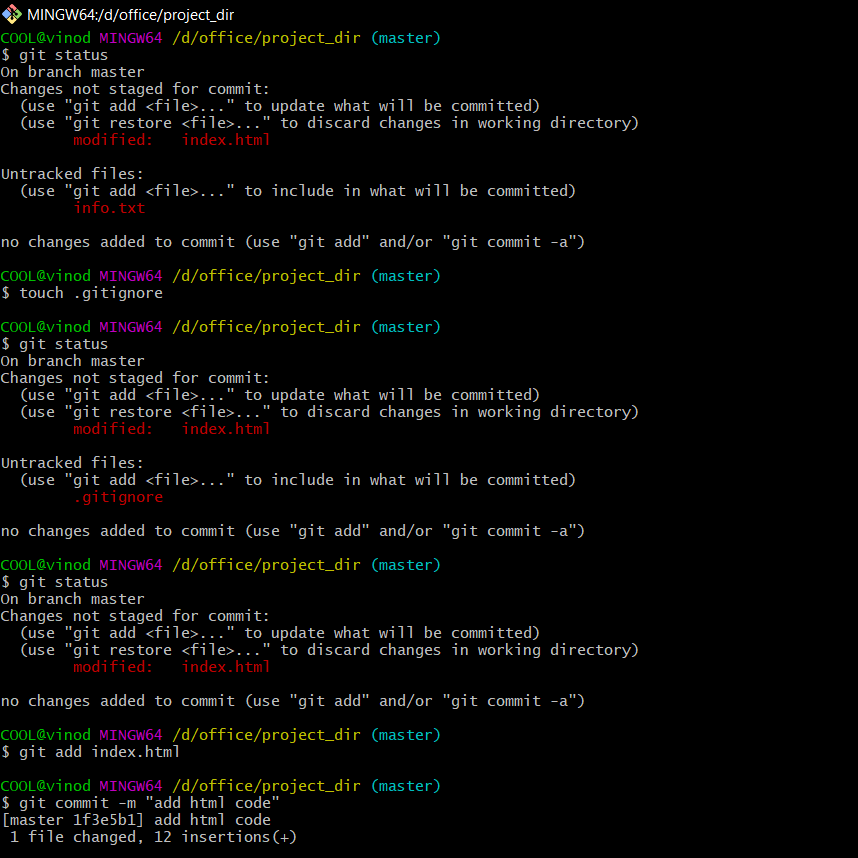
1. Create a directory ‘project\_dir’ & cd to ‘project\_dir’.
2. Initialize git version database. (git init)
3. Create a new file index.html.
4. Check the git status. You should find index.html as untracked file.
5. Stage the index.html file.
6. Commit index.html
7. Make few changes in index.html & create a new file info.txt file.

1-7



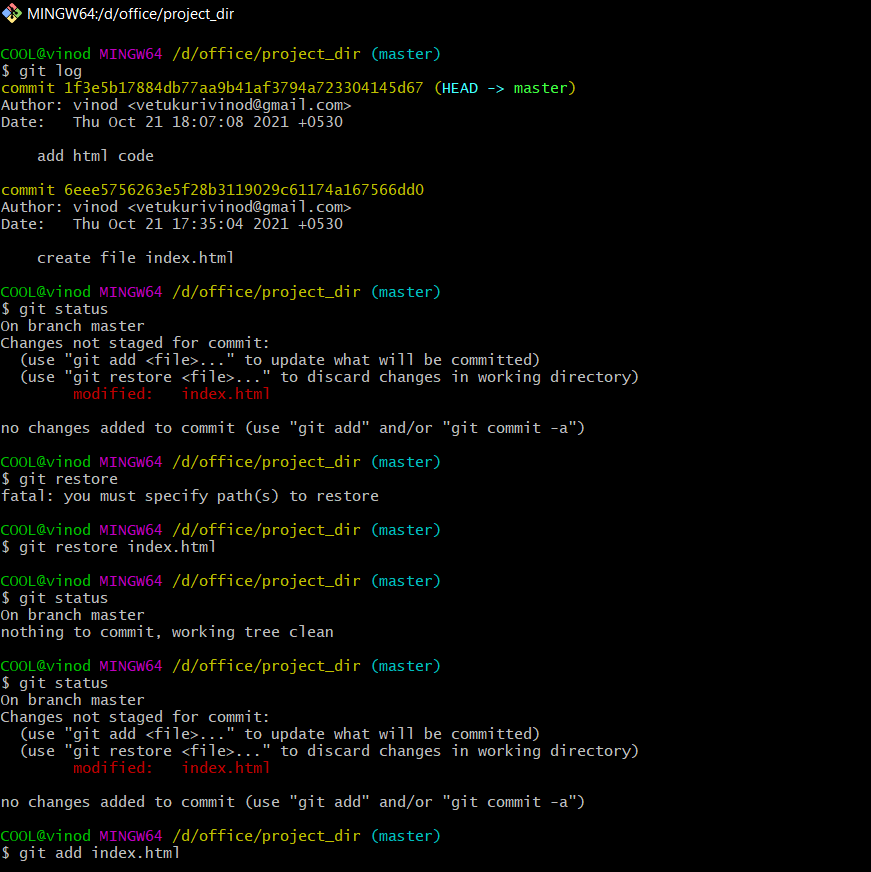
1. Check git status. You should find index.html & info.txt as untracked files.
2. Configure GIT to ignore all txt files.
3. Again check the git status. You should find only index.html as untracked file.
4. State & commit index.html

8-11



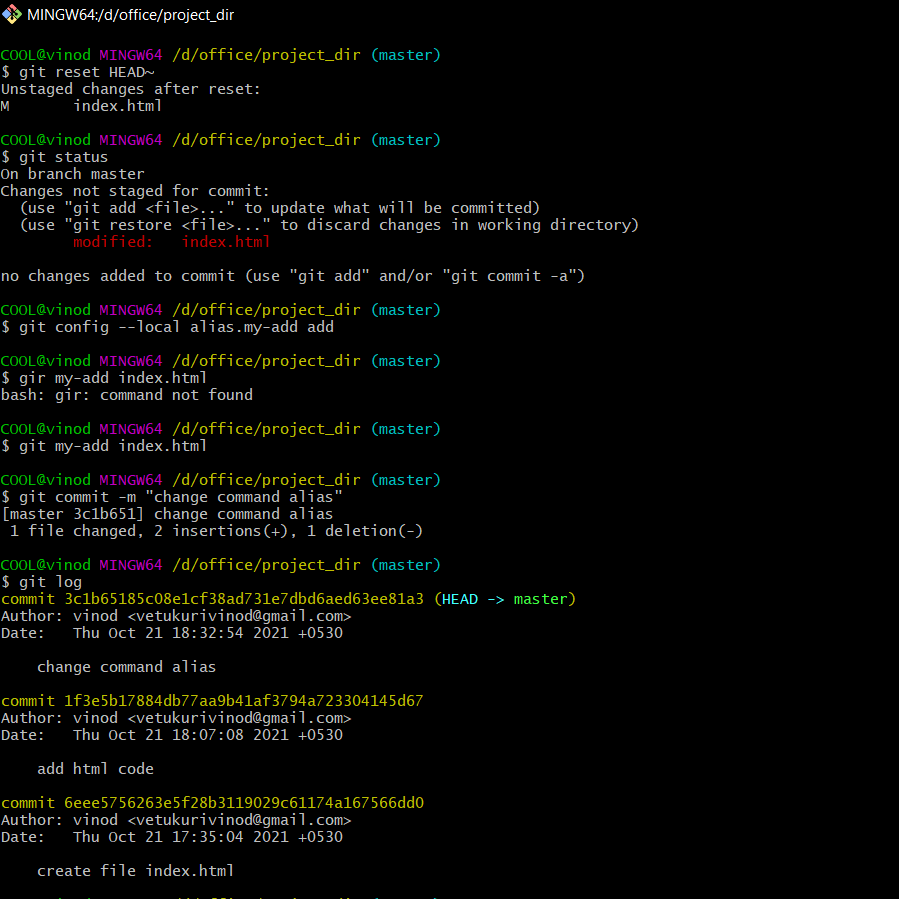
1. Log all your comments so far.
2. Make some changes in index.html.
3. Revert the change made in the previous step using git command.
4. Again change index.html.
5. Stage index.html

12-16

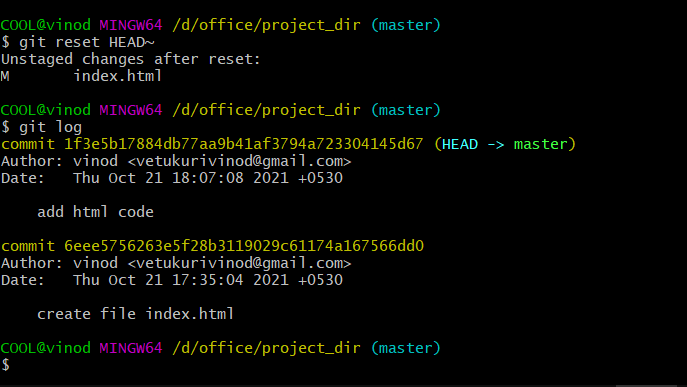


1. Revert back the last stage.
2. Rename ‘add’ command to ‘my-add’.
3. Using my\_add command Stage index.html again & commit the changes.

17-19



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’.
3. Create a file README.txt inside ‘Assignments’ & write few lines about the contents of ‘Assignments’ folder.
4. Commit README.txt file.
5. Now create a new branch ‘html-assignments’.
6. Switch to ‘html-assignments’ branch.
7. Copy all HTML assignments inside ‘Assignments’ folder.
8. Commit HTML assignments into ‘html-assignments’ branch.
9. Make minor changes into few files belonging to ‘html-assignments’ branch.
10. Commit those changed files.
11. Switch to master branch.
12. Make minor changes into README.txt file & commit those changes into master.
13. Again switch to ‘html-assignments’ branch.
14. Make minor changes into few files belonging to ‘html-assignments’ branch.
15. Commit those changes.
16. Switch to master.
17. Merge ‘html-assignments’ branch into master. Confirm all html assignments are shown in master.
18. Finally delete the ‘html-assignments’ branch.

HTML commands

COOL@vinod MINGW64 /d/office

$ mkdir Assignments

COOL@vinod MINGW64 /d/office

$ cd Assignments/

COOL@vinod MINGW64 /d/office/Assignments

$ git init

Initialized empty Git repository in D:/office/Assignments/.git/

COOL@vinod MINGW64 /d/office/Assignments (master)

$ touch README.txt

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git add .

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git commit -m "update readme file"

[master (root-commit) 1e7f2cc] update readme file

1 file changed, 1 insertion(+)

create mode 100644 README.txt

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git branch html-assignments

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git checkout html-assignments

Switched to branch 'html-assignments'

COOL@vinod MINGW64 /d/office/Assignments (html-assignments)

$ git add .

COOL@vinod MINGW64 /d/office/Assignments (html-assignments)

$ git commit -m "copy files"

[html-assignments d35c105] copy files

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

create mode 100644 test1.html

create mode 100644 test2.html

create mode 100644 test3.html

create mode 100644 test4.html

COOL@vinod MINGW64 /d/office/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: test1.html

modified: test2.html

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

COOL@vinod MINGW64 /d/office/Assignments (html-assignments)

$ git add .

COOL@vinod MINGW64 /d/office/Assignments (html-assignments)

$ git commit -m "update file"

[html-assignments 5b83202] update file

2 files changed, 2 insertions(+)

COOL@vinod MINGW64 /d/office/Assignments (html-assignments)

$ git checkout master

Switched to branch 'master'

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git s

send-email show-branch stash switch

shortlog sparse-checkout status

show stage submodule

COOL@vinod MINGW64 /d/office/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")

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git add .

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git commit -m "update readme file"

[master 67b38ff] update readme file

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

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git checkout html-assignments

Switched to branch 'html-assignments'

COOL@vinod MINGW64 /d/office/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: test3.html

modified: test4.html

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

COOL@vinod MINGW64 /d/office/Assignments (html-assignments)

$ git add .

COOL@vinod MINGW64 /d/office/Assignments (html-assignments)

$ git commit -m "update html file"

[html-assignments 80f3147] update html file

2 files changed, 2 insertions(+)

COOL@vinod MINGW64 /d/office/Assignments (html-assignments)

$ git checkout master

Switched to branch 'master'

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git merge html-assignments -m

error: switch `m' requires a value

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git merge html-assignments -m "merging to master"

Merge made by the 'recursive' strategy.

test1.html | 1 +

test2.html | 1 +

test3.html | 1 +

test4.html | 1 +

4 files changed, 4 insertions(+)

create mode 100644 test1.html

create mode 100644 test2.html

create mode 100644 test3.html

create mode 100644 test4.html

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git branch -D html-assignments

Deleted branch html-assignments (was 80f3147).

## 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.

CSS COMMANDS

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git branch css-assignments

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git checkout css-assignments

Switched to branch 'css-assignments'

COOL@vinod MINGW64 /d/office/Assignments (css-assignments)

$ git status

On branch css-assignments

Untracked files:

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

test1.css

test2.css

test3.css

test4.css

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

COOL@vinod MINGW64 /d/office/Assignments (css-assignments)

$ git add .

COOL@vinod MINGW64 /d/office/Assignments (css-assignments)

$ git commit -m "move files"

[css-assignments 5530e4b] move files

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

create mode 100644 test1.css

create mode 100644 test2.css

create mode 100644 test3.css

create mode 100644 test4.css

COOL@vinod MINGW64 /d/office/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")

COOL@vinod MINGW64 /d/office/Assignments (css-assignments)

$ git add .

COOL@vinod MINGW64 /d/office/Assignments (css-assignments)

$ git commit -m "update readme file"

[css-assignments 5e70445] update readme file

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

COOL@vinod MINGW64 /d/office/Assignments (css-assignments)

$ git checkout master

Switched to branch 'master'

COOL@vinod MINGW64 /d/office/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")

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git add .

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git commit -m "update readme file"

[master 9d85908] update readme file

1 file changed, 4 insertions(+), 1 deletion(-)

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git checkout css-assignments

Switched to branch 'css-assignments'

COOL@vinod MINGW64 /d/office/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: test1.css

modified: test2.css

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

COOL@vinod MINGW64 /d/office/Assignments (css-assignments)

$ git add .

COOL@vinod MINGW64 /d/office/Assignments (css-assignments)

$ git commit -m "update css files"

[css-assignments 8bdcb2d] update css files

2 files changed, 2 insertions(+)

COOL@vinod MINGW64 /d/office/Assignments (css-assignments)

$ git checkout master

Switched to branch 'master'

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git merge css-assignments -m

error: switch `m' requires a value

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git merge css-assignments -m "css merging to master"

Auto-merging README.txt

CONFLICT (content): Merge conflict in README.txt

Automatic merge failed; fix conflicts and then commit the result.

COOL@vinod MINGW64 /d/office/Assignments (master|MERGING)

$ get status

bash: get: command not found

COOL@vinod MINGW64 /d/office/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: test1.css

new file: test2.css

new file: test3.css

new file: test4.css

Unmerged paths:

(use "git add <file>..." to mark resolution)

both modified: README.txt

COOL@vinod MINGW64 /d/office/Assignments (master|MERGING)

$ git add .

COOL@vinod MINGW64 /d/office/Assignments (master|MERGING)

$ git commit -m "update readme file"

[master c7d75c7] update readme file

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git merge css-assignments -m "merging css files to master"

Already up to date.

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git branch -D css-assignments

Deleted branch css-assignments (was 8bdcb2d).

COOL@vinod MINGW64 /d/office/Assignments (master)

$ ^C

COOL@vinod MINGW64 /d/office/Assignments (master)

$

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.

JAVA COMMANDS

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git branch js-assignments

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git checkout js-assignments

Switched to branch 'js-assignments'

COOL@vinod MINGW64 /d/office/Assignments (js-assignments)

$ check status

bash: check: command not found

COOL@vinod MINGW64 /d/office/Assignments (js-assignments)

$ git status

On branch js-assignments

Untracked files:

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

test1.js

test2.js

test3.js

test4.js

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

COOL@vinod MINGW64 /d/office/Assignments (js-assignments)

$ git add .

COOL@vinod MINGW64 /d/office/Assignments (js-assignments)

$ git commit -m "update js files"

[js-assignments 420b893] update js files

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

create mode 100644 test1.js

create mode 100644 test2.js

create mode 100644 test3.js

create mode 100644 test4.js

COOL@vinod MINGW64 /d/office/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

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

COOL@vinod MINGW64 /d/office/Assignments (js-assignments)

$ git add .

COOL@vinod MINGW64 /d/office/Assignments (js-assignments)

$ git commit -m "update readme file"

[js-assignments 96cc71e] update readme file

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

COOL@vinod MINGW64 /d/office/Assignments (js-assignments)

$ git checkout master

Switched to branch 'master'

COOL@vinod MINGW64 /d/office/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")

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git add .

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git commit -m "update readme file"

[master ade273c] update readme file

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

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git checkout js-assignments

Switched to branch 'js-assignments'

COOL@vinod MINGW64 /d/office/Assignments (js-assignments)

$ git status

On branch js-assignments

nothing to commit, working tree clean

COOL@vinod MINGW64 /d/office/Assignments (js-assignments)

$ git status

On branch js-assignments

nothing to commit, working tree clean

COOL@vinod MINGW64 /d/office/Assignments (js-assignments)

$ git add .

COOL@vinod MINGW64 /d/office/Assignments (js-assignments)

$ git commit -m "update js files"

On branch js-assignments

nothing to commit, working tree clean

COOL@vinod MINGW64 /d/office/Assignments (js-assignments)

$ git checkout master

Switched to branch 'master'

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git merge js-assignments -m

error: switch `m' requires a value

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git merge js-assignments -m "merging files"

Auto-merging README.txt

CONFLICT (content): Merge conflict in README.txt

Automatic merge failed; fix conflicts and then commit the result.

COOL@vinod MINGW64 /d/office/Assignments (master|MERGING)

$ get status

bash: get: command not found

COOL@vinod MINGW64 /d/office/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: test1.js

new file: test2.js

new file: test3.js

new file: test4.js

Unmerged paths:

(use "git add <file>..." to mark resolution)

both modified: README.txt

COOL@vinod MINGW64 /d/office/Assignments (master|MERGING)

$ git add .

COOL@vinod MINGW64 /d/office/Assignments (master|MERGING)

$ git commit -m "update changes"

[master af26362] update changes

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git checkout js-assignments

Switched to branch 'js-assignments'

COOL@vinod MINGW64 /d/office/Assignments (js-assignments)

$ git status

On branch js-assignments

nothing to commit, working tree clean

COOL@vinod MINGW64 /d/office/Assignments (js-assignments)

$ git checkout master

Switched to branch 'master'

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git merge js-assignments -m "merging to master"

Already up to date.

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git satus

git: 'satus' is not a git command. See 'git --help'.

The most similar command is

status

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git status

On branch master

nothing to commit, working tree clean

COOL@vinod MINGW64 /d/office/Assignments (master)

$ git branch -D js-assignments

Deleted branch js-assignments (was 96cc71e).