Name-Susheel kumar

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’.

Ans-$ mkdir project\_dir

$ cd project\_dir

1. Initialize git version database.(git init)

Ans-

$ git init

Initialized empty Git repository in C:/Users/hp/project\_dir/.git/

1. Create a new file index.html.

Ans- C:/Users/hp/project\_dir/index.html – Index.html file is created

$ Touch index.html

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

Ans- $ 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)

1. Stage the index.html file.

Ans- $ git add .

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.html

1. Commit index.html

Ans- $ git commit -m "first commit"

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

Ans- $ touch info.txt

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

Ans - $ git status

On branch master

Untracked files:

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

README.md/

index.html

info.txt

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

1. Configure GIT to ignore all txt files.

Ans- $ touch .gitignore

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

Ans -$ 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

README.md/

index1.html

info.txt

1. State & commit index.html

Ans- $ git add index.html info.txt

$ git commit -m "restructure pages"

[master f80fb4b] restructure pages

2 files changed, 14 deletions(-)

create mode 100644 info.txt

1. Log all your comments so far.

Ans- $ git --all

unknown option: --all

usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]

[--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]

[-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]

[--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]

[--super-prefix=<path>] [--config-env=<name>=<envvar>]

<command> [<args>]

1. Make some changes in index.html.

$ Vi index.html

$ git commit -m "change index.html"

[master 1c75df2] change index.html

1 file changed, 14 insertions(+)

$ git commit

On branch master

Untracked files:

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

.gitignore

README.md/

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

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

$ git diff head

-S, --gpg-sign[=<key-id>]

GPG sign commit

1. Again change index.html.

$ vi index.html

1. Stage index.html

$ git add index.html

1. Revert back the last stage.

$ git log --oneline

$ git reset 65e6325

Unstaged changes after reset:

M index.html

1. Rename ‘add’ command to ‘my-add’.

$ git mv add my-add

$ git status

On branch lt

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.html

Untracked files:

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

my-add

1. Using my\_add command Stage index.html again & commit the changes.

$ git add .

$ git commit -m"change by adding"

[lt (root-commit) 050e450] change by adding

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

create mode 100644 index.html

create mode 100644 my-add

$ git commit

On branch lt

nothing to commit, working tree clean

1. Revert the last commit.

$ git log --oneline

050e450 (HEAD -> lt) change by adding

$ git reset 050e450

*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.

Ans- DONE

1. Create an empty directory ‘Assignments’ & cd to ‘Assignments’.

Ans-

$ mkdir Assignments

$ cd Assignments

$ git init

Initialized empty Git repository in C:/Users/hp/Assignments/.git/

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

$ touch README.txt

1. Commit README.txt file.

$ git add README.txt

$ git commit -m "add readme file"

[master (root-commit) 79f960e] add readme file

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

create mode 100644 README.txt

$ git commit

On branch master

nothing to commit, working tree clean

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

$ git branch html-assignments

1. Switch to ‘html-assignments’ branch.

$ git checkout html-assignments

Switched to branch 'html-assignments'

1. Copy all HTML assignments inside ‘Assignments’ folder.

$ git commit -m "copy the file"

On branch html-assignments

Untracked files:

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

everyday.html

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

1. Commit HTML assignments into ‘html-assignments’ branch.

$ git add .

$ git commit -m "change html files"

[html-assignments cd4f5be] change html files

1 file changed, 460 insertions(+)

create mode 100644 everyday.html

$ git commit

On branch html-assignments

nothing to commit, working tree clean

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

Ans- I had made some changes manually.

1. Commit those changed files.

$ git commit -m "I make changes"

On branch html-assignments

nothing to commit, working tree clean

$ git commit

On branch html-assignments

nothing to commit, working tree clean

1. Switch to master branch.

$ git checkout master

Switched to branch 'master'

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

$ git add .

$ git commit -m "change in readme"

On branch master

nothing to commit, working tree clean

$ git commit

On branch master

nothing to commit, working tree clean

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

$ git checkout html-assignments

Switched to branch 'html-assignments'

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

Ans- I had made some changes manually.

1. Commit those changes.

$ git commit -m "i changes"

On branch html-assignments

nothing to commit, working tree clean

$ git commit

On branch html-assignments

nothing to commit, working tree clean

1. Switch to master.

$ git checkout master

Switched to branch 'master'

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

$ git merge html-assignments

Updating 79f960e..cd4f5be

Fast-forward

everyday.html | 460

1 file changed, 460 insertions(+)

create mode 100644 everyday.html

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

$ git branch -d html-assignments

Deleted branch html-assignments (was cd4f5be).

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

1. Create a new branch ‘css-assignments’.

$ git branch css-assignments

$ git checkout css-assignments

Switched to branch 'css-assignments'

1. Copy all CSS assignments inside ‘Assignments’ folder.

$ git commit -m "copy the file"

On branch css-assignments

Untracked files:

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

docbook-xsl.css

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

1. Commit CSS assignments into ‘css-assignments’ branch.

$ git add .

$ git commit -m "change css files"

[css-assignments 85f4f5c] change css files

1 file changed, 296 insertions(+)

create mode 100644 docbook-xsl.css

$ git commit

On branch css-assignments

nothing to commit, working tree clean

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

Ans- I had made some changes manually.

1. Commit those changed files.

$ git add .

$ git commit -m "change readme files"

On branch css-assignments

nothing to commit, working tree clean

$ git commit

On branch css-assignments

nothing to commit, working tree clean

1. Switch to master branch.

$ git checkout master

Switched to branch 'master'

1. Make minor changes into README.txt file on line 3 & commit those changes into master.

Ans- I had made some changes manually.

1. Again switch to ‘css-assignments’ branch.

$ git checkout css-assignments

Switched to branch 'css-assignments'

1. Make minor changes into few files belonging to ‘css-assignments’ branch.

Ans- I had made some changes manually.

1. Commit those changes.

$ git add .

$ git commit -m "change in readme"

On branch css-assignments

nothing to commit, working tree clean

$ git commit

On branch css-assignments

1. Switch to master.

$ git checkout master

Switched to branch 'master'

1. Merge ‘css-assignments’ branch into master. Confirm all css assignments are shown in master.

$ git merge css-assignments

Updating cd4f5be..85f4f5c

Fast-forward

docbook-xsl.css | 296

1 file changed, 296 insertions(+)

create mode 100644 docbook-xsl.css

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

$ git branch -d css-assignments

Deleted branch css-assignments (was 85f4f5c).

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

1. Create a new branch ‘js-assignments’.

$ git branch js-assignments

1. Switch to ‘js-assignments’ branch.

$ git checkout js-assignments

Switched to branch 'js-assignments'

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

$ git commit -m "copy the file"

On branch js-assignments

Untracked files:

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

abcd.js

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

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

$ git add .

$ git commit -m "change js files"

[js-assignments f8b4805] change js files

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

create mode 100644 abcd.js

$ git commit

On branch js-assignments

nothing to commit, working tree clean

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

Ans- I had made some changes manually.

1. Commit those changed files.

$ git add .

$ git commit -m "change js files"

On branch js-assignments

nothing to commit, working tree clean

$ git commit

On branch js-assignments

nothing to commit, working tree clean

1. Switch to master branch.

$ git checkout master

Switched to branch 'master'

1. Make minor changes into README.txt file on line 1 & commit those changes into master.

Ans- I had made some changes manually.

1. Again switch to ‘js-assignments’ branch.

$ git checkout js-assignments

Switched to branch 'js-assignments'

1. Make minor changes into few files belonging to ‘js-assignments’ branch.

Ans- In js-assignment branch I had made some changes in the manually.

1. Commit those changes.

$ git add .

$ git commit -m "change js files"

On branch js-assignments

nothing to commit, working tree clean

$ git commit

On branch js-assignments

nothing to commit, working tree clean

1. Switch to master.

$ git checkout master

Switched to branch 'master'

1. Merge ‘js-assignments’ branch into master. Confirm all JavaScript assignments are shown in master.

$ git merge js-assignments

Updating 85f4f5c..f8b4805

Fast-forward

abcd.js | 0

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

create mode 100644 abcd.js

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

$ git branch -d js-assignments

Deleted branch js-assignments (was f8b4805).

*GIT Remoting*

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

SECTION-3 (Pushing assignments to remote repository) - Steps to follow:

1. Create a github account if you do not have already.

done

1. Login on into github account.

done

1. Create new public repository ‘freshersbatch-oct16’.
2. 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.