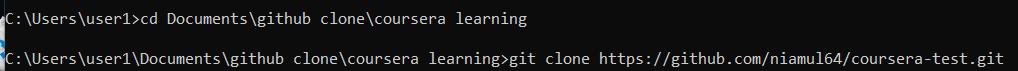
1.Clone a repository : git clone [treposetory link]

Repository link example: <https://github.com/niamul64/coursera-test.git>

Sample clone command:



Now initially we are in ‘on branch master’ if we want to change the branch then command:

 here gh-pages : another branch

2. make directory: mkdir [name of directory]

3. to go inside the directory : cd [name of directory]

4. go back to previous directory: cd ..

5. check which items are not committed : git status

6. to mark which items we need to commit (her ‘ . ’ means all items ): git add .

7. now finally to commit : git commit -m “message”

# ‘-m’ means to type message and in quotation we give a message

8. Now finally to upload the files at github : git push

9. now from browser sync : browser-sync start --server --directory --files "\*"

# “browser-sync start” start the browser sync

# “--server” in server mode

# “--directory” will give a directory list

# file “\*” here file says which file we need to sync here, \* means every thing

\*\*\*\*\* we have to install ‘node js’ and browser sync (inside commnd panel: npm install -g browser-sync). \*\*\*\*\*\*\*\*\*

10. synchronize a local repository wit a git repository. <https://www.coursera.org/learn/bootstrap-4/lecture/q4cvV/exercise-video-online-git-repositories>

### Set the local Git repository to set its remote origin

* At the prompt, type the following to set up your local repository to link to your online Git repository:

git remote add origin <repository URL>

without “<>”

2nd lesson

**More:**

1.

* Check to make sure that Git is installed and available on the command line, by typing the following at the command prompt:

git --version

2. optional:

* To configure your user name to be used by Git, type the following at the prompt:

git config --global user.name "Your Name"

* To configure your email to be used by Git, type the following at the prompt:

git config --global user.email <your email address>

* You can check your default Git global configuration, you can type the following at the prompt:

git config --list

3. <https://www.coursera.org/learn/bootstrap-4/supplement/cgWKo/exercise-instructions-basic-git-commands>

**Basic Git Commands**

* At a convenient location on your computer, create a folder named **git-test**.
* Open this git-test folder in your favorite editor.
* Add a file named *index.html* to this folder, and add the following HTML code to this file:

<!DOCTYPE html>

<html>

<head></head>

<body>

<h1>This is a Header</h1>

</body>

</html>

Initializing the folder as a Git repository

* Go to the git-test folder in your cmd window/terminal and type the following at the prompt to initialize the folder as a Git repository:

git init

Checking your Git repository status

* Type the following at the prompt to check your Git repository's status:

git status

Adding files to the staging area

* To add files to the staging area of your Git repository, type:

git add .

Commiting to the Git repository

* To commit the current staging area to your Git repository, type:

git commit -m "first commit"

Checking the log of Git commits

* To check the log of the commits to your Git repository, type

git log --oneline

* Now, modify the*index.html*file as follows:

<!DOCTYPE html>

<html>

<head></head>

<body>

<h1>This is a Header</h1>

<p>This is a paragraph</p>

</body>

</html>

* Add a sub-folder named **templates** to your **git-test** folder, and then add a file named *test.html* to the templates folder. Then set the contents of this file to be the same as the *index.html* file above.
* Then check the status and add all the files to the staging area.
* Then do the second commit to your repository
* Now, modify the*index.html*file as follows:

<!DOCTYPE html>

<html>

<head></head>

<body>

<h1>This is a Header</h1>

<p>This is a paragraph</p>

<p>This is a second paragraph</p>

</body>

</html>

* Now add the modified index.html file to the staging area and then do a third commit.

Checking out a file from an earlier commit

* To check out the index.html from the second commit, find the number of the second commit using the git log, and then type the following at the prompt:

git checkout <second commit's number> index .html

without ‘<>’ brackets

Resetting the Git repository

* To discard the effect of the previous operation and restore index.html to its state at the end of the third commit, type:

git reset HEAD index.html

* Then type the following at the prompt:

git checkout -- index.html

* You can also use *git reset* to reset the staging area to the last commit without disturbing the working directory.

### 4. Setting up an Online Git repository

* Sign up for an account either at Bitbucket ([https://bitbucket.org](https://bitbucket.org/)) or GitHub ([https://github.com](https://github.com/)).
* Then set up an online Git repository named **git-test**. Note the URL of your online Git repository. Note that private repositories on GitHub requires a paid account, and is not available for free accounts.

### Set the local Git repository to set its remote origin

* At the prompt, type the following to set up your local repository to link to your online Git repository:

git remote add origin <repository URL>

### Pushing your commits to the online repository

* At the prompt, type the following to push the commits to the online repository:

git push -u origin master

### Cloning an online repository

* To clone an online repository to your computer, type the following at the prompt:

git clone <repository URL>

### Conclusions

In this exercise you have learnt to set up an online Git repository, synchronize your local repository with the remote repository, and clone an online repository.