2017-10-06

GIT:

By default, the master branch will created

(master) this will act as the master branch, which will contain all the codes (these codes are stable codes).

They will be a following a future branch strategy whenever there is a release, they will create master for release and they do development of code and branch and then deploy.

1. To create a particular branch: git branch branchname

git branch class

Here, branchname is class.

Now the class is created. But we won’t be able to see the class branch.

1. So, to work in class branch or to see how many branches are there in the master branch that we are working.

git branch

press enter then we can see

git branch

class

\*master

It means. It saying that we are in master and have a branch called class.

1. If we need to go to that particular branch class

git checkout class

switched to branch ‘class’

Now we can see that in braces (class), If we observe in first line in braces (master).. now it changed to (class)

This is the way to create 1 branch

1. If we want to go back to master

git checkout master

now we can see in braces (master)

1. One more command using same checkout, we can create the branch and move to that particular branch.

git checkout –b tester1.0 [[[[[[[[[[here tester 1.0 is the branch name]]]]]]]]]]

now we can see (tester1.0) [[[[[ in braces. Using this command created tester1.0 and is moved to new branch]]]]]

1. To go to master branch again

git checkout master

Now we can see (master)

1. To check log history

git log

now we see a big commit id

1. If we don’t want to see big commit id,

git log –oneline

now it shortens commit id

1. If we have more lines of commit

And wants to see latest 5 commits or latest 1 commit

Git log –oneline -n 1 [[[[[[[[[here 1 is the number of commits that we want to see]]]]]]]]]]]]]]]]]

1. For example, As git works in structured way. Working directory and staging directory.

To put some file in here

First need to create file

Touch file2 [[[[file 2 is created ]]]]

1. git status

is used to see the status of working directory.

Now it will show as untracked file, because file2 is not belongs to that git. Since we created on our local desktop.

If we want to track this file2, we need to put in a staging area.

To do that

git add file2

now do

git status

now we can see that

new file: file2

this is the staging area.

1. If we don’t want to add this file to git

We can revoke or unstage it.

git reset HEAD file 2 [[[[file2 is the file name that need to be revoked] or removed ]]]]]

now do git status ,we can see that file 2 is in untracked.

1. Only above commands are used in real time mainly.
2. ls-a

we can see .git/ file.text file2

1. to see what is there in .git

cd.git/

then we see (git.dir)

then do

ll enter

then we can see all

1. whenever we do git init

that means we are creating a repository or

whenever we cloned

git clone URL that means we are creating a repository

whenever we create a git repository, then .git name will be created automatically and it contains all the work related . all the commands that we did up to now, all those will be done here.

Whenever we commit or whenever we add or whenever we stage , checklogs, create branches everything will be in .git folder.

If .git folder is there then only we can say that it is a repository.

If .git folder is not there then it doesn’t belong to GIT.

1. Cd../

It shows again the first line (master)

1. We can create our own username and email id by using the command

git config --global user.name “tester”

for each and every user we need to have user id to setup accounts in git.

Global represents that across all the repositories under our github, we are asking the access for this user “tester”.

“Tester “is the user name

1. To give access only to a particular repository, just remove 2 hyphons

git config global user.name “tester”

1. To give email-id

git config –global user.email [devopsuniverse@gmail.com](mailto:devopsuniverse@gmail.com)

Now, username “tester” is created and email is created.

1. To push changes to our origin,

git push origin

User name for <http://github.com>’: devopsclass1212

Now we will get as everything up-to-date

1. It goes again to (master)
2. If we want to push that particular branch,

git push origin class

User name for <http://github.com>’: devopsclass1212

Now , we will see

To <http://github.com/devopsclass1212/devopsclass.git>

This is the Repository URL