Date
14/05/2017

DELTA Pg No.
Date / /

file can be seen/tracked in future.

LF will be replaced by CRLF in git - (warning) - What is that and is it important?

Answer: In Unix system the end of a line is represented with a line feed (LF). In windows a line is represented with a carriage return (CR) and a line feed (LF) thus (CRLF). When you get code from git that was uploaded from a unix system they will only have an LF.

If you want to turn this warning off, type this in the git command line:

git config --global core.autocrlf true

If you are on a Linux or Mac system that uses LF line endings, then:

git config --global core.autocrlf input

If you are a windows programmer doing a windows-only project, then:

git config --global core.autocrlf false

Basic commands:

ls or dir ⇒ This is a generic command used to list all the files and folders

status ⇒ This command will tell you what branch you are on, what files have changed, or lists

Date 15/05/2015

DELTA Pg No. Date / /

all the new or modified files to be committed and also hints on what to do next.

add ⇒ It moves file from untracked state to staged state.
Example: git add xyz.php ↵
git add . ↵ (It moves all files available into untracked state to staged state)

commit ⇒ It moves files from staged state to tracked state.
Ex: git commit -m "initial commit" ↵

checkout -- filename ⇒ It discards changes of file i.e. throws the file to last committed state
Ex: git checkout -- xyz.php ↵
git checkout . ↵ (It discards changes of all files)

log ⇒ It lists history commits.
When log history is long you need to press shift+z twice to come out of list

log --oneline ⇒ It also lists history commits but the output will be shorten for you.

diff ⇒ It shows differences between two copies of same file; 1st which is in versioning system/tracked list and 2nd whose modifications are not committed yet.

diff --cached ⇒ In previous command, 2nd copy was in untracked state, but here, we use it when 2nd copy is in staged state.

Date
15/05/2017

DELTA Pg No.
Date / /

Ex: git diff --cached xyz.php ↵

reset ⇒ It removes file(s) from staged state and sends to untracked state but preserves file(s) changes.

Ex: git reset HEAD xyz.php ↵

git reset HEAD . ↵

rm ⇒ When you want to remove a file / permanent delete from git repository system, this command helps you.

or

It removes file from the working tree and from the index.

Ex: git rm index.php ↵ (It will remove/delete the file index.php from your directory/repository)

git commit -m "delete/remove index.php" ↵

↑ and it will update your tracked list.

Now the process is complete.

Date 24/05/201

Configure local git repository to connect remote git repository

DELTA Pg No. Date / /

or

Adding an existing project to GitHub using the command line:

Putting your existing work on GitHub can let you share and collaborate in lots of great ways.

1. Create a new repository on GitHub.
   Goto GitHub.com → Click on [+▽] sign in header section → New repository
   To avoid errors, do not initialize the new repository with README, license, or gitignore files. You can add these files after your project has been pushed to GitHub.
2. Open Git Bash
3. Change the current working directory to your local project root directory. For Ex: D:\xampp\htdocs\2017\magento2 (i.e. inside magento2 directory)
4. Initialize the local directory as a Git repository:
   git init ↵
   This command will create a directory '.git' in your project root directory, having git repository configuration files for the current repository/project.
5. Add the files in your new local repository. This stages them for the first commit.
   git add . ↵
   Add the files in the local repository and stages

Date -24/05/2017

DELTA Pg No. Date / /

them for commit. To unstage a file, use 'git reset HEAD YOUR-FILE'.

6. Commit the files that you have staged in your local repository.
   git commit -m "first commit" ↵
   Commits the tracked changes and prepares them to be pushed to a remote repository. To remove this commit & modify the file, use 'git reset --soft HEAD~1' and commit and add the file again.
7. At the top of your newly created GitHub repository's (i.e. if newly created remote repository name is 'magento2' then https://github.com/jitendrakyadav/magento2) Quick Setup page, click to copy the remote repository URL.
8. In the command prompt, add the URL for the remote repository where your local repository will be pushed:
   git remote add origin REMOTE-REPOSITORY-URL ↵
   // Sets the new remote
   For Ex: git remote add origin https://github.com/jitendrakyadav/magento2.git
9. Push the changes in your local repository to GitHub.
   git push -u origin master ↵
   It pushes the changes from your local repository to the remote repository you specified as the origin.
   This command verifies your GitHub username & password.