Git is the free and open source distributed version control system that’s responsible for everything GitHub related that happens locally on your computer.

SETUP

Configuring user information used across all local repositories

set a name that is identifiable for credit when review version history

*git config --global user.name “[firstname lastname]”*

set an email address that will be associated with each history marker

*git config --global user.email “[valid-email]”*

set default branch name

git config --global init.default branch main

get more help about a specific command

git help command – git help config

SETUP & INIT

Configuring user information, initializing and cloning repositories

initialize an existing directory as a Git repository

git init

STAGE & SNAPSHOT

Working with snapshots and the Git staging area

show modified files in working directory, staged for your next commit

git status

add a file as it looks now to your next commit (stage) – track a file

git add [file]

track all files in an entire directoty

git add .

git add --all

git add -a

untrack a file

git rm --cached [file]

commit your staged content as a new commit snapshot

git commit -m “[descriptive message]”

diff of what is changed but not staged

git diff

diff of what is staged but not yet committed

git diff –staged

unstage a file while retaining the changes in working directory

git reset [file]

unstage a file – back to staging, reverting the add action.

git restore –staged [file]

step over staging phase – moving from working area to commit without passing in staging

git commit -a -m “update file with new commands”

TRACKING PATH CHANGES

Versioning file removes and path changes

git rm [file]

delete the file from project and stage the removal for commit – only works wheb the file is in the commit area. Be careful because it remover the from the working area

Administration GIT

Get git version

git version

Update GIT

git update-git-for-windows

Conclusion

On Git there 3 environments:

Working files

Staging

Commit