|  |  |  |
| --- | --- | --- |
| GitHub Cheat Sheet Commands | | |
| Authenticate | | |
| Set global user name |  | *Git config –global user.name “John Doe”* |
| Set global user email |  | *Git config –global user.email johndoe@example.com* |
|  |  |  |
| Create | | |
| create new repository |  | *git init* |
| clone local repository |  | *git clone /path/to/repository* |
| clone remote repository |  | *git clone username@host:/path/to/repository* |
|  |  |  |
| add & remove | | |
| to add new file |  | *echo "insert your text here" >>filename.ext* |
| add changes to index |  | *git add <filename>* |
| add all changes to index |  | *git add \** |
| Removes files from index and working directory |  | *git rm <filename>* |
|  |  |  |
| commit & sync | | |
| commit changes  using *–a –m* is **=** to git add and git clone |  | *git commit –a –m “commit message”*  -a “all”  -m “message*”* |
| push changes to remote repository |  | *git push origin master* |
| Shaw all remote versions of repository |  | *Git remote* |
| connect local repository to remote repository |  | *git remote add origin <server>* |
| update local repository with remote changes |  | *git pull* |
| Shows status of files in index v working directory |  | *Git status* |
| Reset index and working directory to state of last commit |  | *Git reset*  *e.g. git reset –hard HEAD* |
| Temporarily save changes without committing |  | *Git stash* |
| Fetches all objects from remote repository |  | *Git fetch* |
| Show git object information |  | *Git show* |
|  |  |  |
| branches | | |
| create new branch |  | *git branch insertnamehere* |
| get a list of all branches |  | *git branch*  if “-a” is included, will provide remote branches as  well |
| switch to master branch  to switch to another branch🡺  *git checkout branchname* |  | *git checkout master* |
| delete branch |  | *git branch –d <branch>* |
| push branch to remote repository |  | *git push origin <branch>* |
|  |  |  |
| merge | | |
| merge changes from another branch into master |  | *git merge <branch>* |
| view changes between two branches |  | *git diff <source\_branch> <target\_branch>* |
|  |  |  |
| tagging | | |
| create tag |  | *git tag <tag> <commit id>* |
| get commit ids |  | *git log* |
|  |  |  |
| Misc | | |
| Shows a tree, including mode and name of each item & the SHA-1 value of the blob/tree it points too |  | *git ls-tree* |
| Used to view the type of object through SHA-1 value |  | *git cat-file* |
|  |  |  |
| Search through tree for words/phrases |  | *git prep* |
| Generate patch files or stats of differences between paths/files in repository |  | *git diff* |
| Graphical Tcl/Tk based interface to local git repository |  | *gitk* |
| Direct web browser into local repository |  | *git instaweb* |
| Creates a .tar or .zip including content of single tree |  | *git archive* |
| Repository “Disc cleanup” |  | *git gc* |
| Integrity check on git file system |  | *git fsck* |
| Removes objects that are no longer |  | *git prune* |