|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | |
|  |  | |
| Operational | | |
| 1. git add README   or  git add . (for adding all files that have changed) | Stages a file to be committed | |
| 1. git commit –m ‘about this’ | Commits file locally  Way to link and close and issue w/ a commit:  In –m ‘bla….fixes #xx’ | |
| 1. git push | Push the latest up | |
| git config core.autocrlf "false" | Tell git not to care about line endings. Fix for error “CRLF would be replaced by LF” | |
| git pull | Pull down some changes to your local repository from the web repository | |
| git rm file | To remove a file from repository | |
| git reset --hard origin/master | To roll back commits locally | |
| git reset . | To roll back and add locally while still preserving your local changes. | |
| git submodule update | Updates the submodule to the one defined in the super project  Note: to get new changes in a submodule:  *git checkout master*  *git pull origin master*  Then you will need to update the super project. | |
| git config --global core.autocrlf false | Stop git from trying to change your line endings. This is relevant when going from a LINUX to a Windows environment. Note the --global option  For local one windows first do *git init* to create a local config file and then run *git config core.autocrlf false* Or go manually edit this sucker: *C:\Users\geoderek* | |
| Setting Up | | |
| First you need a github account and the ssh set up | | This sets up your global link: email to github:  *git config --global* [*user.name*](http://user.name) *mwhutchi*  *git config --global user.email* [*mwhutchi@unca.edu*](mailto:mwhutchi@unca.edu)  But, you will also need to generate a SSH key for handshake: see [this](https://help.github.com/articles/generating-ssh-keys)!  How to switch your username: <http://alvinalexander.com/git/git-show-change-username-email-address> |
| *git config --list* | | List your current account info. See git config settings Note: --global will show you the global ones! Without this you are looking at local |
| *cd ~/.ssh*  *ls* | | Check for existing ssh keys. Note: On Linux this is going to be in your */home/user* directory |
| *ssh-keygen -t rsa -C* [*johnderekmorgan@hotmail.com*](mailto:johnderekmorgan@hotmail.com)  *exec ssh-agent bash*  *eval ssh-agent -s*  *ssh-add id\_rsa* | | Add your key! Note: you may need to *ls –a* to be able to see you hidden folders. Also, you may need to start to ssh agent with *sudo ssh-agent bash* [see](http://stackoverflow.com/questions/24154816/git-bash-could-not-open-a-connection-to-your-authentication-agent) |
| Account settings 🡪 ssh keys 🡪 copy to clipboard and paste in | | Add public ssh key to github.com. This will be the contents of the *id\_rsa.pub* file |
| *git init*  note: this is for a new project you are starting in git! | | This creates a new subdirectory named .git that contains all of your necessary repository files — a git repository skeleton. At this point, nothing in your project is tracked yet.    Create a .git directory too |
| Or…  …if just pulling down an existing git project to start work on it…  git clone [git@github.com:nemac/fswms](mailto:git@github.com:nemac/fswms) | | Note: you will have to run *chmod -R g-w html/* as apache is particular about this… |
| git remote add origin git@github.com:geoderek/ol-­­fcav.git | | To connect your local repository to your GitHub account, you will need to set a remote for your repo and push your commits to it |
|  | | Per <http://caiustheory.com/adding-a-remote-to-existing-git-repo> add the following to the bottom of .git/config  [branch "master"]  remote = origin  merge = refs/heads/master |
|  | | When setting up the first time you will need to merge the origin and the master before you can do your first push do that with *git pull origin master* see <https://help.github.com/articles/dealing-with-non-fast-forward-errors>  Also if you get an error about “git replacing LF with CRLF”  Run this: *git config core.autocrlf false* |
|  | | Commit: git commit -m 'initial commit'  Then do that first initial push git push origin master  If that doesn’t work try:  git push origin master --force |
| git add . | | Add all current directory files and below to be commited |
| git branch | | Tells you the branch that you are on |
| git remote –v | | Show the tracked repositories |
| git config --list | | Check ur current git config settings |
| git stash | | Temporarly store local changes so that you can have a virtual clean copy… use git stash pop will merge back in your stashed changes |
| *git checkout 271ef4acbd2f5cd0ee16dee2dcdef176c28e1dfb* | | Check out a specific version… note that this puts you on a branch… use *git checkout master* to get back to the master (pulling latest always).  If you have changes and do not mind losing them: *git reset --hard HEAD* |
| Adding a submodule | | *git submodule add git@github.com:nemac/seldon.git libs/seldon* (see this [blog](http://chrisjean.com/2009/04/20/git-submodules-adding-using-removing-and-updating/)) |
| Getting a submodule into an existing project | | *git submodule update --init –recursive*  And then if you want to make changes to the submodule project and have then tracked to that submodule’s own master you should do the following:  *cd libs/ol-fcav*  *git checkout master* |
|  | | Need to do this after cloning a project that has submodules… this will initialize the submodules….  *git submodule update --recursive --init* |
|  | | To update a submodule project  *ol-fcav/*  *git checkout master*  *git pull* |
| Renaming a github repository | | * Edit the name using the built-in feature under the "Settings" tab at the top of your GitHub repository page. * Remove the origin on your local repository   + git remote rm origin * Next, point to the new one:   + git remote add origin git@github.com:geoderek/seldon.git |
| **To remove a submodule** | | * Delete the relevant section from the .gitmodules file. * Delete the relevant section from .git/config. * Run rm -rf .git/modules/submodule\_name * Run git rm --cached path\_to\_submodule (no trailing slash). * Commit * Delete the now untracked submodule files   + rm -rf path\_to\_submodule |
| Pull upstream changes to submodule…. This would be when you have done some updates in the submodule, and then won’t them to be in the parent project. | | git commit libs/seldon -m "update submodule reference"  git push |
| When your submodule is complaining that it won’t update to what you know is up-today on remote | | *git fetch origin*  *git reset --hard origin/master* |
| ­Install git on Linux | | *apt-get install git* |
| Compile the OL library into single file | | *python build.py full OpenLayersBuild.js* |
| Rollback to specific local commit | | *git reset --hard 950a30db87ef6a7d25ea7090a10dd15494c42d3c* |
| Remove multiple files from git repository without having to list them out individually. | | git rm $(git ls-files --deleted) |
| Remove file from added… | | git rm --cached FILE |

References:

<https://help.github.com/articles/create-a-repo>