

2012-11-05 UF Research Computing Git Worksheet

"\$USER" = your username

```
##### BASIC CYCLE #####  
* Clone the test repository  
$ cd /scratch/hpc/$USER  
$ git clone /project/bio/training/2012-11-05/git/training.git  
$ cd training
```

Alternatively:

```
* Initialize a test working directory as a git repository  
$ cd /scratch/hpc/$USER  
$ cp -a /project/bio/training/2012-11-05/scripts training  
$ cd training  
$ git init  
$ git add *  
$ git commit -m "Initial import"
```

```
* Tell Git about yourself  
$ git config --global user.name "Graduate Student"  
$ git config --global user.email "coolgrad@ufl.edu"
```

```
* Edit a file(s) using a text editor - CMakeLists.txt
```

```
* Commit your changes  
$ git add *  
or  
$ git add CMakeLists.txt  
($ git status) <- see what's changed if you want  
$ git commit -m "Added a comment"
```

```
* View the revision history  
$ git log  
($ git log --oneline)
```

```
* View the revision history with changes  
$ git log -p
```

```
* View the log for a particular revision  
(852df3c87c8299bb867d0d9d251151dadf5046ea)  
$ git log -p 852df3c87 -1
```

```
* Branch the code for a different project  
$ git branch proj2
```

```
* See where you are  
$ git branch
```

```
* Switch branches  
$ git checkout proj2
```

```
* See where you are  
$ git branch
```

```
# Edit and commit

* View the log
$ git log

* Switch back to the master branch
$ git checkout master

* View the log
$ git log

* Merge changes from the proj2 branch
$ git merge proj2

* View the log
$ git log

* Grab a single file from a branch

# Checkout the proj2 branch, edit and commit the CMakeLists.txt file
# Switch back to the master branch
$ git checkout proj2 CMakeLists.txt

* See the status
$ git status

# Now the file can be committed in the master branch

##### END BASIC CYCLE #####

##### REMOTE REPOSITORY #####
* Add a remote repository
$ git remote add origin file:///project/bio/training/2012-11-05/git/scripts.git
* See the remote(s)
$ git remote -v

* Create an empty remote repository
$ mkdir /project/bio/training/2012-11-05/git/scripts.git
$ cd !$
$ git init --bare

* Push from the working tree to the remote repository
$ cd /project/bio/training/2012-11-05/scripts
$ git push origin master

* Clone the repository from a different system
$ git clone ssh://$USER@submit.hpc.ufl.edu/project/bio/training/2012-11-05/git/scripts.git training

* Use a service (gitosis or gitolite based controlled access server)

You can use the world-writable /project/bio/training/sandbox directory

##### END REMOTE REPOSITORY #####
```

```
##### CONFLICT MANAGEMENT #####
* Get changes
$ git pull

* Resolve the merge conflicts if any by editing the file and changing the <<<<
and >>>> sections
E.g.:
<<<<<<< HEAD
# Sample file for UF Research Computing Training on 2012-11-05
=====
# Sample file for UF HPC training on 2012-11-05
>>>>>>> 6602cc584eaeaa1169382406e8e4a1f190710ade

* Commit and push
* Pull each other's changes and have someone push to the repository in the end
##### END CONFLICT MANAGEMENT #####
```