CIS 480 - Extreme Programming
Guide to Git ←- written BY the CIS 480 XP project team!

MODIFIED for CIS 492!!! last updated: 2-23-12

FIRST: set up a Git repository on your nrs-projects TEAM account:

You should only have to do this once.

1. `cd` to a directory **`reponame`** (a name your team decides on for the Git repository) on your nrs-projects TEAM account that you would like to serve as your Git repository
2. Do the command:
   ```
   git init
   ```
3. I believe, before you can actually add anything, that you will have to set the following -- I think you'll have to decide on someone to be the "official" e-mail contact:
   ```
   git config --global user.email "you@example.com"
   git config --global user.name "Team X"
   ```

Setting up a local Git repository on a team member's INDIVIDUAL nrs-projects account (or Mac or Linux system -- maybe Windows, too?)

You should only have to do this once.

1. Log in to nrs-projects with your own account (your HSU username)
2. Get your own copy of your team's git repository:

```
git clone ssh://492teamX@nrs-projects.humboldt.edu/home/492teamX/reponame
```

3. Configure your contact details:
   ```
   git config --global user.name "George Washington"
   git config --global user.email "george.washington@humboldt.edu"
   ```

If you get stuck with a bunch of unintentional merge errors and want to reset your repo:

```
git fetch origin
git reset --hard origin/master
```

Note that you will lose EVERYTHING unsaved (or maybe even saved) in your repo! Keep a backup copy.

Changing files on the repository

This example makes a change to `text.txt`

1. Get the newest updates from the main repo (this changes your local copy, so do this before you make changes to any local files):
   ```
   git pull
   ```
2. Now you can modify test.txt:
   ```
   nano test.txt
   [make a change then ctrl-o to save]
   ```
3. Check the status of your git repo to see if the file is modified (optional):
   ```
   git status
   ```

4. Stage the modified file for a commit (only needed if the file’s not committed to the repo):
   ```
   git add test.txt
   ```
5. Now commit and give it a DESCRIPTIVE log message:
   ```
   git commit -m "extrapolating the metadata"
   ```
6. Push the changes to the main repository:
   ```
   git push origin master
   ```

Other stuff

View the commit log:

```
git log
```

View just the latest commit:

```
git log | head -5
```

Revert any changes you made to a local file (makes it the same as what’s on the main repo):

```
git checkout -- <file>
```

View the difference between your file and the one in the main repo:

```
git diff <file>
```

Resources

http://progit.org/book/ (free online book, very helpful)

read http://progit.org/book/ch1-3.html if you are completely confused about git

http://git-scm.com/documentation (official docs)

http://scottr.org/presentations/git-in-5-minutes/(Git in 5 minutes, a VERY basic overview)