## Git

* Version control system
* Used to track changes and enable collaboration to a set of computer files
  + Tracks history
    - Allows you to revert to a previous version easily
  + Collaborate
    - Synchronizes versions made by multiple people
    - Makes sure that updates are not in conflict with each other

## Creating a Profile (Must be done for each computer)

* git config –global user.name “*name*” – sets up user name
* git config –global user.email “*email*” – sets up email address
* git init – initializes your working folder
  + must be done for each new folder you work in

## Git Commands

* node *filename* – runs the file
* cd – Change directory (use double quotes for folder names with spaces)
* pwd – present working directory
* ls – list all files and folders in pwd
* . – current directory
* .. – parent directory (up one)
* ~ - alias for home directory
* mkdir – makes a new directory in pwd (requires argument to name)
* touch – creates a new file (requires argument to name)
* rm – removes file specified
* clear – clears terminal screen
* history – displays command history
* git –version – tells you git version number
* mv *oldname.ext newname.ext*
  + used to rename a file
* git clone (url argument + name if you want to rename (can also follow with a ‘.’ That says use the pwd) – clones a git repo
* git status – gives you a window to see what files and changes are saved and what isn’t saved yet
  + IMPORTANT – frequent use
* git add {} – specifies files to track
  + A ‘.’ means track all files in pwd
* git commit -m “*my version description*”
  + -m = message
* git branch -a
  + shows the current branch
* git branch *name*(no spaces allowed)
  + adds a branch
* git checkout (branch name)
  + switch to a new branch
* git push –set-upstream origin (branch name)
  + pushes the new branch to github
* git log – displays version history of files
* git checkout *long character string from log* – can revert to previous versions

## Using GitHub

* **remote** is the git hub repo
  + automatically set up as the repo cloned from
* git remote add origin [copy paste the url that you see here]

git push -u origin (master)

* + These commands connect the local folder to github
* git pull origin (master) – ‘pulls’ code from the online github repo into your working folder
* git push origin (master) – ‘pushes’ code from your working folder to the online repo
  + Always **pull before pushing**