Git - Branching & Merging

Git Configuration

When making a commit Git will assign an author name to the commit and you might have to specify your configuration setting to do so. To set up your name and email run:

Git config --global user.name “John Doe”
Git config --global user.email “johndoe@example.com"

Branching

Making a branch of the current repo allows you to make a copy of the project to work on separately from the master branch. The most important thing about using branches is knowing which branch you are always working on, as you now have multiple versions of the project.

Git branch	shows all the branches available
Git branch name	creates a new branch and gives it a name
Git checkout name	switches over to that branch
Git push -u origin name	pushes to the remote repository as a branch, different than the master

Merging

When merging a branch into the master you will first have to make sure that both branches have clean working trees, ie - no changes waiting to commit. Run a git status on both branches to check.

The branch that you are currently in will bring the other branch into when merging. Therefore, you will probably want to be in the master when merging.

Git checkout master	Moves to the master branch
Git merge name	Brings in the branch you want by it’s name to merge with the branch you are working in

Removing a File or Deleting a Branch

Git rm name	removes or deletes a file, this change will also have to be committed
Git commit -m “removed file”	
Git branch -d name	deletes the branch locally
Git push origin --delete name	deletes the branch remotely