Git References

Quick Git

```
$ git init
$ git add .g
$ git commit -m " commit comment"
$ git remote add origin <git@github.com:username/Repositoryname.git>
$ git push -u origin master
```

Getting a Git Repository

`git init` : Creates Git Repository skeleton. There must be a git Repository for any git commands to work

`git init <dirname>` : Creates git Repository in the new directory specified.

`git clone <url>` : Git receives a full copy of nearly all data that the server has. Every version of every file for the history of the project is pulled down by default.

`git status` : tells the user what the state of each file is within the Repository.

- `git status -s` : gives a far more simplified version of the status output.
  - ?? = untracked
  - M = modified
  - MM = modified, staged, modified again. So there are both staged and unstaged changes
  - A = added to staging area.

`git add <filename>` : will add the file to the list of tracked files. If you modify a file after you run git add, you have to run git add again to stage the latest version of the file.

- `git add .` : adds all modified/untracked files to staging area.

Creating a .gitignore file will allow git to ignore files which are listed within this file. Consider using `$ cat` to quickly add ignore files/file extensions.

```
$ cat .gitignore
* .txt
* .[oa]
* ~
```

Git Branching

`git branch <name>` : creates git branch in current Repository.

`git checkout <branchname>` : Moves to the branch name

`git branch` : without a specied name, git branch will list the current branches.