**Introducing yourself to git**

Terminal = gitbash

**git config –global user.name “**(add your name)**”**

**git config –global user.email “**(add your email)**”**

***The command line***

***pwd*** (print working directory)

directory = ***mkdir Test*** = folder named Test

***cd*** *=* change directory

***touch index.html*** = new file named index of type html

***ls*** = list all files and directories in this directory

**ls -a** = view hidden files

***vim “file with extension”*** = open or edit a file using vim

* ***“i”*** to insert text
* ***<!Doctype html>***use the latest html files
* ***“ESC” and “:w”***write a file (save a file)
* ***“:q”***quit a file
* ***“rm [file name]”***remove
* ***“cd ..”*** *step back one directory*
* *“****rm dir*** *[directory name]”*

***Creating a Git repository***

1. Create a directory
2. Move to it
3. **git init** (initialize a git repository)

***Adding files***

1. Create a file (touch index.html) in the repository.

Note: a file with extension css a way to style the web page (style images and editor text)

***Git status***

**git status**

Note: a *working directory* is a directory with a source files under git control, a *local repository* has nothing in it because that’s how we initialize, the comparison is with snapshots. (in the staging area)

**Tracking files**

Branch represent an independent line

**git add [file name]**

*Untrack files*

**git rm –cached [file name]**

*Add multiple files*

**git add .**

**Committing files**

**git commit -m “[message]”**

Note: you should commit every single section of your application (should be in present) and be very clear.

**Viewing history**

**git log** show all the commitments

**Introduction to branching**

A branch is a independent line of development.

**git branch** (see how many branches are in the repository)

**git branch [name]** (creates a branch)

**git checkout [branch name]** (move to this branch)

**Staging and tracking**

Track files are the ones added.

**Viewing file differences**

**git diff [file name]** checkout file differences

**git diff**  checkout file differences but it is messy if you have more than one

**Ignoring files**

**touch .gitignore**

**vim .gitignore (insert the file name, \*.[extension])**

**Commiting Our Work**

**git commit -m “[substantial name]”**

**git add .**

**git status**

**git commit -m “[substantial name]”**

**Merging**

Allows to take 2 independent branches and merge it into a single branch (go to the master branch)

**git merge [branch name that you want to merge in your current branch]**

**Adding a tag**

**git tag -a [tag name] -m “[message]”** (create a tag)

**git tag** (see all tags)