GIT COMMON COMMANDS

GIT BASICS	
command	**output**
cd <directoryname>	changes directory
cd ~	changes to home directory
cd ..	moves up one directory
ls	lists files in folder
pwd	shows current directory
mkdir <foldername>	creates new directory
touch <filename.txt>	crates new file
explorer .	opens specific file
mv <file> <directory>	moves specific file to the directory
mv <old_filename> <new_filename>	renames specific file
rm <file/directory name>	removes files and directories

CREATING A GIT REPOSITORY	
command	**output**
git init	creates a new local repository (on computer)
git clone <URL>	clones/copies an existing remote repo locally (to computer)
git status	determines a status of a repo

REVIEW A REPO'S HISTORY	
command	**output**
git log	displays information about the existing commits
git log --oneline	displays one line to show one commit per line
git log --graph	displays the bullets and lines
git log --all	displays all of the branches of the repository
git log --decorate	displays hidden details from default view
git log --stat <SHA>	displays changed files and the number of added/removed files ("stat"=statistics)
git log -p <SHA>	displays the patch of the code (all changes)
git log -w	ignores white space
git show <SHA>	displays information about the given commit
Using combinations:	git show --stat -p <SHA> **or** git log --oneline --graph --decorate --all

COMMITS	
command	**output**
git add <files_names>	add files from the working directory to the staging index
git add .	stage all files
git add -A	stage all files including deleted
git rm –cached	remove file from staging area
git commit -m "message"	move files from staging index and save them in the repo (commit files)
git diff	(displays the difference between two versions of a file)
	changes are made but aren't in a staging area (similar to git log -p)
.gitignore file	files listed in .gitingnore won't be committed (and not tracked) to repo
	This file should be placed in the same directory as the .git directory.
	(see globbing for more details i.e. samples/*.jpg)

TAGGING BRANCHING AND MERGING	
command	**output**
git tag -a <tag_name> <SHA>	adds a tag to a past commit
git branch	lists all branches
git branch <new_branch_name>	creates a new branch
git branch <new_branch_name> <SHA>	creates a branch on specific SHA (*see fast-forwarding)
git branch -d <branch_name>	deletes a branch
git checkout <branch_name>	switches to specific branch
git checkout -b <new_branch_name>	creates and switch to a new branch
git merge <name_of_branch_to_merge_in>	merges two branches /combines changes on different branches
	Important on which branch are you at the moment.

UNDOING CHANGES	
command	**output**
git commit --amend	alters the most recent commit (typo, forgot a file)
git revert <SHA>	reverses given commit
git reset	erases commit (see tags HEAD^~)

GIT PUSH & GIT PULL (network commands)	
command	**output**
git push origin <branch name>	send changes to the branch of your remote repository
git pull	updates local branch currently checkout to with
	any new commits on remote branch of the same repo
	(combination of git fetch and git merge)