Intro to Git Quick Reference

Key Points

What is Git/GitHub?

- Version control helps track changes to files and projects
- Git and GitHub are not the same

Getting started with Git

- Git repositories contain metadata about files under version control
- This metadata enables us to track changes to files over time
- Git uses a two-stage commit process. Changes to files must first be added to the staging area, then committed to the repository

Sharing your work

- remote repositories on GitHub help you collaborate and share your work
- `push` is a Git command to send changes from the local repository to a remote repository
- `pull` is a Git command to bring changes from a remote repository to the local repository
- `diff` is a Git command to compare an edited file and its most recent commit

Review

- the language of Git can be confusing and intimidating
- rephrasing commands and drawing concepts can clarify Git's workflow

GitHub Pages

- GitHub Pages offer an automated way to create a website that is version controlled and accessible for collaboration
- Collaborating on a GitHub Pages website uses the same Git/GitHub workflow you learned for collaborating via a GitHub repository

Intro to Git Quick Reference

Navigating the Shell

`pwd`	print working directory	`-lh`	list human readable file information
`ls`	list directory	`cd`	change directory
`-l`	list file information		

Interacting with Files

`mkdir`	make directory	`head`	output first parts of a file or files
`cat`	send file to output (in most cases, print to shell)	`tail`	output last parts of a file or files

`mv` rename or move a file or files. Syntax: `mv FILENAME NEWFILENAME`

`cp` copy a file or files. Syntax: `cp FILENAME NEWFILENAME`

`>` redirect output. Syntax: `cat FILENAME1 FILENAME2 > NEWFILENAME`

`rm` remove a file or files. NB: USE WITH CAUTION!!!

Git Commands

`git init` create a new local git repository

`git status` view the status of your files in the working directory and staging area

`git add` stage a file; tell git to start tracking a file, or a series of files

`git commit` save file changes from the staging area permanently to the project history

`git push` upload all commits to a remote repository, such as GitHub

`git log` show history of commits in reverse chronological order

`git diff` show changes made to tracked files

`git pull` download upstream changes and merge them into your local repository

`git remote add origin` add a remote repository named ‘origin’, to upload changes to or download changes from

Intro to Git Quick Reference

Git Commands

Configure Git

`git config –global user.name "[name]"` set name you want attached to your commits

`git config –global user.email "[email address]"` set email for your commits

`git config –global color.ui auto` enable code colouring in your command line window

Create Repositories

`git init [project-name]` create a new local repository

`git clone [url]` download a project and its entire version history

File Changes

`git status` list all new or modified files to be committed

`git diff` show file differences not yet staged

`git add [file]` snapshot the file in preparation for versioning

`git diff –staged` show differences between staging and the last file version

`git reset [file]` unstage the file but preserve its contents

`git commit –m "[message]"` record file snapshots permanently in version history

Group Changes

`git branch` list all local branches in the current repository

`git branch [branch-name]` create a new branch

`git checkout [branch-name]` switch to specified branch and update working directory

`git merge [branch-name]` combine specified branch's history into current branch

`git branch –d [branch-name]` deletes specified branch