Quick Command/Workflow Reference Guide

This is a reference guide for the *most common* commands/workflows you'll use with `git`.

For a more comprehensive command list see the Git_Cheat_Sheet.

Typical workflows

Contributing to an existing project on Github

1. Fork repository on Github using the *fork* button (aka make a *copy* of repository to your own github account)
2. Clone your fork to your computer, aka download from github to your computer (`git clone URL`)
3. Update files as needed in your favorite editor
4. "Stage" (prepare) updated and/or new files to be committed to github (`git add file1 file2`)
5. Commit changes to your local repository, aka take a "snapshot" of that folder (`git commit -m "message"`)
6. Push your local changes back to *your* fork on Github (`git push`)
7. Open a pull request so the owner of the *original* repository can decide to incorporate your changes (pull request button online)

Creating your own github project from scratch

1. Initialize a new repository from a local folder on your computer (`git init` within than folder)
2. "Stage" (prepare/add) files that `git` should be tracking (`git add file1 file2 file3`)
3. Commit those changes (aka take a "snapshot" of that local folder, `git commit -m "message"`)
4. Create a new repository on using the *new repository* button
5. Point your local `git` repository to the Github URL you just made (`git remote add origin URL`)
6. Push your local changes to this new repository (`git push -u origin master`)

Common commands

Creating a repository

Goal	Command	Example/Extra Info
Clone a repository from online	`git clone URL`	`git clone https://github.com/Summer-MIND/mind_2017`
Create a new repository from scratch locally	`git init`	run from *within* the folder you want to create into a repo

Adding, Committing, and Discarding files

Goal	Command	Example/Extra Info
See most recent local change	`git status`	Use this whenever you're in doubt!
Add a new file to git (i.e. start *tracking* a new file)	`git add filename1 filename2`	`git add myscript.py awesome_analysis.py`
Stage (prepare for commit) all tracked files with changes	`git add -u`	Make sure the file is already being tracked
Commit all staged (prepared) file changes	`git commit -m "message"`	`git commit -m "Added new permutation test feature"`
Discard most recent changes and revert to last commit; (changes are still available just "stashed away")	`git stash`	
Rollback to a specific snapshot (commit)	`git revert commitID`	`git revert a20122317890`

Getting and sending to Github

Goal	Command	Example/Extra Info
Pull the latest changes from a repository on Github	`git pull`	
Push the latest *local* changes to a repository on Github	`git push`	