Glossary of Git commands

Add: Add new/changed files to staging area, to be included in the next commit

```
git add .                      # All changed files
git add <file1> <file2>...     # Specific files
```

Commit: Permanently saves a snapshot of your repository.

```
git commit -m "Commit Message"  # Regular commit with commit message
git commit -am "commit Message" # Add and commit in one step
git commit --amend              # Add forgotten file to previous commit
```

Status: Status of git repository (modified or added files present?)

```
git status                 # Verbose information on working directory
git status -s              # Short info
git status -s -b           # Short info + info about current branch
```

Log: Shows a log of your repo's commit history.

```
git log -n 4               # show only last four commits
git log --oneline          # concise log in one line per commit
git log --graph            # log visualization
```

Diff: Compare commits or branches.

```
git diff <Commit>          # Compare current to specific commit
git diff HEAD              # ... e.g. to the last commit
git diff <Commit1> <Commit2> # Compare two commits
git diff HEAD~10 HEAD      # ... e.g. last commit to the 10th previous commit
```

Push: Push local commits to Github

```
git push -u origin <branch>       # First push: Link local to Github branch
git push                          # Push to the associated Github branch
git push origin --delete <branch> # Delete remote branch
```

Pull: Pulls commits on Github to local computer.

```
git pull           # Fetches and merges commits from Github into associated branch
git pull --rebase  # Adds first remote changes, and afterwards local changes (if present)
git pull --ff-only # Only carries out (linear) fast-forward merge
```

Branch: List, create or delete branches

```
git branch                    # List local branches
git branch -r                 # List remote branches
git branch -a                 # List all (local + remote) branches
git branch <newbranch>    # Create new branch
git branch -m <newname>   # Rename current branch
git branch -d                 # Delete local branch
```

Checkout: Switch to a branch or a commit

```
git checkout <branch>        # Switch to specified branch
git checkout <commit>        # Switch to specified commit
git checkout -b <NEWBRANCH>  # Create new branch and switch to it
```

Merge: Merges the specified branch into the currently checked out branch

```
git merge <branch>           # Merge the specified branch into the current branch
```

Reset: Resets the repo to the specified old commit, by rewriting hitory

```
git reset --hard  HEAD~1    # Completly discard last commit
git reset --mixed HEAD~1    # Discard last commit, but keep the changes in the files
git reset --soft  HEAD~1    # Discard last commit, changes in files are kept and staged
```

Revert: Reverts the changes in the specified commit, by creating a new commit.

```
git revert HEAD     # Completly discard last commit
```