

**Setup**

```
git config --global user.name "[username]"
```

Set the Username Git will use to authenticate

```
git config --global user.email "[email]"
```

Set the Email Git will use to authenticate

```
git config --global color.ui auto
```

Set automatic command line coloring for Git

```
git init
```

Initialize the current directory as a git repository

```
git clone [url]
```

Download a local copy of a repository to the machine

**Staging and Snapshots**

```
git status
```

Show modified files that are staged for commit

```
git add [file/directory]
```

Stage a file for the next commit

```
git reset [file]
```

unstage a file for commit while keeping the changes

```
git diff
```

Changes in files that are not staged for commit

```
git diff --staged
```

Changes in files that are staged for commit but not yet committed

```
git commit -m "[commit message]"
```

Commit your staged content as a new commit snapshot

```
git commit --amend
```

Add staged content to a commit snapshot that already exists

**Initializing a Repository**

```
git init
```

Initialize the current directory as a git repository

```
git clone [url]
```

Download a local copy of a repository to the machine

**Programs**

BS Computer Science  
MS Data Science

**Git & GitHub Commands**

Department of Computer Sciences

**File Path Changes**

```
git rm [file]
```

Delete the file from the project and stage the removal for commit

```
git mv [existing-path] [new-path]
```

Change an existing file path and stage the move

```
git log -stat -M
```

Show all commit logs with an indication of any paths that changed

**Branches and Merging**

```
git branch
```

List all branches. The active branch is marked with a '\*'

```
git branch [branch-name]
```

Create a new branch at the current commit

```
git branch -d [branch-name]
```

Delete the specified branch at the current commit

```
git branch -m [existing-branch] [new-branch]
```

Rename the specified branch at the current commit

```
git checkout [branch-name]
```

Switch branches and check it out into the working directory

```
git merge [branch]
```

Merge the selected branch's history into the current branch

**Inspect and Compare**

```
git log
```

Show all commits in the current branch's history

```
git log [branchB]..[branchA]
```

Show the commits on branchA that are not on branchB

```
git log --follow [file]
```

Show the commits that changed 'file', even accross renames

```
git diff [branchB]..[branchA]
```

Show the difference of what is in branchA but not in branchB

```
git show [SHA]
```

Show any object in Git in human-readable format

**Cheat Sheet**

Published ??th August, 2024.  
Updated 24th July, 2024.

**Git & GitHub Commands**

Department of Computer Sciences

**Share and Update**

```
git remote add [alias] [URL]
```

Add a Git URL as an alias

```
git fetch [alias]
```

Fetch down all the branches from that Git remote

```
git merge [alias]/[branch]
```

Merge remote branch into your current branch

```
git push [alias] [branch]
```

Transmit local branch commits to the remote repository branch

```
git pull
```

Fetch and merge any commits from the tracking remote branch

**Temporary Commits**

```
git stash
```

Save modified and staged changes in a stack

```
git stash list
```

List stack-order of stashed file changes

```
git stash pop
```

Write the working changes from the top of the stash stack

```
git stash drop
```

Discard the working changes from the top of the stash stack

**Version Control**

```
git rebase [branch]
```

Apply any commits of current branch ahead of the specified one

```
git reset --hard [commit]
```

Clear staging area, rewrite working tree from the specified commit

```
git checkout -- [file]
```

Discard the changes to 'file' in the working directory

```
git checkout [commit id]
```

Revert to a previous commit

```
git revert [commit id]
```

Undo the specified commit

**Contact**

Dr. James Quinlan  
Chair, Dept. of Computer Science