git Quick Reference

Setting Up Your Local Credentials

```
1. git config --global user.name "Your name"
2. git config --global user.email "Your email"
```

Creating a Local Repository

```
1. git clone <your repository URL>
2. cd <repository name>
```

OR

```
1. mkdir <repository name>
2. cd <repository name>
3. git init
4. git remote add origin <your repository URL>
5. git fetch
6. git checkout master
```

Staging Files

To stage files

```
1. git add <file> OR git add .
```

To unstage files

```
1. git reset HEAD <file>  OR git reset HEAD
```

Committing Files

To commit staged files

```
1. git commit –m “Commit message”
```

To commit unstaged, but tracked files

```
1. git commit -a –m “Commit message”
```

To edit the previous commit

```
1. git commit --amend
```

Modifying Files

To rename a file

```
1. git mv <file> <file>
```

To untrack a file

```
1. git rm --cached <file>
```

To untrack and delete a file

```
1. git rm <file>
```

Tagging Files

To create an annotated tag on the current commit

```
git tag -a <tag name> -m “Tag message”
```

To create a lightweight tag on the current commit

```
git tag <tag name>
```

To create a tag on a previous commit

```
git tag -a <tag name> -m “Tag message” <commit checksum>
```

To see all tags

```
git tag
```

To see the details of a particular tag

```
git show <tag name>
```

Examining Repository Information

To see the current status of the files in the current working directory

```
git status
```

To see the abbreviated version of the current status of the files in the current working directory

```
git status -s
```

To see information about each commit in the local repository

```
git log
```

To see the graphical depiction of the commit history in the local repository

```
git log --all --decorate --oneline --graph
```

Working with Branches

To create a new branch

```
git branch <branch name>
```

To see a list of current branches

```
git branch
```

To delete a branch

```
git branch -d <branch name>
```

To swap to a branch

```
git checkout <branch name>
```

To create a new branch and swap to it

```
git checkout -b <branch name>
```

To create a new branch from a previous commit and swap to it

```
git checkout -b <branch name> <commit checksum>
```

Merging

To merge two branches together (fast-forward merge)

```
1. git checkout <older branch name>
2. git merge <newer branch name>
```

To merge two branches together (three-way merge)

```
1. git checkout <older branch name>
2. git merge <newer branch name>
3. Manually resolve files containing conflicts
4. git add .
5. git commit -m “Commit message”
```

To abort a merge

```
1. git merge --abort
```

To examine the differences between two branches

```
1. git diff <branch name>..<other branch name>
```

To overwrite the current latest commit with a past commit

```
1. git checkout <commit checksum> OR git checkout <tag name>
2. git branch <older branch name>
3. git merge -s ours <current latest branch name>
4. git checkout <current latest branch name>
5. git merge <older branch name>
```

Working with the Remote Server

To download the latest version of the remote repository

```
1. git fetch
2. git merge <current branch> <remote branch>
```

OR

```
1. git pull
```

To download the latest version of a single branch from the remote repository

```
1. git fetch origin <branch name>
2. git merge <current branch> <remote branch>
```

OR

```
1. git pull origin <branch name>
```

To upload the latest version of a branch to the remote repository

```
1. git push origin <branch name>
```

To upload the latest version of all branches to the remote repository

```
1. git push --all origin
```