GIT CONCEPTS

WHY GIT?

Free

Open source

Fast

Small

Backups (implicitly)

Collaborate

VERSION CONTROL SYSTEM

Snapshots in time

Choose what changes to track

Add meaning with messages

Branch out

DISTRIBUTED VERSION CONTROL SYSTEM

Share copies of history

Merge work

No single point of failure

REPOSITORIES

Create

```
git init
```

Clone

```
git clone <username>@<host>:<repository path>
```

Status

```
git status
```

History

```
git log --graph --oneline
```

WORKING AREA

Working Directory

↓

```
git add <files>
```

↓

Index (staging area)

↓

```
git commit -m "<message>"
```

↓

HEAD (last commit of branch)

BRANCHES

Default branch is **master**

Create

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

Switch

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

Merge

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

Delete

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

Merge conflicts require resolving commit

REMOTES

Default server is **origin**

Add

```
git remote add <server name> <server path>
```

Push

```
git push <server name> <branch>
```

Pull

```
git pull <server name> <branch>
```

UNDO

Unstage

```
git reset HEAD <filename>
```

Revert file to state at HEAD

```
git checkout -- <filename>
```

Revert all to HEAD

```
git reset --hard
```

Changing history is possible, but not recommended...

USEFUL LINKS

Online tutorial: Try Git

In-depth tutorial: Git Immersion

Quick guide: git - the simple guide

Alternative look: The Git Parable

Repo hosting: Bitbucket