Git 101

This document briefly details how one can set up a Git repository and use it. This is intended for beginners to Git and is in no way complete nor comprehensive.

I have used CAPS refer to parts to parts that are unique to your usage i.e. project names and file names.

Initialising

First create a repository on github.com or another Git platform. Then either:

Add a directory to your repository:

> 1. Run `git init` from that directory
> 2. Add all files using `git add .` or add particular files using `git add FILE/DIRECTORY`
> 3. Check the status with `git status`, this will show the files in the repository in green
> 4. Commit with `git commit -am "initial version"`
> 5. Link to your repository with `git remote add origin https://github.com/USERNAME/PROJECT.git`
> 6. Push to the online repository with `git push -u origin master`

Download your git repository:

To clone the repo use: `git clone https://github.com/USERNAME/PROJECT.git`

Workflow

`git pull`	**Pull changes**
...	Do some work
...	Get project into working state
`git status`	See what's changed
`git add X`	track new files
...	edit .gitignore to ignore files
`git commit -am "..."`	Take a snapshot
`git push`	Push changes

Branching

Branches allow changes to be made to the existing codebase whilst keeping the default (`master`) branch unchanged. This is ideal for feature additions and code rewrites.

```
git checkout -b NEWBRANCHNAME
git commit -am "CHANGED X AND Y"
git push --set-upstream origin NEWBRANCHNAME
```

Zipping

To zip the current snapshot use:

```
git archive BRANCHNAME -o PROJECT.zip
```

Committing

The `-a` option says to commit all the changes and additions you have made.

The `-m` option says to include a one-line message describing the changes made since the last commit.