Git cheatsheet v1.0

Dean Bodenham

1 Local version control

Step 0: Turn folder into a local Git repo

In an empty or non-empty folder:

```
git init
```

Step 1: Stage files

After making some changes to the files:

```
git add .
```

Another option is: `git add --all`

Step 2: Commit files

Once your want to take a snapshot/create a version:

```
git commit -m 'comment goes here'
```

2 Version management

Check status

Check if up-to-date, or if there are staged changes to commit:

```
git status
```

Check the log

Have a look at the list of commits (times, authors):

```
git log
```

Check the log (short version)

When you just want the commit ID (for reverting):

```
git log --oneline
```

3 Checking out different versions

Checking out an earlier version

If you want to (temporarily) have a look at an earlier commit (with commit ID `x4y5z6`):

```
git checkout x4y5z6
```

or use the `-f` flag if you want to throw away uncommitted changes:

```
git checkout -f x4y5z6
```

This is **very safe**: any changes you make during a `checkout` **will not affect later commits!**

Return to most recent commit

There is a shortcut to return to the most recent commit:

```
git checkout master
```

4 Reverting to earlier version

Revert

Now you decide you want revert to an earlier version. to The foolproof way to revert (NOTE: need to commit afterwards), where the commit IDs can be obtain using `git log`. In this example:

- Latest commit ID: `a1b2c3`
- Desired (older) commit ID: `x4y5z6`

```
git revert --no-commit a1b2c3...x4y5z6
git commit -m 'message about reversion'
```

Aborting a revert

If something goes wrong during a revert, you can abort it using:

```
git revert --abort
```

5 Syncing between machines/using a remote repo

Link to remote repo

After creating a repo on Bitbucket/Github called `reponame`:

```
git remote add origin URL
```

where the url will be something like: `https://bitbucket.org/username/reponame`

push to remote repo

To add local commits to remote repo:

```
git push -u origin master
```

pull from remote repo

To obtain latest commits from remote repo:

```
git pull
```

clone remote repo

When starting a repo from scratch, `clone` will download whole repo (into a folder named `reponame`):

```
git clone URL
```

To delete a local copy of a Git repo

To delete a folder containing a git repo (from outside the folder):

```
rm -rf /path/to/reponame
```

6 Collaboration with others

Update/clone local repo

When starting to work on a repo with others, first `clone`:

```
git clone https://bitbucket.org/username/reponame
```

Or, if you have already cloned, `pull` to update to the latest version:

```
git pull
```

Before pushing a commit

After committing, but BEFORE pushing, you need to first `pull` any changes and then `push` as usual:

```
git pull --rebase
git push -u origin master
```

RESOLVING CONFLICTS

You might have **conflicts** when doing a `pull --rebase`. In such cases, you will see (for example):

```
CONFLICT (content): Merge conflict in file1.txt
```

KEEP CALM (and carry on); this is **easy to solve** by doing the following:

1. Open the listed files, and edit the parts which look like this:

   ```
   <<<<<< HEAD
   X
   ======
   Y
   >>>>>> YOUR-COMMIT-COMMENT
   ```

   `X` and `Y` are two versions of a particular block of text. Pick one version, perhaps edit it, and then remove the other version and the `<<<<<<`, `>>>>>>`, `======` lines.

2. Save the file and go back to the command line.

3. Then:

   ```
   git add .
   ```

4. Then:

   ```
   git rebase --continue
   ```

   You should see a line saying `Applying YOUR-COMMIT-COMMENT`.

5. Then:

   ```
   git push -u origin master
   ```

And that's it!