Example Use *Cheat Sheet*

Create a New Repository

```
git init
```

Grab a Local Copy of GitHub Repo

```
git clone repo-url                  (e.g. 'https://github.com/ergsense/DTECTS_hw.git')
```

Check Repo Status (check for changes)

```
git status
```

Commit new content

```
git add *                           ('*' for all new content, specific names otherwise)
git commit
```

Reset Your Repo (i.e. reset back to HEAD)

```
git reset –hard
```

Switch to a Previous Commit

```
git checkout sha-id
```

Core Components

Use the following ideas to properly achieve source control.

1. Your project is everything housed within the root directory. This includes:
   source code
   all settings (deployment & development, all!)
   description documents
   pictures
   ... quite literally everything that encompasses and contributes
   to the definition of your project/product!

2. Only one version of a project (or any file of the project!) is ever left present in the repository at a given time.
   If you want to retrieve a previous version though, it is simple. 'git checkout *sha-id*'!

3. Keep dev & experimentation local, on your PC. Only push to the server (GitHub) when ready, and complete!

4. Rebase or re-work your repository as softly and rarely as possible
   Only commit when complete. Branches otherwise!

Important Commands

Memorize these, to heart. This is 100% of what is needed to successfully implement & maintain source code control.

Creation

- "git init”
- “git status”

Generation

- “git add <*file*>”
- “git add *”
- “git rm <*file*>”

Commit

- “git commit”
- “git reset --hard” (reset to HEAD)
- “git reset --hard <*commit-id*>” (reset to a specific commit)

Review & Correct

- “gitk --all &” (see Figure 1 for example. My primary repo viewer, quick & easy)
- “git commit --amend” (update an existing commit)
- “git rebase -i HEAD~1” (where ‘1’ is how far back you’d like to rebase)

Useful Commands

These are used often, and they promote clean & organized repository development. Establish the habit early, and often!

Tagging & Tracking

- “git tag *tag_name*” (tag a commit with a tag, a name for later use & reference)
- “git checkout -b *branch_name*” (checkout a new branch)

That’s it!