Advanced Git Cheat Sheet

Command	Explanation & Link
git commit -a	Stages files automatically
git log -p	Produces patch text
git show	Shows various objects
git diff	Is similar to the Linux \`diff\` command, and can show the differences in various commits
git diff --staged	An alias to --cached, this will show all staged files compared to the named commit
git add -p	Allows a user to interactively review patches to add to the current commit
git mv	Similar to the Linux \`mv\` command, this moves a file
git rm	Similar to the Linux \`rm\` command, this deletes, or removes a file

There are many useful git cheatsheets online as well. Please take some time to research and study a few, such as this one.

.gitignore files

.gitignore files are used to tell the git tool to intentionally ignore some files in a given Git repository. For example, this can be useful for configuration files or metadata files that a user may not want to check into the master branch. Check out more at: https://git-scm.com/docs/gitignore.

A few common examples of file patterns to exclude can be found here.

Git Revert Cheat Sheet

git checkout is effectively used to switch branches.

git reset basically resets the repo, throwing away some changes. It’s somewhat difficult to understand, so reading the examples in the documentation may be a bit more useful.

There are some other useful articles online, which discuss more aggressive approaches to resetting the repo.

git commit --amend is used to make changes to commits after-the-fact, which can be useful for making notes about a given commit.

git revert makes a new commit which effectively rolls back a previous commit. It’s a bit like an undo command.

There are a few ways you can rollback commits in Git.

There are some interesting considerations about how git object data is stored, such as the usage of sha-1.

Feel free to read more here:

- https://en.wikipedia.org/wiki/SHA-1
- https://github.blog/2017-03-20-sha-1-collision-detection-on-github-com/

Git Branches and Merging Cheat Sheet

Command	Explanation & Link
git branch	Used to manage branches
git branch <name>	Creates the branch
git branch -d <name>	Deletes the branch
git branch -D <name>	Forcibly deletes the branch
git checkout <branch>	Switches to a branch.

Command	**Explanation & Link**
git checkout -b <branch>	Creates a new branch and switches to it.
git merge <branch>	Merge joins branches together.
git merge --abort	If there are merge conflicts (meaning files are incompatible), --abort can be used to abort the action.
git log --graph --oneline	This shows a summarized view of the commit history for a repo.