**Advanced Git Cheat Sheet**

| **Command** | **Explanation & Link** |
| --- | --- |
| git commit -a | [Stages files automatically](https://git-scm.com/docs/git-commit#Documentation/git-commit.txt---all) |
| git log -p | [Produces patch text](https://git-scm.com/docs/git-log#_generating_patch_text_with_p) |
| git show | [Shows various objects](https://git-scm.com/docs/git-show) |
| git diff | [Is similar to the Linux `diff` command, and can show the differences in various commits](https://git-scm.com/docs/git-diff) |
| git diff --staged | [An alias to --cached, this will show all staged files compared to the named commit](https://git-scm.com/docs/git-diff) |
| git add -p | [Allows a user to interactively review patches to add to the current commit](https://git-scm.com/docs/git-add) |
| git mv | [Similar to the Linux `mv` command, this moves a file](https://git-scm.com/docs/git-mv) |
| git rm | [Similar to the Linux `rm` command, this deletes, or removes a file](https://git-scm.com/docs/git-rm) |

There are many useful git cheatsheets online as well. Please take some time to research and study a few, such as [this one](https://github.github.com/training-kit/downloads/github-git-cheat-sheet.pdf).

**.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](https://gist.github.com/octocat/9257657).