## **Commands by type**

https://git-scm.com/docs

(Out of scope here are commands for Email, Administration, and Plumbing)

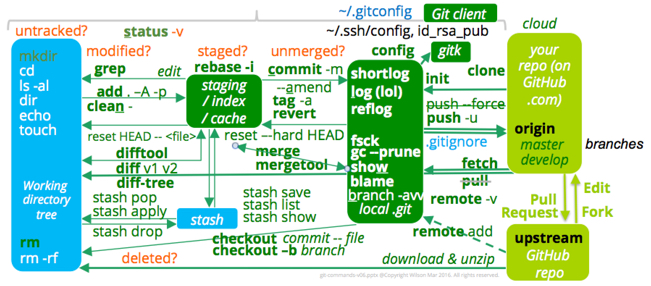
## **Common Git Commands in alphabetical order**

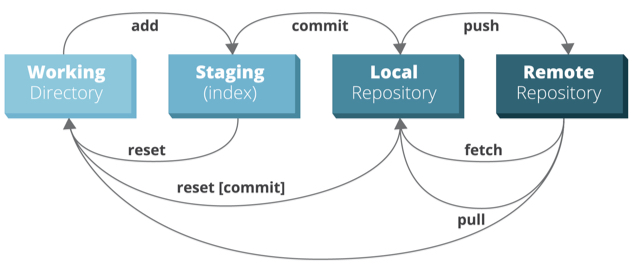
1. [**blame**](https://git-scm.com/docs/git-blame) - Show what revision and author last modified each line of a file.

Ex: **git blame <file name >, git blame -e <file name> ---- it show the email**

1. [**bisect**](https://git-scm.com/docs/git-bisect)- Show what revision and author last modified each line of a file
2. [**branch**](https://git-scm.com/docs/git-branch) - List, create, or delete branches
3. [**checkout**](https://git-scm.com/docs/git-checkout) - Switch branches or restore working tree files
4. [**cherry-pick**](https://git-scm.com/docs/git-cherry-pick) - Apply the changes introduced by some existing commits
5. [**clean**](https://git-scm.com/docs/git-clean) - Remove untracked files from the working tree
6. [**clone**](https://git-scm.com/docs/git-clone) - Clone a repository into a new directory
7. [**commit**](https://git-scm.com/docs/git-commit) - Record changes to the repository
8. [**config**](https://git-scm.com/docs/git-config) - Get and set repository or global options
9. [**describe**](https://git-scm.com/docs/git-describe)**-** Describe a commit using the most recent tag reachable from it
10. [**diff**](https://git-scm.com/docs/git-diff) - Show changes between commits, commit and working tree, etc
11. [**grep**](https://git-scm.com/docs/git-grep) - Print lines matching a pattern
12. **[help](https://git-scm.com/docs/git-help)**
13. [**init**](https://git-scm.com/docs/git-init) - Create an empty Git repository or reinitialize an existing one
14. [**ls-files**](https://git-scm.com/docs/git-ls-files) - Show information about files in the index and the working tree
15. [**log**](https://git-scm.com/docs/git-log) - Show commit logs
16. [**merge**](https://git-scm.com/docs/git-merge) - Join two or more development histories together
17. [**mergetool**](https://git-scm.com/docs/git-mergetool) - Run merge conflict resolution tools to resolve merge conflicts
18. [**pull**](https://git-scm.com/docs/git-pull) - Fetch from and integrate with another repository or a local branch
19. [**push**](https://git-scm.com/docs/git-push) - Update remote refs along with associated objects
20. [**rebase**](https://git-scm.com/docs/git-rebase) - Reapply commits on top of another base tip
21. [**reflog**](https://git-scm.com/docs/git-reflog) - Manage reflog information
22. [**reset**](https://git-scm.com/docs/git-reset) - Reset current HEAD to the specified state
23. [**revert**](https://git-scm.com/docs/git-revert) - Revert some existing commits
24. [**shortlog**](https://git-scm.com/docs/git-shortlog) - Summarize git log output
25. [**stash**](https://git-scm.com/docs/git-stash) - Stash the changes in a dirty working directory away
26. [**status**](https://git-scm.com/docs/git-status) - Show the working tree status
27. [**tag**](https://git-scm.com/docs/git-tag) - Create, list, delete or verify a tag object signed with GPG

**My map (Visual Cheat Sheet)**





The **Git Pull** command pulls down a complete copy of a repository’s change history and checks out the latest revision of files in the local Working Directory.

**Git Fetch** updates changes from the Remote, but does not update the Working Directory.

Locally made changes are added to Staging and **commit** updates change history tracking.

**Different parameters** of the reset command replaces changes in the Working Directory or **in Staging** with what is in Git’s repository.

→ A repo that belongs to another organization is called an **“upstream”** location.

→ If we **edit** that repo, GitHub automatically **forks** it under our own account.

→ If we want to make changes, we would need to file a **Pull Request** from the repo under our account.

→ The Git **init** command processed by the Git client creates the Git folder which holds the history of changes.

→ The Git **clone** command creates that .git folder inside a new folder from files downloaded from GitHub. \*\* The clone command also includes a Git **checkout** command that extracts the latest set of files into the **Working Directory** holding the .git folder.

→ If we run the SSH or Putty command to create keys, the Git client can communicate securely with the cloud service.

→ We run **config** commands to configure the **.gitconfig** file referenced by all Git repositories on our machine to provide our default name and email address to repos.