

Useful Git Commands

Local Repository

- `git config` - configure a user name, email address, editor, and other preferences once per machine.
- `git init` - initializes a repository.
- `git status` - shows the status of a repository.
- `git add` - puts files in the staging area.
- `git commit` - creates a snapshot of the staging area in the local repository.
- `git diff` - displays differences between revisions.
- `git checkout` - recovers old versions of files.
- `git log` - history of commits to this repository

Remote Repository

- `git push` - copies changes from a local repository to a remote repository.
- `git pull` - copies changes from a remote repository to a local repository.
- `git clone` - copies a remote repository to create a local repository with a remote called `origin` automatically set up.

What is `git diff` comparing?

- `git diff` Show differences between your working directory and the staging area.
- `git diff --staged` Show differences between the staging area and the most recent commit.
- `git diff HEAD` Show the differences between your working directory and the most recent commit.

Referring to different commits

Relative

- `HEAD` most recent commit
- `HEAD~1` "parent" of `HEAD`
- `HEAD~N` Nth "parent" of `HEAD`

Absolute

- `<commit ID>`, for example `df156f6766ced77b0da8a857fa2aa1deff65bf63`. You don't need to type the whole thing, Git will accept the first few characters.

Tips

- `git pull` before starting to work to minimize conflicts
- *divide and conquer* : split projects into multiple files to allow limit commits and minimize conflicts
- **Always** write a log message when committing changes.
- `.gitignore` file tells Git what files to ignore.

