

Git'in Started with Git

Erik O. D. Sevre

Git'in Started with Git

There are two ways to start a Git project. You can (1) check it out from a repository, or you can (2) create your own. I will look at both approaches. I will start by creating my own, then describe how to check it in to a one of the popular Git repos online.

Creating a local Git Repository

I will first go through how to do this from the command line. If you have never used the command line please see my "Getting Started with the Terminal" notes.

First I will create a directory called `mygitrepo` and on my system the directory location looks like:

```
/Users/esevre/Dropbox/Notes/git/mygitrepo
```

Currently this directory is empty so I will start by initializing this as a simple git repository.

Add your name and email to the global git config file on your machine.

```
git config --global user.name "Erik Sevre"
git config --global user.email "esevre@gmail.com"
```

Or if you want to check the current username and email address in the global config file you can use the following commands:

```
git config --global user.name
git config --global user.email
```

0.1 Init'in a Git Repository

To create a git repo in our directory described above we simply need to run the following command from the command line in the desired directory

```
git init
```

Now typing `ls -la` allows us to see that there is now a directory called `.git` hidden. This `.git` directory will keep track of all our git changes that we monitor along the way.

Once we have a file in our directory, there are two steps to starting the git repositor. First we need to add the file to the repo, this simply tells the repo to keep track of the file. Then we need to commit our file, this is basically saying "hey git, write down the file the way it appears now!" When we commit we do need to add a comment describing what we are doing with the commit.

So I'm going to add these notes to my git repo, so I copy them to the directory, add them to the git, then commit the file with the following commands:

```
cp ../*.tex .
git add *.tex
git commit -m 'initial setup with starting tex file'
```

To ignore files add a file named `.gitignore` to the top directory of the repo we are working in. Since I am working with `.tex` files I don't want to keep track of all the `.log`, `.aux`, and `.gz` files generated by my TeX editor, but I will add the final PDF to the repo.

```
git add .gitignore *.pdf
git status
```

`git status` will show us the status of all the changes to date so far

This document is created by Erik Sevre: E-mail: esevre@gmail.com