git *before* GitHub

D. Joe Anderson

mastodon.technology/@ritjoe
ritjoe (github)
dzho oftc,freenode,slashnet
deejoe@mail.rit.edu
ritjoe@etrumeus.com

Not “Us > Them”

Historical

Order

21:06 <+qsuscs> !gui
21:06 <@gitinfo> Graphical user interfaces are not supported here.

If you want to get support, it needs to be through the git CLI.
Reasons:

1) Because very few people here use the graphical interface.
2) Because giving instructions for GUI's is difficult.
3) The command line gives you a history of what commands you have executed.

- Terminal
- gnome-term
- PuTTY
- Git bash

- mkdir
- mv
- rm

“holy trinity”

- ls
- cd
- pwd

Editor:

- vi, emacs
- joe, nano
- sed
- >, >>

Folk versioning

myfile1 myfile2

myfile2015-05-09A

myfile.first myfile.latest

myfile.draft myfile.revised

SCCS, RCS, cvs, svn [(C)VCS]

hg, monotone, bzr, arch

BitKeeper [DVCS]


Fork me on GitHub


Fork me on GitHub

Not just GitHub

~~Gitorious~~ Gitlab Bitbucket
~~Gitbull~~ ~~Bullcodes~~ Launchpad
~~Sourceforge~~ ~~Google Code~~
GNU Savannah

gitolite ~~gitosis~~ gitweb, GitLab CE
gogs, gitea, pagure, kallithea

~~git~~:github::email:gmail

~~ves~~:github::email:gmail

diff:github::email:gmail

diff

patch

cat ~/foss/a/file.txt

one

two

three

$ cat ~/foss/file.txt.diff

--- a/file.txt 2017-02-01 00:05:04.986464930 -0500

+++ b/file.txt 2017-02-01 00:05:13.962304178 -0500

@@ -1,3 +1,3 @@

 one

 two

-three

+THREE

cat ~/foss/b/file.txt

one

two

THREE

commit objects *aka* **commits**

branch(es) [master; feature1]

repository/**repos**itories

remote(s) [origin, upstream]

- git init
- git add
- git commit

- git mv
- git rm
- git checkout
- git branch

- git status
- git log
- git diff

- git remote
- git push
- git fetch
- git merge

```
mkdir myfirstgitdir
cd myfirstgitdir
ls; ls -al
git init
ls; ls -al

git log
ls -lR .git/objects
echo red > file.txt  # this is just a quick & easy way to make a file
git status
git add file.txt
git status
git commit -m "my first git file"
git status
git log
ls -lR .git/objects


echo green >> file.txt  # quick & easy way to append a line to a file
git add file.txt
git commit -m "my first revision"
```

```
rm -rf myfirstgit{dir,clone}*

mkdir myfirstgitdir
cd myfirstgitdir
git init

echo red > file.txt
git add file.txt
git commit -a -m "my first git file"

echo green >> file.txt
git add file.txt
git commit -a -m "my first revision"

cd ..
git clone myfirstgitdir myfirstgitclone
cd myfirstgitdir
```

```
echo blue >> file.txt
git add file.txt
git commit -a -m "first half of my first conflict"

cd ../myfirstgitclone
echo BLUE >> file.txt
git add file.txt
git commit -a -m "second half of my first conflict"

git fetch
git merge origin master
echo here I just overwrite the file completely, usually you edit it to resolve the conflict
echo red > file.txt; echo green >> file.txt; echo blue >> file.txt
git commit -a -m "by committing this after the conflict, I assert that it is fixed"

git push origin master:fromclone
cd ../myfirstgitdir
git branch -v
git merge fromclone
git log --oneline --graph
cd ..; ls myfirstgit*
```

Git resources:

Cheat sheets
https://github.com/hbons/git-cheat-sheet
https://jan-krueger.net/git-cheat-sheet-extended-edition/

Git book:
https://git-scm.com/book/en/v2

Forking in github:
https://help.github.com/articles/fork-a-repo/
https://guides.github.com/activities/forking/

Pull requests in github:
https://help.github.com/articles/using-pull-requests/
https://help.github.com/articles/creating-a-pull-request/

Myriad other tutorials and examples online.