EMILY XIE

@emilyxxie

DISSECTING GIT'S GUTS:

Command Cheatsheets

DISSECTING GIT'S GUTS: EXERCISE #1

git init	initialize a git repository
git hash-object -w [filename]:	save given file to git database
git cat-file -p [SHA hash]	inspect git file. “p” stands for pretty, as in human readable
ls .git	see the contents of your .git directory

DISSECTING GIT'S GUTS: EXERCISE #2

git update-index --add [path to file]	add a file to the index, aka the staging area
git ls-files --stage	examine all files in your staging area
git write-tree	write a tree object using what's in the index file aka
find .git/objects -type f	list all of the objects in your git database

DISSECTING GIT'S GUTS: EXERCISE #3

@emilyxxie

echo 'your commit message here' | git commit-tree [tree hash]	write a commit msg, create a commit out of a tree object
find .git/objects -type f	list all of the objects in your git database
git cat-file -p [SHA hash]	inspect git file. "p" stands for pretty, as in human readable
git update-index --add [path to file]	add a file to the index, aka the staging area
git write-tree	write a tree object using what's in the index file aka staging
echo 'your commit message here' | git commit-tree [tree hash] -p [previous commit hash]	write a commit msg, create a commit with it out of a tree object, and link it to a previous commit object. the -p stands for "parent"
git log --stat [SHA HASH]	run a git log on a commit object

DISSECTING GIT'S GUTS: MISC COMMANDS

git cat-file -t [SHA hash]	inspect git file type. the “t” here stands for type