CSE306 Software Quality in Practice

Dr. Carl Alphonce
alphonce@buffalo.edu
343 Davis Hall
Intermediate git

- Concepts to cover in more depth
  - Branch
  - Fetch
  - Merge
Local Machine
  (e.g. your laptop, or timberlake if you’ve ssh’ed in)

Remote
  (e.g. bitbucket, github, CSE servers)

- stash
- workspace
- index staging
- local repository
- remote repository
What you see when working
Cloning a remote

Makes a copy of remote repo in local repo and checks out branch into workspace

git clone
Add a file to the staging area (add it to the index)

- stash
- workspace
- index
- staging
- local
- repository
- remote
- repository

```
git add
```
Create a new commit object with the staged items from the index

```
git commit
```
Pushing files from local repo to remote repo

- stash
- workspace
- index staging
- local repository
- remote repository

`git push`
"git pull is shorthand for git fetch followed by git merge FETCH_HEAD"
[https://git-scm.com/docs/git-pull]
Grab files from remote

git fetch
Create a commit combining the contents of two branches

- stash
- workspace
- index staging
- local repository
- remote repository

`git merge`
Recall the files in .git

-rw-r--r-- 1 alphonce staff 23 Apr 17 13:26 HEAD
-drwxr-xr-x 2 alphonce staff 68 Apr 17 13:26 branches
-rw-r--r-- 1 alphonce staff 328 Apr 17 13:26 config
-rw-r--r-- 1 alphonce staff 73 Apr 17 13:26 description
-drwxr-xr-x 12 alphonce staff 408 Apr 17 13:26 hooks
-rw-r--r-- 1 alphonce staff 137 Apr 17 13:26 index
-drwxr-xr-x 3 alphonce staff 102 Apr 17 13:26 info
-drwxr-xr-x 4 alphonce staff 136 Apr 17 13:26 logs
-drwxr-xr-x 7 alphonce staff 238 Apr 17 13:26 objects
-rw-r--r-- 1 alphonce staff 107 Apr 17 13:26 packed-refs
-drwxr-xr-x 5 alphonce staff 170 Apr 17 13:26 refs
pointer to the current branch
$ more head
ref: refs/heads/master
staging area

-rw-r--r-- 1 alphonce staff 23 Apr 17 13:26 HEAD
-drwxr-xr-x 2 alphonce staff 68 Apr 17 13:26 branches
-rw-r--r-- 1 alphonce staff 328 Apr 17 13:26 config
-rw-r--r-- 1 alphonce staff 73 Apr 17 13:26 description
-drwxr-xr-x 12 alphonce staff 408 Apr 17 13:26 hooks
-rw-r--r-- 1 alphonce staff 137 Apr 17 13:26 index
-drwxr-xr-x 3 alphonce staff 102 Apr 17 13:26 info
-drwxr-xr-x 4 alphonce staff 136 Apr 17 13:26 logs
-drwxr-xr-x 7 alphonce staff 238 Apr 17 13:26 objects
-rw-r--r-- 1 alphonce staff 107 Apr 17 13:26 packed-refs
-drwxr-xr-x 5 alphonce staff 170 Apr 17 13:26 refs
$ git ls-files
README.md
content
(blobs, trees, commits)
$ ls -l objects

```
ls -l objects

```

```
total 0

```

```
drwxr-xr-x  3 alphonce  staff  102 Apr 17 13:26 25
```

```
drwxr-xr-x  3 alphonce  staff  102 Apr 17 13:26 39
```

```
drwxr-xr-x  3 alphonce  staff  102 Apr 17 13:26 9c
```

```
drwxr-xr-x  2 alphonce  staff   68 Apr 17 13:26 info
```

```
drwxr-xr-x  2 alphonce  staff   68 Apr 17 13:26 pack
```

```
$ git cat-file -t 25b4
```

```
commit
```

```
$ git cat-file -t 9ce9
```

```
tree
```

```
$ git cat-file -t 39af
```

```
blob
```

```
drwxr-xr-x  4 alphonce  staff  136 Apr 17 13:26 log
```

```
drwxr-xr-x  7 alphonce  staff  238 Apr 17 13:26 objects
```

```
-rw-r--r--  1 alphonce  staff  107 Apr 17 13:26 packed-refs
```

```
drwxr-xr-x  5 alphonce  staff  170 Apr 17 13:26 refs
```
pointers to commits

-rw-r--r-- 1 alphonce staff 23 Apr 17 13:26 HEAD
drwxr-xr-x 2 alphonce staff 68 Apr 17 13:26 branches
-rw-r--r-- 1 alphonce staff 328 Apr 17 13:26 config
-rw-r--r-- 1 alphonce staff 73 Apr 17 13:26 description
drwxr-xr-x 12 alphonce staff 408 Apr 17 13:26 hooks
-rw-r--r-- 1 alphonce staff 137 Apr 17 13:26 index
drwxr-xr-x 3 alphonce staff 102 Apr 17 13:26 info
drwxr-xr-x 4 alphonce staff 136 Apr 17 13:26 logs
drwxr-xr-x 7 alphonce staff 238 Apr 17 13:26 objects
-rw-r--r-- 1 alphonce staff 107 Apr 17 13:26 packed-refs
drwxr-xr-x 5 alphonce staff 170 Apr 17 13:26 refs
$ ls -l refs
total 0
-rw-r--r-- 1 alphonce staff  107 Apr 17 13:26 packed-refs
drwxr-xr-x  5 alphonce staff  170 Apr 17 13:26 refs
$ ls refs/heads/
master
$ ls refs/remotes/
origin
$ ls refs/remotes/origin/
HEAD master

Local branches

Remote HEAD and branches
Possible states of a file

- edited
- unmodified
- modified
- untracked
- staged
commit preserves contents
(accidental removals can be recovered from)
understanding branching


```
  git branch <name> creates a new pointer
```
Let's take this as our starting point.
git checkout B
create fileB

git add fileB
git commit -m "..."
git checkout master
git checkout B
EXERCISE
Let us first switch to branch ‘a’:

git checkout a
Now edit and commit F2.

```
emacs F2 (call edited version F2')
git add F2'
git commit -m "..."
```
Create FS, add, and commit.

  emacs FS
  git add FS
  git commit -m "..."
Switch to master.

git checkout master
Merge a into master.

```
git merge a
```
Because there was a straight line path from master to a, this is a "FAST FORWARD merge: master is simply moved to point to the same commit as a."
Non-FF merge
(three way merge)

- A new commit is created.
- The branch being merged into is the first parent of the new commit.
- 'master' should always be the first parent, hence switch to master before doing merge.
The non fast-forward merge creates a "merge commit". The branch merged into points to that new commit.
Deleting a branch

```bash
  git branch -d a
```