Intro to git

Dr. Justin F. Brunelle
jbrunelle@cs.odu.edu
http://www.cs.odu.edu/~jbrunelle/cs518

So you want to git...

- Go to github
- Create a repo
- You get some good text...

jbrunelle / **ODUCS418F17**

Quick setup — if you've done this kind of thing before

or HTTPS SSH `https://github.com/jbrunelle/ODUCS418F17.git`

We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# ODUCS418F17" >> README.md
  git init
  git add README.md
  git commit -m "first commit"
  git remote add origin https://github.com/jbrunelle/ODUCS418F17.git
  git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/jbrunelle/ODUCS418F17.git
  git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

Sure you could drag and drop...

- But we're **real** computer scientists!
- Use the command line instructions
- It really is that easy!

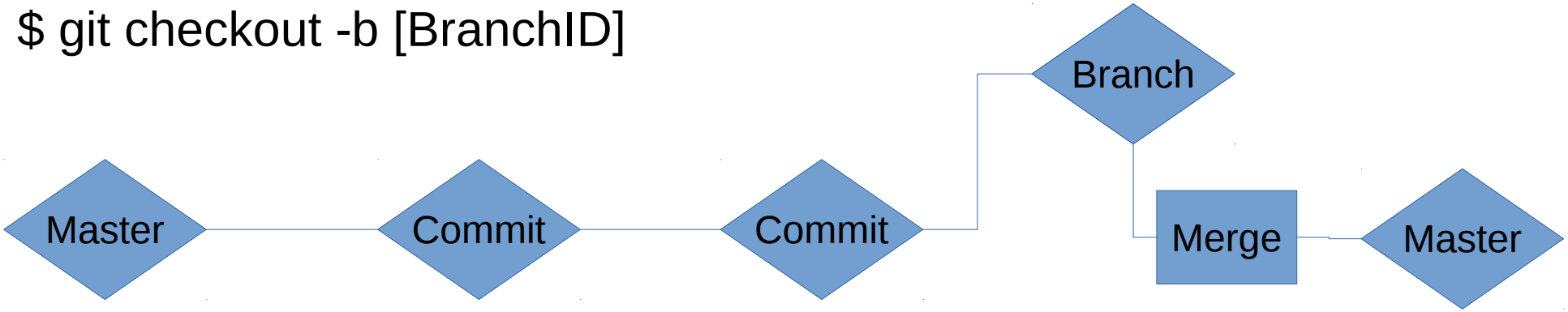
```
$ git init
Initialized empty Git repository in .../ODUCS418F17-master/.git/

$ git add *
$ git commit -m "initializing class repo"
[master (root-commit) 56ce01f] adding examples
 78 files changed, 3105 insertions(+)
 create mode 100644 DockerThings/000-default.conf
 create mode 100644 DockerThings/Dockerfile
 create mode 100644 DockerThings/README.md
 create mode 100644 DockerThings/apache_default
 create mode 100644 DockerThings/create_mysql_admin_user.sh
 create mode 100644 DockerThings/import_sql.sh
 create mode 100644 DockerThings/import_sql.she
 create mode 100644 DockerThings/index.txt
 create mode 100644 DockerThings/my.cnf


On branch master
Untracked files:
        .gitignore


$ git remote add origin https://github.com/jbrunelle/ODUCS418F17.git
$ git push -u origin master
Username for 'https://github.com': jbrunelle
Password for 'https://jbrunelle@github.com':
Counting objects: 86, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (77/77), done.
Writing objects: 100% (86/86), 70.63 KiB | 0 bytes/s, done.
Total 86 (delta 10), reused 0 (delta 0)
remote: Resolving deltas: 100% (10/10), done.
To https://github.com/jbrunelle/ODUCS418F17.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

Why you care

- You'll be managing a large codebase
- You'll probably screw it up at some point
  - Roll backs
  - Branches for assignments
- You may be working in a team
  - Merges
  - Branches
  - Concurrent development

Team Development: Branches

- Purpose: work on a new feature without impacting the stable code
- Command:
  $ git checkout -b [BranchID]



- Merge:
  $ git checkout master
  Switched to branch 'master'
  $ git merge [BranchID]

Team Development: Merges

- $ git merge [BranchID]
  Auto-merging index.php
  CONFLICT (content): Merge conflict in index.php
  Automatic merge failed; fix conflicts and then commit the result.
- $ git status
  …
  <<<<<<< HEAD:index.php
  echo “Justin Rocks”
  =======
  echo “Justin is terrible”
  >>>>>>> [BranchID]:index.php

Team Development: Merges

- Check out “git mergetool”
- https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging
- https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell
- What we haven't covered:
  - Branch switching
  - Multiple merges

Some resources I won't copy...

- http://rogerdudler.github.io/git-guide/
- https://try.github.io/levels/1/challenges/1
- https://git-scm.com/book/en/v2/Getting-Started-Git-Basics