



```
MINGW64/c/Users/Owner/code/pytorch/autoencoder
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        #mnist_gpu.py
        .#mnist_gpu.py
        data/

nothing added to commit but untracked files present (use "git add" to track)

Owner@Galatea MINGW64 ~/code/pytorch/mnist (master)
$ cd ../autoencoder/

Owner@Galatea MINGW64 ~/code/pytorch/autoencoder (master)
$ ls
autoencoder.py

Owner@Galatea MINGW64 ~/code/pytorch/autoencoder (master)
$ git status
On branch master
nothing to commit, working tree clean

Owner@Galatea MINGW64 ~/code/pytorch/autoencoder (master)
$
```

CTD Intro Week 17

git in more depth

GitHub

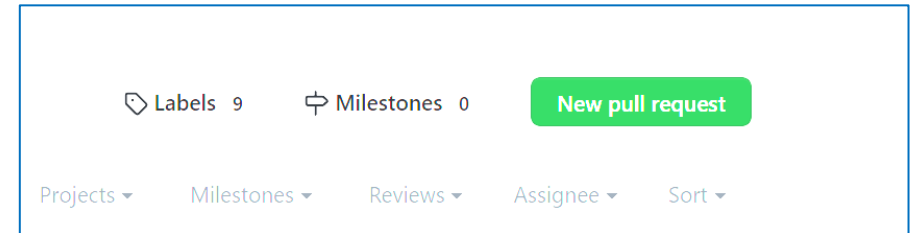
Git Command Review

- Create a new public repository in your github account (e.g. *yourName-classname*)
 - [New repository \(github.com\)](#) – green button on the upper left in your dashboard view
- Clone the github repo to your local machine
 - `git clone https://github.com/YourGithubHandle/your-new-repository.git` (your forked repo)
 - This is run on your local command line in the directory (folder) where you put CTD repositories
 - E.g. `~/code-the-dream`
- `git init` (set up a local repository – not needed if cloning)
- `git status` (which file are modified, etc.)
- `git diff` (what's changed)
- `git log` (all the commit log messages)
- `git branch` (what branches are there?, what's the current branch?)
- `git checkout` (change branches)
 - `git checkout -b branch-name` (create a new branch with current changes)
- `git add` (stage files for commit)
- `git commit -m commit log message` (opens editor if no `-m`)
- `git push` (pushes changes upstream e.g. to github)
- `git pull` (pull changes from upstream e.g. github)



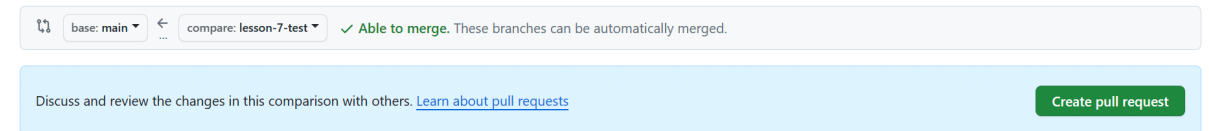
Pull Requests

- Standard workflow for making changes to a shared repository
- Allows your supervisor and peers to review and comment
- Flow:
 - Clone the repository
 - Make a new branch for your changes 'git checkout -b lesson-X'
 - Make and validate your edits
 - Push your branch to github 'git push'
 - may need `git push --set-upstream origin lesson-X` the first time
 - Create a pull request (PR) from your branch
 - Request reviews for your PR
 - You can push more commits to the pull request branch to address review feedback.
 - Merge your pull request when reviews are satisfied



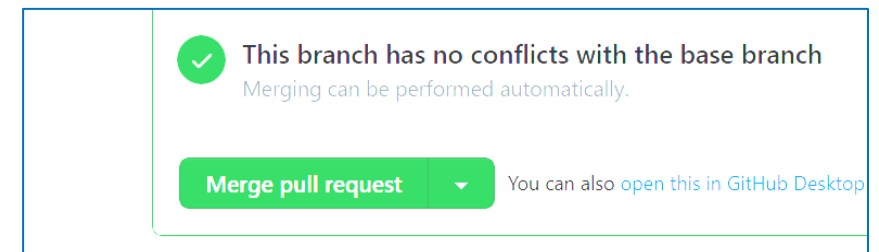
Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).



Added more content for a pull request. #2

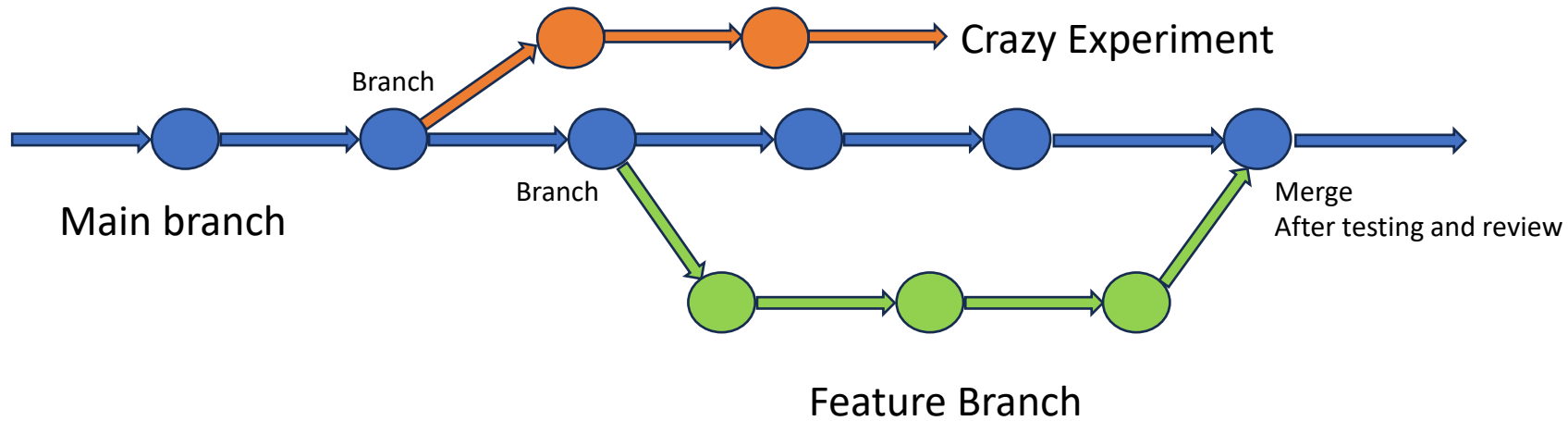
Open toma63 wants to merge 1 commit into `main` from `lesson-7-test`



More git commands and terminology

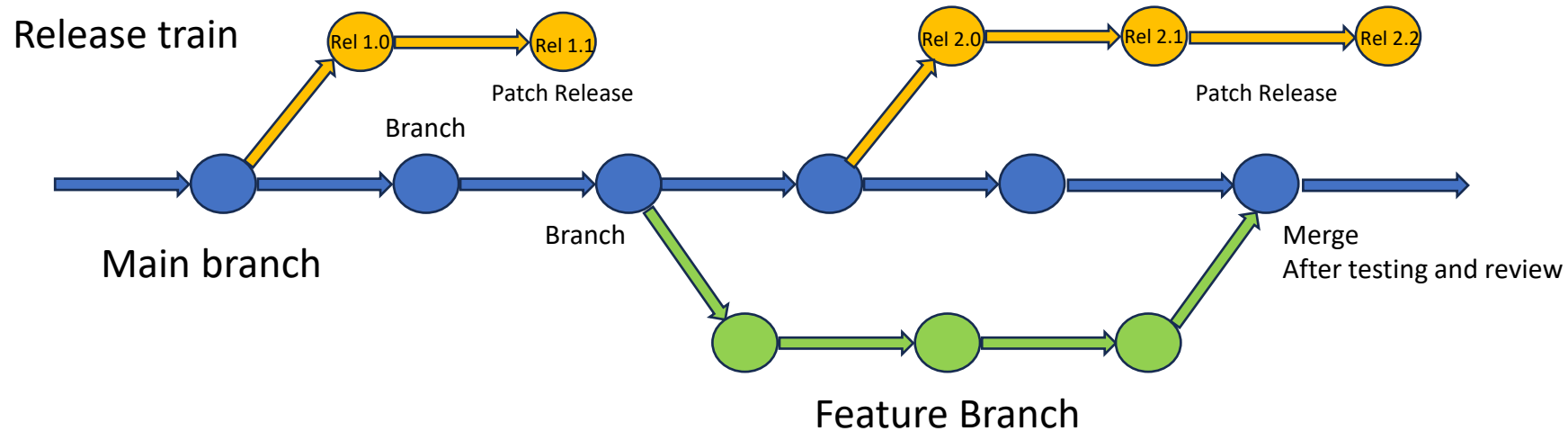
- HEAD – the tip of the current branch
- Remote
 - The remote repository associated with the current repository
 - `git remote -v`
- Relative references
 - HEAD (tip), HEAD^ (one commit before), HEAD~2 (two commits before)
- Commit Hash
 - 160 bit SHA1 secure hash (essentially unique)
 - 40 bit shorthand (just the beginning)
- Revert and restore (and `-staged`)

Git Branches



- Do experiments safely
- Build and review new features
- Manage a release process

Production Release Process



- Main branch
 - Always functional, automated testing and review to merge a PR
- Feature branches
 - Separate development of new features until stabilized
 - Avoid breaking the main branch until stable and well tested
- Release Train
 - Regular cadence
 - Feature which meet the QA deadline are incorporated
 - Not held up for any given feature and hence predictable
 - Special process for critical patch releases

Q & A
Demo
Final Project
Previews

