Class 6: git gud

"Boy I sure do love creating a merge commit every time I pull!"

Overview

1. Announcements
2. Review + Exercises
3. Q&A
4. Basic assignment

Announcements

- Regex due March 12
- Git 2 due March 19
- Regex survey closes tonight

Review

`git reset`

- Allows you to move your HEAD (and current branch) to the specified commit
- Options regarding file states
  - `--soft`
  - `--mixed`
  - `--hard`

Exercise 1

- Download this file

```
https://www.eecs.umich.edu/courses/eecs201/wn2025
|- /files/examples/git2.tar.gz
```

- Extract it, look at the `reset` directory
- Reset to the "Add another line" commit
  - `git log`

Review

`git rebase`

- Allows you to reapply your commits on top of another commit e.g.
  - Earlier commit in history
  - Further in history on top of another branch
- Allows you to perform various operations on the involved commits
- Specify a "base tip" to toss your commits onto
  - This base tip is not involved in any of the rebase operations: it serves as a parent

Exercise 2

- Check the `rebase` directory
- Note how we have a bunch of commits fixing tiny issues for the hello world printout
- Squash these changes into the "Add hello world printout" commit

Review

Conflicts

- Fix the involved files
  - Put them into working condition
  - Reconcile ideas from both branches involved
  - Make sure to delete the version control markers!
- `git status` will give you instructions based on whether your conflict comes from a rebase or a merge

Exercise 3

- Check the `conflict` directory
- Merge `change-iterator` into `master` and resolve the conflict
- The `master` branch changes the iteration count from 10 to 20
- The `change-iterator` branch changes the variable from `n` to `i`

Q&A

Basic assignment