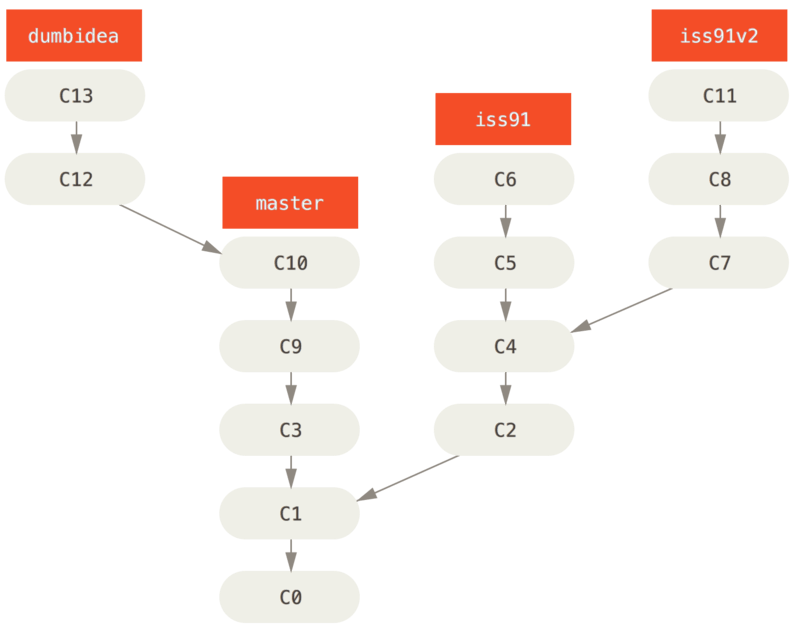
**Questions for Jon** 🎉

1. I was working on a branch that was meant to be a “2.0” version of a pipeline. I accidentally merged and overwrote the main “1.0” branch, almost losing the original pipeline. What can I do to avoid this in the future
   1. You can make "main" or “1.0” a protected branch - [About protected branches - GitHub Docs](https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/defining-the-mergeability-of-pull-requests/about-protected-branches)
2. What is the best way to structure the repository / fork branches to collaboratively work on the same project at the same
   1. Everything is saved, dont panic :)
   2. [](https://git-scm.com/book/en/v2/Git-Branching-Branching-Workflows#_topic_branch)
   3. [A successful Git branching model](https://nvie.com/posts/a-successful-git-branching-model/)
   4. [Comparing Git Workflows: What You Should Know](https://www.atlassian.com/git/tutorials/comparing-workflows)
   5. [Git Explained: Proper Team Etiquette](https://dev.to/milu_franz/git-explained-proper-team-etiquette-1od)
3. What is the best tool to deal with merge problems?
   1. [Meld](https://meldmerge.org/) (Linux / UNIX / Windows / Mac-on-AMD (Intel chip)) / [Meld for OSX by yousseb](https://yousseb.github.io/meld/) (Mac)
   2. In your IDE, e.g. this button  [in VSCode](https://code.visualstudio.com/docs/sourcecontrol/overview)
   3. [8.1 Customizing Git - Git Configuration § External Merge and Diff Tools](https://git-scm.com/book/en/v2/Customizing-Git-Git-Configuration#_external_merge_tools)
      1. near the bottom of this section shows an example output of git mergetool --tool-help which lists some available mergetools
      2. [git-mergetool Documentation § --tool=<tool>](https://git-scm.com/docs/git-mergetool#Documentation/git-mergetool.txt---toollttoolgt)
   4. Copy-paste online, e.g. [Mergely](https://editor.mergely.com/)
   5. [Git - Advanced Merging](https://git-scm.com/book/en/v2/Git-Tools-Advanced-Merging)
4. What is github?
   1. A website that hosts "git" projects. Git is a tool, GitHub is a "Software as a service platform?" (am I using that term correctly?) that provides a bunch of extra features that aren't directly part of git itself (e.g., issues, wiki)
   2. [GitHub Skills](https://skills.github.com/)
5. See visualisation of git tree?
   1. git log --graph
   2. On GitHub, at "network", e.g. <https://github.com/FCP-INDI/C-PAC/network>
   3. In GUI tools
6. how about fork and submit a pull request, instead of working on the same repo but a different branch?
   1. [5.2 Distributed Git - Contributing to a Project](https://git-scm.com/book/en/v2/Distributed-Git-Contributing-to-a-Project)
   2. [Git - Branching Workflows](https://git-scm.com/book/en/v2/Git-Branching-Branching-Workflows#_topic_branch)
   3. [6.2 GitHub - Contributing to a Project](https://git-scm.com/book/en/v2/GitHub-Contributing-to-a-Project)
7. Helpful github tools / features ? does that exist even
   1. [Issue and feature/PR templates](https://docs.github.com/en/communities/using-templates-to-encourage-useful-issues-and-pull-requests/about-issue-and-pull-request-templates)
   2. References ([official git books (free online), videos and cheatsheets](https://git-scm.com/doc))
   3. [Git - GUI Clients](https://git-scm.com/downloads/guis)
   4. [lint\_diffs · PyPI](https://pypi.org/project/lint_diffs/)
   5. blog posts
      1. [Git fundamentals, a complete guide](https://dev.to/leandronsp/git-fundamentals-a-complete-guide-do7)
      2. [7 tips for improving your productivity with Git](https://dev.to/dgenezini/7-tips-for-improving-your-productivity-with-git-ajg)
   6. [GitHub Skills](https://skills.github.com/)
   7. [GitHub Docs](https://docs.github.com/en)
   8. [Learn Git - Tutorials, Workflows and Commands | Atlassian](https://www.atlassian.com/git)
   9. [Git Tutorials and Training](https://www.atlassian.com/git/tutorials)
8. How can i make my github profile better (idk if this is relevant)
   1. Copy Jon ;) <https://github.com/shnizzedy>
      1. [About your profile - GitHub Docs](https://docs.github.com/en/account-and-profile/setting-up-and-managing-your-github-profile/customizing-your-profile/about-your-profile)
      2. basic template: [A template for how we created personal READMEs for Post Graphics team](https://github.com/kaeti/personal-readme)
      3. [Metrics embed - GitHub Marketplace](https://github.com/marketplace/actions/metrics-embed)
      4. [Inspirobot](https://inspirobot.me/) [CI job](https://github.com/shnizzedy/shnizzedy/blob/15eba6e72fa75e2b5f0e430047415fc7b8d641ca/.github/workflows/Metrics.yml#L79-L86)
      5. [Random RickRoll - GitHub Marketplace](https://github.com/marketplace/actions/random-rickroll)
9. How do you feel about GitHub becoming the primary platform for open source software although it is owned and controlled by Microsoft?
   1. Mo money, fewer problems?
10. How to make better PRs (info to include idk)
    1. [About issue and pull request templates - GitHub Docs](https://docs.github.com/en/communities/using-templates-to-encourage-useful-issues-and-pull-requests/about-issue-and-pull-request-templates)
    2. [Adding support resources to your project - GitHub Docs](https://docs.github.com/en/communities/setting-up-your-project-for-healthy-contributions/adding-support-resources-to-your-project)
11. Why are my contributions to non-main branches not appearing in my profile. I want my green boxes 🙁
    1. 😭 [Why are my contributions not showing up on my profile? § Commit was not made in the default or gh-pages branch | GitHub Docs](https://docs.github.com/en/account-and-profile/setting-up-and-managing-your-github-profile/managing-contribution-settings-on-your-profile/why-are-my-contributions-not-showing-up-on-my-profile#commit-was-not-made-in-the-default-or-gh-pages-branch)
12. How often should I commit?
    1. Committing more often is better
    2. Finish a thought: commit
    3. [How to Write a Git Commit Message](https://cbea.ms/git-commit/)
    4. [Creating a commit with multiple authors - GitHub Docs](https://docs.github.com/en/pull-requests/committing-changes-to-your-project/creating-and-editing-commits/creating-a-commit-with-multiple-authors#creating-co-authored-commits-on-the-command-line)
    5. End of day: commit
       1. If you forget, have a cronjob that does it for you. E.g:  
          0 12,17,20 \* \* \* cd /home/Gregory.Kiar/code/gkiar/todo/ && ./recommit.sh

The contents of recommit.sh:

#!/usr/bin/env bash  
  
# Automated with Cron via:  
# crontab -e  
# > 0 12,17,20 \* \* \* cd /path/to/repo && ./recommit.sh

git\_status="$(git status 2> /dev/null)"

if [[ ${git\_status} =~ "working tree clean" ]]; then

exit 0

fi

git pull

git add .

tm=$(date)

git commit -am "Regular cron-sync (${tm})"

git push

1. When should I put emojis in my commit messages?
   * [gitmoji | An emoji guide for your commit messages](https://gitmoji.dev/about)
   * [Commit Message Emoji 👋](https://github.com/dannyfritz/commit-message-emoji)
2. What are the differences, pros and cons between git in the command line and github desktop?
   1. Setting aliases (e.g. ~/.bashrc) - save time (in terminal)
3. When starting a project, at what stage do you integrate it into github, and how do you go about doing this?
   1. Straight away for convenience, push and pull from different servers and machines
   2. git init (locally)
   3. Or create on github and clone repo (create license)
4. A lot of projects contain files from different locations that are too big to upload to github, but are needed to run the scripts. When writing the code to upload to git, is there a good way to organize your folder structure to keep the relative paths within the project? How many non-code files are appropriate to upload to your repository?
   1. [git-lfs](https://git-lfs.com/) is a tool that can be used to add/pull large files on github repositories.
      1. Configure based on a) where you want to store things, and b) what file extensions you want this to apply to.

1. Are there other uses for git besides just managing your code?
   1. Papers 🙂
   2. Example: <https://github.com/gkpapers>
2. At what stage do you make your repository public? How do you manage who can see it if you want to share it with a select few?
   1. Very personal preference
   2. Can add contributors with varying access if you want it private, but allow a few people to be able to join in.