ICS 199 Git Workflow Cheat Sheet

1. Get your project's latest master branch:
    a. If you haven't downloaded your project yet: `git clone <project https URL from GitLab>`
    b. If you already have it, update master branch:
       `git fetch`
       `git diff master origin/master` (if you want to see changes)
       `git merge origin/master`

2. Create your development branch:
    a. `git checkout -b <name of dev branch – be fairly descriptive>`

3. Now do your work, commit early; commit often:
    a. `git commit -a -m "<commit message here>"`

4. At the end of your work session, time to push changes to GitLab. First, update local master:
    a. `git checkout master`
    b. `git fetch`
    c. `git diff master origin/master`
        i. if this outputs nothing, your master branch is already up-to-date. *Go to step 7*.
    d. `git merge origin/master`

5. Now update your dev branch:
    a. `git checkout <your dev branch>`
    b. `git rebase master`

6. If there was no conflict, skip to Step 7, otherwise:
    a. manually edit the file that has a conflict in your dev branch
    b. re-stage the file: `git add <file that had conflict>`
    c. finish the re-basing: `git rebase --continue`

7. Push your dev branch to GitLab (remote repository – the origin):
    a. If you haven't pushed yet: `git push -u origin <your dev branch>`
    b. If you have pushed before: `git push`

8. On GitLab, check to make sure your push passed tests:
    a. If not, fix and go back to Step 3
    b. If so, continue to Step 9

9. If done with work on this dev branch, on GitLab, create a Merge Request - assign yourself or someone else, Submit the Request
10. Accept Merge Request. Your changes should now be on the live site!!!