Code 301: Git Flow

Each day:
- ● navigate in your terminal to your forked 301 repo.
- ● *git checkout master*
- ● *git pull upstream master*
- ● *git checkout -b class-00* (replace the 00 with the actual day's number).

Driver

In your terminal:

cd into the current day's class folder

cd into the **pair-assignment** folder.

cd into the **starter-code** folder.

Type **atom ./** to open this folder as a project in Atom. If needed, use the Atom "Find in Project" feature to locate all the TODO items.

Work through one or two TODO items, testing your code as you go, until you are happy with how it's working.

In your terminal type **git status** to view the files that you have changed. You should only see the files that you have worked on.

Use **git diff** to confirm the differences between the previous version and the most recent version of your file(s).

Type **git add file1 file2** where file1, file2, etc. are the files that you have changed.

Type **git status** to view the files that have been added to your commit. You should only see the files that you worked on.

Type **git commit -m "some meaningful message"** where some meaningful message is a message that explains your commit.

Type **git status** once more to make sure there is nothing left to commit.

Type **git push origin class-00** to push the class-00 branch changes to your forked repo on GitHub.

On GitHub, verify your changes pushed to the branch of the current class day. Now, add your partner as a collaborator (Settings ---> Collaborators & teams), and send them a link to your forked repo.

Switch roles

New Driver: Do **NOT** fork the original driver's repo.

git clone your partner's fork into your **navigator** folder (which should be located in your parent **301** folder).

Use the **mv** command to change the default 301 repo name to driver-name-navigator-name.

git checkout class-00 (replace with correct day's branch).
cd into the *current-class-day/pair-assignment/starter-code*
The new driver now opens the code in Atom and has hands on the keyboard to work through the next one or two TODO items.

When finished, or when time is up, push changes back up to the current class branch, and submit a pull request to the codefellows class repository.