**Scenario #3**

Let's walk through the Git commands that go along with each step in the scenario you just observed in the video.

**Step 1: Andrew commits his changes to the documentation branch, switches to the development branch, and pulls down the latest changes from the cloud on this development branch, including the change I merged previously for the friends group feature.**

**Commit the changes on the documentation branch**

git commit -m "standardized all docstrings in process.py"

**Switch to the develop branch**

git checkout develop

**Pull the latest changes on the develop branch down**

git pull

**Step 2: Andrew merges his documentation branch into the develop branch on his local repository, and then pushes his changes up to update the develop branch on the remote repository.**

**Merge the documentation branch into the develop branch**

git merge --no-ff documentation

**Push the changes up to the remote repository**

git push origin develop

**Step 3: After the team reviews your work and Andrew's work, they merge the updates from the development branch into the master branch. Then, they push the changes to the master branch on the remote repository. These changes are now in production.**

**Merge the develop branch into the master branch**

git merge --no-ff develop

**Push the changes up to the remote repository**

git push origin master

**Resources**

Read [this great article](http://nvie.com/posts/a-successful-git-branching-model/) on a successful Git branching strategy.

**Note on merge conflicts**

For the most part, Git makes merging changes between branches really simple. However, there are some cases where Git can become confused about how to combine two changes, and asks you for help. This is called a merge conflict.

Mostly commonly, this happens when two branches modify the same file.

For example, in this situation, let’s say you deleted a line that Andrew modified on his branch. Git wouldn’t know whether to delete the line or modify it. You need to tell Git which change to take, and some tools even allow you to edit the change manually. If it isn’t straightforward, you may have to consult with the developer of the other branch to handle a merge conflict.

To learn more about merge conflicts and methods to handle them, see [About merge conflicts](https://docs.github.com/en/github/collaborating-with-issues-and-pull-requests/about-merge-conflicts).

;