**Visualizing Git:**

* Go to https://git-school.github.io/visualizing-git/
* Run the following commands int the CLI on the left of the screen:
* git checkout -b develop
* git checkout -b git
* git commit
* git commit
* git checkout develop
* git checkout -b html-css
* git commit
* git commit
* git checkout develop
* git merge git
* git checkout -b javascript
* git commit
* git commit
* git checkout develop
* git merge html-css
* git merge javascript

**GitHub PR Review:**

* Go to https://github.com/TGS-aswigert/tekcamp/pull/1 (or any Pull Request you have access to)
* Break down the information provided on the conversation page
* Note that the Checks page is used with GitHub actions to create a CI/CD pipeline
* On the Commits page, show students that you can hover over the commit date and see the exact time of the commit
* we know if you turned in late homework, and will roll back any late commits so that they are not considered for your grade
* On the Files Changed page, show the changes made to each file and demonstrate how comments are added to a line of code
* Add a suggestion on a line of code
* Either click on the Review changes button and Approve the PR or Request changes or go back to the Conversation page and close the PR
* It’s a good idea to at least walk the students through each of these options as well as the merge PR button on the Conversations page

**Rebasing:**

* Open your preferred CLI and initialize a new example-repo (git init example-repo)
* Change into your example-repo directory (cd example-repo)
* Make an empty initial commit (git commit --allow-empty -m “initial commit”)
* Run git log to show students that master only has a single commit
* Create and switch to a new branch named ‘feature’ (git checkout -b feature)
* Run git log to show students that feature also only has a single commit
* Make a new empty commit (git commit --allow-empty -m “started feature”)
* Make a new empty commit (git commit --allow-empty -m “added to feature”)
* Run git log to show students that feature now has 3 commits (initial commit, started feature, and added to feature)
* Switch to the master branch (git checkout master)
* Run git log to show students that master still only has a single commit
* Add 2 more empty commit to master (git commit --allow-empty -m “fixed bug” git commit --allow-empty -m “added README”)
* Run git log to show students that master now has 3 commits (initial commit, fixed bug, and added README)
* Switch to the feature branch (git checkout feature)
* Run git log to show students that feature still has 3 commits (initial commit, started feature, and added to feature)
* Rebase feature onto master (git rebase master)
* Run git log to show students that feature now has 5 commits (initial commit, fixed bug, added README, started feature, and added to feature)

**Oh My Git!:**

* Go to https://ohmygit.org/ and show them the button to download the game
* Open Oh My Git! on your machine (if not already downloaded, download it from the above website)
* Once game is running, choose “Levels” from the main menu
* Open the first level “Living Dangerously”
* Read out the instructions at the top right of the application
* Click on the file icon for form.txt and the text in the file will appear in its place
* Add a new line to the file and click “save” at the bottom right
* Read the new instructions at the top right, then click “Next Level” on the left
* Continue following the game’s instructions until you have completed the intro levels