

A Quick Word On Git

...

Why Am I having issues?

Definitions

- The **remote** branch is the branch located on Github that I'm uploading to
- Your **local** branch is the branch you work on

Merge Conflicts

- **Merge Conflicts** arise in Git when there are differences between changes I've pushed to the master branch and changes you've made locally
- Merge Conflicts **do not occur** when:
 - I've uploaded new files to the remote branch that didn't exist before
 - You edit files on your local branch, such as homework and classwork files, and I don't push updates to them on the remote branch
- Merge Conflicts **do occur** when:
 - I've made changes to files on the remote branch that you've also made changes to on your local branch
 - I've removed files from the remote branch that were previously there
 - You've removed files from your local branch that are still in the remote branch

Addressing Errors

- error: Entry '<fileName>' not uptodate. Cannot merge. (Changes in working directory)
 - Rename the file mentioned
- error: Entry '<fileName>' would be overwritten by merge. Cannot merge. (Changes in staging area)
 - Run 'git status'
 - See which files are under 'Changed but not updated': 'unmerged'
 - To keep my changes:
 - git checkout --theirs path/to/the/conflicted_file.php
 - git add path/to/the/conflicted_file.php
 - git commit
 - To keep your changes:
 - git checkout --ours path/to/the/conflicted_file.php
 - git add path/to/the/conflicted_file.php
 - git commit

General Rules

- With any files that are very sensitive (projects, homeworks, etc...), back them up by 1) naming them something different than what's in the remote branch, 2) making copies of them
- If you keep your changes, you can always download my files from Github
- You can always re-clone the remote branch if need be
- This is a great opportunity to learn!
- Knowing Git is incredibly useful for the workplace. There are so many different caveats that knowing the edge cases will make you stronger in using Git overall, and less likely to be caught off-guard by weird edge cases, which *will* happen